Host Controller TCP/IP Communication Protocol User Manual

Allen-Bradley by ROCKWELL AUTOMATION

User Manual

Original Instructions

Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

The following icon may appear in the text of this document.

Identifies information that is useful and can help to make a process easier to do or easier to understand.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

Additional Safety Information

Although every effort is made to keep this manual accurate and up-to-date, MagneMotion® and Rockwell Automation® assumes no responsibility for any errors, omissions, or inaccuracies. Information that is provided in this manual is subject to change without notice. Any sample code that is referenced in this manual or included with MagneMotion software is included for illustration only and is, therefore, unsupported.

ATTENTION: For additional safety notices and definitions, see the Notes, Safety Notices, and Symbols section.

Contents

4 Protocol Reference

About This Manual

Overview

This section provides information about the use of this manual, including the manual structure, related documentation, format conventions, and safety conventions.

Purpose

This manual describes the communication protocol and messages that are used between the MagneMotion® high-level controller application and a host controller that is equipped with a TCP/IP interface (typically a PC). This manual also provides basic troubleshooting information. This manual is not intended to provide a design guide for the installation, or a reference for the operation of a transport system.

Use this manual in combination with the other manuals and documentation that accompany the transport system to design, install, configure, test, and operate a transport system.

NOTE: Depending on software version, user access privileges, and application configuration, certain operations that are described in this manual may not be available.

Prerequisites

The information and procedures that are provided in this manual assume the following:

- Familiarity with control system programming. Details on software creation that uses the command and response protocol definitions that are provided in this manual for a production system are beyond the scope of this manual.
- The transport system has been installed and configured, and full documentation for the transport system is available.
- All personnel who configure, operate, or service the transport system are properly trained.

Manual Conventions

Numbers – All numbers are assumed to be decimal unless otherwise noted and use the US number format; that is, one thousand = 1,000.00. Non-decimal numbers (binary hexadecimal) are explicitly stated.

- Binary – Followed by $_2$, for example, $1100\ 0001\ 0101_2$, $1111\ 1111\ 1111\ 1111_2$.
- Hexadecimal – Preceded by 0x, for example, 0xC15, 0xFFFF. AB_{16}, BA_{16}, and other similar annotations are also considered hexadecimal.

Measurements – All measurements are SI (International System of Units). The format for dual dimensions is SI_units [English_units]; for example, 250 mm [9.8 in].

Documentation

The documentation that is provided with the transport system includes this manual, which provides complete documentation for the use of the TCP/IP communication protocol for the transport system. Other manuals in the document set, which are listed in the Additional Resources section, support installation, configuration, and operation of the transport system.

The examples in this manual are included solely for illustrative purposes. Because of the many variables and requirements that are associated with any linear synchronous motor (LSM) system installation, Rockwell Automation cannot assume responsibility or liability for actual use that is based on these examples.

Notes, Safety Notices, and Symbols

Notes, Safety Notices, and Symbols that are used in this manual have specific meanings and formats. Examples of notes, the different types of safety notices and their general meanings, and symbols and their meanings are provided in this section. Adhere to all safety notices provided throughout this manual to help achieve safe installation and use.

Notes

Notes are set apart from other text and provide additional or explanatory information. The text for Notes is in standard type as shown in the following example.

NOTE: A note provides additional or explanatory information.

Safety Notices

Safety Notices are set apart from other text. The symbol on the left of the notice identifies the type of hazard. The text in the message panel identifies the hazard, methods to avoid the hazard, and the consequences of not avoiding the hazard.

Examples of the standard safety notices that are used in this manual are provided in this section. Each example includes a description of the hazard indicated. Labels may also be on or inside the equipment to provide specific precautions.

NOTICE	Identifies an informational notice that indicates practices that are not related to personal injury that could result in equipment or property damage.

IMPORTANT	Identifies information that is critical for the successful application and understanding of the product.

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help to identify a hazard, avoid a hazard, and recognize the consequence.

SHOCK HAZARD: Identifies information about practices or circumstances where a severe shock hazard is present that could cause personal injury or death.

PINCH/CRUSH HAZARD: Identifies information about practices or circumstances where there are exposed parts that move, which could cause personal injury or death.

Additional Resources

Before configuring or running the components, consult the following documentation. You can view or download publications at rok.auto/literature.

Resource	Description
MagneMotion QuickStick and QuickStick HT Design Guide, publication MMI-RM001	This manual explains how to design and configure the track layout and transport system.
MagneMotion Glossary of Terms, publication MMI-RM003	This manual includes definitions of MagneMotion and industry terms.
MagneMotion System Configurator User Manual, publication MMI-UM046	This manual explains how to use the MagneMotion® Configurator to create and modify the Node Controller Configuration File (Configuration File) for the MagneMotion® transport systems.
QuickStick Motors Technical Data, publication MMI-TD051	This manual includes technical specifications for the QuickStick 100 and QuickStick 150 motors.

Resource	Description
MagneMotion Node Controller Interface User Manual, publication MMI-UM001	This manual explains how to use the supplied interfaces to configure and administer node controllers that are used with transport systems. This manual also provides basic troubleshooting information.
MagneMotion LSM Synchronization Option User Manual, publication MMI-UM005	This manual explains how to install, operate, and maintain the LSM Synchronization Option for use with transport systems.
MagneMotion NCHost TCP/IP Interface Utility User Manual, publication MMI-UM010	This manual explains how to use the NCHost TCP/IP Interface Utility to run a transport system for testing and debugging. This manual also explains how to develop Demo Scripts to automate vehicle motion for that testing.
MagneMotion Virtual Scope Utility User Manual, publication MMI-UM011	This manual explains how to install and use the MagneMotion Virtual Scope utility. This utility provides real-time feedback of the change in Linear Synchronous Motor (LSM) performance parameters.
MagneMotion Node Controller Hardware User Manual, publication MMI-UM013	This manual explains how to install and maintain the node controllers that are used with transport systems.
MagneMover LITE Ethernet Motor Configuration and Communication, publication MMI-UM031	This manual describes the network topologies for wiring MagneMover LITE Ethernet motors and for combining both RS-422 and Ethernet motors in the same transport system.
MagneMotion Host Controller TCP/IP Communication Protocol User Manual, publication MMI-UM003	These manuals describe the communication protocols between the high level controller and a host controller. These manuals also provide basic troubleshooting information.
MagneMotion Host Controller EtherNet/IP Communication Protocol User Manual, publication MMI-UM004	
Power Supply Reference Manual 1606-XLS960F-3, publication 1606-RM032	The manual provides the specifications for the 1606 power supplies.
MagneMover LITE User Manual, publication MMI-UM002	This manual explains how to install, operate, and maintain the MagneMover LITE transport system. This manual also provides information about basic troubleshooting.
QuickStick 100 User Manual, publication MMI-UM006	This manual explains how to install, operate, and maintain the QuickStick 100 transport system. This manual also provides information about basic troubleshooting.
QuickStick 150 User Manual, publication MMI-UM047	This manual explains how to install, operate, and maintain the QuickStick 150 motors and magnet arrays. This manual also provides information about basic troubleshooting.
QuickStick HT User Manual, publication MMI-UM007	This manual explains how to install, operate, and maintain the QuickStick High Thrust (QSHT) transport system. This manual also provides information about basic troubleshooting.
EtherNet/IP Network Devices User Manual, publication ENET-UM006	Describes how to configure and use EtherNet/IP devices to communicate on the EtherNet/IP network.
Ethernet Reference Manual, publication ENET-RM002	Describes basic Ethernet concepts, infrastructure components, and infrastructure features.

Resource	Description
System Security Design Guidelines Reference Manual, publication SECURE-RM001	Provides guidance on how to conduct security assessments, implement Rockwell Automation products in a secure system, harden the control system, manage user access, and dispose of equipment.
UL Standards Listing for Industrial Control Products, publication CMPNTS-SR002	Assists original equipment manufacturers (OEMs) with construction of panels, to help ensure that they conform to the requirements of Underwriters Laboratories.
American Standards, Configurations, and Ratings: Introduction to Motor Circuit Design, publication IC-AT001	Provides an overview of American motor circuit design based on methods that are outlined in the NEC.
Industrial Components Preventive Maintenance, Enclosures, and Contact Ratings Specifications, publication IC-TD002	Provides a quick reference tool for Allen-Bradley industrial automation controls and assemblies.
Safety Guidelines for the Application, Installation, and Maintenance of Solid-state Control, publication SGI-1.1	Designed to harmonize with NEMA Standards Publication No. ICS 1.1-1987 and provides general guidelines for the application, installation, and maintenance of solid-state control in the form of individual devices or packaged assemblies incorporating solid-state components.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, rok.auto/certifications.	Provides declarations of conformity, certificates, and other certification details.

Additional Documentation

Release Notes

The Release Notes that are supplied with Rockwell Automation software include special instructions, identification of software versions, identification of new features and enhancements, and a list of known issues. Reading this file before using the software is recommended.

Upgrade Procedure

The Upgrade Procedures that are supplied with Rockwell Automation software provide instructions for upgrading from one version of Rockwell Automation software to another. They also include the procedures for file and driver upgrades that are associated with the software.

Transport System Limits

For information on transport system limits refer to your motor user manual or the node controller user manuals listed in the Additional Resources on page 11.

File Maintenance

Backup Files

It is recommended that regular backups of all files that have been changed and copies of all original and backup files be kept at a remote location for safety.

Creating Backup Files

Backup files are not created automatically. It is the responsibility of the user to create backups of all files by copying them to a secure location.

Restoring from Backup Files

Damaged files can be restored by copying the backup files into the appropriate locations.

Introduction 1

Overview

This chapter provides an overview of the Host Controller TCP/IP Communication Protocol that is used to communicate between the high-level controller and the host controller (typically general-purpose computer-based). This chapter also provides a basic overview of the transport system hardware and software. Additionally, the basic set of tasks that are required to use the TCP/IP Communication Protocol with a transport system are described.

Use this manual to develop an application that uses the TCP/IP Communication Protocol to monitor and control a transport system. Some of the procedures that are provided in this manual as examples vary in actual use depending on the transport system configuration, communications, and other variables.

This manual supports:

- MagneMover® LITE™ transport systems.
- QuickStick® transport systems.
- QuickStick® HT™ transport systems.

TCP/IP Communication Protocol Overview

The Host Controller TCP/IP Communication Protocol is provided by Rockwell Automation to run the transport system during operation. This protocol supports transport system startup, controlling and monitoring vehicle motion, monitoring transport system module status, and handshaking with user-supplied equipment to transfer vehicles on and off the system.

The transport system is a configuration of linear synchronous motors that are placed end-to-end to form long chains or paths. These chains are used to move and position vehicles in a controlled manner at various acceleration/deceleration and velocity profiles while carrying a wide range of payloads with high precision. The transport system consists of the following components at a minimum:

- MagneMover LITE or QuickStick motors.
- Vehicles with magnet arrays.

- Node controllers.
- Power supplies.
- Configuration file defining paths and nodes.
- User-supplied host controller.

Each path in the transport system starts at a node and the motor at that node is connected to a node controller. Any path that a vehicle can exit from must also end in a node that is connected to a node controller. Multiple node controllers can be used for systems with many paths. The high-level controller application runs on one node controller, which interfaces the complete transport system with the host controller. This interface allows the host controller to send commands to and receive status from the transport system. This manual specifies the protocol that the high-level controller software uses to communicate with host controllers via TCP/IP over Ethernet.

Each of the components of the transport system must be fully defined in the Node Controller Configuration File to achieve proper operation of the transport system. Once the transport system is fully defined, it can be monitored and controlled using the TCP/IP Communication Protocol.

NOTE: Changes to the control application may be necessary if any aspect of the transport system changes. Changes could include the number or length of paths, the number or type of nodes, vehicle length, payload weight, or other physical or configuration factors.

Transport System Components Overview

This section identifies the components of a transport system as shown in Figure 1-1 and described after the figure.

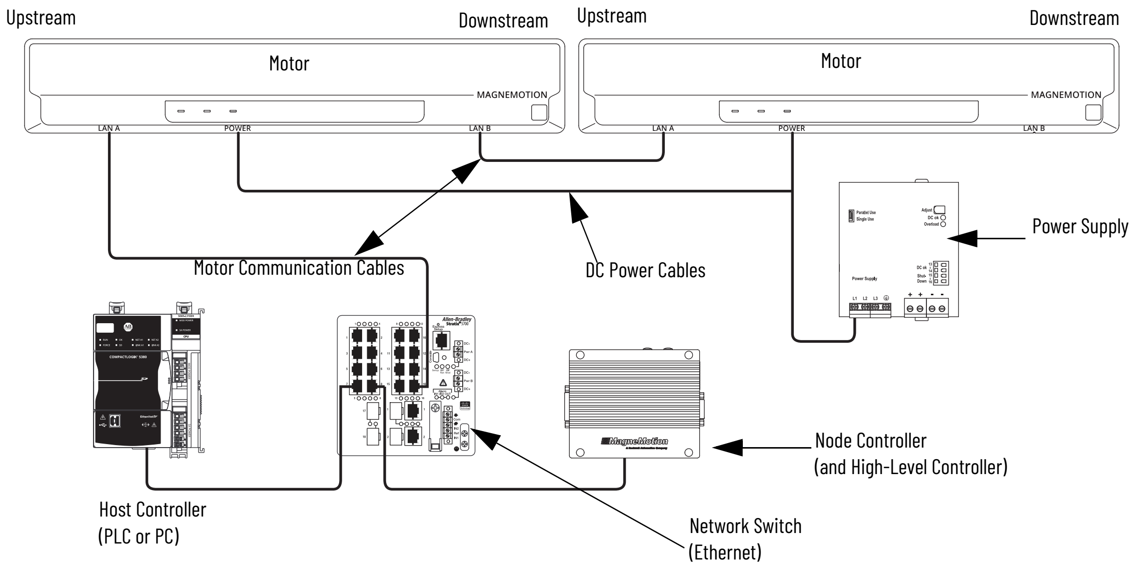


Figure 1-1: Simplified View of the Transport System Components

- **DC Power Cables and Communication Cables** – Distributes DC power to the motors and carries communications, such as RS-422 or Ethernet, between the components of the transport system.
- **High-Level Controller** (HLC) – Software application that is enabled on one node controller. This application handles all communication with the user-supplied host controller and directs communication as appropriate to individual node controllers.
- **Host Controller** – User-supplied controller for control and monitoring of the transport system using TCP/IP communications.
- **Motor/Stator** – Refers to a linear synchronous motor (LSM).
- **Network** – Ethernet network providing communications (TCP/IP) between the host controller, motors, and the HLC (TCP/IP is also used between node controllers).
- **Node Controller** (NC) – Coordinates motor operations and communicates with the HLC. Several types of node controllers are available. All node controllers support Ethernet communication with the host controller and the motors, and depending on the model, provide RS-422 ports for communication with the motors.
- **Power Supply** – Provides DC power to the motors.
- **Vehicle with Magnet Array** – Carries a payload through the transport system as directed. The magnet array is mounted to the vehicle facing the motors and interacts with the motors, which move each vehicle independently.

Transport System Software Overview

Several software applications are used to configure, test, and administer a transport system as shown in Figure 1-2 and described after the figure. See Additional Resources on page 11 for the reference manuals for these applications.

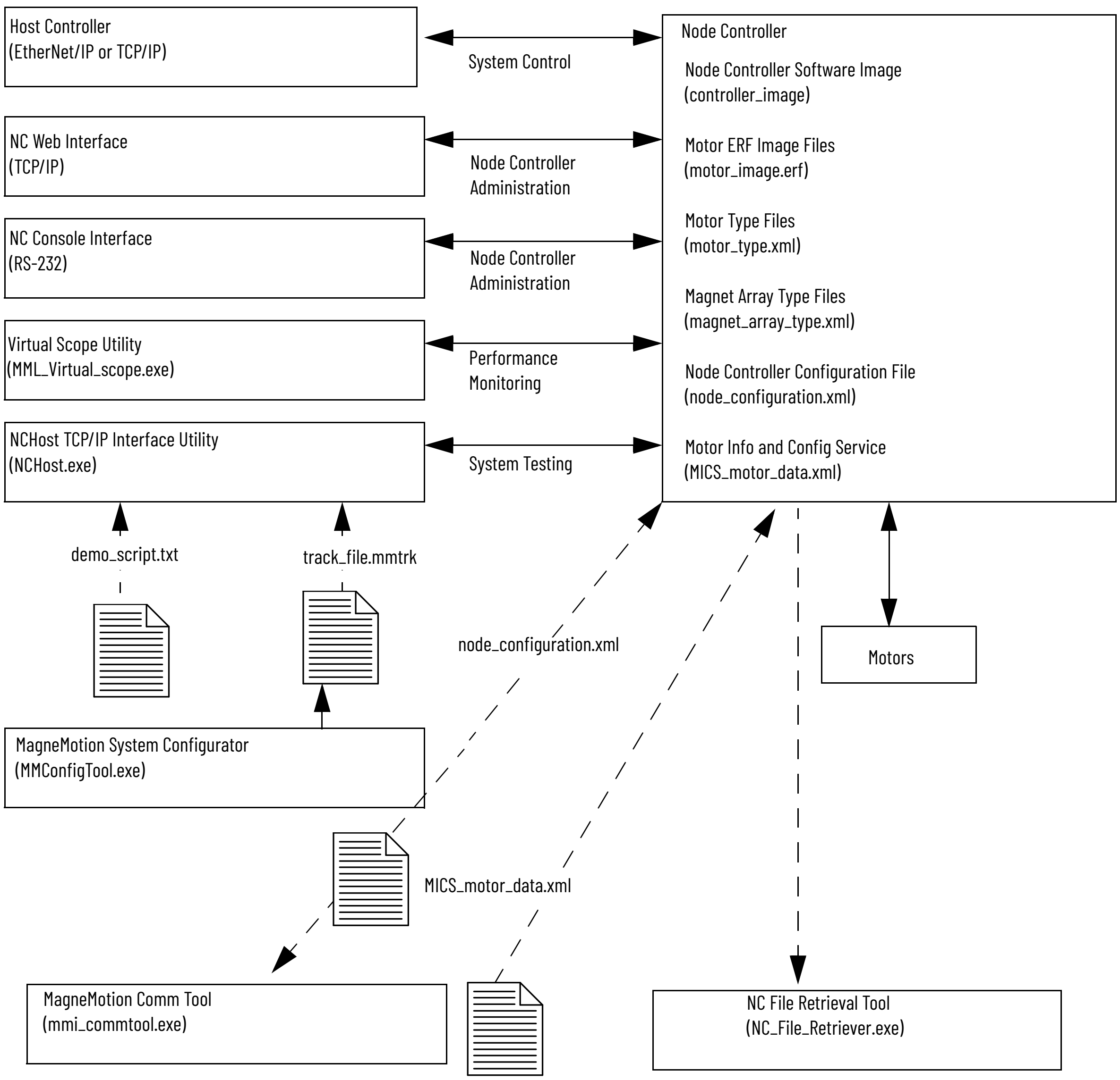


Figure 1-2: Simplified View of Transport System Software Organization

For additional information on utilities or file type information, see *MagneMotion Node Controller Interface User Manual*, publication MMI-UM001.

NOTICE	Modifications to the Image or Type files could cause improper operation of the transport system.

Utilities

- **NC Web Interface –** A web-based software application that is supplied by Rockwell Automation and resident on the node controllers, for administration of the transport system components.
- **NC Console Interface –** A serial communication software application that is supplied by Rockwell Automation and resident on the node controllers, for administration of the node controller.
- **Virtual Scope Utility –** A Windows® software application that is supplied by Rockwell Automation to monitor and record the change of motor performance parameters. These parameters are displayed as waveforms to analyze the performance of the motors.
- **NCHost TCP Interface Utility –** A Windows software application that is supplied by Rockwell Automation to move vehicles for test or demonstration purposes. This application supports system testing without the host controller to verify that vehicles move correctly before integrating the transport system into a production environment.
- **MagneMotion System Configurator (**Configurator Utility)– A Windows software application that is supplied by Rockwell Automation to create or change the Node Controller Configuration File. This tool is available in both MagneMover LITE and QuickStick versions to create or change the Track File and Track Layout File for the transport systems.
- **Ethernet Motor Commissioning Tool –** A Windows software application that is supplied by Rockwell Automation to create and edit MagneMotion Information and Configuration Service (MICS) files.
- **NC File Retrieval Tool –** A Windows software application that is supplied by Rockwell Automation to download configuration and operation files from the specified HLC and all node controllers controlled by the HLC.

File Types

- **Node Controller Software Image File** (IMG file) – The software file for the node controllers (*controller_image*), includes the node controller and high-level controller applications. The Node Controller Software Image file is uploaded to all node controllers in the transport system.
- **Motor ERF Image Files** (ERF file) – The software files for the motors (*motor_image*.erf). The Motor ERF Image files are uploaded to all node controllers in the transport system and then programmed into all motors.
- **Motor Type Files –** XML files (*motor_type*.xml) that contain basic information about the specific motor types being used. The Motor Type files are uploaded to all node controllers in the transport system.
- **Magnet Array Type File –** An XML file (*magnet_array_type*.xml) that contains basic information about the specific magnet array type that is used on the vehicles in

the transport system. The Magnet Array Type file is uploaded to all node controllers in the transport system.

- **Node Controller Configuration Files** (Configuration file) – An XML file (*node_configuration*.xml) that contains all the parameters for the components in the transport system. Multiple Node Controller Configuration Files can be uploaded to all node controllers in the transport system, but only one is active.
- **MagneMotion Information and Configuration Service (MICS) File –** An XML file (*MICS_motor_data*.xml) that contains the network topology parameters for the transport system when using Ethernet communication with the motors. The file includes the MAC address of each motor and the location of each motor on a path. The MICS file is uploaded to all node controllers in the transport system.
- **Restricted Parameters File –** An XML file (*restricted_parameters*.xml) that provides access to restricted configuration elements for specific transport systems. The Restricted Parameters file is uploaded to the HLC. For the development of a custom Restricted Parameters file for a specific transport system, see Rockwell Automation Support.
- **Demo Script –** A text file (*demo_script*.txt) uploaded to the NCHost TCP Interface Utility to move vehicles on the transport system for test or demonstration purposes.
- **Track File –** A text file (*track_file*.mmtrk) that contains graphical path and motor information about the transport system. The Track file is used by the NCHost TCP Interface Utility to provide a graphical representation of the transport system to monitor system operation. The Track file is created for MagneMover LITE transport systems using the Configurator Utility. See Rockwell Automation Support for questions regarding the development of Track files for QuickStick transport systems.
- **Track Layout File –** An XML file (*track_layout*.ndx) that contains the parameters for the graphical representation of a MagneMover LITE transport system. The Track Layout file is used by the Configurator Utility to generate the Node Controller Configuration File and the Track file for MagneMotion systems.

NOTICE	Modifications to the Image or Type files could cause improper operation of the transport system.

Getting Started with the TCP/IP Communication Protocol

Use this manual as a guide and reference for application development when using the TCP/IP Communication Protocol. Follow the steps in this section to get the entire transport system operational quickly with the aid of the other manuals (see Additional Resources on page 11).

NOTE: Make sure that all components and complete design specifications, including the physical layout of the transport system, are available before starting to install or test the transport system.

To get started quickly with the transport system:

1. Download the software for the appropriate transport system and the Utilities from rok.auto/pcdc.

   NOTE: The minimum requirements for running software applications are a general-purpose computer running Microsoft® Windows 7 with .NET 4.0. An Ethernet port (web interface) and an optional RS-232 port (console interface) are required to connect to the node controllers. However, the TCP/IP Communication Protocol can be implemented on a general-purpose computer running any OS.

2. Install the components of the transport system as described in the MagneMotion Node Controller Hardware User Manual, publication MMI-UM013, and either the MagneMover LITE User Manual, publication MMI-UM002, the QuickStick 100 User Manual, publication MMI-UM006, QuickStick 150 User Manual, publication MMI-UM047, or the QuickStick HT User Manual, publication MMI-UM007.

3. Install the Configurator Utility on a computer for user access (see the MagneMotion System Configurator User Manual, publication MMI-UM046).

   A. For MM LITE systems, define the motors and paths and their relationships in the transport system graphically to create the Track Layout File (*track_layout*.ndx) and the Track File (*track_file*.mmtrk).

   B. For all transport systems, create the Node Controller Configuration File (*node_configuration*.xml) to define the components and operating parameters of the transport system.

4. Set the IP address for each node controller and specify the node controller to be used as the high-level controller (see the MagneMotion Node Controller Interface User Manual, publication MMI-UM001).

5. Upload the configuration, image, and type files to each node controller using the node controller web interface (see the MagneMotion Node Controller Interface User Manual, publication MMI-UM001).

   NOTE: Once configured, the node controllers can be used to simulate the transport system.

6. When using motors with Ethernet communication, create the MICS file and provision the motors (see either the MagneMover LITE User Manual, publication MMI-UM002 or the QuickStick and QuickStick HT Design Guide, publication MMI-RM001.

7. Program the motors using the Motor ERF Image files (see the MagneMotion Node Controller Interface User Manual, publication MMI-UM001, and the MagneMotion NCHost TCP Interface Utility User Manual, publication MMI-UM010).

8. Test and debug the transport system by using the NCHost TCP Interface Utility and Demo Scripts (see the MagneMotion NCHost TCP Interface Utility User Manual, publication MMI-UM010). NCHost provides an easy method to verify proper operation and make adjustments such as refining the control loop tuning.

   NOTE: The NCHost TCP Interface Utility is for test and verification trials only. The host controller must be used to control the transport system after verification of functionality.

9. Configure the host controller for transport system control as required to meet the material movement needs of the facility where the system is installed. Use this manual when developing a control application using TCP/IP communications with the host controller. See the following sections for examples:

   - Connecting to the Transport System on page 25.
   - Running the Transport System on page 25.
   - Monitoring Transport System Status on page 39.
   - Vehicle Motion on page 59.

   When using EtherNet/IP communications with the host controller, see the MagneMotion Host Controller EtherNet/IP Communication Protocol User Manual, publication MMI-UM004.

Review manuals.

Familiarize yourself with components.

See Additional Resources on page 11

Go to rok.auto/pcdc

1. Click Downloads by Product.
2. Search for "QuickStick".
3. Select the desired QuickStick product and version.
4. In the selection box, click the Downloads icon.

Design and install the transport system.

For QS and QSHT see, MMI-RM001.

Also see the applicable User Manual for your motor:

- MMI-UM002 (MML)
- MMI-UM006 (QS 100)
- MMI-UM047 (QS 150)
- MMI-UM007 (QSHT)

Install the motor and components.

See the applicable User Manual for your motor:

- MMI-UM002 (MML)
- MMI-UM006 (QS 100)
- MMI-UM047 (QS 150)
- MMI-UM007 (QSHT)

See also, MMI-RM001

Install the node controllers.

See MMI-UM013

Install the MagneMotion System Configurator on a PC for user access.

See MMI-UM046

Review these topics:

- Introduction
- Use the MagneMotion System Configurator
- UI Reference

Create the Node Controller Configuration File.

See MMI-UM046

Review these topics:

- Use the MagneMotion System Configurator
- Transport System Layout
- Transport System Configuration

See also: MMI-UM001

Set the IP address for each node controller and upload the configuration, image, and type files.

See MMI-UM001

Review these topics:

- Node Controller Administration - Console
- Web Interface Reference

For motors with Ethernet, create and upload the MICS file.(1)

See MMI-RM001 and or MMI-UM005.

See also: MMI-UM001

Review this topic:

- Node Controller Administration - Console

Restart Services

Program motors with ERF image files.

See the applicable User Manual for your motor:

- MMI-UM002 (MML)
- MMI-UM006 (QS 100)
- MMI-UM047 (QS 150)
- MMI-UM007 (QSHT)

Test and debug.(2)

See MMI-UM010

Review these topics:

- Using the NCHost TCP/IP Interface Utility
- Demo Scripts

See also:

- MMI-UM002 (MML)
- MMI-UM006 (QS 100)
- MMI-UM047 (QS 150)
- MMI-UM007 (QSHT)

Configure the host controller.(3)

TCP/IP or EtherNet/IP:

- See MMI-UM003 (TCP/IP)
- See MMI-UM004 (EtherNet/IP)

Review these topics:

- Transport System Control
- Application Notes

(1) This configuration does not apply to QS 100 motors.

(2) Use the MagneMotion Virtual Scope Utility to confirm PID tuning for all motors, if required. See the MagneMotion Virtual Scope Utility User Manual, publication MMI-UM011.

(3) Configure LSM Synchronization for QS 100 motors, if necessary. See the MagneMotion LSM Synchronization Option User Manual, publication MMI-UM005.

Figure 1-3: Work Flow for QuickStick System Installation and Commissioning

This page intentionally left blank.

Transport System Control 2

Overview

This chapter provides basic information on using the Host Controller TCP/IP Communication Protocol to monitor and control a MagneMotion® transport system.

See Application Notes on page 59 for detailed information on vehicle motion and using the transport system. See Protocol Reference on page 133 for detailed information on each command. This chapter includes command overviews and provides a representation of the actual command messages. For complete message transmission protocol, see Communications Protocol on page 393.

Connecting to the Transport System

This section describes how to connect the host controller to the high-level controller (HLC) for the transport system to monitor and control the transport system. Once the connection is established, there is no need for authentication.

1. Make a network connection from the host controller to the HLC using 10/100/1000 Base-TX half-duplex or full-duplex twisted-pair Ethernet cable.
2. Use the Ping utility on the host controller to verify the network connection to the HLC.
3. From the host controller, connect to the HLC at TCP port 799, see Communications Protocol on page 393.

Running the Transport System

This section describes how to start and control the transport system. Startup includes verifying that all components of the transport system have started properly and are ready to move the vehicles and all vehicles on the transport system are identified. Control includes commanding vehicle motion as required, and stopping and restarting all motion on the transport system. Most examples that are provided in this section reference the simplified transport system layout that is shown in Figure 2-1.

1. Power up the various components of the transport system (host controller, node controllers, motors, and any user equipment).

2. Restart each node controller as necessary by issuing a Restart Services command through the node controller web interface (see the MagneMotion Node Controller Interface User Manual, publication MMI-UM001).

   NOTE: Node controller restart is only needed when the system configuration is updated, or other type/image files have been changed. Node controllers typically take longer than any path to reset, so if power to the system is cycled (including the node controllers), simply reset and start the paths.

3. Connect to the high-level controller for the transport system (see Connecting to the Transport System on page 25).

4. Verify that all node controllers are running (see the MagneMotion Node Controller Interface User Manual, publication MMI-UM001).

5. Reset all paths in the transport system (see Reset on page 28).

   The HLC sends a status message for each path that shows the command was received. An additional status message is sent that shows completion or failure of the reset and if it failed, the reason for failure.

6. Start all paths in the transport system (see Startup on page 29).

   The HLC sends a status message for each path that shows that the command was received. An additional status message is sent that shows completion or failure of the startup and if it failed, the reason for failure.

7. Send motion commands to the HLC to move the vehicles (see Moving Vehicles on page 49).

   The HLC sends a status message for each motion command that shows that the command was received or rejected, and the reason for the rejection. If accepted, an additional status message is sent when the command completes (the vehicle is within the arrival tolerance of the destination as defined in the Node Controller Configuration File).

   NOTE: The HLC does not send status messages to show that a motion command failed to complete. The host controller is required to monitor the status messages and determine that a motion command has not completed in an expected amount of time and then determine the cause.

8. Send Status Request commands to the HLC to verify the status of the transport system (see Monitoring Transport System Status on page 39).

   The HLC sends status messages as requested.

9. Stop motion on specific paths in the transport system as required (see Suspend on page 30).

10. Restart motion on suspended paths as required (see Resume on page 32).

11. Shut down the transport system for service or when not in use (see Shutting Down the Transport System on page 57).

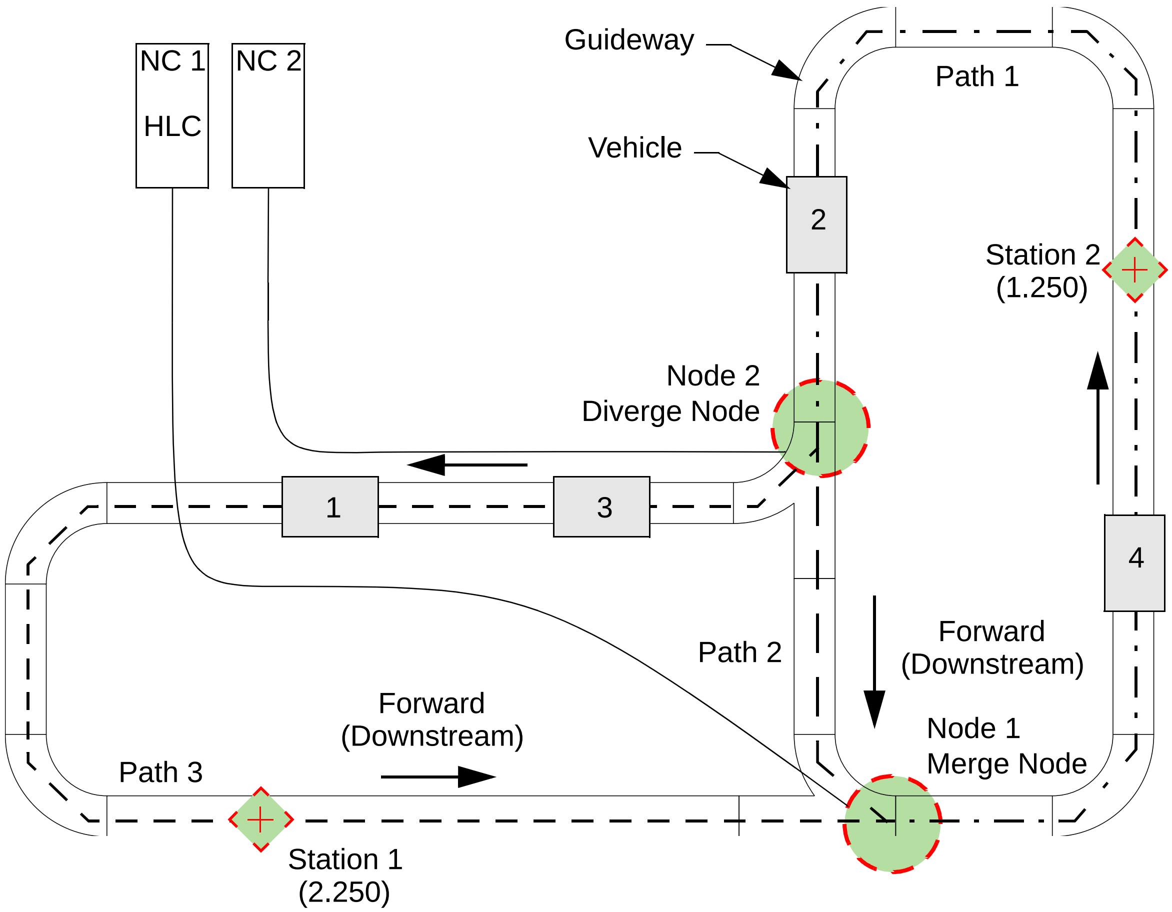


Figure 2-1: Merge-Diverge Transport System for Examples

Transport System Reset, Startup, and Operation

This section describes how to start, or restart the components of the transport system during initial startup or when returning it to service. Vehicle motion cannot be started on a path if the E-stop or Interlock is On for that path. All examples provided reference Figure 2-1.

NOTE: All examples that are provided in this manual omit the MagneMotion protocol framing for ease of reading (see Command Message Framing on page 399). The examples separate all fields by spaces for clarity only.

Reset

Use the Reset command to reset the motors on one or more specified paths. Reset all motors on all paths when a new Node Controller Configuration File has been uploaded to the node controllers. Path status can be monitored as described in Path Status on page 41. The following actions occur during a reset:

- Motor communication is reset.
- Restarts the runtime code on the motors.
- Deletes the vehicle records for vehicles on the specified paths.

All Paths

To reset all paths, use the Reset command with the Path ID set to 0.

```
0xB8 0000
```

The HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the Reset command was accepted for all paths.

```
0xD0 B8 00 0001
0xD0 B8 00 0002
0xD0 B8 00 0003
```

Once the Reset command completes, the HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the Reset command completed successfully for all paths.

```
0xD0 B8 80 0003
0xD0 B8 80 0002
0xD0 B8 80 0001
```

Specific Path

To reset a specific path (for example, path 1), use the Reset command with the Path ID set to 0x0001.

```
0xB8 0001
```

The HLC sends a Command Status message that identifies the path and shows the command status. This example shows that the Reset command was accepted for path 1.

```
0xD0 B8 00 0001
```

Once the Reset completes, the HLC sends a Command Status message that identifies the path. This example shows that the Reset command completed successfully for path 1.

```
0xD0 B8 80 0001
```

Startup

Use the Startup command to initiate the startup sequence on one or more specified paths. Path status can be monitored as described in Path Status on page 41. The following events require initiating a startup:

- When first connecting to a high-level controller.
- After cycling power to the transport system.
- After resetting the HLC.
- After resetting paths.
- To locate all vehicles on the transport system.

NOTE: Paths cannot be started if the E-stop or Interlock is On.

Vehicles may not be detected during startup if they straddle a node. Straddling occurs when part of the vehicle is on the downstream end of one path in a node and the rest of the vehicle is on the upstream end of another path in that node.

Paths cannot be started if there is a vehicle on the path that is longer than the total length of the path unless Overtravel Nodes are used (only supported on QSHT transport systems).

All Paths

To start all paths, use the Startup command with the Path ID set to 0.

```
0xB2 0000
```

The HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the Startup command was accepted for all paths.

```
0xD0 B2 00 0001
0xD0 B2 00 0002
0xD0 B2 00 0003
```

Once the startup command completes, the HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the startup completed successfully for all paths.

```
0xD0 B2 80 0003
0xD0 B2 80 0002
0xD0 B2 80 0001
```

Specific Path

To startup a specific path (for example, path 1), use the Startup command with the Path ID set to 1.

```
0xB2 0001
```

The HLC sends a Command Status message that identifies the path and shows the command status. This example shows that the Startup command was accepted for path 1.

```
0xD0 B2 00 0001
```

Once the startup command completes, the HLC sends a Command Status message that identifies the path. This example shows that the Startup command completed successfully for path 1.

```
0xD0 B2 80 0001
```

Suspend

Use the Suspend Motion command to stop all vehicle motion on the specified paths at the closest allowed position in the current direction of motion. Once stopped, all vehicles are held at their locations. Path status can be monitored as described in Path Status on page 41.

NOTE: Vehicles in motion within a keep-out area in the direction of the keep-out area do not stop until they exit the keep-out area. Vehicles with a destination within the keep-out area stop at their destination.

Stopping motion on one specific path does not affect motion on other paths unless the vehicles on those paths are commanded to the suspended paths. In this case, the vehicle travels as far as it can, then stops and waits for motion on the suspended paths to be resumed.

All Paths

To suspend all paths, use the Suspend Motion command with the Path ID set to 0.

```
0xB4 0000
```

The HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the Suspend Motion command was accepted for all paths.

```
0xD0 B4 00 0001
0xD0 B4 00 0002
0xD0 B4 00 0003
```

Specific Path

To suspend a specific path (for example, path 1), use the Suspend Motion command with the Path ID set to 1.

```
0xB4 0001
```

The HLC sends a Command Status message that identifies the path and shows the command status. This example shows that the Suspend Motion command was accepted for path 1.

```
0xD0 B4 00 0001
```

FastStop

Use the FastStop Motion command to stop all vehicle motion on the specified paths. The motors immediately apply maximum thrust opposing the direction of motion to all vehicles and hold all stopped vehicles at their locations. Path status can be monitored as described in Path Status on page 41.

NOTE: Vehicles in motion within a keep-out area stop within the keep-out area.

Stopping motion on one specific path does not affect motion on other paths unless the vehicles on those paths are commanded to the stopped paths. In this case, the vehicle travels as far as it can, then stops and waits for motion on the stopped path to be resumed.

This command is not supported on MagneMover® LITE transport systems.

All Paths

To stop motion on all paths immediately, use the FastStop Motion command with the Path ID set to 0.

```
0xBC 0000
```

The HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the FastStop Motion command was accepted for all paths.

```
0xD0 BC 00 0001
0xD0 BC 00 0002
0xD0 BC 00 0003
```

Specific Path

To stop motion on a specific path (for example, path 1) immediately, use the FastStop Motion command with the Path ID set to 1.

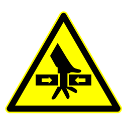
```
0xBC 0001
```

The HLC sends a Command Status message that identifies the path and shows the command status. This example shows that the FastStop Motion command was accepted for path 1.

```
0xD0 BC 00 0001
```

Resume

Use the Resume Motion command to restart vehicle motion on the specified paths. The Resume command is used after a FastStop, a Suspend command, or after an Emergency Stop (E-stop) was issued (for an E-stop, the E-stop button must be manually reset). Once issued, all vehicles on the specified paths resume their motion based on their currently active motion commands. Path status can be monitored as described in Path Status on page 41.



PINCH/CRUSH HAZARD: Moving mechanisms (vehicles) have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

All Paths

To resume all paths, use the Resume Motion command with the Path ID set to 0.

```
0xB3 0000
```

The HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the Resume Motion command was accepted for all paths.

```
0xD0 B3 00 0001
0xD0 B3 00 0002
0xD0 B3 00 0003
```

Specific Path

To resume a specific path (for example, path 1), use the Resume Motion command with the Path ID set to 1.

```
0xB3 0001
```

The HLC sends a Command Status message that identifies the path and shows the command status. This example shows that the Resume Motion command was accepted for path 1.

```
0xD0 B3 00 0001
```

Warm Reset

Use the Warm Reset command to reset the motors on one or more specified paths without interrupting network communications for the motors. Reset all motors on all paths when a new Node Controller Configuration File has been uploaded to the node controllers. Path status can be monitored as described in Path Status on page 41.

The following actions occur during a warm reset:

- Restarts the runtime code on the motors.
- Deletes the vehicle records for vehicles on the specified paths.

NOTE: This command is not supported on QuickStick 100 transport systems. If you send a Warm Reset to a QS 100, it will perform a regular Reset Command.

All Paths

To perform a warm reset of all paths, use the Warm Reset command with the Path ID set to 0.

```
0xBD 0000
```

The HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the Warm Reset command was accepted for all paths.

```
0xD0 BD 00 0001
0xD0 BD 00 0002
0xD0 BD 00 0003
```

Once the Warm Reset command completes, the HLC sends a Command Status message for each path (each line identifies a path and shows the command status). This example shows that the command completed successfully for all paths.

```
0xD0 BD 80 0001
0xD0 BD 80 0002
0xD0 BD 80 0003
```

Specific Path

To perform a warm reset of a specific path (for example, path 1), use the Warm Reset command with the Path ID set to that path (for example, 0x0001).

```
0xBD 0001
```

The HLC sends a Command Status message that identifies the path and shows the command status. This example shows that the command was accepted for path 1.

```
0xD0 BD 00 0001
```

Once the Warm Reset completes, the HLC sends a Command Status message that identifies the path. This example shows that the command completed successfully for path 1.

```
0xD0 BD 80 0001
```

Station Control

Use the stations commands to create or change station locations without changing the Node Controller Configuration File. Station status can be monitored as described in Station Status on page 46.

IMPORTANT	Dynamically created stations or changes to existing stations are not maintained after an HLC restart or after loading and activating a Node Controller Configuration File.

Create Stations

To create a station, use the Create Station command. Specifying a specific station creates a station with that ID. Specify a station ID of 0 to have the HLC assign a station ID to the new station. The example that is shown creates a station 0.5 m from the beginning of path 2 and has the HLC assign a station ID.

```
0xBF 01 01 0002 0000003F 0000
```

The HLC sends a Command Status message that acknowledges the receipt of the Create Station command. The status message shows that the station was created successfully 0.5 m from the beginning of path 2 and assigned an ID of 4.

```
0xD0 BF 00 01 01 0002 0000003F 0004
```

Move Stations

To move an existing station, use the Create Station command and specify the station to move and its new location. The example that is shown moves station 1 from its current position to 1.25 m from the beginning of path 3.

```
0xBF 01 01 0003 0000A03F 0001
```

The HLC sends a Command Status message that acknowledges the receipt of the Create Station command. The status message shows station 1 was moved successfully to 1.25 m from the beginning of path 3.

```
0xD0 BF 00 01 01 0003 0000A03F 0001
```

Delete Stations

To delete an existing station, use the Delete Station command. Specifying a specific station deletes that station. Specify a station ID of 0 to have the HLC delete all stations. The example that is shown deletes all stations.

```
0xBF 01 02 0000
```

The HLC sends a Command Status message that acknowledges the receipt of the Delete Station command (each line identifies a station and shows the command status). This example shows that the command to delete all stations was accepted.

```
0xD0 BF 00 01 02 0001
0xD0 BF 00 01 02 0002
```

Traffic Light Control

Use the traffic lights commands to create or change traffic light settings and locations. Traffic light status can be monitored as described in Traffic Light Status on page 46. There is a limit of one traffic light per motor block and 32 traffic lights per path.

NOTE: Dynamically created traffic lights or changes to existing traffic lights are not maintained after an HLC restart.

Create Traffic Lights

To create a traffic light, use the Create Traffic Light command. Traffic lights are assigned the next available Traffic Light ID and set to green when created to allow vehicles to continue to move beyond the traffic light position. The example that is shown creates a traffic light 1.5 m from the beginning of path 3.

```
0xBF 02 01 0003 0000C03F 0000000A
```

The HLC sends a Command Status message that acknowledges the receipt of the Create Traffic Light command. This example shows that the command for a new traffic light to be created 1.5 m from the beginning of path 3 and assigned an ID of 7 was accepted.

```
0xD0 BF 00 02 01 0003 0000C03F 0000000A 0007
```

Once the Create Traffic Light command completes, the HLC sends a Command Status message that shows the completion status of the command. This example shows that the command to create traffic light 7 1.5 m from the beginning of path 3 completed successfully.

```
0xD0 BF 80 02 01 0003 0000C03F 0000000A 0007
```

Set Traffic Lights

To change the color of an existing traffic light, use the Set Traffic Light command and specify the traffic light and its new color. The example that is shown changes traffic light 1 to red.

```
0xBF 02 02 0001 01 0000000D
```

The HLC sends a Command Status message that acknowledges the receipt of the Set Traffic Light command. This example shows that the command to change the color of traffic light 1 to red was accepted.

```
0xD0 BF 00 02 02 0001 01 0000000D
```

Once the Set Traffic Light command completes, the HLC sends a Command Status message that shows the completion status of the command. This example shows that the command to set traffic light 1 to red completed successfully.

```
0xD0 BF 80 02 02 0001 01 0000000D
```

Delete Traffic Lights

To delete a traffic light, use the Delete Traffic Light command and specify the traffic light to delete. The example that is shown deletes traffic light 3.

```
0xBF 02 04 0003 00000016
```

The HLC sends a Command Status message that acknowledges the receipt of the Delete Traffic Light command. This example shows that the command to delete traffic light 3 was accepted.

```
0xD0 BF 00 02 04 0003 00000016
```

Once the Delete Traffic Light command completes, the HLC sends a Command Status message that shows the completion status of the command. This example shows that the command to delete traffic light 3 completed successfully.

```
0xD0 BF 80 02 04 0003 00000016
```

Motor Controller Control

Use the Motor Inverter Command to control the state of the inverters in specific QSHT motors. The inverters in a motor can be disabled to prevent any commanded motion by that motor until the inverters are enabled. The inverters in the motor are responsible for controlling the coils in a motor block, which moves the vehicle. Inverter status can be monitored as described in Motor Status on page 44.

NOTE: This command is only supported on QuickStick HT transport systems.

Disable Motor Controllers

To disable an inverter in a QSHT motor, use the Motor Inverter Command and specify the motor, the inverter within the motor, and its state (disable). The example that is shown disables inverter 1 in motor 1 on path 5.

```
0xBF 0B 01 0005 0001 0001 01 00000001
```

The HLC sends a Command Status message that acknowledges the receipt of the Motor Inverter Control command. This example shows that the command to disable inverter 1 in motor 1 on path 5 was accepted.

```
0xD0 BF 00 0B 01 0005 0001 0001 01 00000001
```

Once the Motor Inverter Control command completes, the HLC sends a Command Status message that shows the completion status of the command. This example shows that the command to disable the specified inverter completed successfully.

```
0xD0 BF 80 0B 01 0005 0001 0001 01 00000001
```

Enable Motor Controllers

To enable an inverter in a QSHT motor, use the Motor Inverter Command and specify the motor, the inverter within the motor, and its state (enable). The example that shown enables inverter 1 in motor 1 on path 5.

```
0xBF 0B 01 0005 0001 0001 00 00000002
```

The HLC sends a Command Status message that acknowledges the receipt of the Motor Inverter Control command. This example shows that the command to enable inverter 1 in motor 1 on path 5 was accepted.

```
0xD0 BF 00 0B 01 0005 0001 0001 00 00000002
```

Once the Motor Inverter Control command completes, the HLC sends a Command Status message that shows the completion status of the command. This example shows that the command to enable the specified inverter completed successfully.

```
0xD0 BF 80 0B 01 0005 0001 0001 00 00000002
```

Propulsion Power Supply Control

Use the Set Propulsion Power State command to control the state of specific propulsion power supplies connected to QSHT 5700 Inverters. Propulsion Power Supply status can be monitored as described in Propulsion Power Supply Status on page 47.

NOTE: This command is only supported on QSHT transport systems using the QSHT 5700 Inverters.

Set Propulsion Power Supply

To change the state of an existing Kinetix 2198-Pxxx DC-bus propulsion power supply, use the Set Propulsion Power State command. Specify the type of power supply, its ID, and its new state. The example that is shown sets the state of power supply 1, which is a DFE, to running.

```
0xBF 0D 02 0001 01 04 00000001
```

The HLC sends a Command Status message that acknowledges the receipt of the Set Propulsion Power Supply State command. This example shows that the command to change the state of power supply 1 to running was accepted.

```
0xD0 BF 00 0D 02 0001 01 04 00000001
```

Once the Set Propulsion Power Supply State command completes, the HLC sends a Command Status message that shows the completion status of the command. This example shows that the command to set power supply 1 to running completed successfully.

```
0xD0 BF 80 0D 02 0001 01 04 00000001
```

Node Controller Digital I/O Control

Use the Set Node Controller Digital I/O Outputs command to set the digital I/O outputs on a specific node controller. Specify an output data mask to identify output bits to be masked (not changed by this command) by entering the mask as a hexadecimal value (mask bit set low). Specify the output data as a hexadecimal value (only those bits enabled by the output data mask are output). Digital I/O status can be monitored as described in Node Controller Status on page 40.

To set bits 04…07 of the digital outputs on a node controller (NC 1), use the Set Node Controller Digital I/O Outputs command. Set the node controller to 1, the output data mask to 0x000000F0, and the output data to 0x000090A0 (the node controller provides 16 bits of digital I/O so only 2 bytes are used).

```
0xBF 07 02 0001 000000F0 000090A0
```

The HLC sends a Command Status message that acknowledges the receipt of the Digital Output command, which shows the node controller ID, output data mask, and output data.

```
0xD0 BF 00 07 02 0001 000000F0 000090A0
```

Monitoring Transport System Status

This section describes how to monitor the components of the transport system. All examples provided reference Figure 2-1.

1. Connect to the high-level controller for the transport system (see Connecting to the Transport System on page 25).

2. Use the host controller to send a Status Request command to the HLC formatted for the type of status being requested.

High-Level Controller

High-Level Controller Status

To check the high-level controller status, use the Status Request command with the Request Type set to 0 (HLC Status). When requesting the status of the HLC, the ID and Motor ID do not need to be included (there is only one HLC and motor status is not being requested).

```
0xB5 00
```

The HLC sends a High-Level Controller Status message that contains the HLC status. This example shows that the Status Request for HLC status was accepted and shows that it is present, operational, and that EtherNet/IP™ communication is not being used.

```
0xD1 03 00
```

Extended High-Level Controller Status

To check the extended high-level controller status, use the Get Extended High-Level Controller Status command.

```
0xBF 0C 04
```

The HLC sends an Extended High-Level Controller Status message that contains the extended HLC status. This example shows that the Status Request for Extended HLC Status was accepted. It shows that the HLC is operational, provides the software version, shows that managed configurations are not being used, the Node Controller Configuration File is valid, and that EtherNet/IP communication is not being used.

```
0xDF 0C 04 03 0F 0B 0B 00 01 00
```

Node Controllers

Node Controller Status

To check the status of node controllers, use the Status Request command with the Request Type set to 1 (Node Controller Status). For a specific node controller, set the ID field to the ID of that node controller. For all node controllers, set the ID field to 0. When requesting the status of node controllers, the Motor ID does not need to be included (motor status is not being requested).

```
0xB5 01 0000
```

The HLC sends a Node Controller Status message for each requested node controller (each line provides the status for a node controller that is defined in the Node Controller Configuration File). This example shows that the Status Request for node controller status was accepted and shows that all node controllers are present and operational.

```
0xD2 0001 01 03
0xD2 0002 01 03
0xD2 0003 01 03
```

Extended Node Controller Status

To check the extended status of node controllers, use the Get Extended Node Controller Status command. For a specific node controller, set the Node Controller ID field to the ID of that node controller. For all node controllers, set the Node Controller ID field to 0.

```
0xBF 0C 03 0000
```

The HLC sends an Extended Node Controller Status message that contains the extended HLC status for each requested node controller (each line provides the status for an NC that is defined in the Node Controller Configuration File). This example shows that the Status Request for Extended Node Controller Status was accepted and shows that it is operational, provides the software version, shows that managed configurations are not being used, and the Node Controller Configuration File is valid.

```
0xDF 0C 03 00 0001 03 0F 0B 0B 00 01
0xDF 0C 03 00 0002 03 0F 0B 0B 00 01
```

Node Status

All Nodes

To check the status of all nodes, use the Status Request command with the Request Type set to 2 (Node Status) and the ID set to 0.

```
0xB5 02 0000
```

The HLC sends a Node Status message for each node (each line provides the status and additional information for a node that is defined in the Node Controller Configuration File). This example shows that the Status Request for node status was accepted, all nodes are present, the type of the nodes, there are no vehicles in the nodes, and the switches are in their last requested position and operational.

```
0xD3 0001 01 01 0000 01 01 03
0xD3 0002 01 02 0000 02 02 03
```

Specific Node

To check the status of a specific node (for example, node 1), use the Status Request command with the Request Type set to 2 (Node Status) and the ID set to 1.

```
0xB5 02 0001
```

The HLC sends a Node Status message that identifies the node and shows its status and additional node-specific information. This example shows that the Status Request for node status was accepted, the node is present, it is a Merge Node, there is no vehicle in the node, and the switch is in its last requested position (straight) and operational.

```
0xD3 0001 01 01 0000 01 01 03
```

Path Status

All Paths

To check the status of all paths, use the Status Request command with the Request Type set to 3 (Path Status) and the ID set to 0.

```
0xB5 03 0000
```

The HLC sends a Path Status message for each path (each line provides the status for a path that is defined in the Node Controller Configuration File). This example shows that the Status Request for path status was accepted, all paths are present and operational, the number of vehicles on each path, and all communications are good.

```
0xD4 0001 01 02 02 00 00 00
0xD4 0002 01 00 02 00 00 00
0xD4 0003 01 02 02 00 00 00
```

Specific Path

To check the status of a specific path (for example, path 1), use the Status Request command with the Request Type set to 3 (Path Status) and the ID set to 1.

```
0xB5 03 0001
```

The HLC sends a Path Status message that identifies the path and shows its status. This example shows that the Status Request for path status was accepted, path 1 is present and operational, there are two vehicles on the path, and all communications are good.

```
0xD4 0001 01 02 02 00 00 00
```

Vehicle Status

Basic Vehicle Status

To check the basic status of all vehicles, use the Status Request command with the Request Type set to 4 (Vehicle Status) and the Vehicle ID set to 0.

```
0xB5 04 0000
```

The HLC sends a Vehicle Status message for each vehicle (each line provides the status for a vehicle on the transport system). This example shows that the Status Request for vehicle status was accepted, there are four vehicles present (four responses). The responses show the paths that the vehicles are on, that they are not in motion, their position on the paths, and that they are detected on the motor.

```
0xD5 0001 01 0003 0000 6666663F 00000000 00 01 00000000
0xD5 0002 01 0001 0000 00002C40 00000000 00 01 00000000
0xD5 0003 01 0003 0000 0000003F 00000000 00 01 00000000
0xD5 0004 01 0001 0001 0000403F 00000000 00 01 00000000
```

To check the basic status of a specific vehicle, use the Status Request command with the Request Type set to 4 (Vehicle Status) and the ID set to that vehicle (for example, vehicle 1).

```
0xB5 04 0001
```

The HLC sends a Vehicle Status message that identifies the vehicle and shows its status. This example shows that the Status Request for vehicle status was accepted, vehicle 1 is present on path 3, it is not in motion, its position on the path, and it is detected on the motor.

```
0xD5 0001 01 0003 0000 6666663F 00000000 00 01 00000000
```

Extended Vehicle Status

The extended vehicle status returns the same status information as the Vehicle Status message and also includes additional status information. To check additional vehicle status information for all vehicles, use the Get Extended Vehicle Status command with the Vehicle ID set to 0.

```
0xBF 03 06 0000
```

The HLC sends an Extended Vehicle Status, message for each vehicle (each line provides the status for one vehicle on the transport system). This example shows that the Extended Vehicle Status Request command was accepted, there are four vehicles present (four responses), the paths they are on, their current destination, their position on the paths, and they are not in motion. Vehicles 1, 2, and 4 have no current command, they are detected on the motor, they have no current destinations, they are not following another vehicle, and they are using PID set 0. Vehicle 3 has a current command to follow vehicle 4, it is detected on the motor, it has no current destination, and it is using PID set 0.

```
DF 03 06 0001 01 0003 0003 D2FFFF3E 00000000 00 0001
00000000 D2FFFF3E 0000 E23E323D 0000 00 00 5BE47F3F
A6FDFF3E 00000000

DF 03 06 0002 01 0002 0000 FDFFBF3F 00000000 00 0001
00000000 FDFFBF3F 0000 E23E323D 0000 00 00 00000000
00000000 00000000

DF 03 06 0003 01 0001 0000 FDFFBF3F 00000000 B7 0601
8399993E FDFFBF3F 0004 E23E323D 0000 00 00 5BE47F3F
00000000 00000000

DF 03 06 0004 01 0001 0001 5E66E63F 00000000 00 0001
00000000 5E66E63F 0000 E23E323D 0000 00 00 5BE47F3F
00000000 00000000
```

To check additional vehicle status information for a specific vehicle, use the Get Extended Vehicle Status command with the Vehicle ID set to that vehicle (for example, vehicle 1).

```
0xBF 03 06 0001
```

The HLC sends an Extended Vehicle Status, message for the vehicle. This example shows that the Extended Vehicle Status Request command was accepted, for the vehicle, the path it is on, its current destination, its position on the path, and that it is not in motion. The vehicle has no current command, it is detected on the motor, it has no current destination, it is not following another vehicle, and it is using PID set 0.

```
0xDF 03 06 0001 01 0003 0003 D2FFFF3E 00000000 00
0001 00000000 D2FFFF3E 0000 E23E323D 0000 00 00
5BE47F3F A6FDFF3E 00000000
```

Motor Status

All Motor Status

To check the status of all motors on all paths, use the Status Request command with the Request Type set to 5 (Motor Status), the Path ID set to 0 (all paths), and the Motor ID set to 0 (all motors).

```
0xB5 05 0000 00
```

The HLC sends a Motor Status for each motor (each line provides the status and additional information for a motor that is defined in the Node Controller Configuration File). This example, for a MagneMover LITE system, shows that the Status Request for motor status was accepted. It shows the path that each motor is on, the path status, the motor position on the path, motor presence, motor type, and motor fault data. The fault data includes; number of blocks/driver boards in the motor, any scheduler errors, communication status, motor status, and so on. Only the first and last motor responses are shown to save space.

```
0xD7 0001 01 01 01 02 04 00 00 00 00 00 00 00 00 00 00
0xD7 0001 01 02 01 02 04 00 00 00 00 00 00 00 00 00 00
.
.
.
0xD7 0004 01 04 01 02 04 00 00 00 00 00 00 00 00 00 00
```

Specific Motor Status

To check the status of a specific motor (for example, motor 1 on path 1), use the Status Request command with the Request Type set to 5 (Motor Status), the Path ID set to 1, and the Motor ID set to 1.

```
0xB5 05 0001 01
```

The HLC sends a Motor Status message that identifies the motor and shows its status.

This example, for a MagneMover LITE system, shows that the request for motor status was accepted, the path that the motor is on, the path exists, the motor is the first motor on the path, and the motor is present. It shows it is a MagneMover LITE motor, there is one driver board, no scheduler errors, upstream and downstream communications are good, overall motor status is good, there are no motor controller faults, and there are eight driver boards with no faults (1 m motor).

```
0xD7 0001 01 01 01 02 04 00 00 00 00 00 00 00 00 … 00 00
```

This example, for a QuickStick system, shows that the request for motor status was accepted, the path that the motor is on, the path exists, the motor is the first motor on the path, and the motor is present. It shows it is a QuickStick motor, there are five blocks (500 mm motor), no scheduler errors, upstream and downstream communications are good, overall motor status is good, and there are no block faults on any of the blocks.

For QS 100:

```
0xD7 0002 01 01 01 01 05 00 00 00 00 00 00 … 00 00
```

For QS 150:

```
0xD7 0002 01 01 01 05 05 00 00 00 00 00 00 … 00 00
```

This example, for a QSHT system, shows that the request for motor status was accepted, the path that the motor is on, the path exists, the motor is the first motor on the path, and the motor is present. It shows it is a QuickStick HT motor, there are two blocks (1 m motor), no scheduler errors, upstream and downstream communications are good, overall motor status is good, and there are no motor controller, HES, or inverter faults (1 m motor) on either block, and no Ethernet comm faults. There are no safety faults or digital inputs since the QSHT controller does not support them.

```
0xD7 0001 01 01 01 03 02 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00000000 00 00 00
```

This example, for a QSHT 5700 system, shows that the request for motor status was accepted, the path that the motor is on, the path exists, the motor is the first motor on the path, and the motor is present. It shows it is a QuickStick HT motor, there are two blocks (1 m motor), no scheduler errors, upstream and downstream communications are good, overall motor status is good, and there are no motor controller, HES, or inverter faults (1 m motor) on either block, no Ethernet comm faults, no safety faults, or active digital inputs.

```
0xD7 0002 01 01 01 03 02 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00000000 00 00 00
```

Only Motors With A Fault

To identify those motors with one or more active faults, use the Status Request command with the Request Type set to 5 (Motor Status), the Path ID set to 0 (all paths), and the Motor ID set to 255 (only motors with faults).

```
0xB5 05 0000 FF
```

The HLC sends a Motor Status *message only for each motor with a fault condition (each line provides the status and additional motor information for a motor that is defined in the Node Controller Configuration File). This example shows that the request for motor status was accepted, the path that each motor in a fault condition is on, and the path status. It also provides the motor position on the path, motor presence, motor type (MagneMover LITE), number of driver boards, any scheduler* *errors, upstream and downstream communication status, motor status that shows motion is suspended by the node controller, and the motor fault data. Only the responses from the first and last motors on the path are shown to save space.*

```
0xD7 0001 01 01 01 02 01 00 00 00 08 000000000000
.
.
.
0xD7 0004 01 04 01 02 01 00 00 00 08 000000000000
```

Station Status

To check the status of all stations, use the Get Station Status command with the ID set to 0. Stations are defined using either the Create Station command or in the Node Controller Configuration File.

```
0xBF 01 03 0000
```

The HLC sends a Station Status message for each station. This example shows that the Station Status request was accepted, there are two stations defined (see Figure 2-1 on page 27), station 1 is at 2.25 m on path 3 and station 2 is at 1.25 m on path 1.

```
0xDF 01 01 00 0001 0003 00001040
0xDF 01 01 00 0002 0001 0000A03F
```

Traffic Light Status

To check the status of all traffic lights, use the Get Traffic Light Status command with the ID set to 0. Traffic lights are defined using the Create Traffic Light and Set Traffic Light commands as shown in Traffic Light Control on page 35.

```
0xBF 02 03 0000
```

The HLC sends a Traffic Light Status message for each traffic light. This example shows that the Traffic Light Status request was accepted. It shows that there are two traffic lights defined, traffic light 1 is 1.5 m from the start of path 1 and set to red and traffic light 2 is 1.0 m from the start of path 3 and set to green.

```
0xDF 02 01 00 0001 0001 0000C03F 01
0xDF 02 01 00 0002 0003 0000803F 00
```

Motor Controller (Inverter) Status

NOTE: This function is only supported on QuickStick HT transport systems.

To check the status of any motor controller, request the appropriate motor index from the Motor Status response. The QSHT Motor Fault data provides the status of the following inverter disable faults; Inverter disabled by command (QSHT and QSHT 5700), Inverter disabled by power supply not ready (QSHT 5700).

Motor controllers can be controlled as described in Motor Controller Control on page 36.

Moving Path Node Status

NOTE: This function is not supported on MagneMover LITE transport systems.

To check the status of the paths in any of the Moving Path nodes, request the appropriate node ID using the MP Get Path End Status command. Moving Path nodes can be controlled as described in Moving Path Node on page 114.

This example references the Moving Path node that is shown in Figure 3-45 on page 124.

```
0xBF 09 01 0003
```

The HLC sends an MP Path End Status Report message for each path in all Moving Path nodes showing there are four paths in one Moving Path node (four responses). Each line provides the status and additional information for a path that is connected to the Moving Path node as defined in the Node Controller Configuration File. This example shows that the Moving Path Status Request was accepted. It shows that it is node 3, all four paths are present, the current command count for each path, link information for each path end, vehicle information for each path, and the command counter.

```
0xDF 09 01 00 0003 0001 0000000A 04 02 01 0000 0000 0003
0008 0000 0000 0000 0000 0000
0xDF 09 01 00 0003 0002 00000000 01 01 01 0000 0000 0000
0000 0000 0000 0000 0000 0000
0xDF 09 01 00 0003 0003 00000009 06 03 02 0004 0000 0001
0000 0000 0000 0000 0000 0000
0xDF 09 01 00 0003 0004 00000000 01 01 02 0004 0000 0000
0000 0000 0000 0000 0000 0000
```

Propulsion Power Supply Status

NOTE: This function is only supported on QSHT 5700 transport systems.

To check the status of any propulsion power supply, request the appropriate Power Supply ID using the Get Propulsion Power Status command. To check the status of all propulsion power supplies in a QSHT 5700 transport system, use the Get Propulsion Power Status command with the ID set to 0. Propulsion Power Supplies can be controlled as described in Propulsion Power Supply Control on page 37.

```
0xBF 0D 01 0000
```

The HLC sends a Propulsion Power Supply Status message for each power supply. This example shows that the Propulsion Power Supply Status request was accepted, there are two power supplies defined, they are each a DFE type, and they are both in the precharge state.

```
0xDF 0D 01 00 0001 01 01
0xDF 0D 01 00 0002 01 01
```

Node Controller Digital I/O Status

To check the status of the node controller digital I/O, request the appropriate node controller ID using the Node Controller Digital I/O Status command. The digital I/O is controlled as shown in Node Controller Digital I/O Control on page 38.

NOTE: This function is not supported on MagneMover LITE transport systems.

Asynchronous Notification of Input Bit Changes

To have a state change on any of the Digital Inputs of a node controller that supports digital I/O generate a notification use Set Node Controller Digital I/O Notification Mask. Set the node controller to the specific NC (for example, NC 1) and the Notification Mask to 0x0000FFFF (the node controller provides 16 bits of digital I/O so only 2 bytes are used).

```
0xBF 07 03 0001 0000FFFF
```

The HLC sends a Command Status *message that acknowledges the receipt of the Set Node Controller Digital I/O Notification Mask command that identifies the node controller and shows its Input Notification Mask.*

```
0xD0 BF 00 07 03 0001 0000FFFF
```

Whenever any digital I/O input on node controller 1 changes state and its corresponding Notification Mask bit is set, the HLC sends a Node Controller Digital I/O Status message to the host controller.

This example shows that the input data on node controller 1 has changed, node controller 1 has 16 bits of digital I/O and the current input data, output data, and the Notification Mask setting.

```
0xDF 07 01 00 0001 16 16 00000C59 000090A0 0000FFFF
```

Read Digital Inputs On All Node Controllers

To check the status of the digital I/O on all node controllers, use the Get Node Controller Digital I/O Status command with the Node Controller ID set to 0.

```
0xBF 07 01 0000
```

The HLC sends a Node Controller Digital I/O Status *message for each node controller (each line identifies a node controller that is defined in the Node Controller Configuration File and shows its digital I/O status). This example shows that the Get Node Controller Digital I/O Status command was accepted. It shows that node controller 1 has 16 digital I/O inputs and 16 digital I/O outputs and the current input data, output data, and the Notification Mask setting, and that node controller 2 does not have digital I/O hardware.*

```
0xDF 07 01 00 0001 16 16 0000AAAA 00005555 0000FFFF
0xDF 07 01 00 0002 00 00 00000000 00000000 00000000
```

Read Digital Inputs On A Specific Node Controller

To check the status of the digital I/O on a specific node controller (for example, NC 1), use the Get Node Controller Digital I/O Status command with the Node Controller ID set to 0.

```
0xBF 07 01 0001
```

The HLC sends a Node Controller Digital I/O Status *message that identifies the node controller and shows its digital I/O status). This example shows that the* Get Node Controller Digital I/O Status *command was accepted, node controller 1 has 16 digital I/O inputs and 16 digital I/O outputs and the current input data, output data, and the Notification Mask setting.*

```
0xDF 07 01 00 0001 16 16 0000AAAA 00005555 0000FFFF
```

Moving Vehicles

This section describes how to move vehicles in the transport system. To move a vehicle, a destination must be provided as either a position on a path or as a predefined station.

NOTE: If the acceleration or velocity values in a motion command are higher than the limit set in the Node Controller Configuration File, the command is rejected.

If the velocity values are higher than the values for a specific motor, the velocity value for the motor is used while the vehicle is on that motor.

If a motion command is issued to a vehicle already in motion, and the command has a lower acceleration than the previous command to that vehicle the command is rejected.

PINCH/CRUSH HAZARD: Moving mechanisms (vehicles) have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

Direction of Motion

Vehicles can be moved either forward or backward on the transport system guideway. The direction of motion for a motion command is typically specified to make sure that the vehicle moves as expected. When bidirectional is specified as the direction for vehicle motion it only applies to the initial selection of direction by the transport system, the vehicle does not change direction during its move.

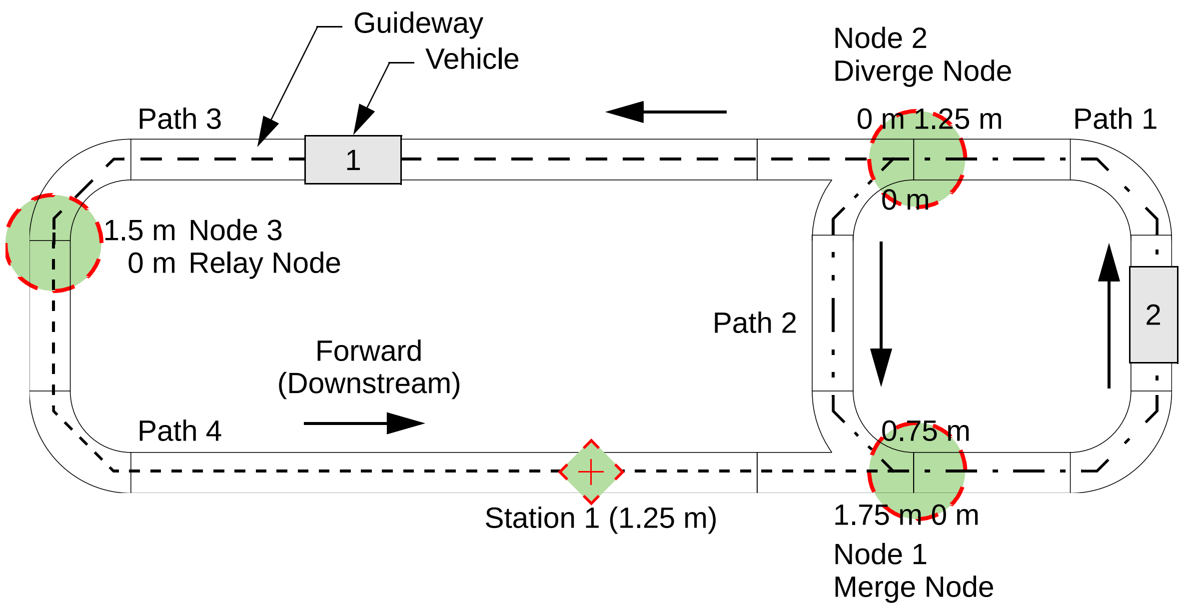


Figure 2-2: Transport System for Motion Examples

Forward – Vehicles move downstream only, useful to implement a unidirectional loop. If the destination is not reachable (for example, the paths do not form a loop), the command is rejected with a command status of 0x41 and the vehicle does not move.

Examples:

A. For vehicle 1 in Figure 2-2 to reach the beginning of path 3 it moves from path 3 to path 4, from path 4 to path 1, then from path 1 to path 3. Vehicle 2 must move out of the way for vehicle 1 to complete its motion.

B. For vehicle 1 in Figure 2-2 to reach the beginning of path 2 it moves from path 3 to path 4, from path 4 to path 1, then from path 1 to path 2. Vehicle 2 must move out of the way for vehicle 1 to complete its motion.

Backward – Vehicles move upstream only, useful to implement a unidirectional loop in the backwards direction. If the destination is not reachable (for example, the paths do not form a loop), the command is rejected with a command status of 0x41 and the vehicle does not move.

Examples:

A. For vehicle 1 in Figure 2-2 to reach the beginning of path 3 it moves directly to that position.

B. For vehicle 1 in Figure 2-2 to reach the beginning of path 2 it moves from path 3 to path 1, from path 1 to path 2, then to the beginning of path 2. Vehicle 2 must move out of the way for vehicle 1 to complete its motion.

Bidirectional – Vehicles can move in either direction as required to get to the destination in the shortest distance. Once vehicle motion is initiated, the vehicle continues in the initial direction. If the destination is on a path other than the path where the vehicle is located, the forward direction takes precedence for a transport system that is a closed-loop.

Examples:

A. For vehicle 1 in Figure 2-2 to reach the beginning of path 3 it moves backwards directly to that position.

B. For vehicle 1 in Figure 2-2 to reach the beginning of path 2 it moves forward from path 3 to path 4, from path 4 to path 1, then from path 1 to path 2. Vehicle 2 must move out of the way for vehicle 1 to complete its motion. Once the vehicle starts moving in a specific direction, it does not change direction.

Move to Position

Move the specified vehicle to a specified position, relative to the start of the specified path. The path must be previously defined in the Node Controller Configuration File.

To move a vehicle to a position on a path, use the Move Vehicle To Position command with an Order Number (required by all motion commands for tracking). Specify the Vehicle ID (for example, vehicle 1), Flags and Direction for the move (for example, Forward using PID set 0 “unloaded” values), the destination position (measured from the start of the path), the Path ID, the Acceleration, and the Velocity.

`0xB1 00000002 0001 01 0000003E 0004 0000803F 0000803F`
(move #2, vehicle 1, unloaded and forward, position 0.125 m, path 4, acceleration 1.0 m/s^2, velocity 1.0 m/s)

The HLC sends a Command Status message that acknowledges the receipt of the motion command that shows the Order Number, Vehicle ID, Position, Path ID, Acceleration, Velocity, and Flags and Direction.

`0xD0 B1 00 00000002 0001 0000003E 0004 0000803F 0000803F 01`
(command accepted, move #2, vehicle 1, position 0.125 m, path 4, acceleration 1.0 m/s^2, velocity 1.0 m/s, unloaded and forward)

Once vehicle motion completes, the HLC sends a Command Status message that shows completion of the motion command and includes the Order Number, Vehicle ID, Current Position, Path ID, current Acceleration, current Velocity, and Flags and Direction.

`0xD0 B1 80 00000002 0001 2EA5EE3D 0004 00000000 00000000 01`
(command completed successfully, move #2, vehicle 1, position 0.116 m, path 4, acceleration 0 m/s^2, velocity 0 m/s, unloaded and forward)

Move to Station

Move the specified vehicle to a specified station. The station and the path it is on must be previously defined in the Node Controller Configuration File.

To move a vehicle to a station, use the Move Vehicle To Station command with an Order Number (required by all motion commands for tracking). Specify the Vehicle ID for the specific vehicle (for example, vehicle 1), Flags and Direction set for the move (for example, bidirectional using PID set 0 “unloaded” values), the destination Station ID, the Acceleration, and the Velocity.

`0xB0 00000001 0001 00 0001 0000803F 0000803F`
(move #1, vehicle 1, unloaded and bidirectional, station 1, acceleration 1.0 m/s^2, velocity 1.0 m/s)

The HLC sends a Command Status *message that acknowledges the receipt of the motion command that shows the Order Number, Vehicle ID, Station ID, Path ID, Acceleration, Velocity, and Flags and Direction.*

`0xD0 B0 00 00000001 0001 0001 0004 0000803F 0000803F 00`
(command accepted, move #1, vehicle 1, station 1, path 4, acceleration 1.0 m/s^2, velocity 1.0 m/s, unloaded and bidirectional)

Once vehicle motion completes, the HLC sends a Command Status *message that shows completion of the motion command that shows the Order Number, Vehicle ID, Station ID, Path ID, Acceleration, Velocity, and Flags and Direction.*

`0xD0 B0 80 00000001 0001 0001 0004 00000000 00000000 00`
(command completed successfully, move #1, vehicle 1, station 1, path 4, acceleration 0 m/s^2, velocity 0 m/s, unloaded and bidirectional)

Move to Station Using an Offset

Move the specified vehicle to a specified station with an offset from that station. The station and the path it is on must be previously defined in the Node Controller Configuration File. Figure 2-3 shows station 1 and the station offset.

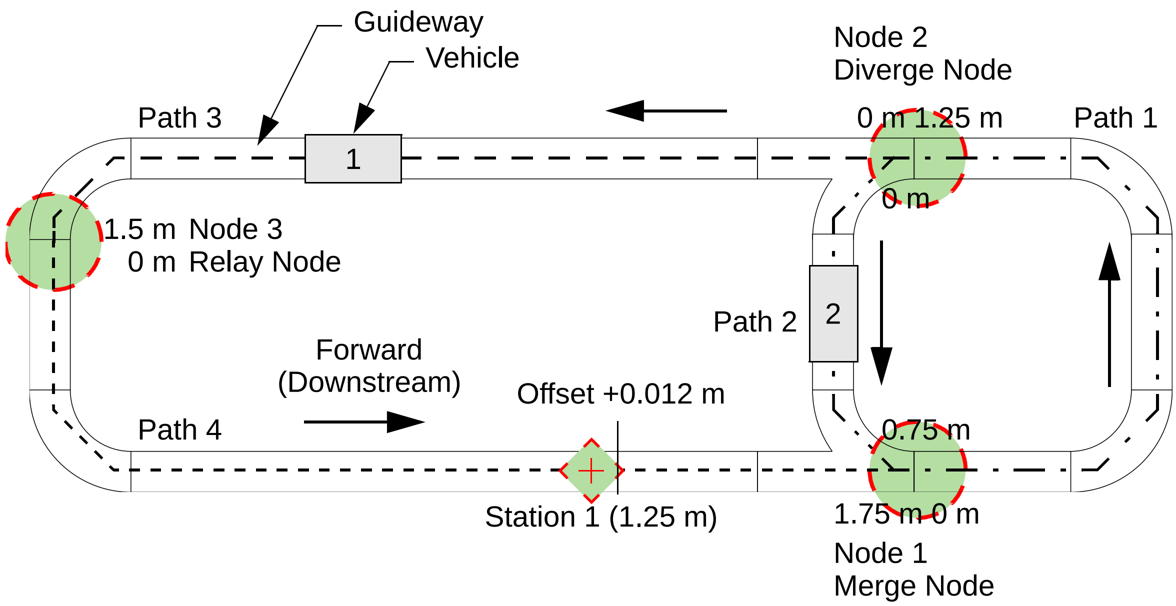


Figure 2-3: Move to Station Using an Offset

To move a vehicle to a station with an offset from the station location, use the Move Vehicle To Station command with an Order Number (required by all motion commands for tracking). Specify the Vehicle ID (for example, vehicle 1), Flags and Direction for the move (for example, bidirectional using PID set 0 “unloaded” values), the destination Station ID, the Acceleration, the Velocity, and the Station Offset.

```
0xB0 00000001 0001 00 0001 0000803F 0000803F A69B443C
```

(move #1, vehicle 1, unloaded and bidirectional, station 1, acceleration 1.0 m/s^2, velocity 1.0 m/s, offset +0.012)

The HLC sends a Command Status message that acknowledges the receipt of the motion command that shows the Order Number, Vehicle ID, Station ID, Path ID, Acceleration, Velocity, Flags and Direction, and Offset.

```
0xD0 B0 00 00000001 0001 0001 0004 0000803F 0000803F 00 A69B443C
```

(command accepted, move #1, vehicle 1, station 1, path 4, acceleration 1.0 m/s^2, velocity 1.0 m/s, unloaded and bidirectional, offset +0.012)

Once vehicle motion completes, the HLC sends a Command Status message that shows completion of the motion command and includes the Order Number, Vehicle ID, Station ID, Path ID, Acceleration, Velocity, and Flags and Direction.

```
0xD0 B0 80 00000001 0001 0001 0004 00000000 00000000 00 A69B443C
```

(command completed successfully, move # 1, vehicle 1, station 1, path 4, acceleration 0 m/s^2, velocity 0 m/s, unloaded and bidirectional, offset +0.012)

Platooning

Platooning provides support to move a vehicle to a position relative to another vehicle and couple to it to that vehicle to create a platoon. The lead vehicle in the platoon is moved, which causes all vehicles in the platoon to follow it without brick-wall headway allowing close spacing between the vehicles. Once there is no longer a need for the platoon, the vehicles are decoupled from it. Platoons can also be created where a vehicle within the platoon (either one vehicle or the lead vehicle for a new platoon) can automatically decouple from the platoon at a specified location.

Create a Platoon without Auto-decouple

To create a platoon, move the following vehicle into position behind the lead vehicle using a Move Vehicle To Position command. The following example (see Figure 2-2 for the system layout) shows the following vehicle being positioned. The command is accepted, and the command is completed once the vehicle stops moving.

```
0xB1 00000001 0002 01 CDCC4C3E 0003 0000803F 0000003F
```

(move #1, vehicle 2, unloaded and forward, position 0.8 m, path 3, acceleration 1.0 m/s^2, velocity 0.5 m/s)

```
0xD0 B1 00 00000001 0002 CDCC4C3E 0003 0000803F 0000003F
01
0xD0 B1 80 00000001 0002 DF614B3F 0003 5BE47F3F A6FDFF3E
01
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was received (0x00) and that it completed (0x80).

Couple the following vehicle to the lead vehicle using a Vehicle Follow Order command. The following example shows the following vehicle being coupled to the lead vehicle without automatic decoupling. The command is accepted, and the command is completed once the vehicles are coupled.

```
0xB7 00000002 0002 01 CDCC4C3E 0001 00 0000803F 0000003F
0000 00000000 0000 00000000
```

(move #2, vehicle 2, forward, follow distance 0.2 m, follow vehicle 1, unloaded, acceleration 1.0 m/s^2, velocity 0.5 m/s, decouple path 0, decouple position 0)

```
0xD0 B7 00 00000002 0002 01 CDCC4C3E 0001 00 0000803F
0000003F 0000 00000000 0000 00000000
0xD0 B7 81 00000002 0002 01 9ECC4C3E 0001 00 5BE47F3F
A6FDFF3E 0000 00000000 0000 00000000
```

The HLC sends Command Status messages that show the Vehicle Follow Order command was received (0x00) and the vehicle is coupled to the platoon (0x81).

Move a Platoon without Auto-decouple

Move the platoon as required using either a Move Vehicle To Station or a Move Vehicle To Position command. The following example (see Figure 2-2 for the system layout) shows the lead vehicle in the platoon moving to Station 1. The command is accepted, and the command is completed once the vehicle stops moving.

```
0xB0 00000003 0001 01 0001 0000803F 0000003F 00000000
```
(move #3, vehicle 1, unloaded and forward, station 1, acceleration 1.0 m/s^2, velocity 0.5 m/s)

```
0xD0 B0 00 00000003 0001 0001 0004 0000803F 0000003F 01
00000000
0xD0 B0 80 00000003 0001 0001 0004 5BE47F3F A6FDFF3E 01
00000000
```

The HLC sends Command Status messages that show the Move Vehicle To Station command was received (0x00) and that it completed (0x80).

Decouple a Platoon

With the platoon not moving, request the current position of the vehicles in the platoon using either a Status Request or an Get Extended Vehicle Status command. The following example (see Figure 2-2 for the system layout) shows a status request for all vehicles and the status responses showing their locations.

```
0xB5 04 0000
```
(vehicle status request, all vehicles)

```
0xD5 0001 01 0003 0003 EBFF7F3F 00000000 B0 01 0001
0xD5 0002 01 0003 0000 C4CC4C3F 00000000 B7 01 9ECC4C3E
```

The HLC sends a Vehicle Status message for each vehicle that provides information about the vehicle, including its current location.

Move the following vehicle to its current position, using a Move Vehicle To Position command set for bidirectional motion to decouple it from the platoon. The following example shows the following vehicle being positioned at its current location. The command is accepted, and the command completes, no motion occurs.

NOTE: Bidirectional motion must be specified when decoupling a vehicle from the platoon to make sure the command can complete.

```
0xB1 00000018 0002 00 6666863F 0004 0000803F 0000003F
```
(move #18, vehicle 2, unloaded and bidirectional, position 1.05 m, path 4, acceleration 1.0 m/s^2, velocity 0.5 m/s)

```
0xD0 B1 00 00000018 0002 6666863F 0004 0000803F 0000003F 00
0xD0 B1 80 00000018 0002 6366863F 0004 5BE47F3F A6FDFF3E 00
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was received (0x00) and that it completed (0x80).

Create a Platoon with Auto-decouple

To create a platoon, use a Move Vehicle To Position command to move the following vehicle into position behind the lead vehicle. To couple that vehicle to the lead vehicle, use a Vehicle Follow Order command to create a platoon with the vehicle following the lead vehicle. The vehicle being coupled is configured to decouple automatically from the platoon and stop at 0.6 m on path 1. The following example shows the following vehicle being coupled to the lead vehicle, the command is accepted, and the command completes.

```
0xB7 00000019 0002 01 CDCC4C3E 0001 00 0000803F 0000003F
0001 9A99193F
```

(move #19, vehicle 2, forward, follow distance 0.2 m, follow vehicle 1, unloaded, acceleration 1.0 m/s^2, velocity 0.5 m/s, decouple path 1, decouple position 0.6 m)

```
0xD0 B7 00 00000019 0002 01 CDCC4C3E 0001 00 0000803F
0000003F 0001 9A99193F
0xD0 B7 81 00000019 0002 01 9ECC4C3E 0001 00 5BE47F3F
A6FDFF3E 0001 9099193F
```

The HLC sends Command Status messages that show the Vehicle Follow Order command was received (0x00) and the vehicle is coupled to the platoon (0x81).

Move a Platoon with Auto-decouple

The following example (see Figure 2-2 for the system layout) shows a Move Vehicle To Position command that is issued to move the lead vehicle in the platoon to 0.8 m on path 3, the command is accepted, and the command completes. This example shows that once the platoon reaches the correct location on path 1, vehicle 2 automatically decouples from the platoon (a decoupling Command Status is sent). The vehicle stops at 0.6 m on path 1 (a vehicle move to position complete command status is sent).

1. `0xB1 0000001A 0001 01 0000A03F 0003 0000803F 0000003F`
   (move #1A, vehicle 1, unloaded and forward, position 0.8 m, path 3, acceleration 1.0 m/s^2, velocity 0.5 m/s)

   ```
   0xD0 B1 00 0000001A 0001 0000A03F 0003 0000803F 0000003F
   01
   ```

   The HLC sends a Command Status message that shows the Move Vehicle To Position command was received (0x00).

2. ```
   0xD0 B7 82 00000019 0002 01 9ECC4C3E 0001 00 5BE47F3F
   A6FDFF3E 0001 9099193F
   ```
   (command status move #19, vehicle 2, decoupling)

   The HLC sends a Command Status message for the Vehicle Follow Order that shows the vehicle is decoupling from the platoon (0x82).
   ```

3. `0xD0 B1 80 00000019 0002 7E03183F 0001 5BE47F3F A6FDFF3E 01`
(command status move #19, vehicle 2, at destination location)

   The HLC sends a Command Status message that shows the decoupled vehicle is at its destination.

4. `0xD0 B1 80 0000001A 0001 96D19E3F 0003 5BE47F3F A6FDFF3E 01`

   The HLC sends a Command Status message that shows the Move Vehicle To Position command completed (0x80).

Shutting Down the Transport System

Whenever the transport system is being shut down it is important that all motion is stopped and the system is powered down correctly.

1. Issue a Suspend Motion command to all paths, which causes all vehicles to come to a controlled stop at the closest allowed position in the current direction of motion.

```
0xB4 0000
```

   The HLC sends a Command Status message for each path. This example shows that the Suspend Motion command was accepted for all paths.

```
0xD0 B4 00 0001
0xD0 B4 00 0002
0xD0 B4 00 0003
0xD0 B4 00 0004
```

2. Issue a Status Request command for all vehicles to verify that all motion has stopped.

```
0xB5 04 0000
```

   The HLC sends a Vehicle Status message for each vehicle. This example shows their current location and that they are not in motion.

```
0xD5 0001 01 0003 0000 6666663F 00000000 00 01 00000000
0xD5 0002 01 0001 0000 00002C40 00000000 00 01 00000000
0xD5 0003 01 0003 0000 0000003F 00000000 00 01 00000000
0xD5 0004 01 0001 0001 0000403F 00000000 00 01 00000000
```

3. Once all motion has stopped, issue a Reset command for all paths, which clears all vehicle records.

    ```
    0xB8 0000
    ```

    The HLC sends a Command Status *message for each path. This example shows that the Reset command was accepted for all paths.*

    ```
    0xD0 B8 00 0001
    0xD0 B8 00 0002
    0xD0 B8 00 0003
    0xD0 B8 00 0004
    ```

    Once the Reset command completes, the HLC sends a Command Status *message for each path. This example shows successful completion of the Reset command for all paths.*

    ```
    0xD0 B8 80 0001
    0xD0 B8 80 0002
    0xD0 B8 80 0003
    0xD0 B8 80 0004
    ```

4. Turn off power to the motors.

5. Turn off power to the node controllers.

6. Turn off power to the host controller.

Application Notes 3

Overview

This chapter provides examples and step-by-step procedures that are related to specific tasks using MagneMotion® motors in a transport system.

Application Notes are provided for specific tasks in the operation of MagneMover® LITE and QuickStick® transport systems. Simple step-by-step procedures and examples for each application that is covered are provided. These procedures reference detailed information about the steps that is located elsewhere in this manual.

Vehicle Motion

Brick-wall Headway

All vehicles on a transport system are moved with a brick-wall headway between vehicles so that a safe stopping distance between vehicles is always maintained. Brick-wall headway makes sure that if one vehicle stops unexpectedly (for example, an object falls on the guideway) any vehicle following it is able to stop without colliding with it. It is possible to override this move profile if necessary in special circumstances.

Vehicle Positioning

The location a vehicle is commanded to must allow the entire vehicle (as defined in the Node Controller Configuration File) to remain on the transport system (over the motor). The only time any portion of a vehicle can be commanded to a location off a motor is during vehicle insertion or removal through a Terminus Node. If a vehicle is commanded to a position that causes any portion of the vehicle to leave the end of the guideway, the vehicle stops before it reaches that position. This position control makes sure that no part of the vehicle leaves the end of the guideway and since it never reaches its destination it never completes the motion command.

Example: Figure 3-1 shows a simple path that starts at a Simple Node with no connecting path. Commanding the 200 mm vehicle that is shown in the figure to a position of 0 m on the path causes the vehicle to stop at the 100 mm position. The motion command never completes and no Command Status for completion is sent as the command stays pending. In this case, the Obstructed bit in the Vehicle Status for the vehicle is set, which shows that the vehicle cannot move to complete the command.

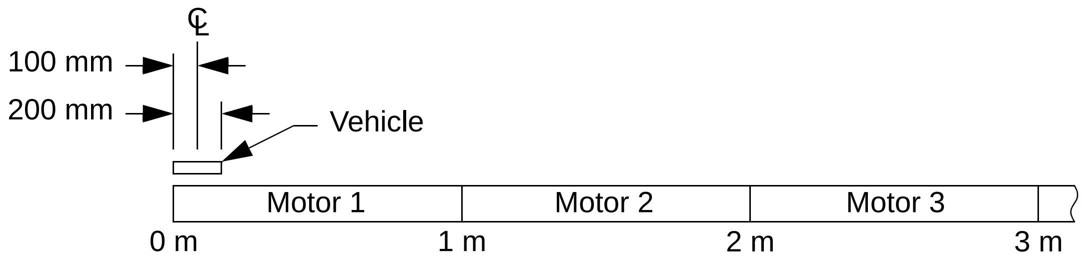


Figure 3-1: Vehicle Positioning Example

Safe Stopping Distance

Standard vehicle control and operation makes sure that vehicles always have a safe stopping distance, referred to as brick-wall headway. Figure 3-2 shows acceleration, velocity, and position versus time for the standard vehicle motion profile. Permission for vehicle motion is granted as required on a block-by-block basis. This permission keeps vehicles on their move profile (solid heavy line) and provides a safe stopping distance (dashed heavy line) based on the current velocity and commanded acceleration of the vehicle. This stopping distance can be found by dividing the square of the current velocity of a vehicle by twice its acceleration ($V^2/2a$).

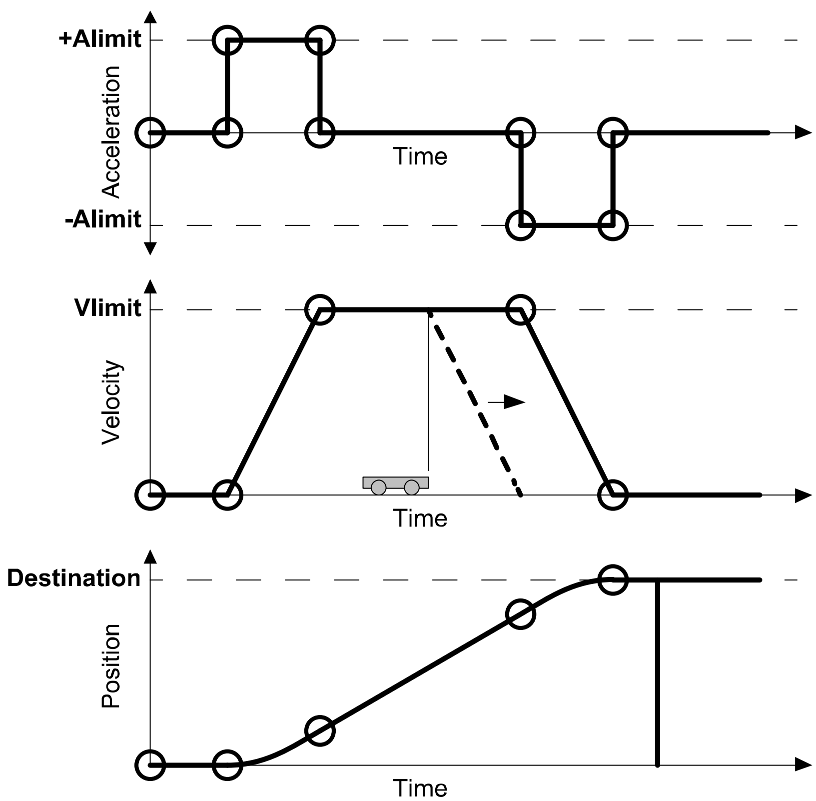


Figure 3-2: Vehicle Motion Profile

Thrust Limitations

When a vehicle is commanded with a higher acceleration rate than the motor can provide, the vehicle falls behind its ideal move profile while accelerating. Figure 3-3 shows both the ideal move profile (solid line) and the degraded move profile (dashed line).

In addition, and more critically, the vehicle is not able to decelerate at the specified rate and overshoots its destination as shown by the dashed line in Figure 3-3. This behavior can result in vehicles colliding with other vehicles or switch components, or loss of control of a vehicle as it exits the area where it has permission to move. Thus, it is important to avoid commanding a move with an acceleration that is higher than the deceleration capability of the system.

The precise deceleration capability depends on vehicle mass (including payload), center of gravity location, speed, and track geometry. Furthermore, the thrust capability of the motors is reduced in proximity to the gaps between motors.

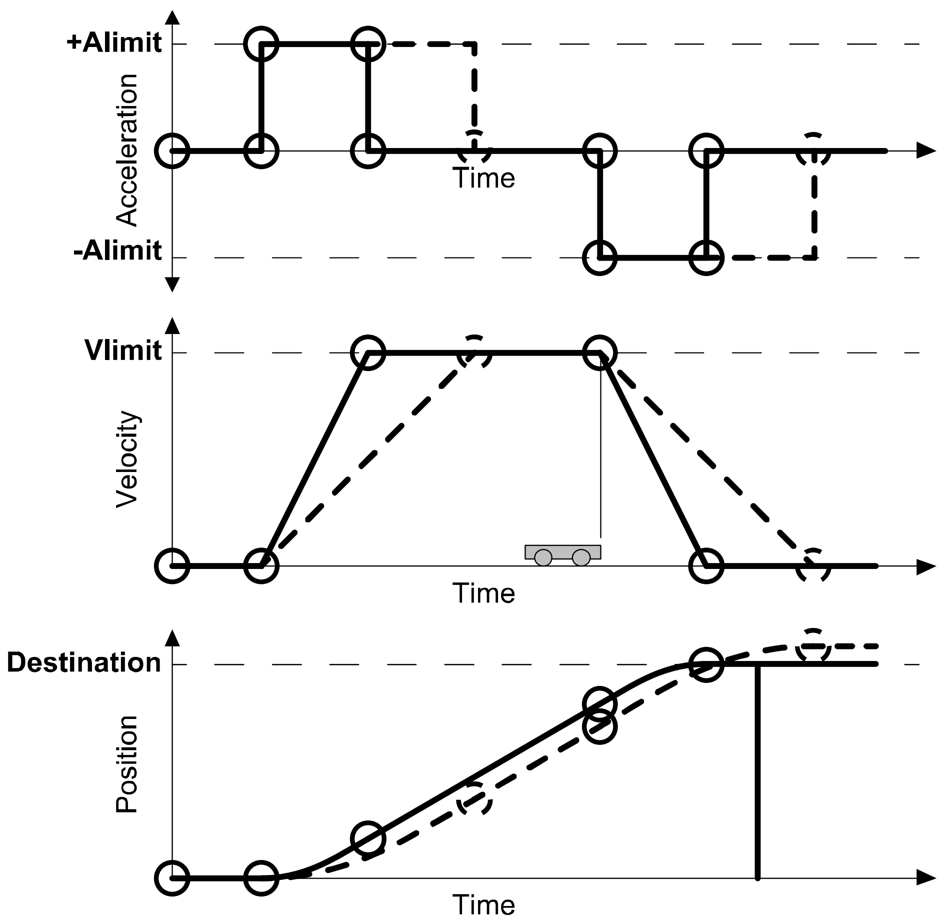


Figure 3-3: Vehicle Motion Profile Showing Thrust Limitations

Inserting and Removing Vehicles

Rockwell Automation recommends that a Terminus Node be used for all insertion and removal of vehicles on the transport system. Terminus Nodes provide all needed handshaking with the remote equipment for smooth transfer of the vehicle and proper identification or deletion of the vehicle on the transport system.

In some situations, it is not possible to move the vehicle fully past the end of a path where a Terminus Node would be located. An alternate method to insert or remove vehicles on the transport system is described in this section.

NOTE: In the following examples, the insert path and the remove path can be the same path if vehicles are both inserted and removed on that path.

Vehicle Insertion

1. Define a short path (minimum length of one motor and a maximum length of one vehicle length) as the insert path.
2. Issue a Reset command to the insert path.
3. Place the vehicle to be inserted onto the insert path.
4. Issue a Startup command to the insert path.
5. Review the Vehicle Status or Extended Vehicle Status, response to check the status of all vehicles.

   The Vehicle Status responses identify the Vehicle ID of the new vehicle now on the insert path.
6. Command the vehicle from the insert path onto the transport system.

Vehicle Removal

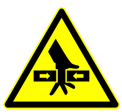
1. Define a short path (minimum length of one motor and a maximum length of one vehicle length) as the remove path.
2. Issue a Move Vehicle command to locate the vehicle on the remove path.
3. Issue a Reset command to the remove path.
4. Remove the vehicle from the remove path.
5. Issue a Startup command to the remove path.
6. Review the Vehicle Status or Extended Vehicle Status, response to check the status of the removed vehicle.

   The Vehicle Status response shows that the Vehicle ID of the removed vehicle is no longer being used.

E-stops

An E-stop uses a user-installed locking button that an operator can press if an emergency situation arises to suspend all motion on the specified paths. Once the emergency is resolved, motion can be restored.

SHOCK HAZARD: The E-stop only executes the actions that are described, it is not the same as an EMO (Emergency Off), which removes all power to the transport system.



PINCH/CRUSH HAZARD: Moving mechanisms (vehicles) have no obstruction sensors.

An E-stop does not suspend motion of a vehicle within a keep-out area or motion of a vehicle that has already been granted permission to move through a Terminus Node.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

Initiating a Host Controller E-stop

The E-stop should be configured to operate through the host controller, which requires the host controller to monitor the circuit and initiate the desired action when signaled.

NOTE: If an E-stop is wired to, and monitored by, the host controller, it is the responsibility of the host controller to command the system to perform the desired actions.

Upon receiving an E-stop request, the host controller must issue either a Suspend Motion or FastStop Motion command to the appropriate paths. If a Suspend Motion command is issued, all motors suspend vehicle target requests and permissions. All vehicles then come to a controlled stop at the furthest position they have permission to go to and are held in position by the motors. If a FastStop Motion command is issued, all vehicles immediately decelerate with maximum thrust opposing motion. All motors suspend vehicle target requests and permissions and all vehicles are held in position by the motors. The HLC updates the status for each path, which shows that motion is suspended for each path.

NOTE: When a Suspend Motion command is issued, vehicles in motion within a keep-out area in the direction of the keep-out area do not stop until they exit the keep-out area. Vehicles with a destination within the keep-out area stop at their destination.

When a FastStop Motion command is issued, vehicles in motion within a keep-out area stop within the keep-out area.

Stopping time for each vehicle is dependent on the load, available thrust, and the velocity and acceleration settings.

Power is not removed from the motors.

To check the status of a path, use a Status Request with **Request Type** set to 3. This request returns a Path Status with the **Path Motion Status** as either "Suspended by Command" or "FastStop Active" for the stopped paths.

Motion cannot resume until the host controller issues a Resume command to the paths that are associated with the E-stop.

If the host controller-based E-stop is configured so that it shuts off motor power when activated, the host controller must issue either a Suspend Motion or FastStop Motion command before shutting off power. Once all motion has stopped, the host controller can remove power. Time to wait before shutting off power can be determined based on the acceleration and velocity that is configured for the transport system. Make sure that the vehicles have sufficient time to come to a controlled stop.

- If only the motor propulsion power is shut off (motor logic and motor propulsion power must be wired separately), there is an under-voltage fault. When power is restored, the fault is cleared.
  - For MagneMover LITE, use a Status Request with **Request Type** set to 5, which returns a Motor Status that reports a "Propulsion power not ready" fault.
  - For QuickStick 100 motors, use a Status Request with **Request Type** set to 5, which returns a Motor Status that reports a "Soft Start not complete" fault.
  - For QuickStick HT motors, use a Status Request with **Request Type** set to 5, which returns a Motor Status that reports a "Soft Start switch off" fault.
- If both motor propulsion and motor logic power are shut off, a Status Request with **Request Type** set to 5 returns a "Motor Not Responding" fault. When power is restored, the fault is cleared.

Recovering from a Host Controller E-stop

Once the events that necessitated the E-stop have been resolved, motion can be restarted. To make sure of proper startup, the E-stop must be cleared by releasing the button that was pressed.

If power was not removed from the motors, once the events that necessitated the E-stop have been resolved the host controller must issue a Resume Motion command to the suspended paths. All vehicles on the suspended paths then resume their motion, which is based on all currently active motion commands.

If power was removed from the motors:

- If only the motor propulsion power was removed, all vehicles on the suspended paths immediately resume their motion once propulsion power is restored. The motion is based on all currently active motion commands.
- If both motor propulsion and motor logic power were removed, the motors must be restarted as described in Startup on page 29 once power is restored.

Interlocks

An interlock is a user-installed circuit that another piece of equipment in the facility activates to suspend all motion on the specified paths temporarily. Once the other equipment completes its task and deactivates the interlock, motion is automatically restored.

NOTE: This is to support legacy node controllers with digital I/O. New systems should control their interlock functionality through the host controller.

Initiating an Interlock

When the node controller detects that the interlock circuit is activated, it commands all paths that are associated with that interlock to suspend vehicle motion. All motors suspend vehicle target requests and permissions. All vehicles then come to a controlled stop at the furthest position they have permission to go to and are held in position by the motors. The HLC updates the status for each path, which shows the interlock is active for the stopped paths.

NOTE: Vehicles in motion within a keep-out area in the direction of the keep-out area do not stop until they exit the keep-out area. Vehicles with a destination within the keep-out area stop at their destination.

Stopping time for each vehicle is dependent on the load, available thrust, and the velocity and acceleration settings.

Power is not removed from the motors.

To check the status of a path, use a Status Request with **Request Type** set to “3”. This request returns a Path Status with the **Path Motion Status** as “Interlock Active” for the stopped paths.

Motion cannot resume until the associated equipment releases the interlock.

Recovering from an Interlock

Once the events that necessitated the interlock have been resolved, motion is restarted by deactivating the interlock signal. All vehicles on the suspended paths immediately resume their motion, which is based on all currently active motion commands.

Emergency Machine Off

The equipment that uses a transport system can have an Emergency Machine Off (EMO) circuit that is user-installed and configured. These circuits typically include a locking EMO button configured such that pressing the button removes all power to the transport system. The power that is removed typically includes motor propulsion power and possibly motor logic, node controller, and HLC power. On some transport systems, pneumatic power must also be removed.

Initiating an EMO

Upon loss of propulsion power all vehicles slow to a stop. The amount of time it takes to stop is dependent on vehicle velocity, load, and friction in the vehicle guideway.

Recovering from an EMO

Once the events that necessitated the EMO have been resolved, the transport system can be restarted (see Startup on page 29). To make sure of proper startup, the EMO must be cleared by releasing the button that was pressed before restarting the transport system.

Platooning

Vehicle platooning allows moving multiple vehicles simultaneously without the restriction of brick-wall headway between the vehicles by controlling only the lead vehicle. Brick-wall headway is still maintained before the lead vehicle in the platoon. Platooning is useful for stations that have multiple processes, as it allows a set of vehicles to move into, and out of, position in the station simultaneously, which can minimize change over time. For detailed examples of platooning, see Platooning on page 54.

Create platoons by coupling a following vehicle to a leading vehicle. Once a platoon of two or more vehicles is created, additional vehicles are added to it by coupling the additional vehicles to the end of the platoon, or by coupling the platoon to a new lead vehicle. Once a platoon is created, control of the vehicle at the front of the platoon (the leader) controls the behavior of all vehicles in the platoon (acceleration, velocity, position).

NOTE: Platooning is only available for MagneMover LITE transport systems on version 15.11.10 and later.

PINCH/CRUSH HAZARD: Moving mechanisms (vehicles) have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

This feature currently supports the following functions:

- Vehicles can be coupled into a platoon only when the platoon is not moving.
- Vehicles can be uncoupled from a platoon either when the platoon is not moving or when the Vehicle Follow Order for a specific vehicle is configured for auto-decouple.
- All vehicles in a platoon must be the same size (that is, all single vehicles or all tandem vehicles).
- Motion on any path and motion across a node is supported.
- Coupling vehicles to a platoon that is located across a node is supported.
- Uncoupling a vehicle from a platoon that is located across a node is supported.
- The vehicle follow order to the HLC must include acceleration and velocity values.
- Platoons cannot reverse direction. To change the direction of motion, the platoon must stop and be uncoupled. Once uncoupled, recouple the vehicles in the opposite direction. The last vehicle in the original platoon becomes the lead vehicle in the new platoon.

Platooning Operations Overview

The platooning feature couples a following vehicle to the vehicle in front of it at an ordered position offset to mimic the move profile (acceleration and velocity) of the platoon leader. This feature can be used to create a platoon of *N* vehicles to move out of a station while a new platoon of *N* vehicles moves in.

ATTENTION: Vehicles following another vehicle in a platoon do not support collision avoidance (brick-wall headway) between vehicles in the platoon.

When a platoon is moving through a node, the following vehicles in the platoon continue to follow the lead vehicle. Ownership of the node is only released after the entire platoon passes through the node.

The functioning of Single Vehicle areas, Keepout areas, and Traffic Lights is not guaranteed with vehicles that are in a platoon. Only the lead vehicle obeys these functions.

A vehicle ordered to follow another vehicle with a platooning command will obey the specified platoon following distance, and will NOT use wide vehicle spacing. A platoon spacing distance must be specified that is large enough to avoid collisions along the route that the platoon will travel (for example, through curves).

- When using multiple platoons, the distance from the following platoon to the leading platoon must obey brick-wall headway.
- Platoons can be “coupled” and “uncoupled” to move vehicles into and out of a station without the constraint of brick-wall headway between the vehicles in the platoon.
- All vehicles in a platoon must be the same size and use the same type of magnet array.
- When coupling a vehicle into a platoon, the center-to-center distance between vehicles must keep the vehicles from contacting each other when moving through a curve.
- Attempting to couple across a node that another platoon is crossing is rejected.
- For MagneMover LITE, to reduce the effect of competing forces between adjacent magnet arrays and the coils in the motor, the recommended minimum center-to-center space from leading to following vehicles when creating a platoon is:
  - 100 mm for standard 62 x 62 mm single pucks with a 77 mm configured length.
  - 173 mm for standard 62 x 150 mm tandem pucks with a 165 mm configured length.

If the vehicle length is longer than the minimal puck length described above, the minimum gap between vehicles must be the value that is returned from the following formula.

$MinGap = 30.0\ \text{mm} + VehicleLength$

Where:

MinGap – Minimum gap between vehicles in a platoon (in mm).

VehicleLength – Length of the vehicle (in mm).

The gap that is calculated must be added to the actual vehicle length when deriving minimal vehicle spacing in the platoon, subject to practical minimal distance recommendations.

If the vehicle overhangs the puck, the vehicle edges can be placed closer than the distance that the formula defines. However, if the vehicles are placed too close to each other, tolerances in the vehicle following the profile can cause vehicles to touch.

- For QuickStick 100 (using firmware version 15.9.19) to reduce the effect of competing forces between adjacent magnet arrays and the coils in the motor, the recommended minimum center-to-center space from leading to following vehicles when creating a platoon is:

  $MinGap = 112\ \text{mm} + VehicleLength + LargestDownstreamGap$

  Where:

  MinGap – Minimum gap between vehicles in a platoon (in mm).

  VehicleLength – Length of the vehicle (in mm).

  LargestDownstreamGap – Length of the largest downstream gap (in mm).

  The gap that is calculated must be added to the actual vehicle length in deriving minimal vehicle spacing in the platoon, subject to practical minimal distance recommendations.

- To make effective use of platooning, the maximum distance between the vehicles in a platoon should be no greater than:

  $MaxDistance = MaxCommandedVelocity^2 / (CommandedAcceleration * 2)$

  Where:

  MaxDistance – Maximum distance between vehicles in a platoon (in meters).

  MaxCommandedVelocity – Maximum velocity defined in the Node Controller Configuration File (in m/s).

  CommandedAcceleration – Maximum acceleration being used in a motion command for the platoon (in m/s^2).

- When coupling a vehicle into a platoon, the maximum difference between the physical distance and the requested distance between the vehicles (the distance that is specified in the command) must be within 30 mm.

- When coupling a vehicle to a platoon using the vehicle follow order, the vehicle and the platoon must be stopped. Stopped is defined as vehicle motion must be below the **Arrival Velocity Tolerance** defined on the **Global Settings** page in the Configurator when creating the Node Controller Configuration File.
- When uncoupling a vehicle from a platoon using a vehicle position order, the platoon must be stopped (vehicle motion must be below the **Arrival Velocity Tolerance**) and the order must specify bidirectional motion.
- When uncoupling a vehicle from a platoon using a vehicle follow order with a decouple destination, the vehicle decouples from the platoon when the platoon approaches the decouple destination. The vehicle then stops at the decouple destination when the platoon passes that position.
- The leading vehicle in a platoon must be leading in the direction of travel.
- Each vehicle added to the platoon must be added in the direction of travel of the platoon.
- Deleting the lead vehicle from a platoon causes the platoon to stop due to lack of profile data from the leader. The next vehicle in the platoon becomes the new leader. To continue motion of the platoon, send a vehicle motion order to the new leader.

  If the vehicle behind the lead vehicle was the end of a two-vehicle platoon, the platoon no longer exists.
- Platoons cannot couple or uncouple vehicles while using Sync.
- Sync can only be used on the lead vehicle of a platoon, and the sync profile must continue in the direction of the platoon. Followers in the platoon ignore all sync messages.

Creating Platoons

Platoons can be created for vehicle motion either downstream or upstream. Once a platoon is created, it is limited to motion in the direction that is specified when it was created. The direction of a platoon can be reversed by decoupling the vehicles in the platoon and creating a platoon moving in the opposite direction (see Change Platoon Direction on page 77).

To couple a vehicle into a platoon, or uncouple a vehicle from a moving platoon, use the Move Vehicle To Position and the Vehicle Follow Order commands as described in the following examples.

Follow A Vehicle Moving Downstream

Create a two-vehicle platoon that moves downstream by specifying a vehicle for another vehicle to follow.

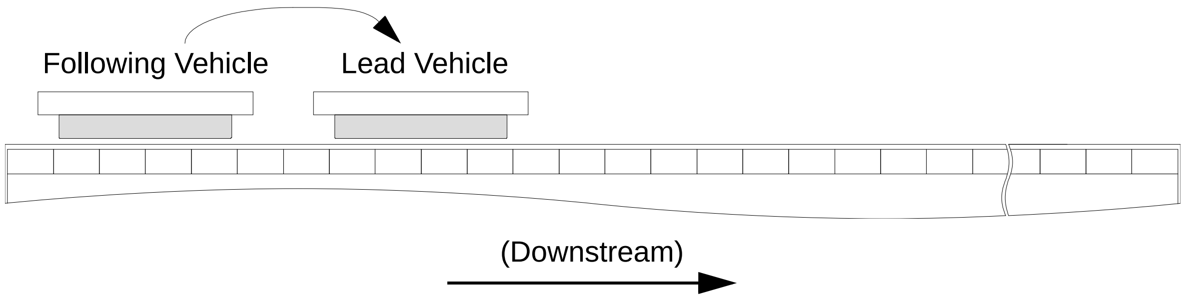


Figure 3-4: Create a Downstream Platoon

1. While the vehicle that is going to be the platoon leader is stopped, command the following vehicle to move downstream into position behind it (upstream of it) using a Move Vehicle To Position command. The vehicle follow order command must specify this distance between the vehicles as the Follow Distance.

   The following vehicle moves to its commanded position.

2. Couple the following vehicle to the lead vehicle and specify the distance (from Step 1) between the vehicles and the direction of motion (downstream) using a Vehicle Follow Order command.

   The vehicles are linked into a platoon at the specified distance between the vehicles.

3. Command the lead vehicle to move downstream as required.

   The following vehicle follows behind the lead vehicle at the specified distance.

To add additional vehicles to the platoon, see Add Additional Vehicles To The End Of A Platoon on page 73.

Follow A Vehicle Moving Upstream

Create a two-vehicle platoon that moves upstream by specifying a vehicle for another vehicle to follow.

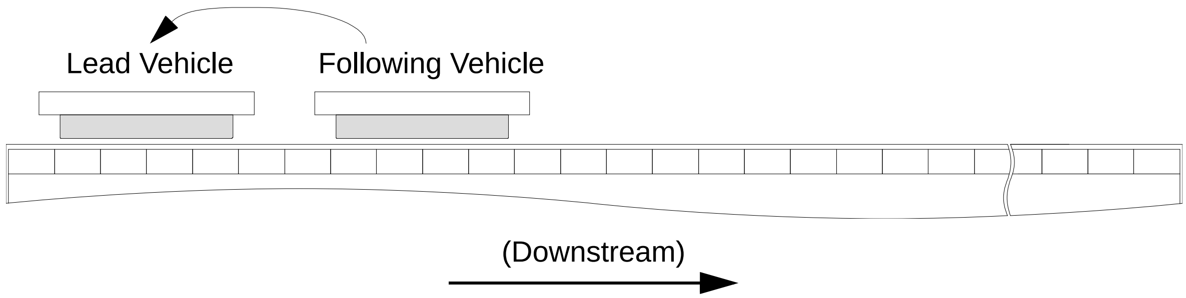


Figure 3-5: Create an Upstream Platoon

1. While the vehicle that is going to be the platoon leader is stopped, command the following vehicle to move upstream into position behind it (downstream of it) using a Move Vehicle To Position command. The vehicle follow order command must specify this distance between the vehicles as the Follow Distance.

   The following vehicle moves to its commanded position.

2. Couple the following vehicle to the lead vehicle and specify the distance (from Step 1) between the vehicles and the direction of motion (upstream) using a Vehicle Follow Order command.

   The vehicles are linked into a platoon at the specified distance between the vehicles.

3. Command the lead vehicle to move upstream as required.

   The following vehicle follows behind the lead vehicle at the specified distance.

To add additional vehicles to the platoon, see Add Additional Vehicles To The End Of A Platoon on page 73.

Add Additional Vehicles To The End Of A Platoon

Extend a multi-vehicle platoon that moves either downstream or upstream by adding additional vehicles to the end of the platoon. The additional vehicles are added in the direction of motion of the platoon by specifying the last vehicle in the platoon as the vehicle to follow for the vehicle being added. The vehicle being added can be either one vehicle or the lead vehicle in another platoon.

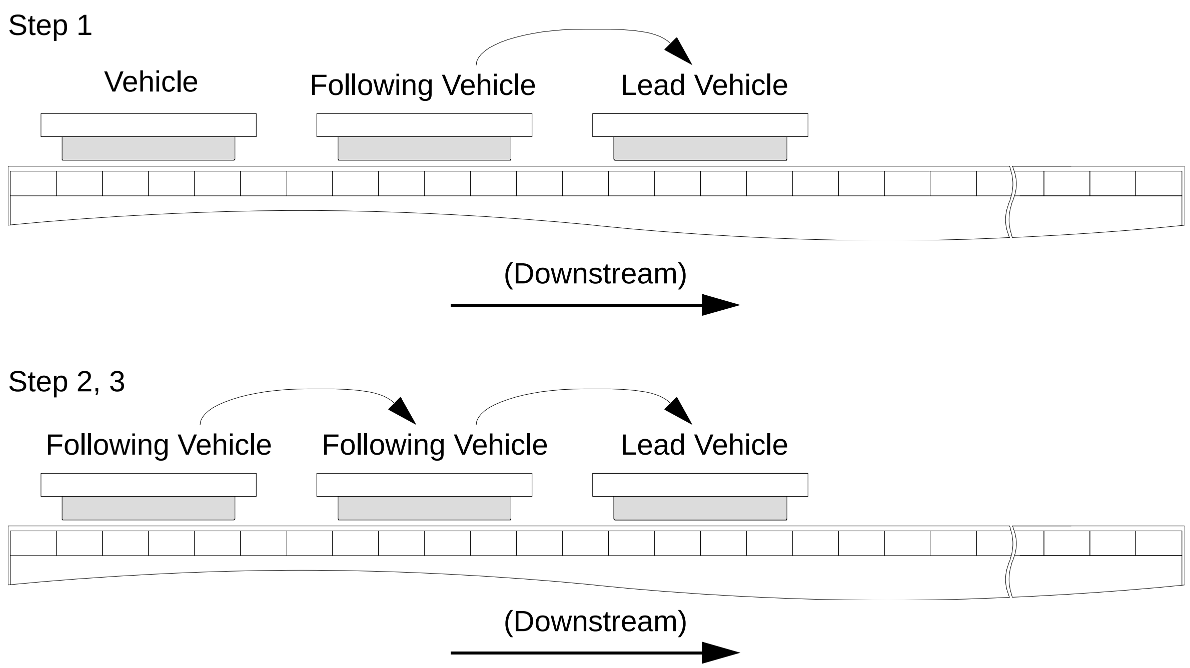


Figure 3-6: Add Vehicles to an Existing Platoon

1. While the platoon is stopped, command the next following vehicle (either one vehicle or the lead vehicle in another platoon) to move into position behind the platoon using a Move Vehicle To Position command. The vehicle follow order command must specify this distance between the vehicles as the Follow Distance.

   The following vehicle moves to its commanded position.

2. Couple the next following vehicle to the platoon by specifying the last vehicle in the platoon as the vehicle to follow. Specify the distance (from Step 1) between the vehicles and the direction of motion of the platoon using a Vehicle Follow Order command.

   The vehicles are linked into a platoon at the specified distance between the vehicles.

3. Command the lead vehicle as required.

   All following vehicles follow behind the lead vehicle at their specified distances.

Add A New Leader To A Platoon

Extend a multi-vehicle platoon that moves either downstream or upstream by coupling the lead vehicle in the platoon to a vehicle, which creates a new lead vehicle for the platoon. The platoon is coupled to the vehicle in the direction of motion of the platoon by specifying the vehicle for the current lead vehicle in the platoon to follow. This vehicle can be either one vehicle or the last vehicle in another platoon.

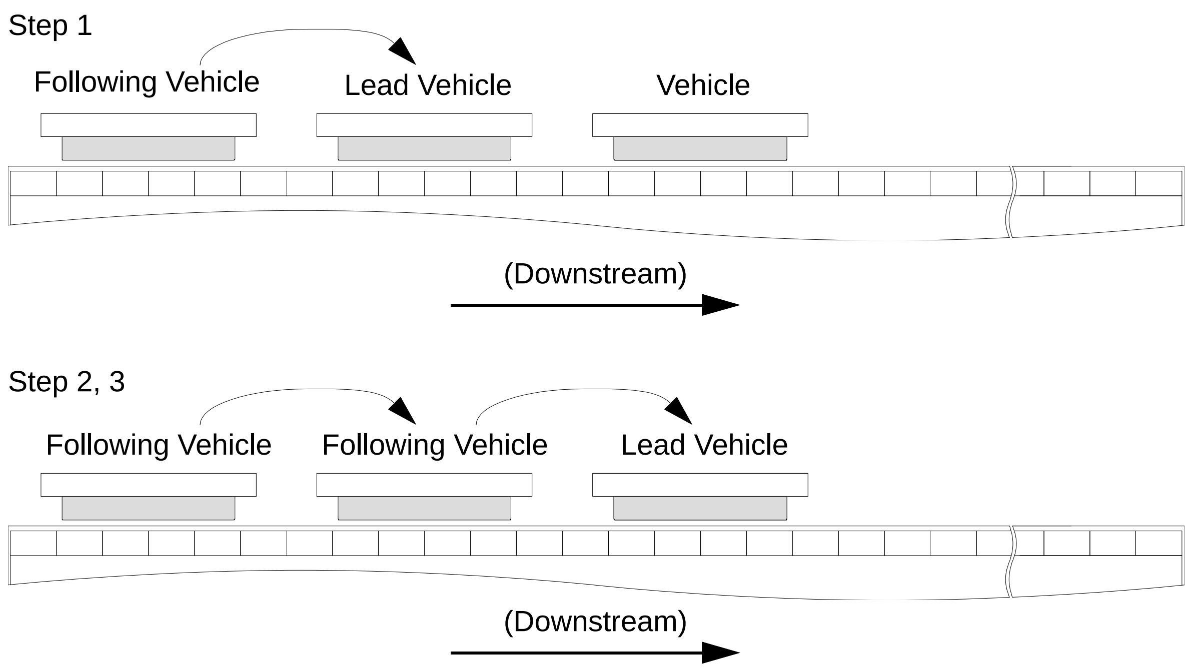


Figure 3-7: Add New Lead Vehicle to a Platoon

1. While the vehicle that becomes the new lead vehicle (or the last vehicle in another platoon) is stopped, command the lead vehicle in the existing platoon to move into position behind that vehicle using a Move Vehicle To Position command. The vehicle follow order command must specify this distance between the vehicles as the Follow Distance.

   The existing platoon follows the lead vehicle as it moves to its commanded position.

2. Couple the platoon by specifying the new lead vehicle (or the last vehicle in a platoon) as the vehicle for the platoon leader to follow. Specify the distance (from Step 1) between the vehicles and the direction of motion of the platoon using a Vehicle Follow Order command.

   The platoon is linked to its new lead vehicle at the specified distance between the vehicles.

3. Command the new lead vehicle as required.

   All following vehicles follow behind the lead vehicle at their specified distances.

Uncoupling Platoons

Existing platoons can be uncoupled once the need for platooning is complete. Platoons can be split into their individual vehicles, or they can be split into smaller platoons.

Separate A Platoon Into Vehicles

Once a platoon has been created, it can be separated into uncoupled vehicles when the need for platooning is complete. Remove a vehicle from a platoon by sending it a new vehicle motion command.

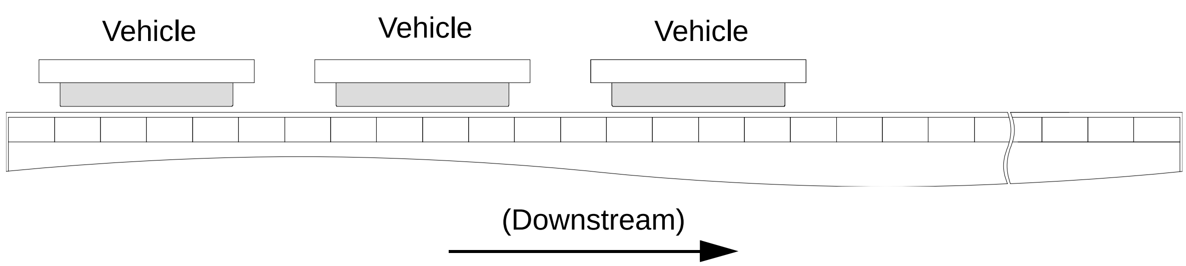


Figure 3-8: Separate A Platoon

1. With the platoon stopped, command vehicles to be uncoupled from the platoon.
   - To separate a vehicle from the vehicle that it is following in the platoon without moving the vehicle, command it to move to its current target as returned in a Vehicle Status or an Extended Vehicle Status, response. The Move Vehicle To Position command must specify bidirectional motion.

     The vehicle moves to its current target position (no actual motion occurs), which uncouples it from the vehicle that it is following. If there is a vehicle that is coupled to this vehicle, it remains coupled until it is uncoupled.
   - To separate the last vehicle from the platoon and move the vehicle, command it to move to its new location. The Move Vehicle To Position command must specify motion in the direction away from the platoon.

     The vehicle moves to its new position, which uncouples it from the vehicle that it is following.
2. Once the vehicles are uncoupled, command them to their new positions using Move Vehicle To Position commands.

   Each vehicle moves individually as commanded and normal brick-wall headway is maintained between all vehicles.

Separate A Platoon Into Smaller Platoons

Once a platoon has been created, sections can be uncoupled from the original platoon, creating smaller platoons when the need for the larger platoon is complete. To split the platoon, uncouple the lead vehicle of the new smaller platoon from the original platoon by sending it a new vehicle motion command.

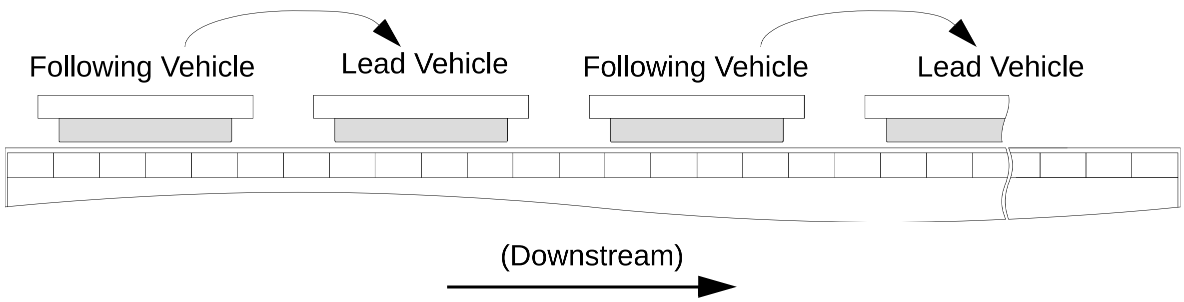


Figure 3-9: Separate A Platoon Into Smaller Platoons

1. With the platoon stopped, determine where to split the platoon into two. Command the vehicle that becomes the lead vehicle for the new platoon to move to its current target as returned in a Vehicle Status or an Extended Vehicle Status, response. The Move Vehicle To Position command must specify bidirectional motion.

   The vehicle moves to its current position (no actual motion occurs), which uncouples it from the vehicle that it is following. All vehicles that are coupled to this vehicle remain coupled and this vehicle becomes the leader of a new platoon.

2. Once the platoon is separated into smaller platoons, command the lead vehicle of each new platoon to its new position using Move Vehicle To Position commands.

   Each platoon moves individually as commanded, normal brick-wall headway is maintained between platoons.

Separate a Vehicle from a Platoon at a Specified Position

When a platoon is created, the follow order for a vehicle can specify a decouple position. When the platoon with the following vehicle with that position specified as part of its follow order approaches that position, the vehicle decouples from the platoon, slows down, and stops at that position. Any vehicles that are following the vehicle that decouples from the platoon continue to follow it after it decouples and stops. The remaining platoon continues to move under its original motion order.

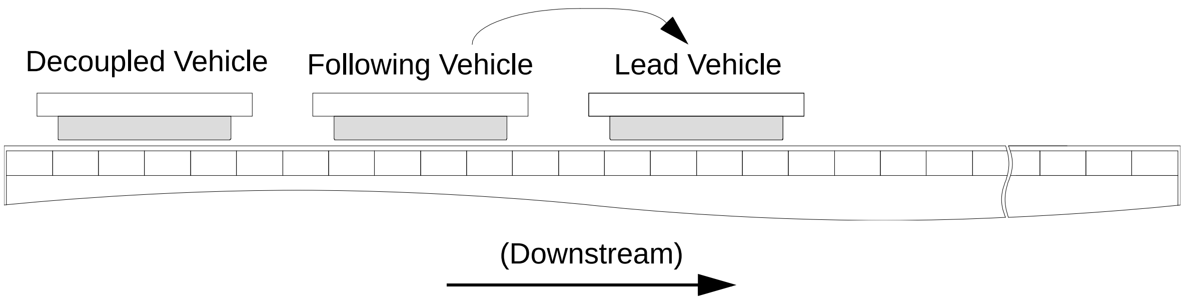


Figure 3-10: Separate a Vehicle from a Moving Platoon

1. While the platoon is stopped, command the following vehicle that is going to decouple from the platoon to move into position behind it (see Creating Platoons on page 70). The vehicle follow order command must specify this distance between the vehicles as the Follow Distance.

2. Couple the following vehicle to its lead vehicle and specify the distance (from Step 1) between the vehicles and the direction of motion using a Vehicle Follow Order command. This vehicle follow order must also define the decouple path and the final position for this vehicle.

3. Command the lead vehicle to move as required, the following vehicle follows it at the specified distance.

   When the platoon approaches the decouple position the vehicle decouples from the platoon, slows down, and stops at that position. Any vehicles that are following the vehicle that decouples from the platoon continue to follow it after it decouples and stops.

Change Platoon Direction

Once a platoon has been created to move in a specific direction, the direction of the platoon can be reversed by stopping it, separating the vehicles in the platoon, and creating a platoon in the opposite direction. The lead vehicle for the new platoon is the vehicle that was the last following vehicle.

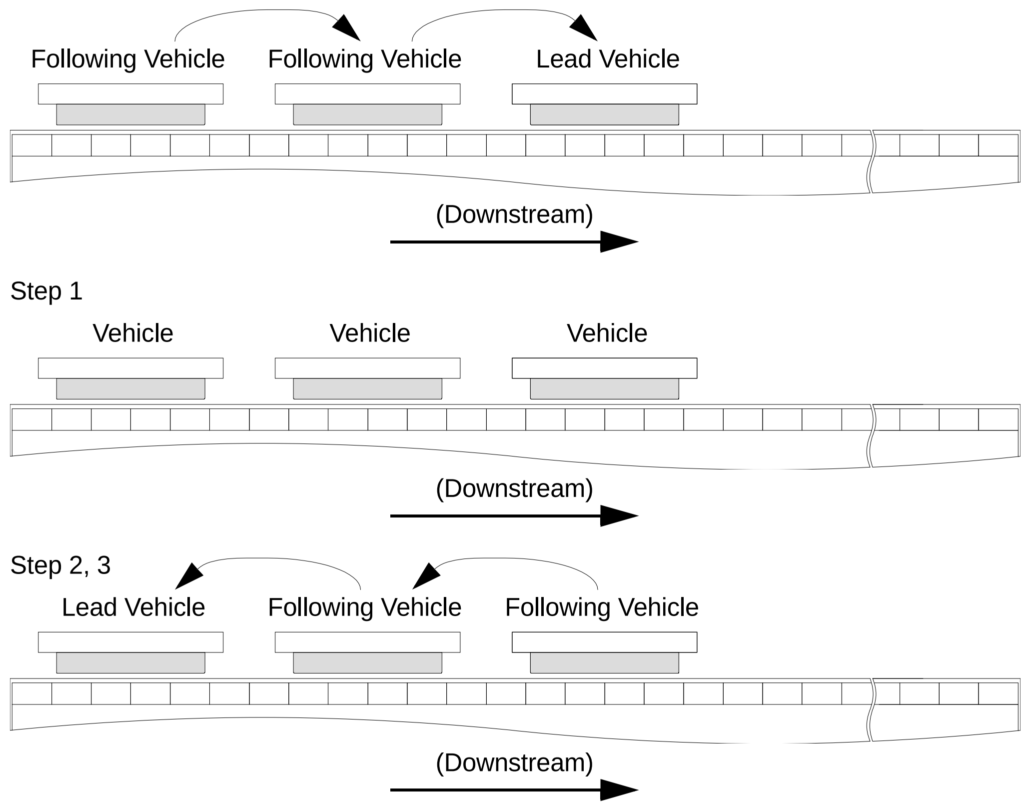


Figure 3-11: Change Platoon Direction

1. With the platoon stopped, command each vehicle to move to its current target as returned in a Vehicle Status or an Extended Vehicle Status, response. The Move Vehicle To Position command must specify bidirectional motion.

   Each vehicle moves to its current target position (no actual motion occurs), which uncouples it from the vehicle that it is following.

2. Use the last vehicle from the previous platoon as the new platoon leader. Couple the new following vehicle to the new lead vehicle and specify the distance (from Step 1) between the vehicles and the new direction of motion using a Vehicle Follow Order command.

   The vehicle is linked into the new platoon as specified.

3. Couple the next following vehicle to the new platoon by specifying the last vehicle in the new platoon as its vehicle to follow. Specify the distance (from Step 1) between the vehicles and the new direction of motion using a Vehicle Follow Order command.

   The vehicle is linked into the new platoon as specified.

4. Repeat Step 3 for each additional vehicle in the platoon.

5. Command the new lead vehicle as required, all following vehicles follow it at their specified distances.

   The following vehicles follow behind the lead vehicle at their specified distances.

Platooning Across a Node

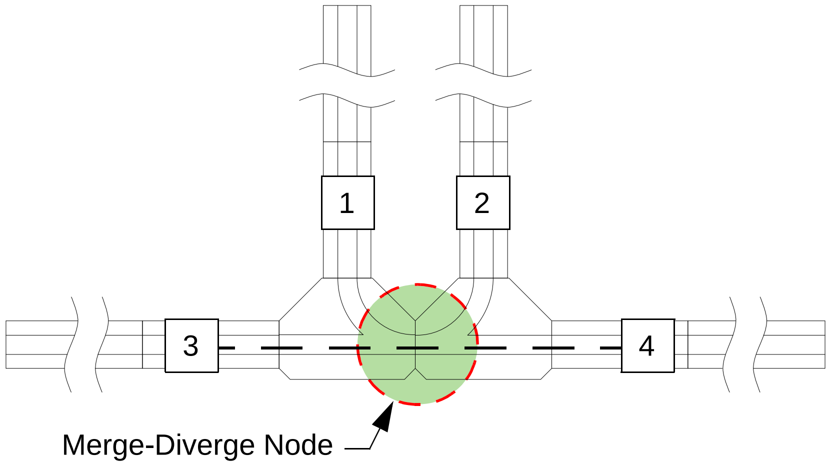


Figure 3-12: Platooning Across a Node

Attempting to couple across a node that is already part of a platoon causes the Vehicle Follow Order command to be rejected.

The example in Figure 3-12 shows a platoon with vehicle 3 coupled to vehicle 4 crossing a node. If a Vehicle Follow Order to couple vehicle 1 to vehicle 2 is sent, the order is rejected since the vehicle 3 - vehicle 4 platoon has locked the Merge-Diverge Node.

Traffic Lights

The Traffic Light feature allows the creation of virtual traffic lights for use during system operation. Traffic lights are used to control vehicle motion at defined positions in the transport system and are associated with specific motor blocks. For examples of traffic light use, see Traffic Light Control on page 35 and Traffic Light Status on page 46.

When the traffic light is set to green, vehicles are granted permission to move beyond the traffic light location. When the traffic light is set to red, vehicles are not granted permission to enter the motor block where the traffic light is located as shown in Figure 3-13.

There is a limit of one traffic light per motor block and 32 traffic lights per path. The motor block lengths for each motor type are provided in Table 3-1 for reference. Commands to create traffic lights on a motor block where a traffic light is already located or create more than 32 traffic lights on a path are rejected.

IMPORTANT	Traffic lights and their settings are not persistent. All traffic lights are created or set using the Traffic Light commands and must be recreated and reset in the following situations: • Whenever the node controller is restarted or rebooted. • Whenever any paths where traffic lights are located are reset.

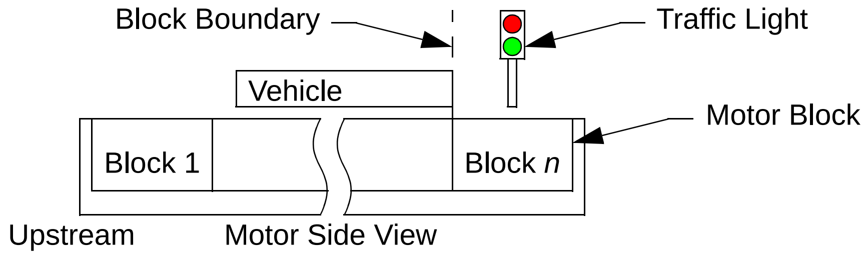


Figure 3-13: Traffic Lights and Motor Blocks

Table 3-1: Block Length

Motor Type	Block Length
MagneMover LITE	16.47 mm
QuickStick	96 mm
QuickStick HT	480 mm

System Monitoring

System monitoring describes a mechanism for retrieving system metrics that can be used to characterize the health of various components within a transport system.

When collecting system metrics on a larger scale and over longer periods of time, such metrics can be used to discover meaningful patterns for improving system operation and early fault detection. By aggregating such data, system metrics facilitate predictive analytics, factory optimization, and efficiency analysis decisions.

The goal of system monitoring is to provide the user with the information to discover, diagnose, and solve installation issues and monitor the operation of running systems.

System Metrics

During normal operation, a host controller can survey the health of system components within a transport system. For example, each motor can be directed to report key system metrics such as board temperature and propulsion voltage continually.

System metrics are enumerated using a unique identifier and organized into simple behavior patterns.

Behavior Patterns

System Metrics are organized into the following behavioral patterns:

1. The Variable pattern represents a simple value, such as constants, configuration, or other internal parameter information. Variables only return the last updated value when read.

2. The Counter pattern represents a positive integer value that increases monotonically until it reaches its maximum value, when it wraps around and starts increasing again from zero. Counters only return the last updated value when read.

3. The Gauge pattern represents a dynamic value that increases or decreases and contains the following properties: last value, minimum value, maximum value, and average value for the requested sampling interval (see Table 3-2).

Table 3-2: Gauge Pattern Properties

Gauge Property	Description
Last Value	The last value that updated this metric.
Minimum Value	The lowest value that this metric experienced.
Maximum Value	The highest value that this metric experienced.
Average Value	The statistical mean that is calculated by dividing the sum of the updated values by the number of updates for this metric.

When the software updates a Gauge metric, it updates all four properties to maintain statistical consistency. Gauges return all four values when read.

Component Types

System metrics can be collected from any of the primary components comprising a transport system. Each component is grouped according to its assigned component type.

Available component types are described in Table 4-3 on page 230.

Metric Identifiers

A unique, 16-bit number that is called a Metric Identifier (see Table 4-5 on page 235) is used to is identify each metric. Metric identifiers uniquely identify the behavior and characteristics of the metric it references and are consistent across system components.

Metric Instance

Some metrics (for example, communication counters) have multiple unique instances that are present for each of the communication ports. These kinds of metrics use the instance field to select which port the metric is referencing. For these metrics, the instance field is mandatory, and is used to select the correct occurrence of the value. When a metric consists of one value and does not require an instance, the instance field value must be zero.

Accessing the System Monitoring Interface

To access System Monitoring commands, the host controller must connect to the high-level controller using TCP port 801 using industry standard socket communication. This port has been designated for System Monitoring commands only and cannot be used for any other purpose.

System Monitoring API

The System Monitoring API is a subscription-based mechanism that allows host controllers to request specific metric data at a custom interval or on demand. The examples that are provided show monitoring of motors.

The host controller can subscribe to a specific system metric to report at the specified interval continuously until stopped with an unsubscribe command. The time interval between samples can range from 1 second to over 18 hours. A command option in the SM Subscription Command can be used to instruct the motor to clear the metric data before sampling starts.

Each motor maintains an internal list of active metric subscriptions, up to a maximum of four entries as represented in Figure 3-14. Each subscription slot, along with tracking data, maintains its own interval time and subscription options. On reset or power cycle, the motor starts with all subscription slots empty.

	0	--empty--
	0	--empty--
	0	--empty--
	0	--empty--

Figure 3-14: Empty Metric Subscription Slots

Subscribe

The host controller subscribes to a system metric for a motor using its Path ID and Motor ID by sending an SM Subscription Command message to the HLC, which passes it to the appropriate Node Controller. The subscription command also specifies the metric identifier and metric instance, its sampling interval in seconds, and command options.

Each subscription command consumes a subscription slot in the motor. If a subscription command is received when there are no available subscription slots, the motor responds with an SM Subscription Response message with a status of No record available.

When the subscription command is received, the motor stores subscription parameters in one of its available subscription slots. If the subscription command specifies the "Clear Metric Data on Sampling Start" command option, the targeted metric data is cleared at the beginning of each sampling interval.

An SM Subscription Response message is sent to the host controller to signal command failure or command completion. In addition, metric data updates are continuously reported to the host controller using SM Subscription Data Response messages as specified.

The sequence diagram in Figure 3-15 shows a subscription operation to sample motor temperature every 300 seconds (5 minutes). After 300 seconds, a metric data update is sent and repeated every 300 seconds thereafter until an unsubscribe command message is sent to stop it.

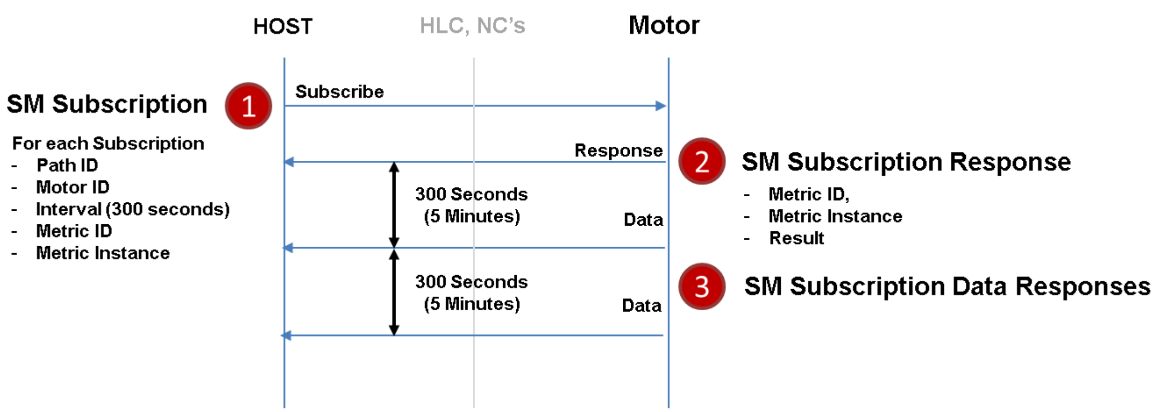


Figure 3-15: Subscribe Sequence Diagram

When the subscription operation completes, the motor has consumed the first subscription slot to monitor board temperature as shown in Figure 3-16.

	300	Temperature
	0	--empty--
	0	--empty--
	0	--empty--

Figure 3-16: Metric Subscription Slot Allocated for Temperature Monitoring

Unsubscribe

When a subscription is no longer needed, an SM Subscription Command message specifying an unsubscribe operation (Subscription Interval = 0) stops metric data updates and clears the motor's subscription slot for reuse. When completed, the motor responds with an SM Subscription Response message.

The sequence diagram in Figure 3-17 shows an unsubscribe operation.

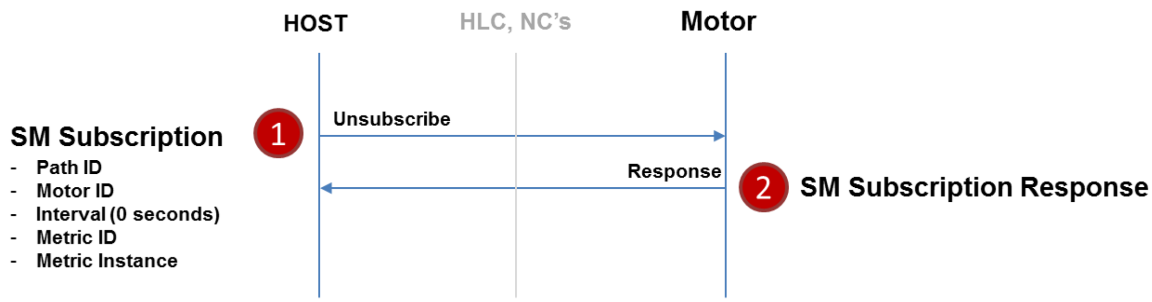


Figure 3-17: Unsubscribe Sequence Diagram

When the unsubscribe operation completes, the motor frees the specified subscription slot for reuse as shown in Figure 3-18.

	0	--empty--
	0	--empty--
	0	--empty--
	0	--empty--

Figure 3-18: Metric Subscription Slot Freed for Reuse

Polling

The host controller can poll metric data on demand using the SM Poll Command. Requesting data on demand is useful when the interval desired is longer than the maximum interval supported in the subscription command. The SM Poll Command when combined with the "Clear Metric Data on Sampling Start" command option, provides the same statistical consistency as subscription-based monitoring.

Care must be used when using the "Clear Metric Data on Sampling Start" command option when polling metric data with active subscriptions. Clearing metric data using the SM Poll Command affects the consistency of the data being reported in the subscription. When polling, specify that metric data is not cleared, which makes sure that the subscription-based metric data remains consistent.

The sequence diagram in Figure 6 shows a poll operation.

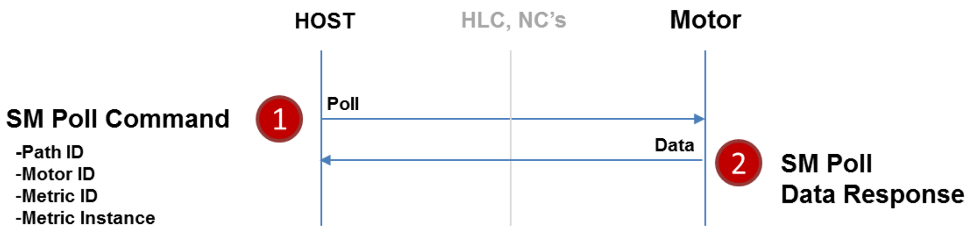


Figure 3-19: Poll Sequence Diagram

Motor Types

The transport systems are a configuration of linear synchronous motors that are placed end-to-end to form long chains or paths. These chains are used to move and position material carriers (vehicles) in a controlled manner at various acceleration/deceleration and velocity profiles while carrying various payloads with high precision. The Host Controller TCP/IP Communication Protocol supports the following types of motors.

NOTE: Motors from different product lines cannot be mixed in the same transport system.

MagneMover LITE

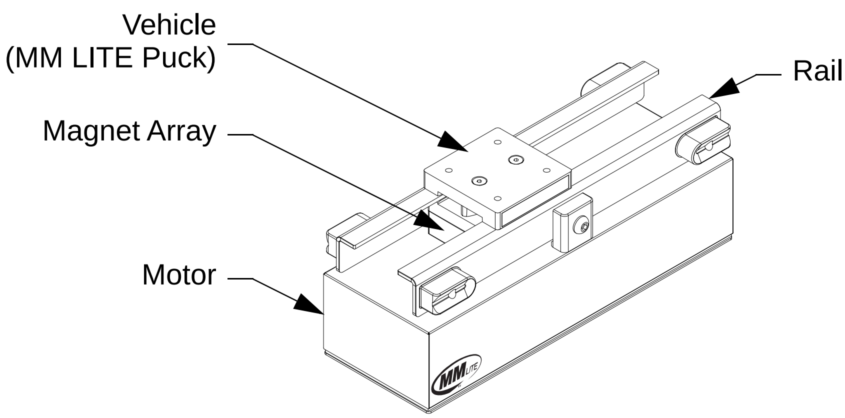


Figure 3-20: MagneMover LITE Motor

The MagneMover LITE motor consists of the motor with internal control electronics and an integrated rail (the 250 mm motor is shown in Figure 3-20). Motors are mounted end-to-end on a frame with the rails on the motors connected together to create the guideway.

The motor controller for each MagneMover LITE motor is located inside the MM LITE motor. The motor controller monitors the motor and controls vehicle position. The controller also communicates vehicle position and other information to the other motors in the path and to the node controller.

The vehicle that is used with the MagneMover LITE motors is a preconfigured puck with an integrated magnet array that slides on the rails. The motors interact with the magnet array to move and track the puck.

QuickStick (QuickStick 100 and QuickStick 150)

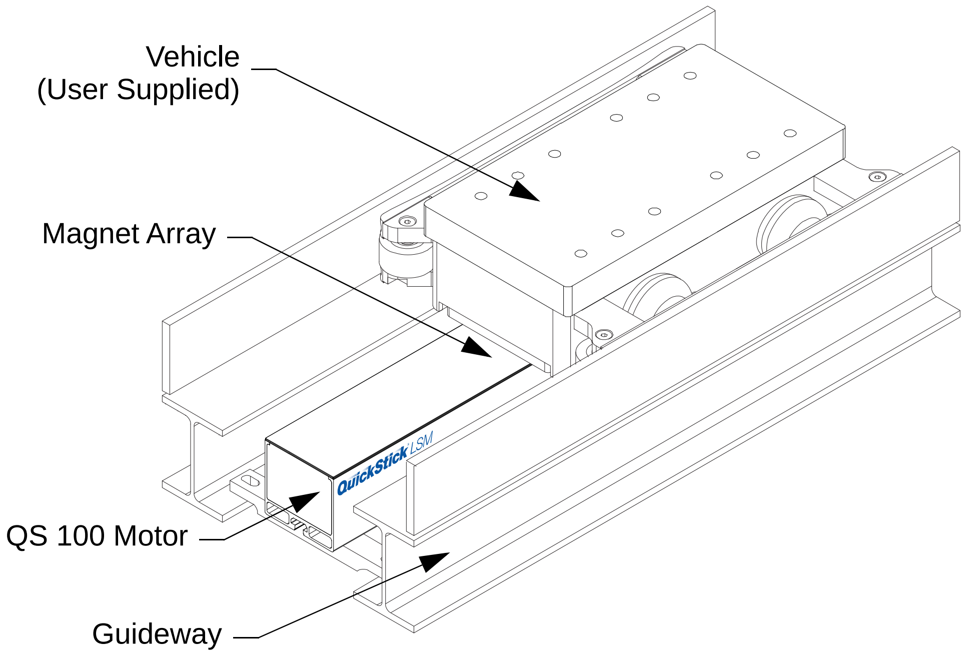


Figure 3-21: QuickStick 100 Motor

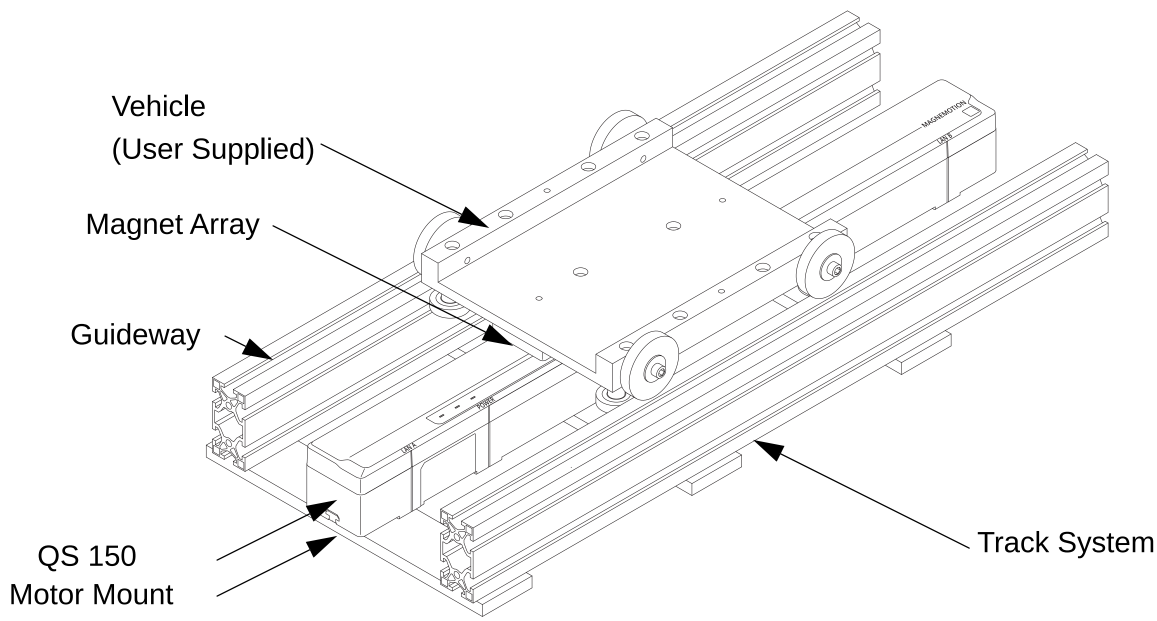


Figure 3-22: QuickStick 150 Motor

The QuickStick motors consists of the motor with internal control electronics. Motors are mounted end-to-end on a user-supplied frame with a user-supplied guideway.

The motor controller for each QuickStick motor is located inside the motor. The motor controller monitors the motor and controls vehicle position. The controller also communicates vehicle position and other information to the other motors in the path and to the node controller.

The vehicle that is used with the QuickStick motors is user-defined with a magnet array that is mounted on the surface closest to the motors. The vehicle rides on the guideway, which is typically mounted to the frame the motors are mounted on. The motors interact with the magnet array to move and track the vehicle.

QuickStick High Thrust

For definitions of motors, motor controller, or inverters see MagneMotion Glossary of Terms, publication MMI-RM003.

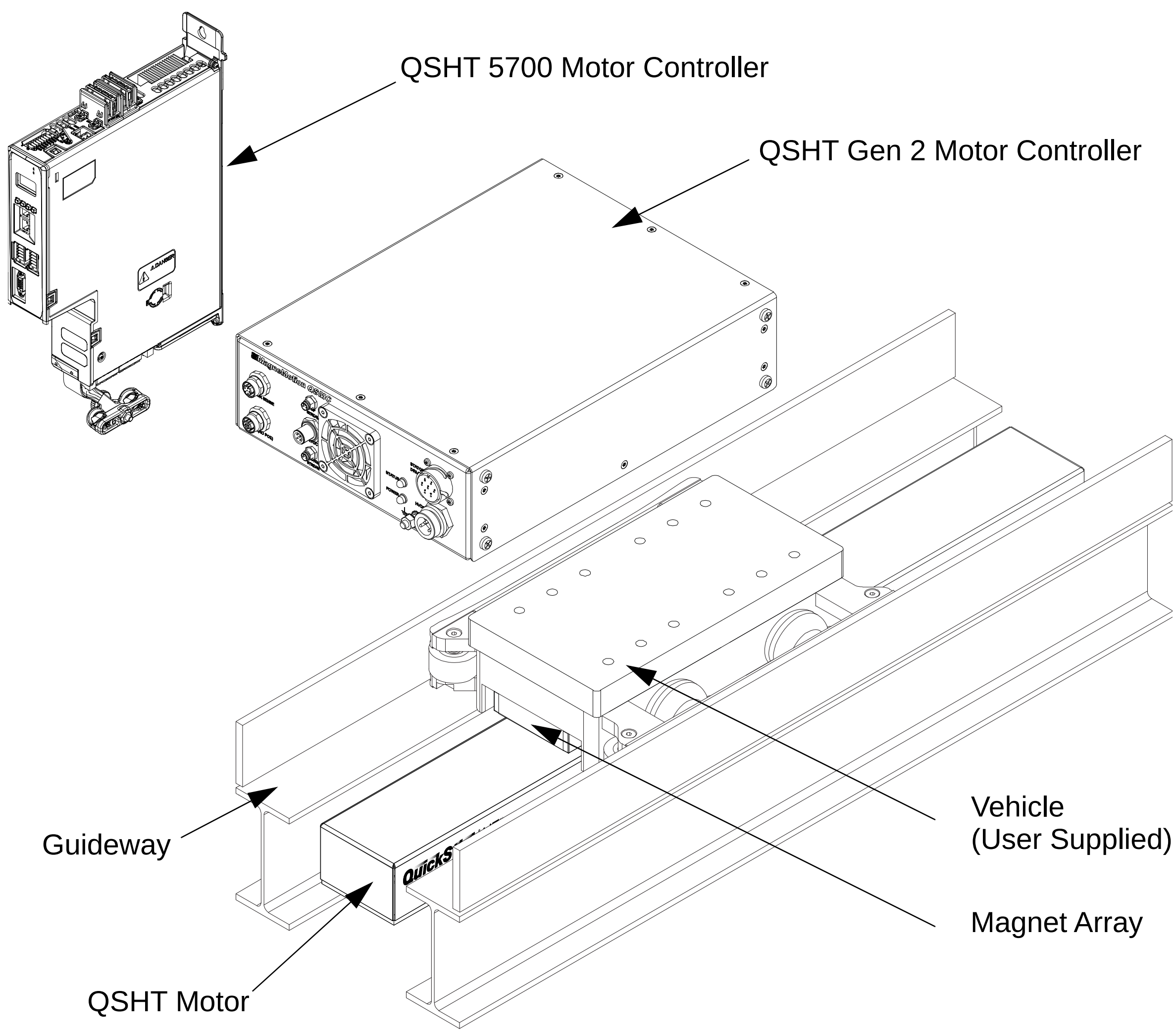


Figure 3-23: QuickStick High Thrust Motor

The QuickStick High Thrust motor consists of the motor with external control electronics (the 1 m motor is shown in Figure 3-23). Motors are mounted end-to-end on a user-supplied frame with a user-supplied guideway.

The motor controller for each QuickStick HT motor is located external to the QSHT motor. The motor controller monitors the motor and controls vehicle position. The external control electronics for the QSHT motors are the remotely located QSHT Gen 2 motor controller and QSHT 5700 motor controller. The controllers monitor and power the QSHT motors and control vehicle position. The controller also communicate vehicle position and other information to the other controllers in the path and to the node controller.

The vehicle that is used with the QuickStick HT motors is user-defined with a magnet array that is mounted on the surface closest to the motors. The vehicle rides on the guideway, which is typically mounted to the frame the motors are mounted on. The motors interact with the magnet array to move and track the vehicle.

Path Resets and Vehicle Deletion

This section covers reset path and vehicle delete operations. It describes the actions taken by the MMI LSM Transport Control System on path reset and delete vehicle commands. If an application requires path resets on a portion of an transport system, information on issues to be aware of and recommended reset procedures are discussed.

Reset All Paths

When the host controller sends a path reset with the path ID set to zero (0):

- The HLC sends a reset_path_request for all paths, causing the Node Controllers to put all paths in reset state.
- During each path's reset process, vehicle records in the motors for that path are deleted.
- Additionally, the HLC detects that all paths are in reset state at the same time. Subsequently, when the HLC detects that all paths are no longer in reset, it sends a reset_node_request to all Node Controllers clearing any vehicle node ownership record. Thus, records for all vehicles on the system are deleted.

Reset a Single Path

When the host controller sends a path reset to a specific path ID:

- The HLC sends a reset_path_request to the Node Controller responsible for the specified path.
- During the path's reset process, vehicle records in the motors for that path are deleted. Vehicles on the target path that are not involved in a node will be reliably deleted.

Delete a Vehicle

When the host controller sends a delete vehicle command:

- The vehicle records are purged from all motors, nodes, and paths.

Issues with Partial System Resets and Vehicle Deletes

1. If a vehicle is spanning adjacent paths at a node, its records will be deleted from the path being reset, but retained on the adjacent paths if they are not reset as well. This can cause startup to fail to properly locate the offending vehicle, possibly without a startup failure indication.

2. If a vehicle owns a node when the path is reset, the node will continue to be owned by the vehicle even after the vehicle records are deleted from the motors in the path.

3. Relay nodes are not owned but issue 1 above applies at relay nodes.

4. When a vehicle is deleted, permission to occupy the blocks that it owned can be granted to another vehicle. As a result a vehicle under command can collide with an unlocated vehicle. It is important that paths be suspended prior to deleting any vehicles on the path. It is also important that paths reporting unlocated vehicle faults be suspended.

Managing Individual Path Resets

To manage individual path resets, the host controller must use one of the following procedures.

Option 1:

1. Suspend the path that will be reset, termed the target path in this discussion.
2. Suspend any paths that share membership in the target path's upstream and downstream nodes.
3. Delete any vehicles that own the nodes at the end of the target path.
4. If there is a relay node at an end of the target path, delete any vehicle that is located on the opposite side of the relay node within ½ a vehicle length of the relay node.
5. Reset the target path.
6. Resume all suspended paths and ensure that they are all operational.
7. Startup the path that was reset and confirm the deleted vehicles are recovered prior to putting the system back in production.

Option 2:

1. Suspend the path that will be reset, termed the target path in this discussion.

2. Suspend any paths that share membership in the target path's upstream and downstream nodes.

3. If the target path's end is a relay node, also reset the path on the other side of the relay node. This process will create a string of target paths linked by relay nodes but terminated by switch, simple, or terminus nodes.

4. Delete any vehicles that own the nodes at either end of the string of target paths.

5. Reset the string of target paths.

6. Resume the suspended paths and ensure they are operational.

7. Startup the paths that were reset and confirm that the deleted vehicles are recovered prior putting the system back into production.

Node Type Descriptions and Usage

Nodes define the beginning of all paths and the connections between paths. See the MagneMotion System Configurator User Manual, publication MMI-UM046, for a detailed description of nodes, node types, and node parameters. There are several behaviors that are related to nodes that must be considered when moving vehicles.

- A vehicle is said to be navigating the node (or owning the node) when the node is within the motor permissions that are required for brick-wall headway for the vehicle (see Safe Stopping Distance on page 60).
- Vehicles are considered to have left the node (released ownership) once the center of the vehicle is 1/2 vehicle length plus 1/4 motor cycle past the node (end of the motor) in the direction of travel for the vehicle. These clearance distances are shown in Table 3-1 as the distance from the end of the motor to the end of the vehicle.

Table 3-1: Node Clearances

Motor Type	Clearance Distance
MagneMover LITE	12.35 mm
QuickStick	12 mm
QuickStick HT	30 mm

- Nodes create different loads on the node controllers depending on the type of node. Multi-port nodes types (Relay, merge, diverge, or moving path nodes) are nodes that connect multiple paths and require multiple connections to the node controller in serial systems. In Ethernet systems, these connections are managed via the network and do not require dedicated cables. These types of nodes result in the same amount of load on the node controller even if they have different numbers of paths that are connected to the node controller. Single-port node types (Simple, Overtravel, Terminus, and Gateway) present a negligible load. When using two Terminus Nodes to pass vehicles between paths, they count as one multi-port node due to the increased communication load. See the MagneMotion Node Controller Hardware User Manual, publication MMI-UM013, for additional information about loading.

- Node clearance distances and entry gate distances are used with switching nodes. These configurable anti-collision parameters define the clearance for vehicles entering and exiting the nodes. These clearances are used to avoid collisions with other vehicles on adjoining paths or with any mechanisms that are related to the node.

Relay nodes allow a vehicle to move from the end of a path to the beginning of another path. This motion can either be on one path creating a loop, or from one path to another extending the length of the string of motors.

Switching nodes allow a vehicle to move from the end of the path it is on to the beginning of another path through a switch mechanism. Switches in MagneMover LITE systems are preconfigured motor modules that provide either a diverge from one path to two paths or a merge from two paths to one path. The MM LITE™ switch modules use an integrated switch mechanism that does not require any user action as described in Merge and Diverge Nodes on page 108. Switches in QuickStick systems require an external switch mechanism that the host controller controls as described in Moving Path Node on page 114.

The transport systems support the following types of nodes. All node types support bidirectional motion through the node. Not all systems support all node types. Node types define their use and are presented in this section in order from least to most complex.

Simple Node

A Simple Node is used to begin a standalone path. This type of path is not connected to anything else at the upstream end and no vehicles enter or exit through the node. The upstream end of the path where the node is located requires one connection to the node controller. See Figure 3-24, where the shaded circle represents the Simple Node. Paths can also start from other node types.

Support for this node type is provided in:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

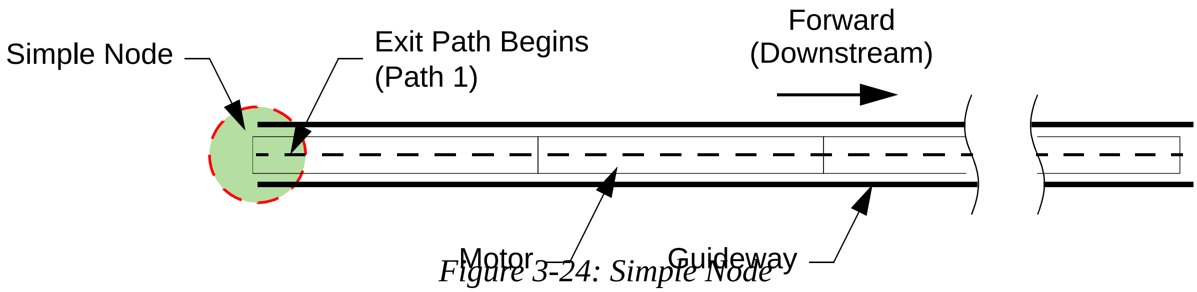


Figure 3-24: Simple Node

Node Operation

- Normal vehicle clearances apply.

Relay Node

A Relay Node is used to provide a simple connection where the end of one path connects to the start of another path. Relay Nodes are typically used to create a loop or a long continuous path. The two path ends require two connections to the same node controller. This type of node is used to break up large paths, create a simple loop, or separate E-stop/Interlock zones. See Figure 3-25, where the shaded circle represents the Relay Node.

Support for this node type is provided in:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

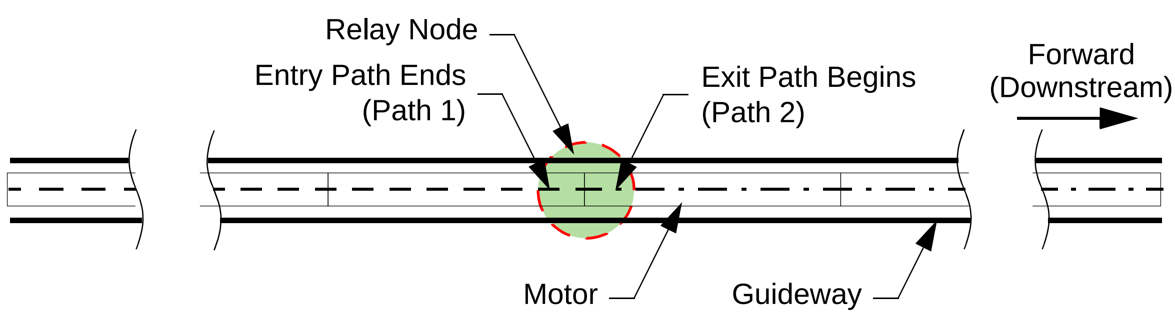


Figure 3-25: Relay Node

Node Operation

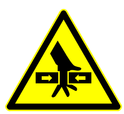
- Normal vehicle clearances apply across the node.
- Vehicles moving in the same direction can queue across the node.
- The move profile for the vehicle is maintained across the node so the vehicle crosses the node at a consistent velocity and acceleration.

Terminus Node

A Terminus Node is used on a path where vehicles want to enter or exit a path through a blind handoff. Terminus Nodes can be placed at either the upstream or downstream end of the path. See Figure 3-26, where the shaded circle represents the Terminus Node at the downstream end of the path. This section describes how to move a vehicle on or off the transport system through a Terminus Node, including all handshaking.

Handshaking between the HLC and the host controller is used to insert and extract vehicles on a path from equipment separate from the transport system. The host controller sets and clears signals that are affiliated with a Terminus Node using the Set Signal command. The last recorded signal state is viewed by sending a Status Request command for the appropriate Node ID. The Node Status response provides the current state of the signals.

NOTE: When a Terminus Node output state changes, the node controller automatically sends a Node Status response.



PINCH/CRUSH HAZARD: Moving mechanisms (vehicles) have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

Support for this node type is provided in:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

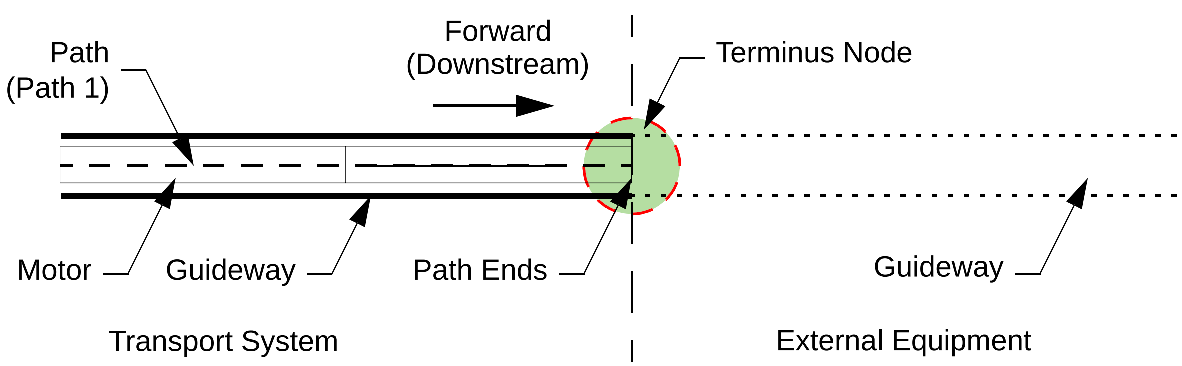


Figure 3-26: Terminus Node

Node Operation

- Normal vehicle clearances apply on the transport system side the node.
- Vehicles moving in the same direction can queue up to the node.
- Vehicles are not allowed to enter the node unless there is a full vehicle length of space free at the end of the path where the node is located.
- The move profile for the vehicle is maintained across the node so the vehicle transfers from the node to the user-supplied equipment at a consistent velocity and acceleration. Once the vehicle exits the transport system, the motors no longer provide any motive force or tracking of the vehicle

Handshake for Vehicle Entry onto the Transport System

This handshake is used for inserting a vehicle onto a transport system path at a Terminus Node. The handshake is designed to let only one vehicle at a time be inserted at a Terminus Node without risk of collision (simultaneous insertions at other termini are allowed). Once the vehicle enters the Terminus Node, it stops once half the vehicle is on the motor unless there is a new motion command. Only issue the new command after the **ENTRY_ALLOWED** signal goes High.

Signal Descriptions

Signal Name	Input/Output	Description
ENTRY_CLEAR	Output	Signal from the high-level controller that, last time it was checked, sufficient room to insert a vehicle at the Terminus Node exists. When the **ENTRY_CLEAR** signal is high, entry permission has not been granted, it is only a signal that the entry space was clear the last time it was checked. If a vehicle moves into this space in the meantime, or an error occurs, **ENTRY_ALLOWED** does not go high when entry is requested.
ENTRY_REQUESTED	Input	Signal from the host controller that a vehicle on the external equipment is ready for entry.
ENTRY_ALLOWED	Output	Signal from the high-level controller that motion permission has been acquired for the vehicle and the vehicle can enter.

Entry Handshake

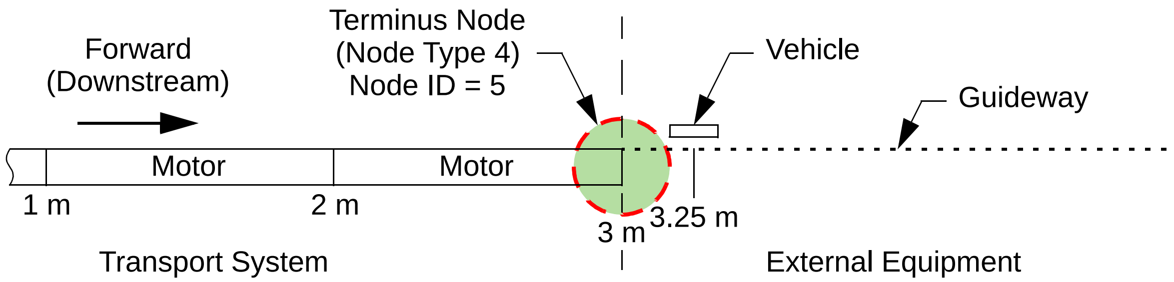


Figure 3-27: Terminus Node Entry Example

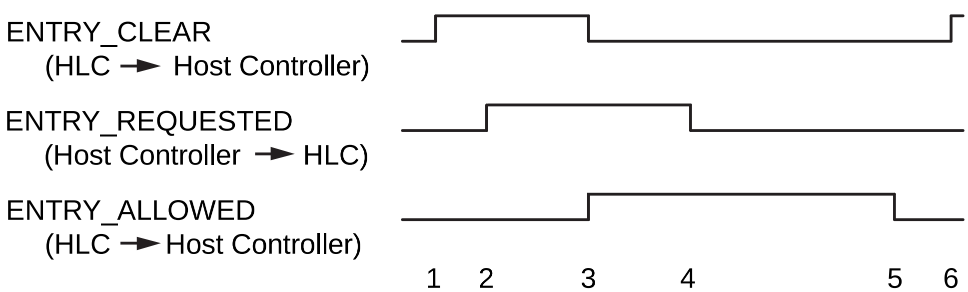


Figure 3-28: Entry Handshake Timing

1. Whenever the entry area is clear the high-level controller sets the **ENTRY_CLEAR** signal High (see Figure 3-28).

   NOTE: If the status of the entry area changes, the HLC changes the **ENTRY_CLEAR** signal.

   The host controller sends a Status Request command to the high-level controller to check the status of the Terminus Node (see Figure 3-27).

   `0xB5 02 0005`
   (status request, node 5)

   *The HLC responds with a Node Status message that acknowledges the receipt of the request and shows the node status (**ENTRY_CLEAR** = High (a vehicle can be inserted)).*

   `0xD3 0005 01 04 0000 10 00 00`
   (node status, node 5, present, terminus, no vehicle, entry clear, not used, not used)

2. The host controller sets the **ENTRY_REQUESTED** signal High using the Set Signal command to indicate that it has a vehicle ready to enter.

   NOTE: Assigning a Vehicle ID that is already in use causes the command to be rejected. Therefore, specifying a Vehicle ID of 0 is recommended, which commands the system to assign it.

`0xB6 0005 00 01 0000`
(set signal, node 5, signal = ENTRY_REQUESTED, level = High, vehicle 0)

*The HLC responds with a Command Status message that acknowledges the receipt of the signal change for node 5 setting **ENTRY_REQUESTED** = High and assigns a Vehicle ID.*

`0xD0 B6 00 0005 00 01 0003`
(command status, set signal, accepted, node 5, enter request, level = High, vehicle 3)

3. The HLC sets the **ENTRY_CLEAR** signal Low and sets the **ENTRY_ALLOWED** signal High to indicate permission for the vehicle to enter the path at the Terminus Node. This status shows that space has been reserved on the motor at the Terminus Node for the vehicle.

   The host controller sends a Status Request command to the high-level controller to check the Terminus Node status.

   `0xB5 02 0005`
   (status request, node 5)

   *The HLC responds with a Node Status message that acknowledges the receipt of the request and shows that a Vehicle ID has been assigned. This example shows Vehicle ID 3 is assigned, **ENTRY_CLEAR** = Low, **ENTRY_ALLOWED** = High, and **ENTRY_REQUESTED** = High.*

   `0xD3 0005 01 04 0003 21 00 00`
   (node status, node 5, present, terminus, vehicle 3, entry clear, not used, not used)

   The high-level controller is now ready to accept a motion command for the vehicle.

4. The host controller acknowledges entry permission has been granted by setting the **ENTRY_REQUESTED** signal Low using the Set Signal command.

   `0xB6 0005 00 00 0003`
   (set signal, node 5, signal = ENTRY_REQUESTED, level = Low, vehicle 3)

   *The HLC sends a Command Status message that acknowledges the receipt of the signal change for node 5. This example shows that **ENTRY_REQUESTED** = Low and provides the assigned Vehicle ID.*

   NOTE: In simulation mode, when **ENTRY_REQUESTED** is changed to Low by the Host Controller, the receiving Path takes control of the mover; the mover is considered to have arrived. The **ENTRY_ALLOWED** bit is changed to Low by the HLC; the mover is received.
   On a physical track, when the Host Controller sets **ENTRY_REQUESTED** to Low, the **ENTRY_ALLOWED** remains High until the mover is received.

`0xD0 B6 00 0005 00 00 0003`
(command status, set signal, accepted, node 5, enter request, level = Low, vehicle 3)

If desired, the host controller sends a Move Vehicle To Position command to move the vehicle to a position on the path and out of the Terminus Node. If a motion command is not sent, the vehicle stops as soon as it fully enters the transport system at the Terminus Node.

`0xB1 00000005 0003 00 40200000 0001 0000803F 0000803F`
(move 5, vehicle 3, unloaded and bidirectional, position 2.5 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

The HLC responds with a Command Status message that acknowledges the receipt of the motion command.

`0xD0 B1 00 00000005 0003 40200000 0001 0000803F 0000803F 02`
(command status, move to position, accepted, move 5, vehicle 3, position 2.5 m, path 1, acceleration 1.0 m/s^2, velocity 1.0 m/s, unloaded and backwards)

The vehicle is allowed to enter (the host controller commands the user-supplied equipment to insert the vehicle onto the transport system).

NOTE: If a vehicle motion command is sent before the vehicle is inserted, the vehicle starts moving under that command as soon as it is inserted. To have a motion command queued this way, requires specifying the Vehicle ID when the Set Signal command is sent in Step 4.

While the vehicle is in motion through the node, the host controller can send a Status Request command to the HLC to check the Terminus Node status.

`0xB5 02 0005`
(status request, node 5)

The HLC responds with a Node Status message that acknowledges the receipt of the request. This example shows that vehicle 3 is in the node, ***ENTRY_CLEAR*** *= Low,* ***ENTRY_ALLOWED*** *= High, and* ***ENTRY_REQUESTED*** *= Low.*

`0xD3 0005 01 04 0003 20 00 00`
(node status, node 5, present, terminus, vehicle 3, entry not clear, not used, not used)

5. Once the vehicle clears the node, the HLC confirms that the entry request is over (**ENTRY_REQUESTED** is Low) and the vehicle has entered by setting the **ENTRY_ALLOWED** signal Low.

   Upon completion of the motion command, the HLC sends a Command Status message that shows the command has completed successfully.

```
0xD0 B1 80 00000005 0003 40200000 0001 00000000
00000000 00
```
(command status, move to position, completed successfully, move 5, vehicle 3, position 2.5 m, path 1, acceleration 0 m/s^2, velocity 0 m/s, unloaded and bidirectional)

6. The host controller must wait for the **ENTRY_ALLOWED** signal to be Low and the **ENTRY_CLEAR** signal to be High before sending another entry request.

Handshake for Vehicle Exit from the Transport System

This handshake is used for removing a vehicle from a transport system path at a Terminus Node. The handshake is designed to let only one vehicle at a time leave a path through a Terminus Node without risk of collision (simultaneous exits at other termini are allowed). The vehicle exiting at the Terminus Node must have been given a command to move beyond the end of the path in order for the vehicle to exit. The vehicle then waits on the motor for the appropriate handshake to take place before exiting. The destination in the command must place the entire vehicle (not just the center of the vehicle) beyond the end of the motor.

Signal Descriptions

Signal Name	Input/Output	Description
EXIT_ALLOWED	Input	Signal from the host controller that the user-supplied equipment is ready to receive a vehicle from the Terminus Node (the motor is allowed to push the vehicle off the path).
EXITING	Output	Signal from the high-level controller that motion permission has been acquired for the vehicle and the vehicle is exiting.

Exit Handshake

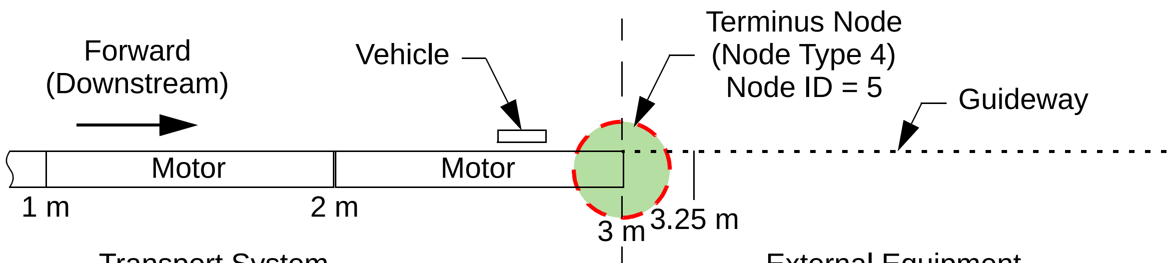


Figure 3-29: Terminus Node Exit Example

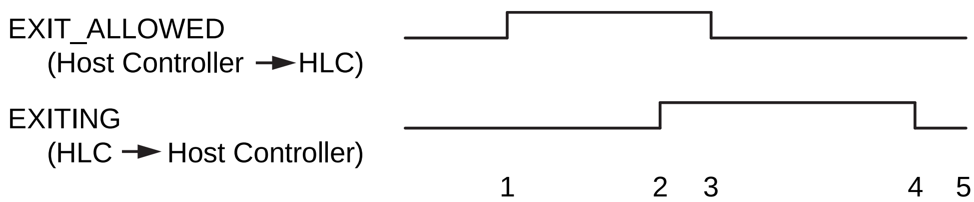


Figure 3-30: Exit Handshake Timing

1. The host controller sets **EXIT_ALLOWED** High for the Terminus Node (see Figure 3-29) using the Set Signal command to signal that it is ready for a vehicle to exit. When high, this flag specifies space is available for a vehicle on the user-supplied equipment beyond the Terminus Node of a transport system.

    `0xB6 0005 01 01 0000`
    (set signal, node 5, signal = EXIT_ALLOWED, level = High, no vehicle specified)

    *The HLC responds with a Command Status message that acknowledges the receipt of the signal change for node 5 setting **EXIT_ALLOWED** = High.*

    `0xD0 B6 00 0005 01 00 0000`
    (command status, set signal, accepted, node 5, Exit Allowed, level = High, no vehicle)

    The host controller sends a Status Request command to the high-level controller to check the Terminus Node status before commanding the vehicle to enter the node.

    `0xB5 02 0005`
    (status request, node 5)

    *The HLC responds with a Node Status message that acknowledges the receipt of the request and shows that **EXIT_ALLOWED** = High, and **EXITING** = Low.*

    `0xD3 0005 01 04 0000 02 00 00`
    (node status, node 5, present, terminus, no vehicle, *EXIT_ALLOWED = High, and EXITING = Low*, not used, not used)

    The host controller sends a Move Vehicle To Position to move the vehicle past the end of the path. The minimum commanded distance past the end of the motor must be at least one vehicle length.

    `0xB1 00000004 0003 01 40500000 0001 0000803F 0000803F`
    (move 4, vehicle 3, unloaded and forward, position 3.25 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

    The HLC responds with a Command Status message that acknowledges the receipt of the motion command.

`0xD0 B1 00 00000004 0003 40500000 0001 0000803F 0000803F 01`
(command status, move to position, accepted, move 4, vehicle 3, position 3.25 m, path 1, acceleration 1.0 m/s^2, velocity 1.0 m/s, unloaded, and forwards)

NOTE: This motion command never completes (return a 0xD0 B1 80...) as the vehicle has left the transport system.

2. The high-level controller notes that the **EXIT_ALLOWED** signal is High and has a vehicle that has begun exiting and sets the **EXITING** signal High.

   While the vehicle is in motion through the node, the host controller can send a Status Request command to the high-level controller to check the Terminus Node status.

   `0xB5 02 0005`
   (status request, node 5)

   The HLC responds with a Node Status message that acknowledges the receipt of the request and shows the node status.

   `0xD3 0005 01 04 0003 42 00 00`
   (node status, node 5, present, terminus, vehicle 3, *EXIT_ALLOWED = High, and EXITING = High*, not used, not used)

3. The host controller acknowledges that a vehicle is leaving (**EXITING** signal has gone High) by setting the **EXIT_ALLOWED** signal Low using the Set Signal command. The host controller commands the user-supplied equipment to remove the vehicle from the transport system.

   `0xB6 0005 01 00 0000`
   (set signal, node 5, signal = EXIT_ALLOWED, level = Low, no vehicle)

   *The HLC responds with a Command Status message that acknowledges the receipt of the signal change for node 5 setting **EXIT_ALLOWED** = Low.*

   `0xD0 B6 00 0005 01 01 0000`
   (command status, set signal, accepted, node 5, Exit Allowed, level = Low, no vehicle)

4. The high-level controller confirms that the **EXIT_ALLOWED** signal is Low and the vehicle has left by setting the **EXITING** signal Low.

5. Once the vehicle has been moved out of the node and onto the user-supplied equipment, the host controller checks the Terminus Node status using the Status Request command.

   `0xB5 02 0005`
   (status request, node 5)

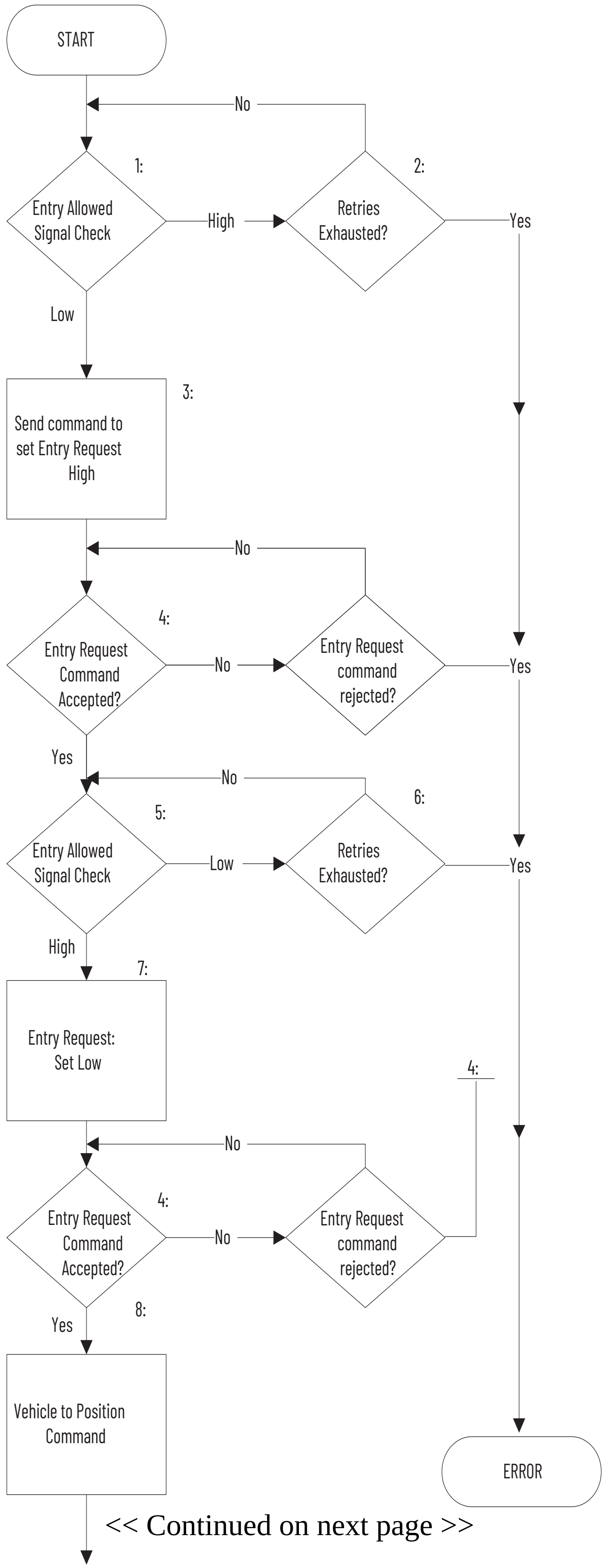
The HLC responds with a Node Status message that acknowledges the receipt of the request and shows that the node is present and available for transfer.

`0xD3 0005 01 04 0000 10 00 00`
(node status, node 5, present, terminus, no vehicle, ENTRY_CLEAR = High, EXIT_ALLOWED = Low, not used, not used)

6. The host controller can now set **EXIT_ALLOWED** High again when the user-supplied equipment has space for another vehicle.

Example of Two Back to Back Terminus Nodes

One terminus node is receiving a vehicle from another terminus node.



<< Continued on next page >>

<u>Explanation of steps in the flow chart:</u>

1: Entry Allowed (high) indicated that permission has been given for a vehicle may enter the node.

2: For Retries Exhausted you need to determine how long to wait for the node to be clear. Entry allowed will be high when a vehicle has been given permission to enter and while the vehicle is in the process of entering. Entry Allowed will be low when a vehicle has completed the entry into the node.

3: This is the command from the Host to the HLC to set the Entry Request bit high for the terminus node

4: This is the standard response for command accepted. This should always be received. If there is an error in the format of the command then it will be rejected.

5: Entry Allowed (high) indicated that permission has been given for a vehicle may enter the node.

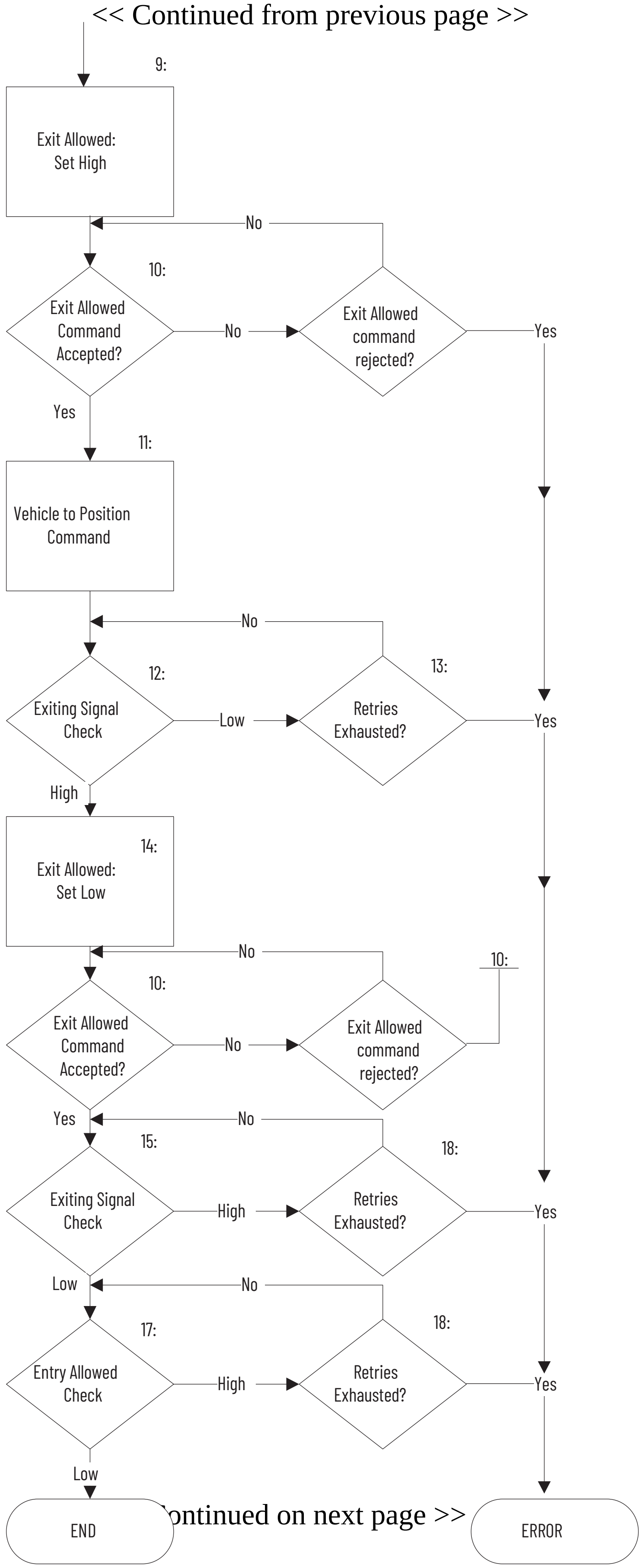
6: For Retries Exhausted you need to determine how long to wait for the node to be clear. Entry allowed will be high when a vehicle has been given permission to enter and while the vehicle is in the process of entering. Entry Allowed will be low when a vehicle has completed the entry into the node.

7: This is to confirm that Entry_Allowed was set high by the system setting Entry_ Requested to low.

This completed the set-up for a vehicle to enter a terminus node.

8: This command is to order the vehicle which is entering the terminus node to it's target position. This can be done so when the vehicle starts to enter the terminus node it has a destination. Otherwise the vehicle will stop once entered until a move command is issued.

<< Continued from previous page >>



<< Continued on next page >>

Explanation of steps in the flow chart (cont.):

9: This is the host commanding the terminus node that there is space ready for the vehicle to move beyond the terminus end of the motor. This is the host giving permission that it is OK for the vehicle to leave.

10: This is the standard response for command accepted. This should always be received. If there is an error in the format of the command then it will be rejected.

11: This command is sending the vehicle off the end of the terminus EXIT node.

12: This will be high while the vehicle has permission to move, is moving or still owns the node.

13: For Retries Exhausted you need to determine how long to wait for the node to be clear. Exiting will be high when a vehicle has been given permission to exit and while the vehicle is in the process of exiting. Exiting will be low when a vehicle has completed the exit from the node.

14: This is the host acknowledging that Exiting has been set high and the vehicle has permission to leave.

15: This is to check if the vehicle has completed the exit and no longer owns the node.

16: This is a timer to see if there is a reason that the vehicle is not exiting. You should also check the vehicle status to be sure it is moving.

17: This confirms that the vehicle has finished entering the Entry Allowed node.

18: For Retries Exhausted you need to determine how long to wait for the node to be clear. Entry allowed will be high when a vehicle has been given permission to enter and while the vehicle is in the process of entering. Entry Allowed will be low when a vehicle has completed the entry into the node.

Gateway Node

A Gateway Node is used to connect a path in one Control Group to a path in another Control Group within a larger transport system. Each Control Group has separate high-level controllers and can have separate host controllers. The end of the path where the node is located requires one connection to the node controller. See Figure 3-31, where the shaded circles represent the Gateway Nodes.

NOTE: The Gateway Nodes definitions in the Node Controller Configuration File for each Control Group must reference the matching Gateway Node in the other Control Group to achieve proper vehicle transfer.

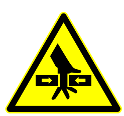
Only two Gateway Nodes can be configured per node controller. One at an upstream path end and one at a downstream path end.

If the Vehicle ID of the vehicle entering a Control Group through a Gateway Node exists in that Control Group, entry is refused.

Both node controllers that are connected to the Gateway Node must be connected to the same network subnet.

The HLC can run on the same hardware as the node controller (for example, an NC LITE), but may need to be on another NC depending on the communications load.

Handshaking through the local subnet is used by the node controllers in each Control Group responsible for the Gateway Node to pass vehicles from Control Group to Control Group.



PINCH/CRUSH HAZARD: Moving mechanisms (vehicles) have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

Support for this node type is provided in:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

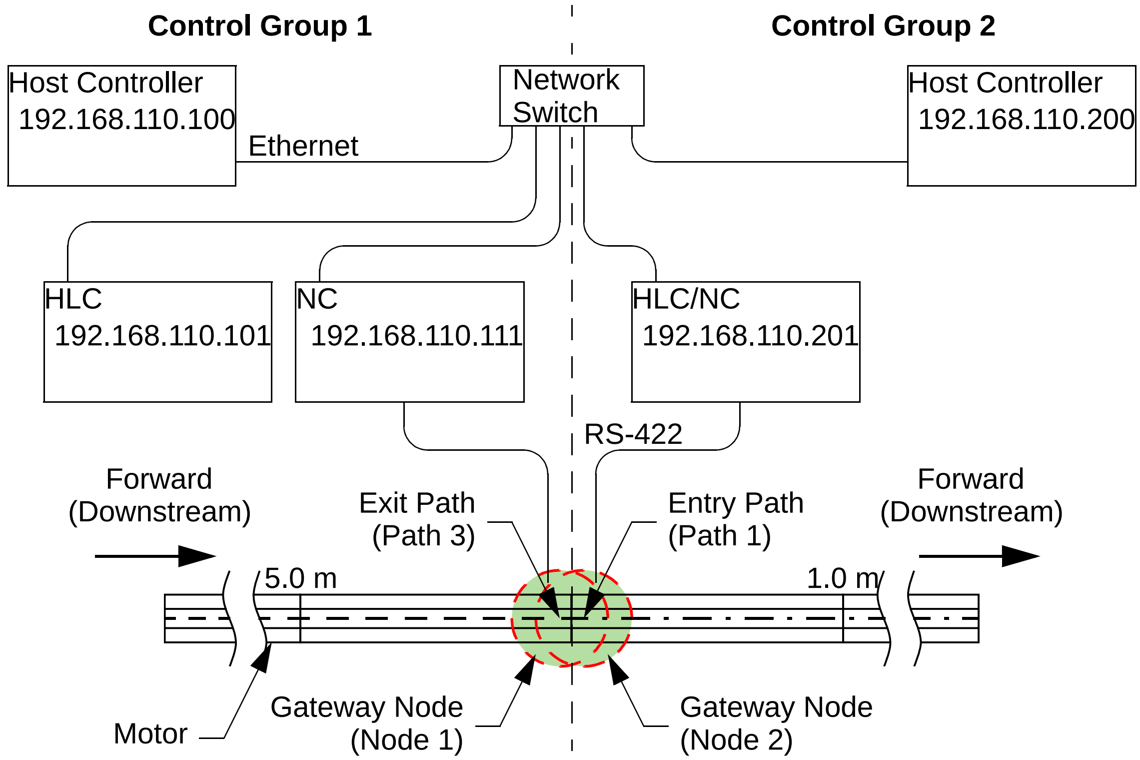


Figure 3-31: Gateway Node

Node Operation

- Normal vehicle clearances apply across the node.
    - As a vehicle approaches the Gateway Node, the node controller requests permission from the target HLC to enter the other Control Group. As long as the path in the target Control Group is operational and no duplicate vehicle ID exists, entry is granted.
    - If entry is granted, the vehicle continues to move through the node.
    - If entry is not granted, the vehicle decelerates and stops before crossing the node. The leading edge of the vehicle stops at the node boundary.
- Vehicles moving in the same direction can queue up to and through the node.
    - If multiple vehicles are sent to the node, vehicles can queue before, across, and through the node. The queuing is similar to a Relay Node.
- The move profile for the vehicle is maintained across the node so the vehicle crosses the node at a consistent velocity and acceleration.
    - The vehicle ID is maintained across the Gateway Node if the sending and receiving node controllers are configured in the same HLC control group.
    - If the destination is just beyond the node, the vehicle decelerates to follow the move profile in the same manner as on any path.

- The default destination for the Gateway Node is defined in the Node Controller Configuration File for entering vehicles.
    - The Node Controller Configuration Files for the Control Groups that share a Gateway Node must have default destination paths and positions defined. These positions become the default locations for any vehicle that enters the node in either direction.
    - The host controller can supersede the default position with a vehicle motion command as soon as the vehicle record is detected.
- Node controllers coordinate exit and entry.
    - The host controller does not control the access through the Gateway Node, the node controllers in the two Control Groups handle all access handshaking. Using the node controllers to handle all handshaking eliminates the need to have entry/exit requests coordinated by the host controllers when a vehicle moves through the pair of Gateway Nodes.
- The host controller detects the Gateway Node ownership when a vehicle enters it via Node Status messages, which provide notification that a vehicle is entering.
    - Once target requests are granted, a vehicle record with the current Vehicle ID is generated and is accessible to the host controller.
    - Once the vehicle record exists, the host controller can command the vehicle to a destination. This motion command is the same as any other vehicle command (move to position or move to station).

Moving Downstream Through a Gateway Node

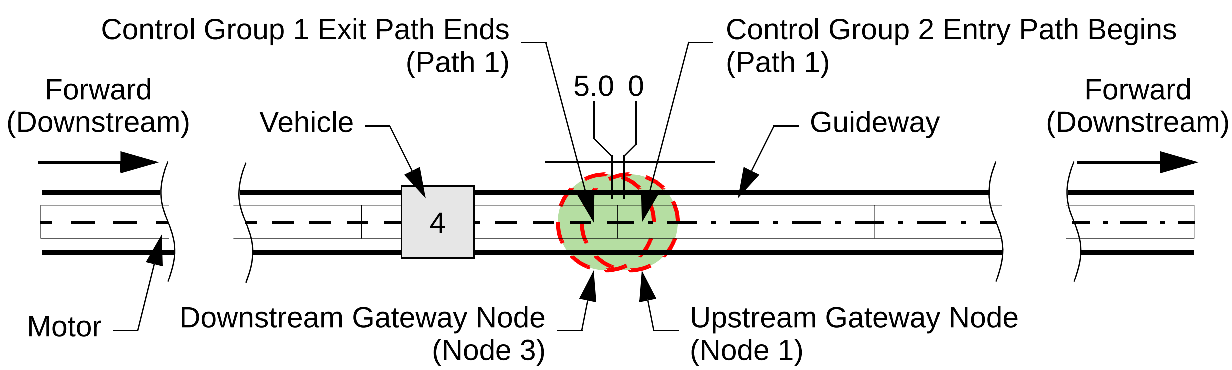


Figure 3-32: Gateway Node Motion

1. The host controller issues a vehicle motion command that moves the vehicle so the leading edge of the vehicle crosses the border of the Gateway Node.

   `0xB1 00000001 0004 11 0000A040 0001 0000803F 0000803F`
   (move 1, vehicle 4, loaded and forward, position 5.0 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

The HLC responds with a Command Status message that acknowledges the receipt of the motion command.

`0xD0 B1 00 00000001 0004 0000A040 0001 0000803F 0000803F 11`
(command status, move to position, accepted, move 1, vehicle 4, position 5.0 m, path 1, acceleration 1 m/s^2, velocity 1 m/s, loaded and forward)

Once the vehicle has permission to enter the node from the target Control Group the move continues through the node. When the target position for the vehicle is past the Gateway Node, the following occurs:

- The node controller in the first Control Group responsible for the Gateway Node deletes the vehicle record for the vehicle and changes the Vehicle ID for the node to idle.
- The node controller in the second Control Group responsible for the Gateway Node creates a vehicle record for the vehicle and changes the Vehicle ID for the node to the entering vehicle.

NOTE: If the Vehicle ID of the vehicle entering a Control Group through a Gateway Node exists in that Control Group, entry is refused and the vehicle stops before the node border.

2. To prepare for vehicle entry through the Gateway Node, the host controller for the second Control Group monitors the status of the Gateway Node for a new Vehicle ID.

   `0xB5 02 0001`
   (status request, node 1)

   The HLC responds with a Node Status message that acknowledges the receipt of the request and shows that the node is present and vehicle 4 is in the node.

   `0xD3 0001 01 09 0004 00 00 00`
   (node status, node 1, present, gateway, vehicle 4, not used, not used, node enabled)

3. Once a new Vehicle ID is detected, the host controller issues a vehicle motion command (Move to Position or Move to Station) for the new vehicle.

   `0xB1 00000027 0004 11 0000803F 0001 0000803F 0000803F`
   (move 27, vehicle 4, loaded and forward, position 1.0 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

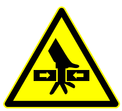
   NOTE: If no command is issued before the vehicle completes its transfer, the vehicle travels to the default destination defined in the Node Controller Configuration File for the first Control Group and stops.

   The motion command completion message is transmitted to the new transport system (where the vehicle is located) not the system that issued the command.

Merge and Diverge Nodes

Merge or Diverge nodes allow vehicles to move from one path to another. The downstream orientation of the switch mechanism determines if the switch is configured as either a Merge Node or a Diverge Node. The paths in a Merge Node are called Straight Entry, Curve Entry, and Merged Exit as shown in Figure 3-33. The paths in a Diverge Node are called Single Entry, Straight Exit, and Curve Exit as shown in Figure 3-34.

NOTE: Merge and Diverge nodes are only available on MagneMover LITE systems.



PINCH/CRUSH HAZARD: Moving mechanisms (vehicles and switch mechanisms) have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

When the motion from one path to another path is in the same direction (for example, motion on all paths is forward, or downstream), no special considerations are required. This type of motion (forward or backward through a switch) is the normal use for a switch.

When the motion from one path to another path is not in the same direction, the motion must be configured as a pair of moves. Each of these moves is a separate command where the motion is in only one direction. This type of motion (reversing the direction of a vehicle through a switch) must be used carefully. Motion permissions are granted upon request and another vehicle could get permission through the switch before the original vehicle requests permission for its second move. Permissions are granted up to the point where the same block is being requested for both vehicles. Once both vehicles are requesting the same block, motion stops and both vehicles report they are obstructed.

NOTE: Even if a motion command is configured as bidirectional, two moves are required as a vehicle cannot change direction in the middle of a move. Use caution when attempting to create motion profiles of this type as there is nothing in the system to prevent a deadlock. Deadlocks can occur when, due to permissions already granted, no new permissions can be given and the vehicles that are involved are not able to move.

Merge Node

A Merge Node is where the downstream ends of two paths merge into the upstream end of a third path. This type of node is used to merge paths or create multiple loops. See Figure 3-33, where the shaded circle represents the Merge Node.

Support for this node type is provided in MagneMover LITE transport systems.

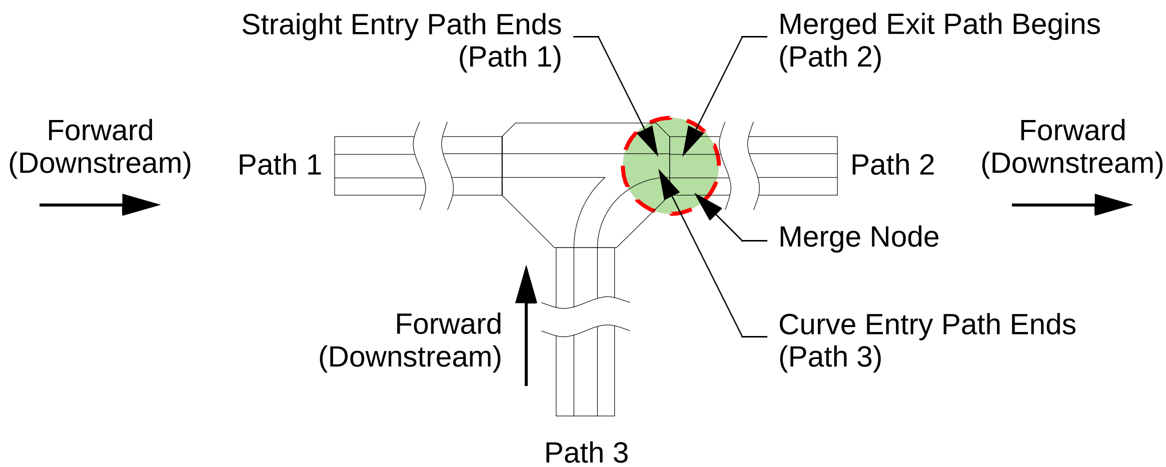


Figure 3-33: Merge Node

Node Operation

- Normal vehicle clearances apply across the node.
- Vehicles moving in the same direction can queue across the node.
- The move profile for the vehicle is maintained across the node so the vehicle crosses the node at a consistent velocity and acceleration.

Reversing Through a Merge Node

Reversing the direction of a vehicle through an MM LITE Merge Node (for example, path 1 to path 3 as shown in Figure 3-33) requires two moves. The first move is forward from path 1 to path 2. In MagneMover LITE systems, the switch mechanism is automatically positioned for the appropriate entry path. The second move is backward from path 2 to path 3 where motion continues backwards. As part of the second move in MagneMover LITE systems, the switch mechanism is automatically positioned for the appropriate entry path.

NOTE: The first motion command must move the vehicle far enough past the switch to clear the switch mechanism and any configured clearance distance.

`0xB1 00000002 0004 01 0000803E 0002 0000803F 0000803F`
(move 2, vehicle 4, unloaded and forward, position 0.25 m, path 2, acceleration 1 m/s^2, velocity 1 m/s)

The HLC responds with a Command Status message that acknowledges the receipt of the motion command.

`0xD0 B1 00 00000002 0004 0000803E 0002 0000803F 0000803F 01`
(command status, move to position, accepted, move 2, vehicle 4, position 0.25 m, path 2, acceleration 1 m/s^2, velocity 1 m/s, unloaded and forward)

Once vehicle motion completes, the HLC sends a Command Status message that indicates completion of the motion command.

`0xD0 B1 80 00000002 0004 0000803E 0002 00000000 00000000 01`
(command status, move to position, completed successfully, move 2, vehicle 4, position 0.25 m, path 2, acceleration 0 m/s^2, velocity 0 m/s, unloaded and forward)

Once the first move has completed, the second move can be commanded.

`0xB1 00000003 0004 02 0000B840 0003 0000803F 0000803F`
(move 3, vehicle 4, unloaded and backward, position 5.75 m, path 3, acceleration 1 m/s^2, velocity 1 m/s)

The HLC responds with a Command Status message that acknowledges the receipt of the motion command.

`0xD0 B1 00 00000003 0004 0000B840 0003 0000803F 0000803F 02`
(command status, move to position, accepted, move 3, vehicle 4, position 5.75 m, path 3, acceleration 1 m/s^2, velocity 1 m/s, unloaded and backward)

Once vehicle motion completes, the HLC sends a Command Status message that indicates completion of the motion command.

`0xD0 B1 80 00000003 0004 0000B840 0003 00000000 00000000 02`
(command status, move to position, completed successfully, move 3, vehicle 4, position 5.75 m, path 3, acceleration 0 m/s^2, velocity 0 m/s, unloaded and backward).

Diverge Node

A Diverge Node is where the downstream end of a path splits into the upstream ends of two other paths. This type of node is used to split a path or create multiple loops. See Figure 3-34, where the shaded circle represents the Diverge Node.

Support for this node type is provided in MagneMover LITE transport systems.

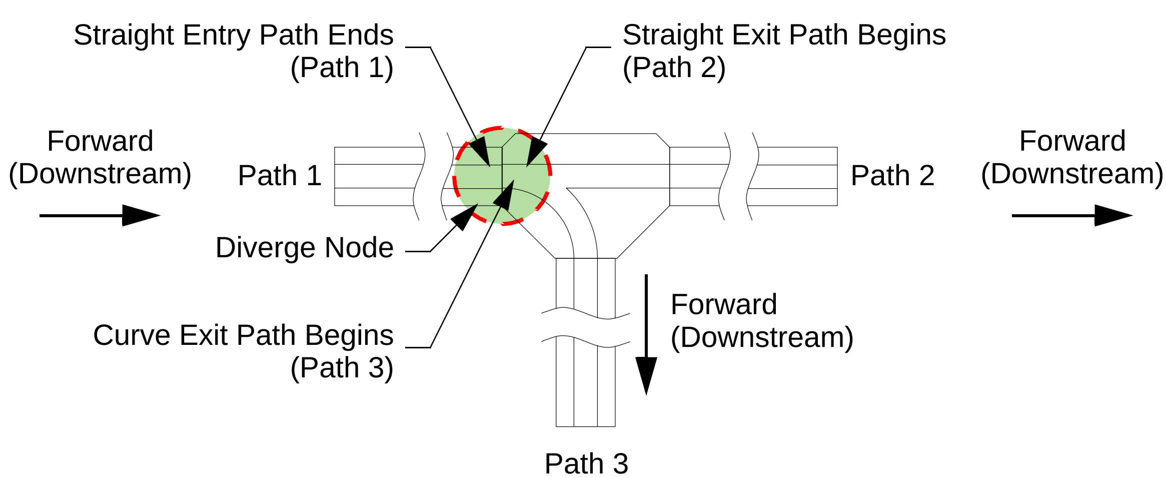


Figure 3-34: Diverge Node

Node Operation

- Normal vehicle clearances apply across the node.
- Vehicles moving in the same direction can queue across the node.
- The move profile for the vehicle is maintained across the node so the vehicle crosses the node at a consistent velocity and acceleration.

Reversing Through a Diverge Node

Reversing the direction of a vehicle through an MM LITE Diverge Node (for example, path 2 to path 3 as shown in Figure 3-34) requires two moves. The first move is backward from path 2 to path 1. In MagneMover LITE systems, the switch mechanism is automatically positioned for the appropriate exit path. The second move is forward from path 1 to path 3 where motion continues forwards. As part of the second move in MagneMover LITE systems, the switch mechanism is automatically positioned for the appropriate exit path.

NOTE: The first motion command must move the vehicle far enough to clear the switch mechanism and any configured clearance distance.

`0xB1 00000002 0004 02 0000B840 0001 0000803F 0000803F`
(move 2, vehicle 4, unloaded and backward, position 5.75 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

The HLC responds with a Command Status message that acknowledges the receipt of the motion command.

`0xD0 B1 00 00000002 0004 0000B840 0001 0000803F 0000803F 02`
(command status, move to position, accepted, move 2, vehicle 4, position 5.75 m, path 1, acceleration 1 m/s^2, velocity 1 m/s, unloaded and backward)

Once vehicle motion completes, the HLC sends a Command Status message that indicates completion of the motion command.

`0xD0 B1 80 00000002 0004 0000B840 0001 00000000 00000000 02`
(command status, move to position, completed successfully, move 2, vehicle 4, position 5.75 m, path 1, acceleration 0 m/s^2, velocity 0 m/s, unloaded and backward)

Once the first move has completed, the second move can be commanded.

`0xB1 00000003 0004 01 0000C03F 0003 0000803F 0000803F`
(move 3, vehicle 4, unloaded and forward, position 1.5 m, path 3, acceleration 1 m/s^2, velocity 1 m/s)

The HLC responds with a Command Status message that acknowledges the receipt of the motion command.

`0xD0 B1 00 00000003 0004 0000C03F 0003 0000803F 0000803F 01`
(command status, move to position, accepted, move 3, vehicle 4, position 1.5 m, path 3, acceleration 1 m/s^2, velocity 1 m/s, unloaded and forward).

Once vehicle motion completes, the HLC sends a Command Status message that indicates completion of the motion command.

`0xD0 B1 80 00000003 0004 0000C03F 0003 00000000 00000000 01`
(command status, move to position, completed successfully, move 3, vehicle 4, position 1.5 m, path 3, acceleration 0 m/s^2, velocity 0 m/s, unloaded and forward).

Merge-Diverge Node

A Merge-Diverge Node is used where the downstream ends of two paths connect to the upstream ends of two other paths. A merge-diverge is created using two switches, where the common path of each switch is connected together. See Figure 3-35, where the shaded circle represents the Merge-Diverge Node.

Support for this node type is provided in MagneMover LITE transport systems using high-payload switches only.

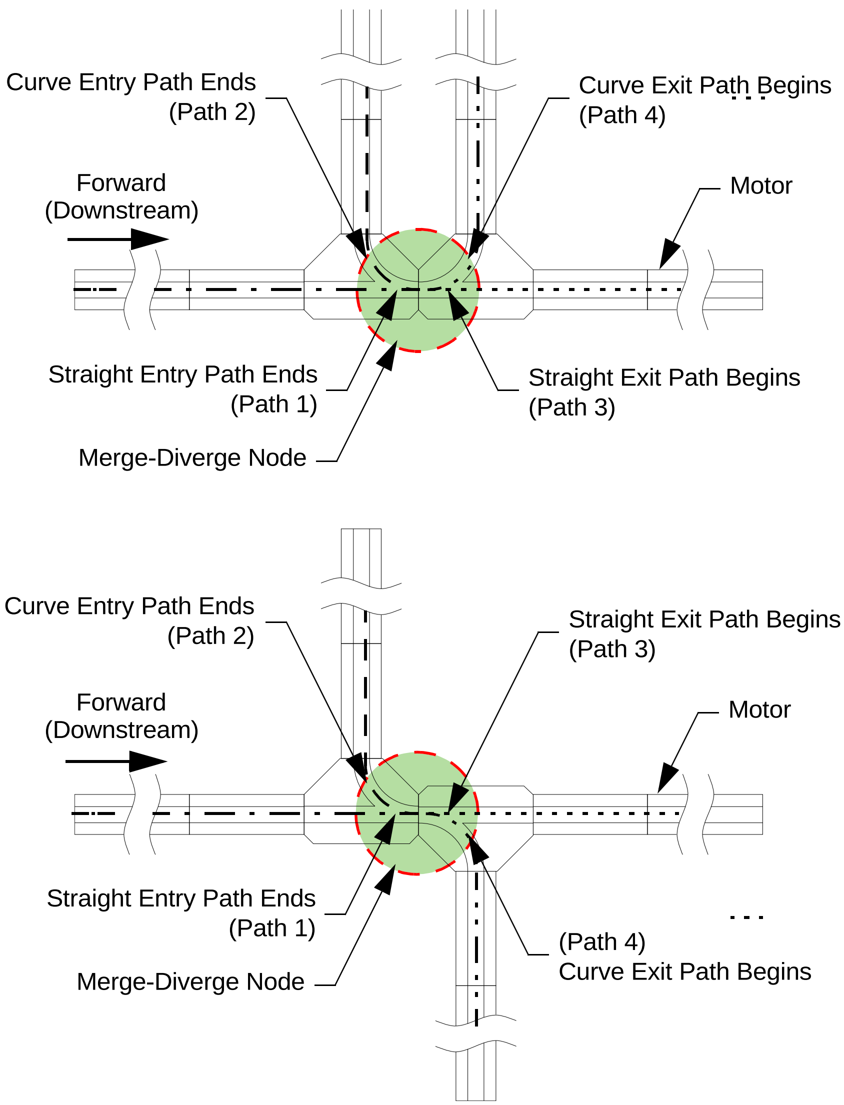


Figure 3-35: Merge-Diverge Node

Node Operation

- Normal vehicle clearances apply across the node.
- Vehicles moving in the same direction can queue across the node.
- The move profile for the vehicle is maintained across the node so the vehicle crosses the node at a consistent velocity and acceleration.

Overtravel Node

An Overtravel Node is used to permit part of a vehicle to move past the end of the motor at the end of a path. Additional support structure (guideway) for the vehicle must be provided, or the supports (wheels) for the vehicle must not be allowed to move past the end of the path. See Figure 3-36 and Figure 3-37, where the shaded area represents Overtravel Nodes.

The overtravel node is used for the following applications:

- Providing space for vehicle motion during startup (Figure 3-36).
- Allowing motion of a vehicle that is longer than the path (Figure 3-37).

Support for this node type is provided in:

- QuickStick transport systems.
- QuickStick HT transport systems.

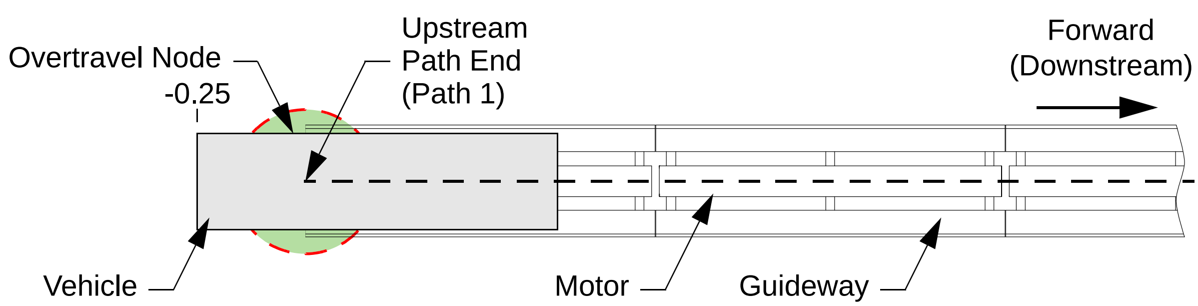


Figure 3-36: Overtravel Node, Startup Top View

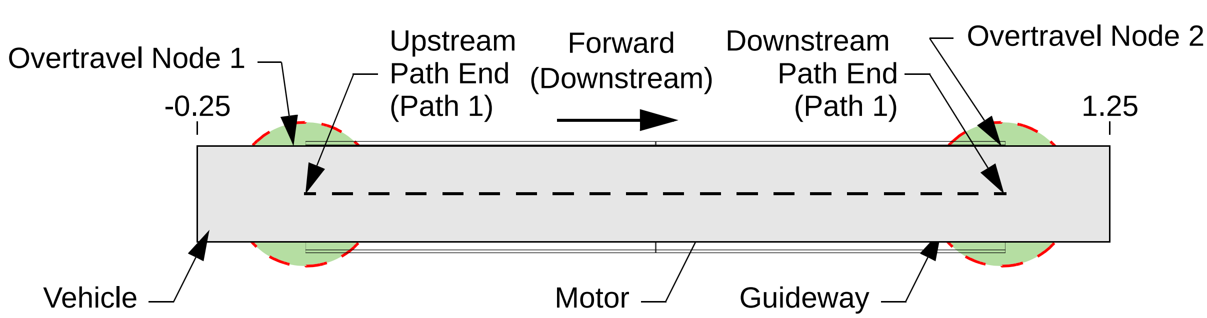


Figure 3-37: Overtravel Node, Extended Vehicle Top View

Node Operation

- Normal vehicle clearances apply.

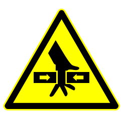
Moving Path Node

A Moving Path Node is used to connect the ends of multiple paths using a path that a user-supplied drive mechanism moves. This mechanism can be any user-supplied mechanism, including a QuickStick path.The Moving Path Node enables vehicles to move between multiple guideways. The paths in the Moving Path Node are called entry and exit paths. Either type of path can be moved but the connecting paths must all be of the other type (that is, entry paths move and all exit paths are fixed). Up to 12 paths can be configured as either entry paths or exit paths. See Figure 3-38, where the shaded circle represents the Moving Path Node.

Use of the Moving Path Node requires the host controller to command the drive mechanism to position one of the moving paths so that it aligns with one of the fixed paths. The host controller must then use the MP Link Command to connect the two paths to allow vehicle motion. Once the vehicle has moved beyond the node, the Moving Path must be unlinked before it can be moved to a new position.

Support for this node type is provided in:

- QuickStick transport systems.
- QuickStick HT transport systems.



PINCH/CRUSH HAZARD: Moving mechanisms (vehicles and moving paths) have no obstruction sensors.

Do not operate the transport system or any switching mechanisms without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

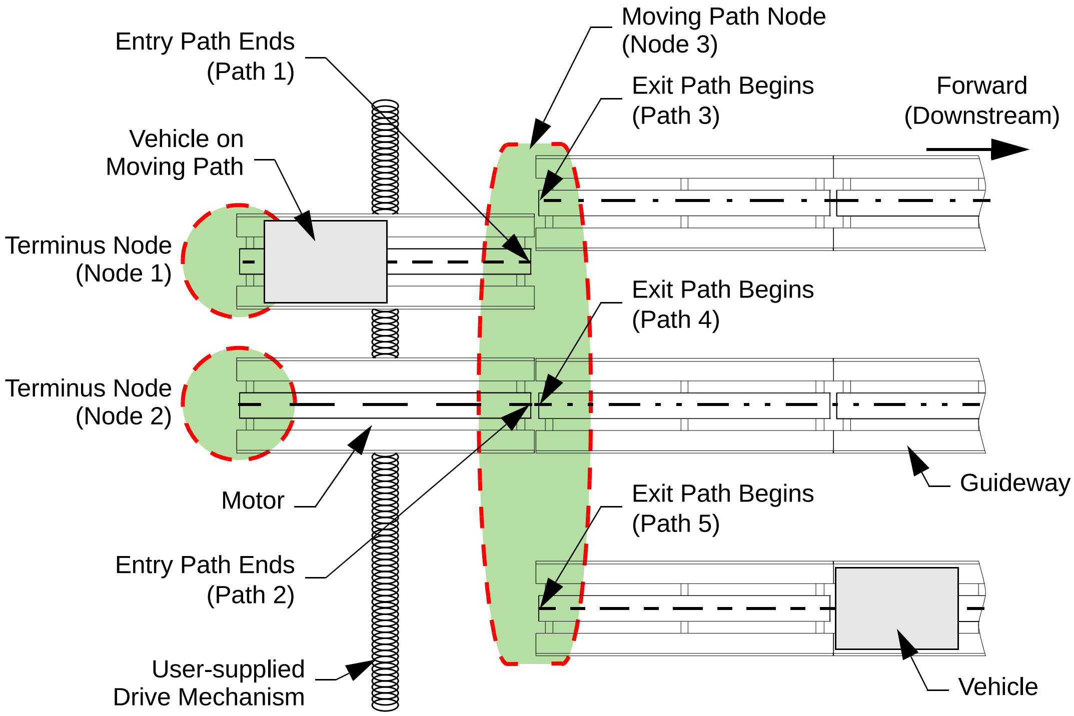


Figure 3-38: Moving Path Node

This mechanism can either move a section of guideway and motors with vehicles on them as shown in Figure 3-39 or a mechanical switch that diverts vehicle motion.

Node Operation

- Entry gates keep vehicles from entering the protected area around the path junctions unless the linked path provides the shortest route to its destination. To be granted permission to pass an entry gate, a vehicle must be under an order to a destination beyond the end of the Entry Path.
- A linked junction cannot be unlinked unless all vehicles are beyond the configured clearance distances of the junction.
- When positioning the moving path is not required, normal vehicle-to-vehicle clearances apply across the linked junction allowing vehicles to move across the junction.
- When positioning the moving path is not required, the move profile for the vehicle is maintained across the node so the vehicle crosses the node at a consistent velocity and acceleration.
- When positioning the moving path is not required, vehicles can queue across the node.
- Vehicles moving in the same direction can queue on the entry/exit paths.
- Vehicles can queue on the moving path.
- Normal vehicle-to-vehicle clearances apply across the linked junction allowing vehicles on the same route to platoon across the junction.
- Vehicles can queue across a junction.

- The move profile for the vehicle is maintained across the node so the vehicle crosses the junction at a consistent velocity and acceleration only when the move is such that the moving path does not need to move.

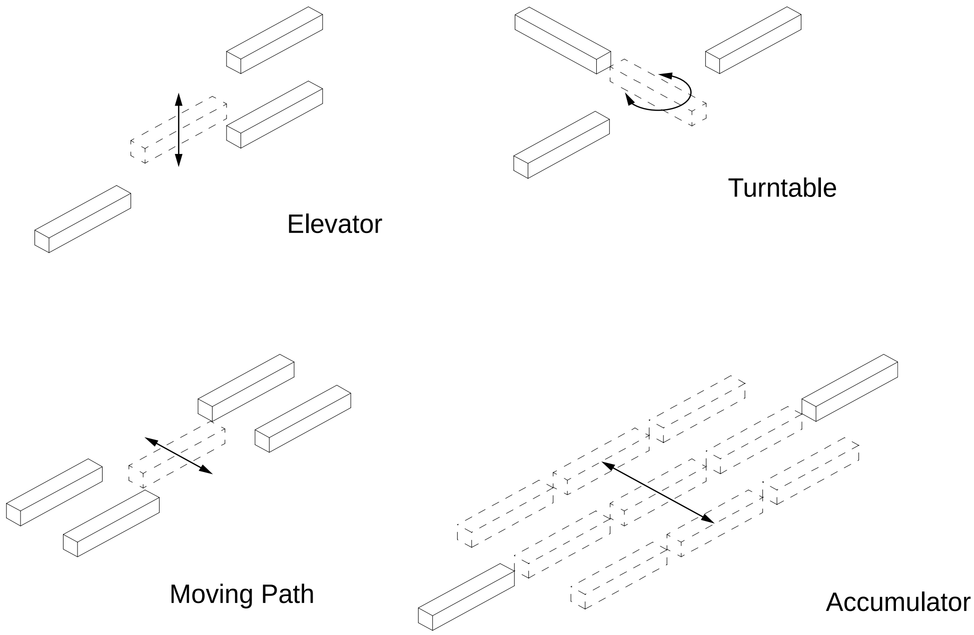


Figure 3-39: Examples of Moving Path Node Configurations Using Moving Guideways

Path Startup Considerations

During path startup in QuickStick 100 transport systems, unlocated vehicles move to determine their precise location. If the edge of a vehicle is over a motor, the vehicle does not always move to locate for QSHT and QS 150, for QS 100 they will always move. This move is termed a “locate move”. The node controller determines if there is room for the locate move. If there are already-located vehicles that are blocking the move, they must be moved to make room to locate the unlocated vehicle. This move is termed a “push move”. If there is no room to locate a vehicle in the downstream direction, the node controller searches for room in the upstream direction.

If a moving path junction is linked, the locating vehicle is allowed to locate into the linked junction. If a moving path junction is unlinked, the area past the entry gate is not available for a locate move or a push move. However, vehicles that extend past an entry gate can locate in the direction of the junction if there is room. Vehicles near an entry gate at an unlinked junction can only locate away from the junction.

It is possible that after an unlocated vehicle that was within the gate area is located it is still within the gate area at an unlinked path end. If a vehicle is positioned this way, the path end status transitions to the Unlinked Vehicle Present state and the Device Status field (bits 0…3 in the Node Status message) is set to “Junction Fault”. The host controller can either move the vehicle clear of the gate to cause the path end state to transition to Unlinked, or link the junction. Once none of the member path ends of the Moving Path node are in the Unlinked Vehicle Present state, the node device status transitions to “Operational”.

NOTE: A vehicle in the gate area has the potential to interfere with a path that is being moved to align it.

QuickStick Startup using Moving Paths

Figure 3-40 describes the steps involved in the startup sequence for a QuickStick system. For QS 100 system vehicles are ALWAYS required to move to locate. For QS 150 and QSHT vehicles only have to move if a magnet array edge is not over a motor. The distance required to move will depend on the vehicle location, vehicle length and magnet array length. If there is not enough room to locate, previously located vehicles are “pushed” one motor block to create space for a new vehicle to locate. Startup will first attempt to locate vehicles by moving them in the downstream direction. If there is not enough room, the process will attempt to move vehicles in the upstream direction. This will continue until all vehicles are located or fail when there is not enough room to locate a vehicle.

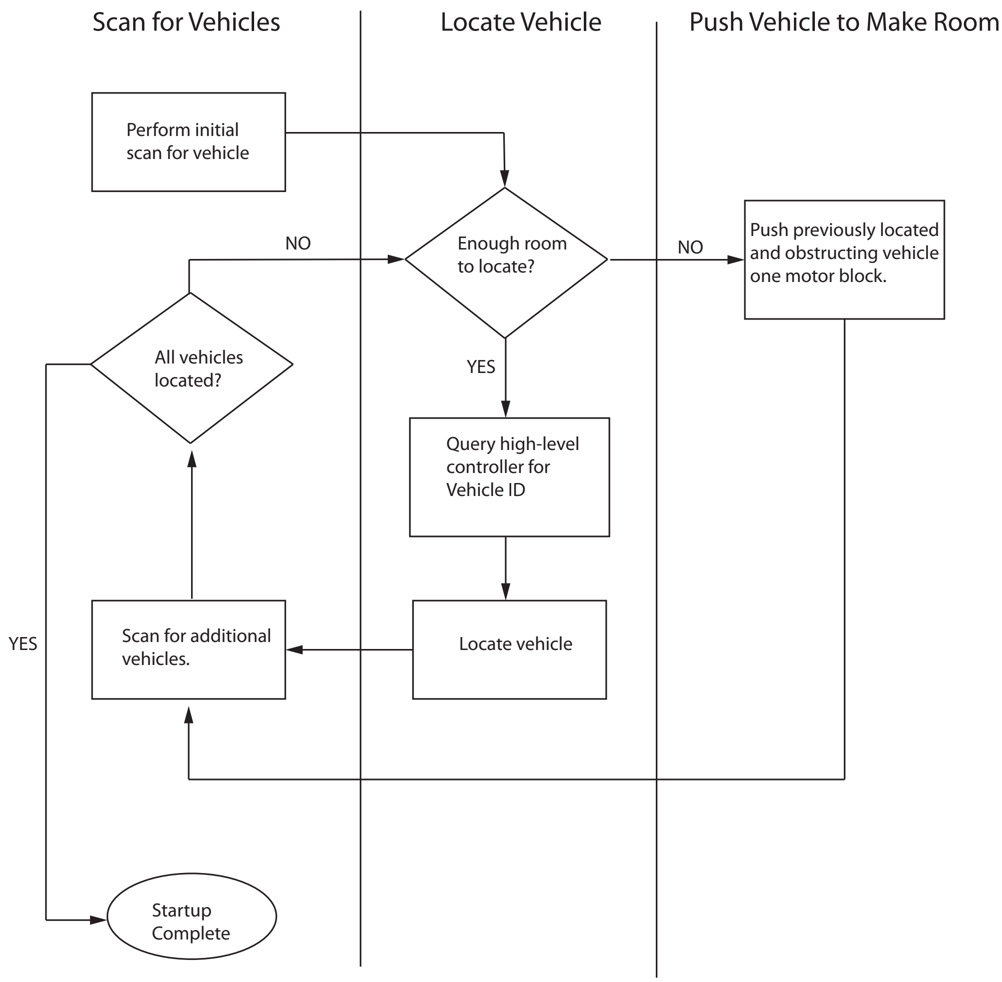


Figure 3-40: Startup Sequence - Worst Case

Moving Path Startup

For QuickStick systems with the Moving Path Node that fail to startup, the moving Paths can be used to assist neighboring fixed Paths for startup. This will increase the available number of blocks used for startup. Figure 3-41 explains the startup routine and how it should be implemented with Moving Paths.

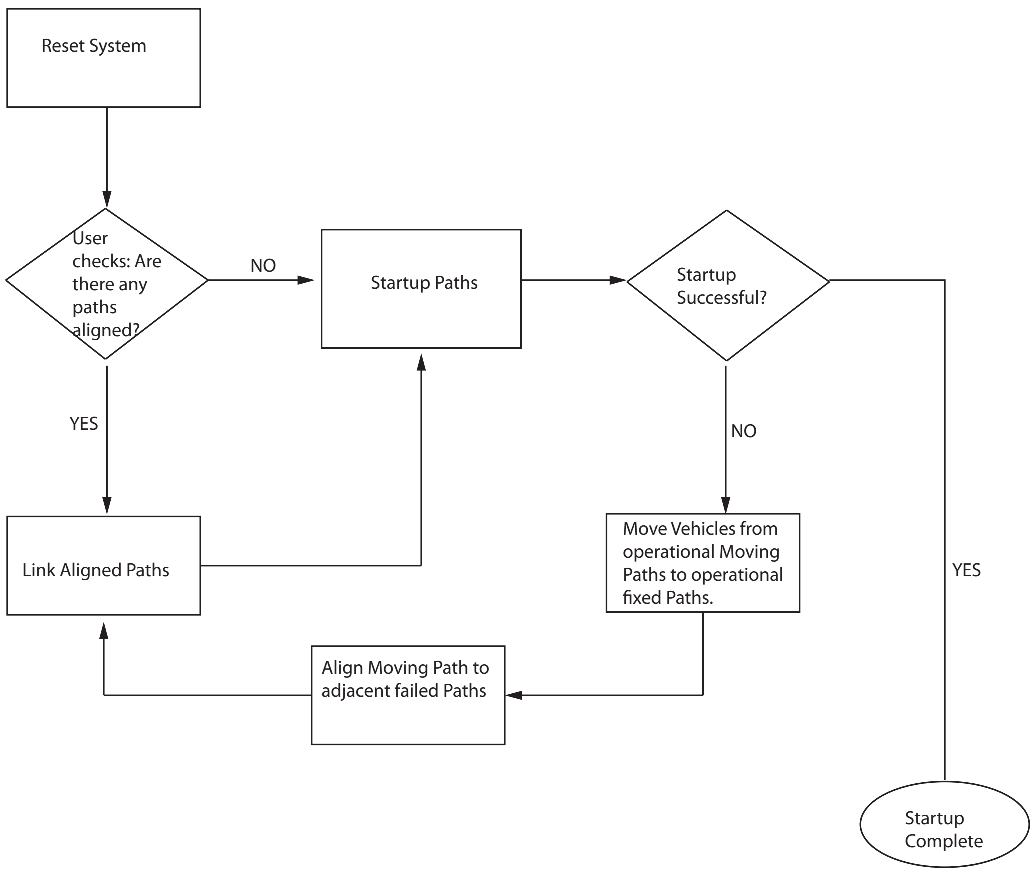


Figure 3-41: Startup Routine with Assistance from a Moving Path

During a Path 0 Reset Command, all Nodes go to the unlinked state. The host controller should check the location of all the moving Paths. If there are aligned Paths, a vehicle could be located between Paths. For this reason, it is important to "link" aligned Paths before sending a startup command. If a Path fails startup, moving Paths can be aligned and linked to those Paths and retry the startup routine. Figure 3-10 shows how linking two Paths allow for vehicles to startup across the junction by having more space to locate.

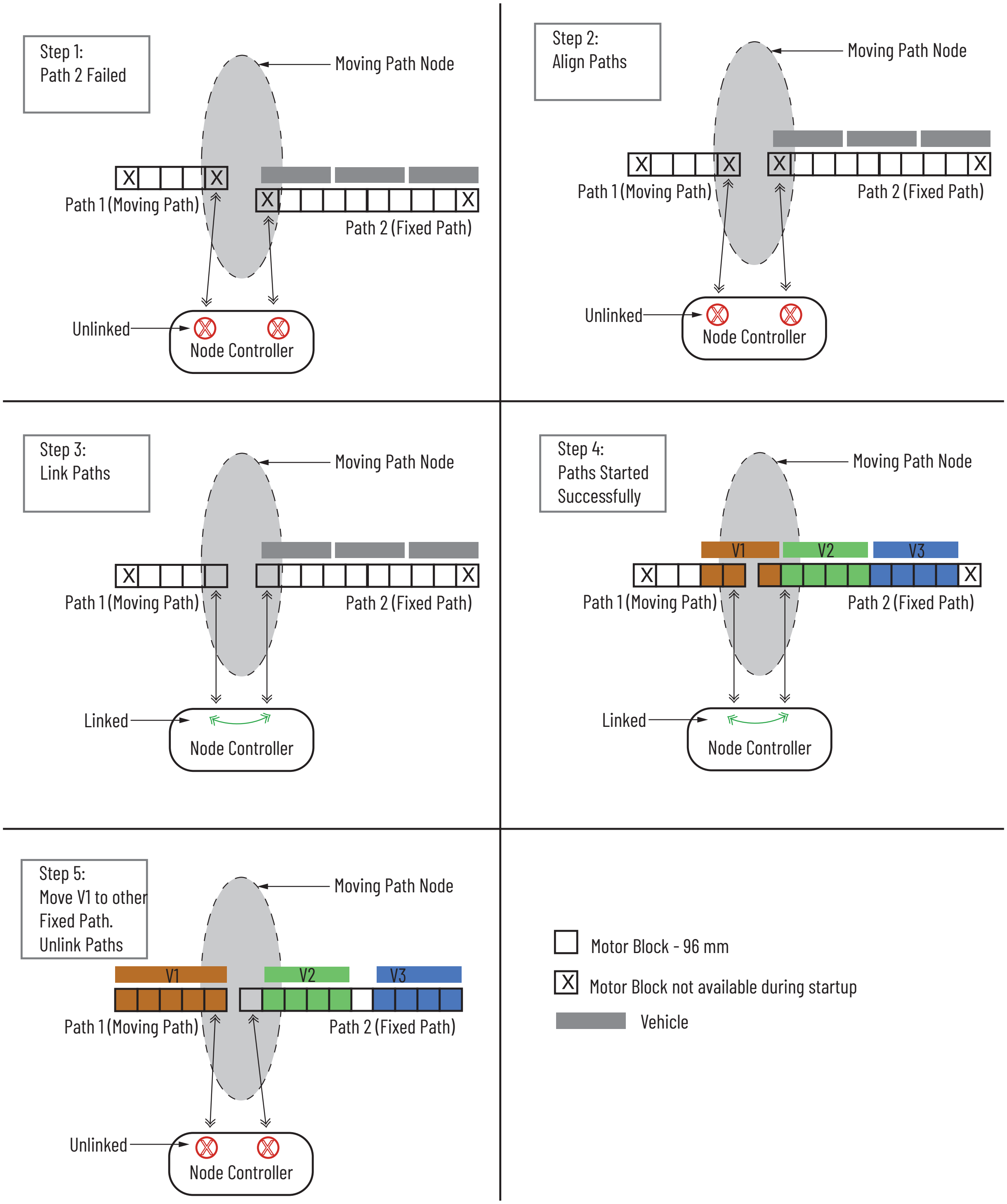


Figure 3-42: Startup Sequence for a Packed Path using a Moving Path for Assistance

NOTE: The X in the motor blocks show unavailable blocks due to the entry gate/clearance distance or vehicle overhang.

Path Linking/Unlinking

When two path ends are linked to form a junction, the node controller forwards motor-to-motor messages to allow vehicles to navigate the junction.

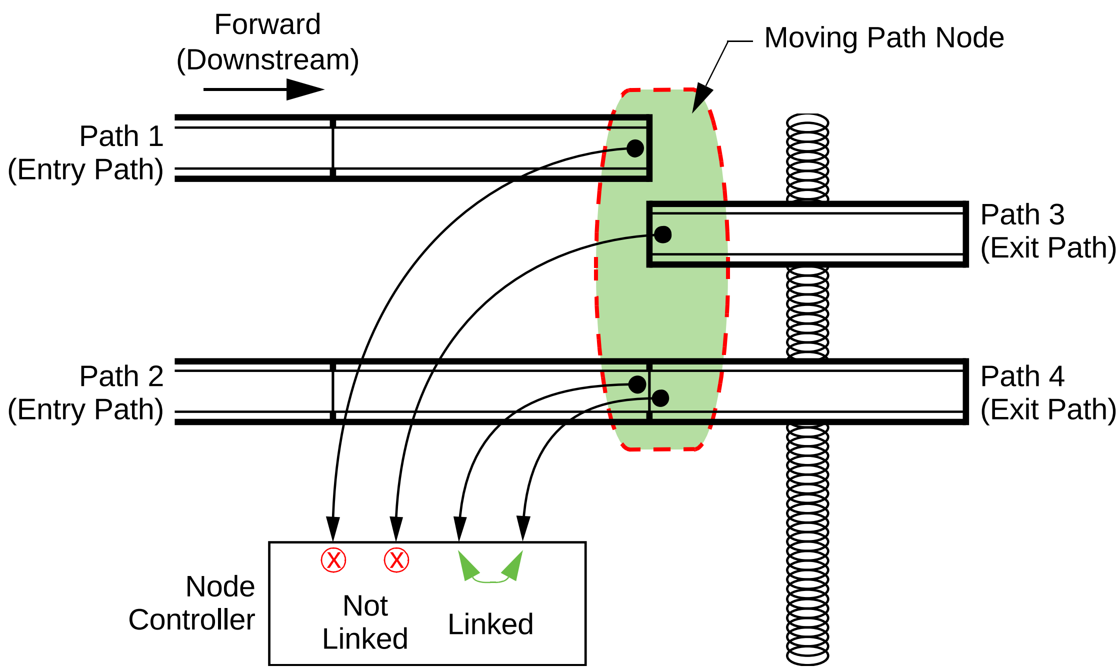


Figure 3-43: Path Linking/Unlinking in Moving Path Node

Linking Paths in a Moving Path Node

When the paths in a Moving Path node junction are linked, the node controller forwards motor-to-motor messages between the path ends that are linked. This linking allows vehicle headway to advance through the junction and permits vehicle motion through the junction. When unlinked, motor-to-motor messages are not forwarded, which causes vehicles that are ordered through a junction to become obstructed, preventing vehicle motion past the Entry Gate until the junction is linked.

To link a moving path and a fixed path (for example, path 2 to path 4 as shown in Figure 3-44) to permit vehicles to move from one path to the other an MP Link Command must be used.

```
0xBF 09 02 0003 0002 0004 0000 0000 00000001
```
(node 3, control path 2, peer path 4, keep paths linked, alignment request 0, command count 1)

```
0xD0 BF 00 09 02 0003 0002 0004 0000 0000 00000001
0xD0 BF 80 09 02 0003 0002 0004 0000 0000 00000001
```

The HLC sends Command Status messages that show the MP Link Command was accepted (0x00) and that it completed (0x80).

Host Control of Link (Last Allowed Vehicle ID = 0)

If the host controller links the path with an MP Link Command that has Last Allowed Vehicle ID set to “0”, the path end remains in the linked state until an MP Unlink Command is received. While in the linked state, vehicles are allowed to navigate the node if the linked path offers a route to the destination for the vehicle.

The host controller must send an MP Unlink Command to unlink the path. If the host controller sends the unlink command while a vehicle is navigating the junction, the junction state changes to Linked Unlink Pending and no additional vehicles are granted permission to navigate the junction. As soon as the vehicle clears the junction, the path end state transitions to the Unlinked state.

Host Control of Link (Last Allowed Vehicle ID = Specific Vehicle ID)

If the host controller links the path with an MP Link Command that has Last Allowed Vehicle ID set to a specific vehicle ID, the path end remains in the linked state until the specified vehicle is granted permission to enter the junction. The path end state transitions to Linked Unlink Pending as soon as the last allowed vehicle is granted permission to enter. No other vehicles are allowed permission to navigate the junction after the last allowed vehicle enters.

Once in the Linked Unlink Pending state, the path transitions to the Unlinked state as soon as all vehicles are clear of the junction. The host controller does not need to send an unlink command.

NOTE: Additional link commands can be issued to modify the Last Allowed Vehicle ID for the linked paths but must specify the same Control Path ID and Peer Path ID used initially to establish the junction.

Unlinking the Paths in a Moving Path Node

Once use of the node is complete, the linked paths in a Moving Path node must be unlinked to permit linking to different paths in the node. When the node is not configured to automatically unlink after a specific vehicle, an MP Unlink Command must be used to unlink the paths.

```
0xBF 09 03 0003 0001 00000002
```
(node 3, control path 1, command count 2)

```
0xD0 BF 00 09 03 0003 0001 00000002
0xD0 BF 80 09 03 0003 0001 00000002
```

The HLC sends Command Status messages that show the MP Unlink Command was accepted (0x00) and that it completed (0x80).

Moving Path Node Considerations

When using a moving path switch, the host controller is responsible for positioning the guideway for the moving path to provide the required routing of the vehicles. The Moving Path node provides support to route the vehicles from the entry paths to their exit paths by linking the paths, it does not provide any mechanism control or monitoring.

A user-supplied mechanism supports and moves the guideway for the moving path and aligns the end of the moving path with a fixed guideway, which allows vehicles to move on and off the moving path. The paths in a moving path switch can be configured in different layouts including moving paths and turntables. When using a moving path style configuration (see Figure 3-44), any user-supplied mechanism can be used to position the guideway, including a QuickStick path. When using a turntable configuration, a user-supplied rotary indexer is used to position the guideway.

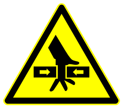
Either entry or exit paths can move but the connecting paths must all be of the other type (that is, entry paths move and all exit paths are fixed). There is a maximum of 12 paths in a Moving Path node, and the ends of all paths in the node must be connected to the same node controller. All paths in the node must maintain the downstream to upstream relationship of the motors. See Figure 3-44 where the large shaded area represents the Moving Path node and the entry paths can move. There can also be a Moving Path node at each end of a moving path where vehicles move onto or off the moving path as shown in Figure 3-45.

Use of a moving guideway requires the host controller to command the drive mechanism to position one of the moving paths so that it aligns with one of the fixed paths. The host controller must then issue an MP Link Command to connect the two paths to allow vehicle motion. Once the vehicle has moved beyond the node, the host controller must issue an MP Unlink Command before the Moving Path can be moved to a new position.

Figure 3-44 shows a configuration with one Moving Path node. Vehicles can move between the three fixed exit paths using either of the two moving entry paths. Vehicles can enter the moving paths either through the Terminus Nodes or from the fixed paths.

Figure 3-45 shows a configuration with dual Moving Path nodes. Vehicles can move from either of the two fixed entry paths in Moving Path node 3 to any of the fixed exit paths in Moving Path node 4 using either of the two moving paths. In this type of configuration, the moving paths are associated with both Moving Path nodes. For Moving Path node 3, these paths are moving exit paths. For Moving Path node 4, these paths are moving entry paths.

In the example shown in Figure 3-45, if the vehicle is moving from Moving Path node 3 entry path 2 to Moving Path node 4 exit path 6 the moving path does not need to move. Once path 4 is linked to both path 2 and path 6, vehicles can move directly from path 2 to path 6.



PINCH/CRUSH HAZARD: Moving mechanisms (motors and moving paths) have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result in the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

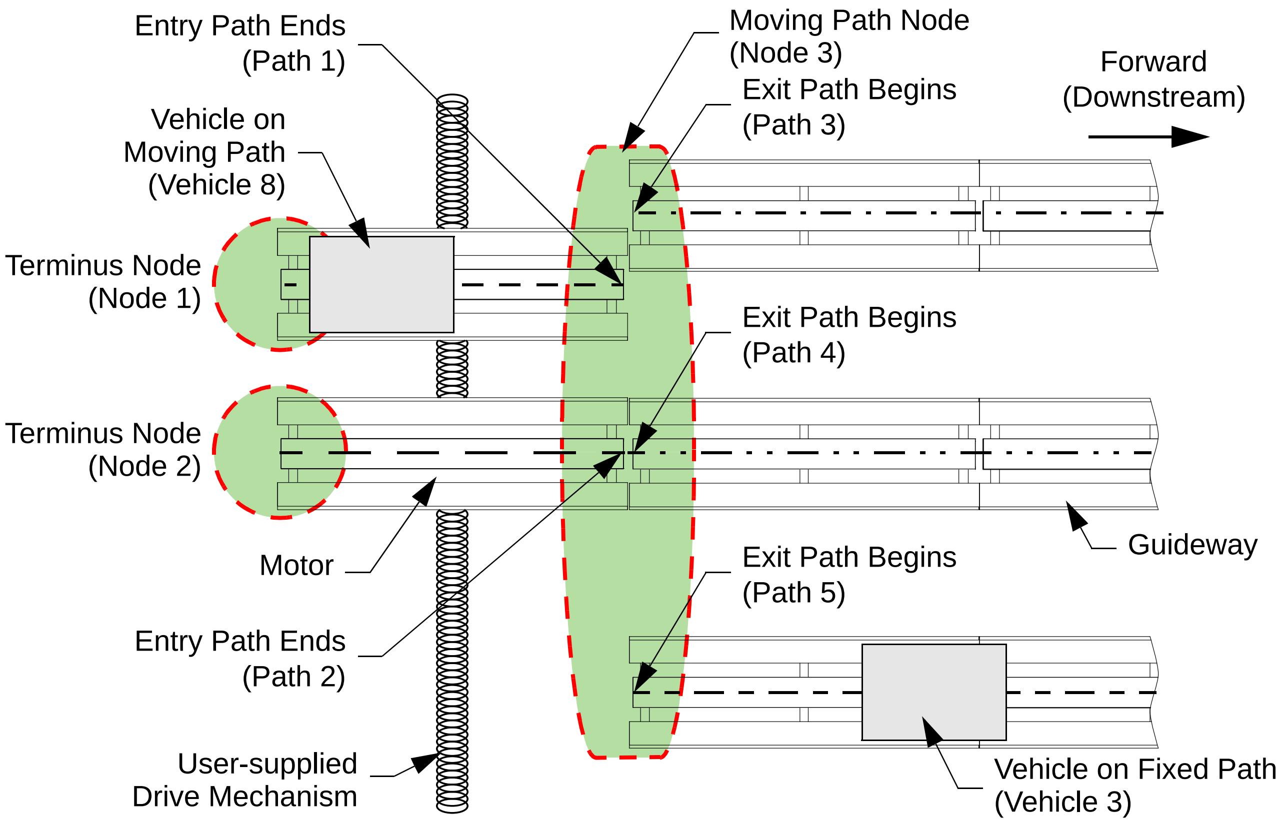


Figure 3-44: Moving Path Node, Top View

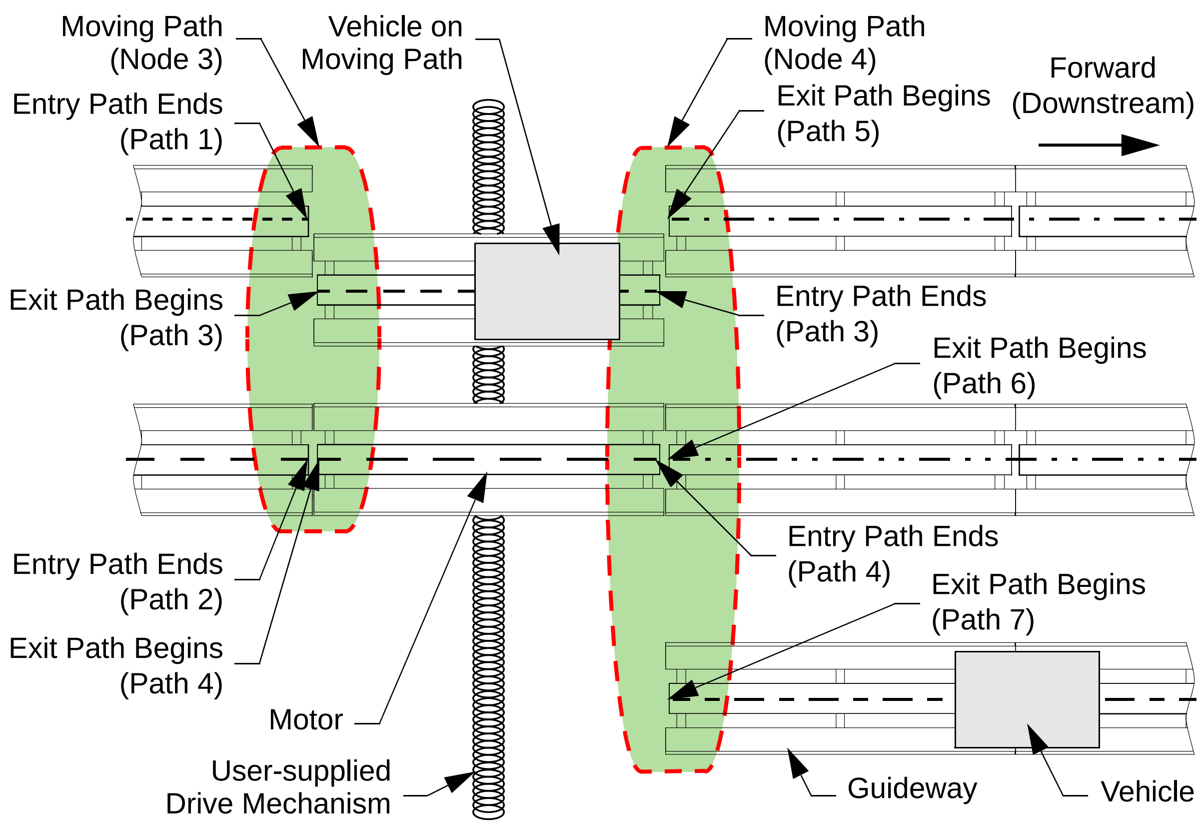


Figure 3-45: Moving Path Node Pair, Top View

Forward Through a Moving Path Node

Forward motion through a Moving Path Node (for example, path 1 to path 3 as shown in Figure 3-44) requires positioning the moving path so it aligns with the fixed path. Once the paths are aligned, the paths are linked as described in Linking Paths in a Moving Path Node. Once the paths are linked, this move is configured as a basic move from the entry path to the exit path where motion continues forwards. Once the move is complete, unlink the paths as described in Unlinking the Paths in a Moving Path Node.

`0xB1 00000002 0008 01 0000803E 0003 0000803F 0000803F`
(move 2, vehicle 8, unloaded and forward, position 0.25 m, path 3, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 00000002 0008 0000803E 0003 0000803F 0000803F 01
0xD0 B1 80 00000002 0008 0000803E 0003 00000000 00000000 01
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was accepted (0x00) and that it completed (0x80).

Backward Through a Moving Path Node

Backward motion through a Moving Path node (for example, path 5 to path 2 as shown in Figure 3-44) requires positioning the moving path so it aligns with the fixed path. Once the paths are aligned, the paths are linked as described in Linking Paths in a Moving Path Node. Once the paths are linked, this move is configured as a basic move from the exit path to the entry path where motion continues backwards. Once the move is complete, unlink the paths as described in Unlinking the Paths in a Moving Path Node.

```
0xB1 00000002 0004 02 0000B840 0002 0000803F 0000803F
```

(move 2, vehicle 4, unloaded and backward, position 5.75 m, path 2, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 00000002 0004 0000B840 0002 0000803F 0000803F 02
0xD0 B1 80 00000002 0004 0000B840 0002 00000000 00000000 02
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was accepted (0x00) and that it completed (0x80).

Reversing Through a Moving Path Node

Reversing the direction of a vehicle through a Moving Path node (for example, path 3 to path 4 as shown in Figure 3-44) requires two moves. The first move is backward from path 3 (the fixed path) to path 1 (the moving path) and requires positioning the moving path so it aligns with the fixed path. Once the paths are aligned, the paths are linked as described in Linking Paths in a Moving Path Node. After the paths are linked, this move is configured as a basic move backwards from path 3 to path 1 to center the vehicle on the moving path. Once the move is complete, unlink the paths as described in Unlinking the Paths in a Moving Path Node.

NOTE: The first motion command must move the vehicle fully onto the moving path to clear the guideway it had been on, which allows the path to move. Typically the vehicle is centered on the moving path. Or, the vehicle must be positioned far enough on the moving path that the appropriate node clearance (see Table 3-1) is achieved.

The second move is forward from path 1 to path 4 and requires positioning path 1 (the moving path) so it aligns with path 4 (the fixed path). Once the paths are aligned, the paths are linked as described in Linking Paths in a Moving Path Node. After the paths are linked, this move is configured as a basic move forwards from path 1 to path 4. Once the vehicle exits path 1 the paths are unlinked as described in Unlinking the Paths in a Moving Path Node.

NOTE: The first move command must move the vehicle far enough onto the moving path so that it is clear of the moving path mechanism and any configured clearance distance.

```
0xB1 000000D2 0004 02 0000003F 0001 0000803F 0000803F
```

(move D2, vehicle 4, unloaded and backward, position 0.5 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 000000D2 0004 0000003F 0001 0000803F 0000803F 02
0xD0 B1 80 000000D2 0004 0000003F 0001 00000000 00000000 02
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was accepted (0x00) and that it completed (0x80).

Once the first move completes, path 1 and path 3 must be unlinked (see Unlinking the Paths in a Moving Path Node). Once the unlink completes and the moving path (with the vehicle on it) is aligned with the new fixed path, path 1 and path 4 must be linked (see Linking Paths in a Moving Path Node). Once the link completes, the vehicle can be moved.

NOTE: The second move command must move the vehicle far enough onto the fixed path so that it is clear of the moving path mechanism and any configured clearance distance.

```
0xB1 000000D3 0004 01 0000C03F 0004 0000803F 0000803F
```

(move D3, vehicle 4, unloaded and forward, position 1.5 m, path 4, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 000000D3 0004 0000C03F 0004 0000803F 0000803F 01
0xD0 B1 80 000000D3 0004 0000C03F 0004 00000000 00000000 01
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was accepted (0x00) and that it completed (0x80).

Once the second move completes, path 1 and path 4 must be unlinked (see Path Linking/Unlinking on page 120).

Moving Path Example 1: Moving Onto a Moving Path

In the example in Figure 3-46, with one Moving Path node, the vehicles move from a fixed path (defined in the Moving Path node configuration as the Entry Path) and enter the moving path (defined in the Moving Path node configuration as the Exit Path).

The example in Figure 3-46 has two fixed paths, path 1 and path 2, which connect to two moving paths, path 3 and path 4.

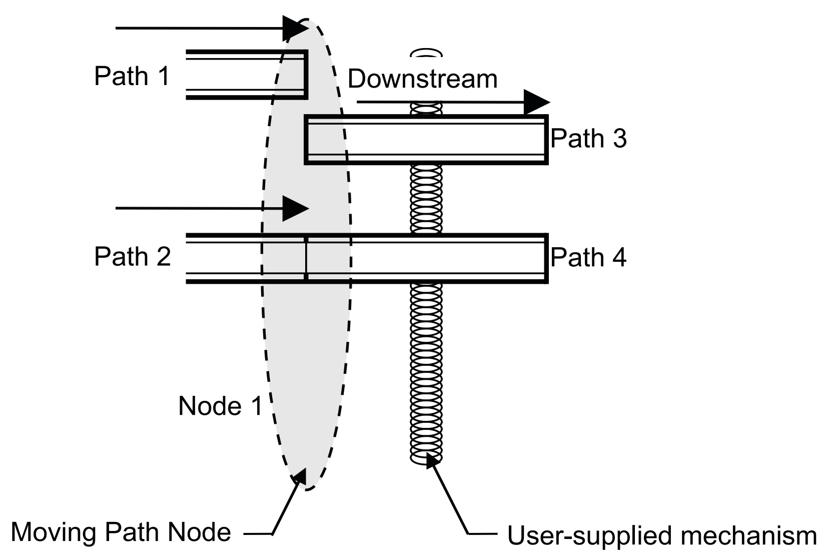


Figure 3-46: Moving Onto a Moving Path

1. To move onto either path 3 or path 4, the host controller must order the vehicle to path 3 or path 4.

2. Once the vehicle headway reaches the entry gate, the HLC then requests alignment of the destination path that is specified in the vehicle order, setting MP Path End Status Report for the path the vehicle is on to the unlinked_alignment_requested state, see Table 4-17 on page 319.

3. The host controller then commands the mechanism to position the moving path as requested, such that two paths are physically aligned.

4. Once the moving path are aligned, the host controller issues an MP Link Command to link the two paths and the vehicle can move across the junction between the paths.

5. Depending upon the usage of the linked paths, once the vehicle clears the junction, the paths can be unlinked, or the paths can remain linked (see Path Linking/Unlinking on page 120).

Moving Path Example 2: Moving Across a Moving Path

This Moving Path node application allows one or more moving paths to span a gap between one or more fixed paths. Figure 3-47 shows an example of a Moving Path node pair (node 1 and node 2). Node 1 is a Moving Path node linking the downstream end of fixed paths, path 1 and path 2, to the upstream end of moving paths, path 3 or path 4. Node 2, is a Moving Path node linking the downstream end of moving paths, path 3 and path 4, to the upstream end of fixed paths, path 5, path 6, or path 7.

Considering movement in the downstream direction (indicated by the arrows) the upstream end of Exit Paths 3 and 4 (the moving paths) offer equivalent routes for vehicles navigating node 1 to destinations on paths 5, 6, or 7. Either can satisfy a vehicle move from path 1 or path 2 to path 5, path 6, or path 7. Paths 3 and 4 are termed equivalent-route paths. At node 2, still moving downstream, the host controller must align to either path 5, path 6, or path 7 depending upon the vehicles destination. Paths 5, 6, and 7 are termed specific-route paths.

Similarly, considering movement in the upstream direction, the downstream end of path 3 or 4 offer equivalent routes to path 1 and path 2 when navigating node 2. Paths 1 and 2 require specific alignment at node 1.

Two node configuration items must be set, one for the set of Entry Paths and one for the set of Exit Paths, designating the set as specific-route or equivalent-route. The HLC assumes that any moving path in an equivalent-route set can satisfy the route. Because the host controller application chooses the path to align to satisfy an equivalent route alignment request from the HLC, the host controller can provide application-specific management strategy for utilization of moving paths.

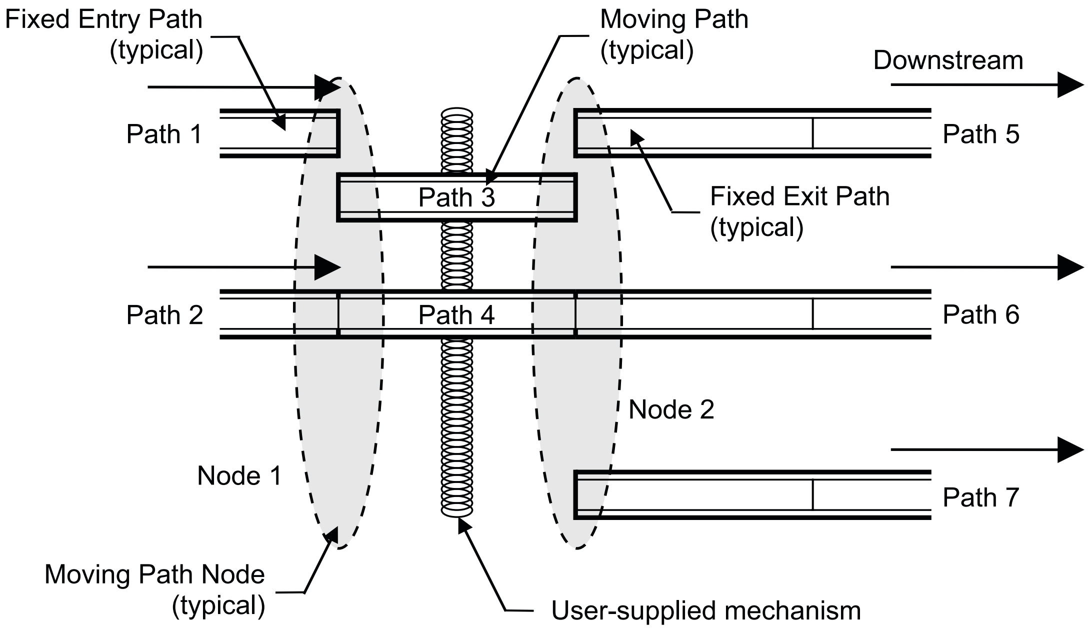


Figure 3-47: Moving Across a Moving Path

1. To move onto one of the Exit Paths from either of the Entry Paths, path 1 or path 2, the host controller must order the vehicle to either path 5, path 6, or path 7.

2. Once the vehicle headway reaches the entry gate, the HLC requests alignment of a moving path to complete the vehicle order by setting MP Path End Status Report for the path the vehicle is on to the unlinked_alignment_requested state, see Table 4-17 on page 319 for Moving Path Node End States.

3. The host controller determines which path to move and align and then commands the mechanism to position the moving path as requested.

4. Once the moving path is positioned, the host controller notifies the HLC that the paths are aligned by issuing an MP Link Command to link the two paths so the vehicle can move across the junction between the paths onto the moving path.

5. Once the last allowed vehicle is granted permission to navigate the junction, the HLC sets MP Path End Status Report for the path to linked_unlinked_pending state to notify the host controller that permission to navigate the junction is not granted to any additional vehicles.

6. Once all vehicles that are involved in this junction clear the junction, the paths are unlinked (see Path Linking/Unlinking on page 120).

7. The vehicle on the moving path continues to move toward the other end of the moving path and the path end state of that path end transitions to unlinked_alignment_requested this time specifying the specific path end where the host controller must align the moving path.

8. The host controller commands the mechanism to position the moving path and aligns it with the Exit Path. The moving path can be moved while the vehicle is in motion.

9. Once the moving path is positioned, the host controller issues an MP Link Command to link the two paths and the vehicle can move across the junction between the paths onto the Exit Path.

10. Depending upon the usage of the linked paths, once the vehicle clears the junction, the paths can be unlinked, or the paths can remain linked (see Path Linking/Unlinking on page 120).

Moving Path Switch Configuration

When using a mechanism to move a section of guideway with the vehicle on it, the host controller is responsible for positioning the guideway to provide the required routing of the vehicles. The Moving Path node provides support to route the vehicles from the entry paths to the moving path and then to their exit paths by linking the paths, it does not provide any position control or monitoring.

When using a switching mechanism to divert vehicle motion, the host controller is responsible for positioning the switch mechanism to provide the required routing of the vehicles. The Moving Path node provides support to route the vehicles from the entry paths to their exit paths by linking the paths, it does not provide any mechanism control or monitoring.

Moving path can be used to control a switch that can merge two paths together (see Figure 3-48) or diverge one path into two (see Figure 3-49).

NOTE: Merge/Diverge Switches are only available as switch modules on MagneMover LITE transport systems.

ATTENTION:

Moving Path node types are not the same nodes that are used for Merge, Diverge, or Merge-Diverge.

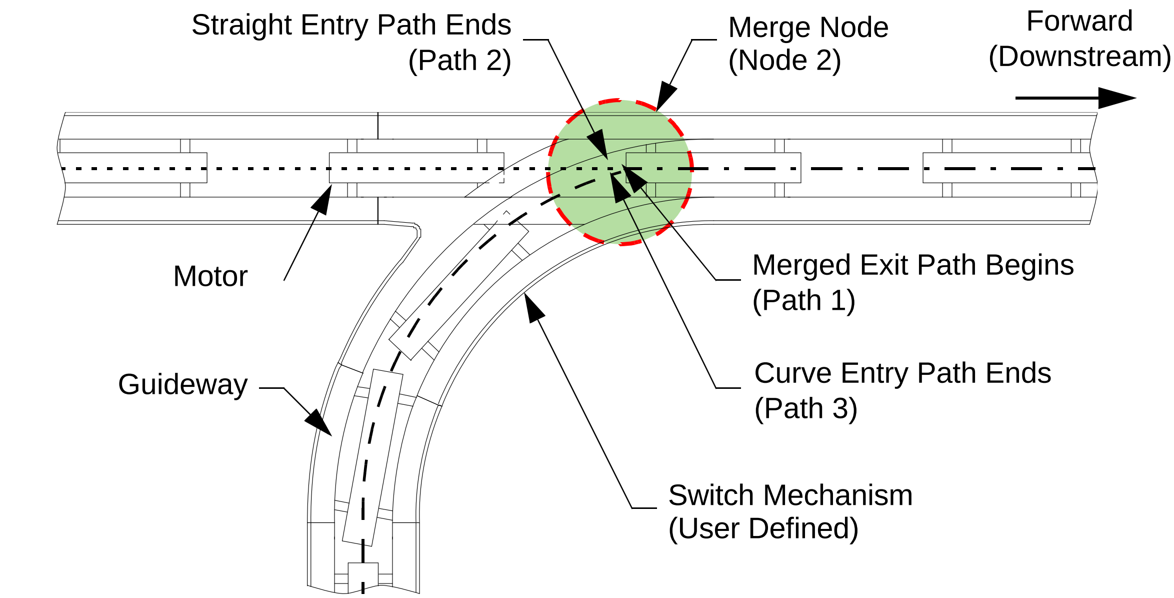


Figure 3-48: Moving Path Node Merge, Top View

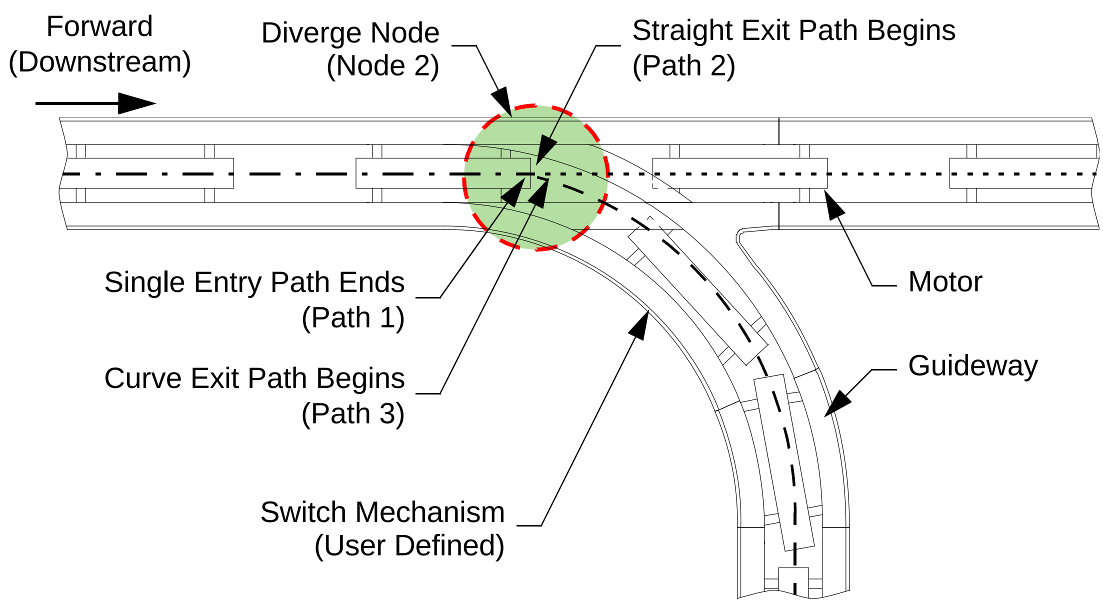


Figure 3-49: Moving Path Node Diverge, Top View

Forward Through a Moving Path Based Switch

Forward motion through a Moving Path based switch (for example, path 2 to path 1 as shown in Figure 3-48) requires linking the paths as described in Path Linking/Unlinking on page 120. Once the paths are linked, this move is configured as a basic move from the entry path to the exit path where motion continues forwards. Before the move can be performed, the switch mechanism must be positioned appropriately. Once the move is complete, unlink the paths.

```
0xB1 000000F2 0004 01 0000C03F 0001 0000803F 0000803F
```

(move F2, vehicle 4, unloaded and forward, position 1.5 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 000000F2 0004 0000C03F 0001 0000803F 0000803F 01
0xD0 B1 80 000000F2 0004 0000C03F 0001 00000000 00000000 01
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was accepted (0x00) and that it completed (0x80).

Backward Through a Moving Path Based Switch

Backward motion through a Moving Path based switch (for example, path 1 to path 2 as shown in Figure 3-48 and Figure 3-49) requires linking path 1 to path 2 as described in Path Linking/Unlinking on page 120. Once the paths are linked, this move is configured as a basic move from the exit path to the entry path where motion continues backwards. Before the move can be performed, the switch mechanism must be positioned appropriately. Once the move is complete, unlink the paths as described in Path Linking/Unlinking on page 120.

```
0xB1 000000F2 0004 02 0000B840 0002 0000803F 0000803F
```

(move F2, vehicle 4, unloaded and backward, position 5.75 m, path 2, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 000000F2 0004 0000B840 0002 0000803F 0000803F 02
0xD0 B1 80 000000F2 0004 0000B840 0002 00000000 00000000 02
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was accepted (0x00) and that it completed (0x80).

Reversing Through a Moving Path Based Switch

Reversing the direction of a vehicle through a Moving Path based switch requires two moves. This example shows reversing through a Moving Path merge switch (path 2 to path 3 as shown in Figure 3-48). The first move is forward from path 2 to path 1 and requires linking the paths as described in Path Linking/Unlinking on page 120. Before the first move can be performed, the switch mechanism must be positioned appropriately.

The second move is backward from path 1 to path 3 and requires unlinking path 1 and path 2 as described in Path Linking/Unlinking on page 120. Once the paths are unlinked, path 1 can be linked to path 3. Before the second move can be performed, the switch mechanism must be positioned appropriately. Once the move is complete, unlink the paths.

NOTE: The first motion command must move the vehicle far enough past the switch to clear the switch mechanism and any configured clearance distance.

```
0xB1 000000D2 0004 01 0000803E 0001 0000803F 0000803F
```

(move D2, vehicle 4, unloaded and forward, position 0.25 m, path 1, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 000000D2 0004 0000803E 0001 0000803F 0000803F 01
```

```
0xD0 B1 80 000000D2 0004 0000803E 0001 00000000 00000000 01
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was accepted (0x00) and that it completed (0x80).

Once the first move has completed, path 1 and path 2 must be unlinked (see Path Linking/Unlinking on page 120). Once the unlink completes, path 1 and path 3 must be linked. Once the link completes, the vehicle can be moved.

```
0xB1 000000D3 0004 02 0000B840 0003 0000803F 0000803F
```

(move D3, vehicle 4, unloaded and backward, position 5.75 m, path 3, acceleration 1 m/s^2, velocity 1 m/s)

```
0xD0 B1 00 000000D3 0004 0000B840 0003 0000803F 0000803F 02
0xD0 B1 80 000000D3 0004 0000B840 0003 00000000 00000000 02
```

The HLC sends Command Status messages that show the Move Vehicle To Position command was received (0x00) and that it completed (0x80).

Once the second move has completed, path 2 and path 3 must be unlinked (see Path Linking/Unlinking on page 120).

Protocol Reference 4

Overview

This chapter provides detailed descriptions of the command and response messages for the Host Controller TCP/IP Communication Protocol. The high-level controller (HLC) provides a broad range of command and response options for MagneMover® LITE (MML™) and QuickStick® transport systems. The high-level controller also provides the host controller the ability to monitor, and be notified of, status changes within the transport system.

NOTE: This Protocol Reference reflects the version of the software that is indicated in Changes at the front of the manual. Specific versions of the Node Controller Software Image File may not implement all Host Controller TCP/IP Communication Protocol features described in this manual. See the Release Notes that are supplied with the Software for more information.

All MagneMotion® product lines support all TCP/IP commands and responses except as stated in this reference.

Introduction

This chapter provides an overview of the control software communications between the host controller and the high-level controller (HLC) and detailed information on the command set. The method of communication uses TCP/IP communication protocols (see Communications Protocol on page 393). The command set may include commands that are not documented in this chapter. Any undocumented commands are used by Rockwell Automation to support internal functions and are not intended for customer use. Use only the commands that are documented in this Protocol Reference.

This communication protocol is provided for communications between the HLC and the host controller, which is established using a TCP/IP connection. Only one device can be connected through the control port at this interface to the HLC at a time.

For information about Logix and Data Types, see Logix 5000 Controllers Design Considerations, publication 1756-RM094.

Host Controller to HLC Communications

The commands that are shown in Table 4-1 are sent from the host controller to the HLC as asynchronous requests for the transport system to perform an action. The HLC handles these requests by routing the command to the appropriate node controller for execution and sending the responses that are shown in Table 4-8 to the host controller.

Table 4-1: Host Controller to HLC Commands

Description	Message Type	Extension Type / Subtype	Page
Move Vehicle To Station	0xB0	—	136
Move Vehicle To Position	0xB1		141
Startup	0xB2		146
Resume Motion	0xB3		148
Suspend Motion	0xB4		150
Status Request	0xB5		152
Set Signal	0xB6		158
Vehicle Follow Order	0xB7		161
Reset	0xB8		167
Delete Vehicle	0xB9		170
Set Control Loop Parameters	0xBA		173
Set Node Parameters	0xBB		177
FastStop Motion	0xBC		179
Warm Reset	0xBD		181
Create Station	0xBF	0x01 / 0x01	184
Delete Station	0xBF	0x01 / 0x02	187
Get Station Status	0xBF	0x01 / 0x03	190
Create Traffic Light	0xBF	0x02 / 0x01	193
Set Traffic Light	0xBF	0x02 / 0x02	196
Get Traffic Light Status	0xBF	0x02 / 0x03	199
Delete Traffic Light	0xBF	0x02 / 0x04	202
Lock Vehicle	0xBF	0x03 / 0x01	204
Vehicle Command	0xBF	0x03 / 0x04	206

Table 4-1: Host Controller to HLC Commands (Continued)

Description	Message Type	Extension Type / Subtype	Page
Get Extended Vehicle Status	0xBF	0x03 / 0x07	208
	0xBF	0x03 / 0xFF	
Get Node Controller Digital I/O Status	0xBF	0x07 / 0x01	211
Set Node Controller Digital I/O Outputs	0xBF	0x07 / 0x02	214
Set Node Controller Digital I/O Notification Mask	0xBF	0x07 / 0x03	217
MP Get Path End Status	0xBF	0x09 / 0x01	219
MP Link Command	0xBF	0x09 / 0x02	222
MP Unlink Command	0xBF	0x09 / 0x03	226
SM Subscription Command	0xBF	0x0A / 0x01	229
SM Poll Command	0xBF	0x0A / 0x03	238
Motor Inverter Command	0xBF	0x0B / 0x01	241
Set Node Controller Configuration	0xBF	0x0C / 0x01	245
Restart Node Controllers	0xBF	0x0C / 0x02	247
Get Extended Node Controller Status	0xBF	0x0C / 0x03	249
Get Extended High-Level Controller Status	0xBF	0x0C / 0x04	252
Get Propulsion Power Status	0xBF	0x0D / 0x01	254
Set Propulsion Power State	0xBF	0x0D / 0x02	256

NOTE: The examples separate all fields by spaces; commands and responses are separated by blank lines for clarity only. See General Message Formats on page 400 for the actual format of all commands.

Move Vehicle To Station

Message: 0xB0

Source/Destination

Host Controller ► HLC

Purpose

Moves a vehicle to a specified station or to an offset from a specified stations position when the optional "offset" field is supplied.

NOTE: The high-level controller, node controllers, and the path the vehicle resides on must be in the operational state. If the path where the vehicle is located is not in an operational state, the command is rejected. If any component required to execute the command is not in an operational state, the command does not complete.

If the commanded acceleration or velocity values are higher than the limit that is defined in the Node Controller Configuration File, the command is rejected.

If the commanded velocity value is higher than the value for a specific motor, the motor velocity value overrides the commanded value while the vehicle is on that motor.

If a motion command is issued to a vehicle already in motion and the command has a lower acceleration than the previous command to that vehicle, the command is rejected.

NOTE: These stations are added in the node controller configuration file or dynamically created, and do not function the same as the station commands [raM_MM_Tec_Station, raM_MM_Tec_SupervisedStation} that are found in the MagneMotion libraries from the PCDC. It is highly recommended for new systems that stations are defined and controlled using the station AOIs (add-on instructions) in the MagneMotion library.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	20 – Offset field not supplied 24 – Offset field supplied
3	Message Type	1	0xB0
4	Order Number	4	0x0…0xFFFFFFFF
8	Vehicle ID	2	1…65535
10	Flags and Direction	1	Bits 0…3 are direction values Bits 4…7 are the PID set index
11	Station ID	2	1…2048
13	Acceleration/Deceleration	4	0.001…60.0 (m/s^2, floating-point)
17	Velocity	4	0.001…2.5 (m/s, floating-point)
21	Offset	4	-1.0…+1.0 (m, floating-point)
Varies	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Move Vehicle to Station command.

Order Number – Host controller derived unique order number that is used for tracking, which is included in the command status response messages the high-level controller returns.

Vehicle ID – The ID of the vehicle to move. The ID must be a nonzero positive integer that references a vehicle that exists in the transport system. This value is system-dependent, see MagneMotion Node Controller Interface User Manual, publication MMI-UM001.

Flags and Direction – Bits 0…3 indicate the direction to move the vehicle. Bits 4…7 indicate the PID set index to use when executing the motion command.

Bit	Description
0…3	**Direction:** 0 – **Bidirectional** – If the destination is on the same path the vehicle resides on, the vehicle can move either direction as required to get to the destination in the shortest distance. If the destination is on a path other than the path where the vehicle is located, the forward direction takes precedence for a transport system that is a closed-loop. 1 – **Forward** – Force the vehicle to move forward (downstream) only, useful to implement a unidirectional loop. If the destination is not reachable, the command fails and the vehicle is not moved. 2 – **Backward** – Force the vehicle to move backward (upstream) only, useful to implement a unidirectional loop in the backwards direction. If the destination is not reachable, the command fails and the vehicle is not moved. 3…15 – **Reserved**
4…7	**PID Loop Set Index:** 0 – Use user-defined PID set 0 – Unloaded PID values. 1 – Use user-defined PID set 1 – Loaded PID values. 2 – Use user-defined PID set 2. 3 – Use user-defined PID set 3. 4 – Use user-defined PID set 4. 5 – Use user-defined PID set 5. 6 – Use user-defined PID set 6. 7 – Use user-defined PID set 7. 8 – Use user-defined PID set 8. 9 – Use user-defined PID set 9. 10 – Use user-defined PID set 10. 11 – Use user-defined PID set 11. 12 – Use user-defined PID set 12. 13 – Use user-defined PID set 13. 14 – Use user-defined PID set 14. 15 – Use user-defined PID set 15 – Startup PID values. This PID set is automatically used during startup. If it is not defined, PID set 0 is scaled by 25% and used for startup.

Station ID – The ID of the station for the vehicle destination. The station is either defined statically in the Node Controller Configuration File or defined dynamically using a Create Station command.

Acceleration/Deceleration – A positive number (in m/s^2) that defines the maximum acceleration and deceleration rate for the motion command (expressed as a 32-bit single-precision floating-point number {see NOTE}). This value is checked against the limit that is defined in the Node Controller Configuration File, and if higher the command is rejected. If the velocity is not defined in the NC file, the default is 1.0 m/s. If this value is within the limit that is defined in the Node Controller Configuration File but higher than the value for any specific motor in the route of the vehicle, the value from that motor is used and the command is updated to that lower value.

This value is system-dependent, see MagneMotion Node Controller Interface User Manual, publication MMI-UM001.

NOTE: All floating-point numbers use little-endian format, see Data Encoding on page 401.

Velocity – A positive number (in m/s) that defines the maximum velocity for vehicle motion during the move command (expressed as a 32-bit single-precision floating-point number). This value is checked against the limit that is defined in the Node Controller Configuration File, and if higher the command is rejected. If the velocity is not defined in the NC file, the default is 1.0 m/s. If this value is within the limit that is defined in the Node Controller Configuration File but higher than the value for any specific motor in the route of the vehicle, the value for that motor overrides the command value while the vehicle is on that motor.

This value is system-dependent, see MagneMotion Node Controller Interface User Manual, publication MMI-UM001.

Offset – An optional field used to specify an offset from a station as the destination for the vehicle (expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401). Offsets are relative to the station position. Positive values provide offsets downstream from a station, negative values provide offsets upstream from a station.

NOTE: Using offsets that specify positions that go beyond the end of the designated path for the station are not supported.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a “Command Accepted” or a “Command Rejected” response as appropriate (see Command Status message). If the command is accepted, the HLC executes a move order for the specified vehicle.

On completion of the command, when the HLC is notified that the vehicle has arrived at the specified station, the HLC sends a “Command Complete” response.

NOTE: If the vehicle never reaches its destination, this command never completes.

See Also

[Move Vehicle To Position on page 141]
[Vehicle Follow Order on page 161]
[Delete Vehicle on page 170]
[Command Status on page 261]

Example

The following example (see Figure 2-1 on page 27 for the system layout) shows a Move Vehicle to Station command. The command is vehicle order 1. It commands vehicle 4 to move using unloaded PID values for a bidirectional move to station 2 at an acceleration of 1 m/s^2 and a velocity of 1 m/s. The HLC responses show that the command was accepted and that it completed successfully.

```
0xB0 00000001 0004 00 0002 0000803F 0000803F

0xD0 B0 00 00000001 0004 0002 0001 0000803F 0000803F 00
0xD0 B0 80 00000001 0004 0002 0001 5BE47F3F A6FD7F3F 00
```

Move Vehicle To Position

Message: 0xB1

Source/Destination

Host Controller ► HLC

Purpose

Moves a vehicle to a specified position, relative to the start of a path.

NOTE: The high-level controller, node controllers, and the path the vehicle resides on must be in the operational state. If the path where the vehicle is located is not in the operational state, the command is rejected. If any component required to execute the command is not in an operational state, the command does not complete.

If the commanded acceleration or velocity values are higher than the limit that is defined in the Node Controller Configuration File, the command is rejected.

If the commanded velocity value is higher than the value for a specific motor, the motor velocity value overrides the commanded value while the vehicle is on that motor.

If a motion command is issued to a vehicle already in motion and the command has a lower acceleration than the previous command to that vehicle, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	24
3	Message Type	1	0xB1
4	Order Number	4	0x0…0xFFFFFFFF
8	Vehicle ID	2	1…65535
10	Flags and Direction	1	Bits 0…3 are direction values Bits 4…7 are the PID set index
11	Position	4	-41.0…+41.0 (m, floating point)
15	Path ID	2	1…65535
17	Acceleration/Deceleration	4	0.001…60.0 (m/s^2, floating-point)
21	Velocity	4	0.001…2.5 (m/s, floating-point)
25	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Move Vehicle to Position command.

Order Number – Host controller derived unique order number that is used for tracking, which is included in the command status response messages the high-level controller returns.

Vehicle ID – The ID of the vehicle to move. The ID must be a nonzero positive integer that references a vehicle that exists in the transport system. This value is system-dependent, see MagneMotion Node Controller Interface User Manual, publication MMI-UM001.

Flags and Direction – Bits 0…3 indicate the direction to move the vehicle. Bits 4…7 indicate the PID set index to use when executing the motion command.

Bit	Description
0…3	**Direction:** 0 – **Bidirectional** – If the destination is on the same path the vehicle resides on, the vehicle can move either direction as required to get to the destination in the shortest distance. If the destination is on a path other than the path where the vehicle is located, the forward direction takes precedence for a transport system that is a closed-loop. 1 – **Forward** – Force the vehicle to move forward (downstream) only, useful to implement a unidirectional loop. If the destination is not reachable, the command fails and the vehicle is not moved. 2 – **Backward** – Force the vehicle to move backward (upstream) only, useful to implement a unidirectional loop in the backwards direction. If the destination is not reachable, the command fails and the vehicle is not moved. 3…15 – **Reserved**
4…7	**PID Loop Set Index:** 0 – Use user-defined PID set 0 – Unloaded PID values. 1 – Use user-defined PID set 1 – Loaded PID values. 2 – Use user-defined PID set 2. 3 – Use user-defined PID set 3. 4 – Use user-defined PID set 4. 5 – Use user-defined PID set 5. 6 – Use user-defined PID set 6. 7 – Use user-defined PID set 7. 8 – Use user-defined PID set 8. 9 – Use user-defined PID set 9. 10 – Use user-defined PID set 10. 11 – Use user-defined PID set 11. 12 – Use user-defined PID set 12. 13 – Use user-defined PID set 13. 14 – Use user-defined PID set 14. 15 – Use user-defined PID set 15 – Startup PID values. This PID set is automatically used during startup. If it is not defined, PID set 0 is scaled by 25% and used for startup.

Position – The destination (in meters) of the vehicle relative to the start of the specified path. The position is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401). Zero position is defined as the midpoint of the vehicle at the beginning of the path. When decoupling a vehicle from a stationary platoon, the position that is specified must be the current target for the vehicle as returned through the Extended Vehicle Status. Negative positions can only be used with node types that support movement to a position before the beginning of a path.

Path ID – The ID of the path where the **Position** is located. The ID must be a nonzero 16-bit positive integer that references a path that exists in the configuration.

Acceleration/Deceleration – The maximum acceleration/deceleration rate for the vehicle (in m/s^2) for the motion command, expressed as a 32-bit single-precision floating point number using little-endian format, see Data Encoding on page 401). This value is checked against the limit that is defined in the Node Controller Configuration File, and if higher, the command is rejected. If the velocity is not defined in the NC file, the default is 1.0 m/s. If this value is within the limit that is defined in the Node Controller Configuration File but higher than the value for any specific motor in the route of the vehicle, the value for that motor is used and the command is updated to that lower value.

This value is system-dependent, see MagneMotion Node Controller Interface User Manual, publication MMI-UM001.

Velocity – The maximum velocity for the vehicle (in m/s) to use when moving. This value is a positive number (0.001…2.5 m/s, expressed as a 32-bit single-precision floating point number using little-endian format, see Data Encoding on page 401). This value is checked against the limit that is defined in the Node Controller Configuration File, and if higher, the command is rejected. If the velocity is not defined in the NC file, the default is 1.0 m/s. If this value is within the limit that is defined in the Node Controller Configuration File, but higher than the value for any specific motor in the route of the vehicle, the value for that motor overrides the command value while the vehicle is on that motor.

This value is system-dependent, see MagneMotion Node Controller Interface User Manual, publication MMI-UM001.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends a “Command Accepted” or a “Command Rejected” response as appropriate (see Command Status message). If the command is accepted, the HLC executes a move order for the specified vehicle.

On completion of the command (when the HLC is notified that the vehicle has arrived) at its specified destination, the HLC sends a “Command Complete” response.

NOTE: If the vehicle never reaches its destination, this command never completes.

See Also

Move Vehicle To Station on page 136
Vehicle Follow Order on page 161
Delete Vehicle on page 170
Command Status on page 261

Example

The following example (see Figure 2-1 on page 27 for the system layout) shows a Move Vehicle to Position command. The command is vehicle order 2, which commands vehicle 4 to move using unloaded PID values for a bidirectional move to a position of 0.125 m on path 3 at an acceleration of 1 m/s^2 and a velocity of 1 m/s. The HLC responses show that the command was accepted and that it completed successfully.

```
0xB1 00000002 0004 00 0000003E 0003 0000803F 0000803F

0xD0 B1 00 00000002 0004 0000003E 0003 0000803F 0000803F 00
0xD0 B1 80 00000002 0004 4719EE3D 0003 5BE47F3F A6FD7F3F 00
```

Startup

Message: 0xB2

Source/Destination

Host Controller ➛ HLC

Purpose

Initiates the startup sequence for locating all vehicles on the specified path. If the Path ID specified is zero, the startup sequence for locating all vehicles is executed on all paths in the transport system.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xB2
4	Path ID	2	0…65535
6	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Startup command.

Path ID – The ID of the path to execute startup processing on. The ID must be either a nonzero positive integer that references a path that exists in the configuration or zero to execute startup processing on all paths in the transport system.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a “Command Accepted” or a “Command Rejected” response (see Command Status message) for each path that is specified in the command. If the command is accepted, it is forwarded to the appropriate node controllers for execution.

On completion of the command, the HLC sends a “Command Complete” or a “Command Failed” response for each path that is specified in the command.

See Also

Reset on page 167
Command Status on page 261

Example

The following example (see Figure 2-1 on page 27 for the system layout) shows a Startup command that is issued for all paths. The HLC responses show that the command was accepted for each path and that it completed successfully for each path.

```
0xB2 0000

0xD0 B2 00 0001
0xD0 B2 00 0002
0xD0 B2 00 0003

0xD0 B2 80 0001
0xD0 B2 80 0002
0xD0 B2 80 0003
```

Resume Motion

Message: 0xB3

Source/Destination

Host Controller ➛ HLC

Purpose

Enables motion after it has been disabled with a Suspend Motion or FastStop Motion command on the specified path. If the Path ID specified is zero, motion is resumed on all paths in the transport system. When the Resume is issued, motion based on all currently active motion commands continues.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

The path that is specified must have completed startup. When resuming all paths (Path ID of zero specified), all paths must have completed startup.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xB3
4	Path ID	2	0…65535
6	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Resume Motion command.

Path ID – The ID of the path to resume motion on. The ID must be either a nonzero positive integer that references a path that exists in the configuration or zero to resume motion on all paths in the transport system.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or "Command Rejected" response (see Command Status message) for each path that is specified in the command. If the command is accepted, it is forwarded to the appropriate node controllers for execution.

See Also

Suspend Motion on page 150
FastStop Motion on page 179
Command Status on page 261

Example

The following example (see Figure 2-1 on page 27 for the system layout) shows a Resume motion command that is issued for all paths. The HLC responses show that the command was accepted for each path.

```
0xB3 0000

0xD0 B3 00 0001
0xD0 B3 00 0002
0xD0 B3 00 0003
```

Suspend Motion

Message: 0xB4

Source/Destination

Host Controller ► HLC

Purpose

Suspends all motion on the specified path. If the Path ID specified is zero, motion is suspended on all paths in the transport system. Vehicles immediately decelerate to a controlled stop at the location that they were last given permission to move to. Previously commanded motion does not resume until a Resume Motion command is received. The control loop is still enabled while motion is suspended holding all vehicles in place. Additional vehicles are not allowed to enter the path while it is suspended.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

A request to suspend motion on a path can be issued at any time.

Vehicles in motion within a keep-out area do not stop until they exit the keep-out area.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xB4
4	Path ID	2	0…65535
6	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Suspend Motion command.

Path ID – The ID of the path to resume motion on. The ID must be either a nonzero positive integer that references a path that exists in the configuration or zero to resume motion on all paths in the transport system.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a “Command Accepted” or “Command Rejected” response (see Command Status message) for each path that is specified in the command. If the command is accepted, it is forwarded to the appropriate node controllers for execution.

See Also

Resume Motion on page 148
FastStop Motion on page 179
Command Status on page 261

Example

The following example (see Figure 2-1 on page 27 for the system layout) shows a Suspend Motion command that is issued for all paths. The HLC responses show that the command was accepted for each path.

```
0xB4 0000

0xD0 B4 00 0001
0xD0 B4 00 0002
0xD0 B4 00 0003
```

Status Request

Message: 0xB5

Source/Destination

Host Controller ➛ HLC

Purpose

Requests the status for the item that is specified in the Request Type. For each Request Type, the status can be requested for one specific item or for all items of that type.

NOTE: The high-level controller, node controllers, and the components that are specified must be in the operational state.

When a status request is received for all items in a specific type, a separate status response is sent for each item.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	4, 6, 7
3	Message Type	1	0xB5
4	Request Type	1	0…5
5	ID	2	0…65535
7	Motor Index	1	0…255
Varies	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Status Request command.

Request Type – The type of status request.

Value	Requests Status For
0	**High-Level Controller** – The ID and Motor Index fields can be omitted and are ignored by the HLC if included.
1	**Node Controller** – The ID field specifies the node controller, the Motor Index field can be omitted and is ignored by the HLC if included.
2	**Node** – The ID field specifies the node, the Motor Index field can be omitted and is ignored by the HLC if included.
3	**Path** – The ID field specifies the path, the Motor Index field can be omitted and is ignored by the HLC if included.
4	**Vehicle** – The ID field specifies the vehicle, the Motor Index field can be omitted and is ignored by the HLC if included.
5	**Motor** – The ID field specifies the path that the motors are on and the Motor Index field specifies the specific motor(s) on the path.

ID – The ID of the item that is specified in **Request Type** that status is being requested for. The ID must be either a nonzero positive integer that references a specific item that exists in the configuration or zero to request all elements of the specified type.

When the **Request Type** is 5 (Motor Status), this field specifies the ID of the path where the motors are located. Zero specifies all paths. The **Motor Index** specifies the motor or motors on that path.

Motor Index – The ID of the motor or motors on the specified path when the **Request Type** is 5 (Motor Status). This field must be included when the **Request Type** is 5 (Motor Status). This field can be omitted for a Request Type of 0, 1, 2, 3, or 4.

Value	Description
0	All motors on the specified path.
1…40	Individual Motor Index on the specified path that references a motor that exists in the configuration.
255	All motors with active faults on the specified path.*

* If the host controller specifies 255 to query only motors with active faults and a path has no motors with active faults, one motor status report is sent with the proper **Path ID**. The **Motor Index** is set to 255, which indicates no motors on the path have faults active. See the Motor Status message for more details.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends the response that is shown for the **Request Type** submitted.

For a **Request Type** of 0, a high-level controller status response is sent (see High-Level Controller Status).

For a **Request Type** of 1, 2, 3, or 4, one or more status response messages for the requested type are sent depending on the ID specified in the request. See the specific responses; Node Controller Status, Node Status, Path Status, or Vehicle Status.

For a **Request Type** of 5, one or more status response messages for the requested Motor Index are sent depending on the Motor Index specified in the request. See the Motor Status response.

If a request is made for an item that does not exist in the transport system, a status response message for the corresponding type is returned with the "Present" flag at the beginning of the message set to zero, which indicates no such item exists.

- For example, if node status is requested for a non-existent node, a Node Status response message is returned with the **Node Present** flag set to zero indicating no such node exists.
- For motor status, presence flags are included for both the specified Path ID and Motor Index (see Motor Status).

See Also

Get Station Status on page 190
Get Traffic Light Status on page 199
Get Extended Vehicle Status on page 208
Get Node Controller Digital I/O Status on page 211
MP Get Path End Status on page 219
Get Extended Node Controller Status on page 249
Get Extended High-Level Controller Status on page 252
Get Propulsion Power Status on page 254
High-Level Controller Status on page 271
Node Controller Status on page 273
Node Status on page 275
Path Status on page 281
Vehicle Status on page 284
Motor Status on page 289

Examples

High-Level Controller Status

The following example (see Figure 2-1 on page 27 for the system layout) shows a Status Request command that is issued for the high-level controller. The HLC response provides the status information for the HLC, which shows the HLC is operational and not configured for EtherNet/IP communication.

```
0xB5 00

0xD1 03 00
```

Node Controller Status

The following example (see Figure 2-1 for the system layout) shows a Status Request command that is issued for all node controllers. The HLC responses provide the status information for each node controller, which shows all node controllers are present and operational.

```
0xB5 01 0000

0xD2 0001 01 03
0xD2 0002 01 03
```

Node Status

The following example (see Figure 2-1 for the system layout) shows a Status Request command that is issued for all nodes. The HLC responses provide the status information for each node, which shows all nodes are present and operational, node 1 is a Merge Node, and node 2 is a Diverge Node.

```
0xB5 02 0000

0xD3 0001 01 01 0000 00 00 03
0xD3 0002 01 02 0000 00 00 03
```

Path Status

The following example (see Figure 2-1 for the system layout) shows a Status Request command that is issued for all paths. The HLC responses provide the status information for each path, which shows all paths are present and operational, paths 1 and 3 each have two vehicles, motion is enabled, and upstream and downstream communication links are OK.

```
0xB5 03 0000

0xD4 0001 01 02 02 00 00 00
0xD4 0002 01 00 02 00 00 00
0xD4 0003 01 02 02 00 00 00
```

Vehicle Status

The following example (see Figure 2-1 for the system layout) shows a Status Request command that is issued for all vehicles. The HLC responses provide the status information for each vehicle, which shows all four vehicles, the paths they are on, their current position and motion commands. The response for vehicle 4 shows that it completed a Move to Station command (0xB0).

```
0xB5 04 0000

0xD5 0001 01 0003 0000 6666663F 00000000 00 01 00000000
0xD5 0002 01 0001 0000 00002C40 00000000 00 01 00000000
0xD5 0003 01 0003 0000 0000003F 00000000 00 01 00000000
0xD5 0004 01 0001 0001 0000403F 00000000 B0 01 00000000
```

The following example (see Figure 2-1 for the system layout) shows a Status Request command that is issued for all vehicles. The HLC responses provide the status information for each vehicle, which shows all four vehicles, the paths they are on, their current position and motion commands. The response for vehicle 3 shows that it is following vehicle 4.

```
0xB5 04 0000

0xD5 0001 01 0003 0003 d2ffff3e 00000000 00 01 00000000
0xD5 0002 01 0002 0000 fdffbf3f 00000000 00 01 00000000
0xD5 0003 01 0001 0000 fdffbf3f 00000000 B7 01 8399993e
0xD5 0004 01 0001 0001 5e66e63f 00000000 00 01 00000000
```

Motor Status

The following example (see Figure 2-1 for the system layout) shows a Status Request command that is issued for all motors on all paths in an MM LITE transport system. The HLC responses provide the status information for the motors, which shows all motors are present and operational.

NOTE: Example output includes only faults for driver boards 1…4.

```
0xB5 05 0000 00

0xD7 0001 01 01 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0001 01 02 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0001 01 03 01 02 04 00 00 00 00 00 00 00 00 00 00
0xD7 0001 01 04 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0001 01 05 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0001 01 06 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0001 01 07 01 02 01 00 00 00 00 00 00 00 00 00 00

0xD7 0002 01 01 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0002 01 02 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0002 01 03 01 02 01 00 00 00 00 00 00 00 00 00 00

0xD7 0003 01 01 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0003 01 02 01 02 04 00 00 00 00 00 00 00 00 00 00
0xD7 0003 01 03 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0003 01 04 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0003 01 05 01 02 01 00 00 00 00 00 00 00 00 00 00
0xD7 0003 01 06 01 02 04 00 00 00 00 00 00 00 00 00 00
0xD7 0003 01 07 01 02 01 00 00 00 00 00 00 00 00 00 00
```

Set Signal

Message: 0xB6

Source/Destination

Host Controller ► HLC

Purpose

Sets and clears signals that are affiliated with a Terminus Node, which is used to move a vehicle on or off a path to or from user-supplied equipment. See Node Status for the current state of the signals. See Terminus Node on page 93 for a description of handshaking using these signals.

NOTE: The high-level controller, node controllers, and the path where the Terminus Node is located must be in the operational state. If not, the command is rejected. If any component required to execute the command is not in the operational state, the command does not complete.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	9
3	Message Type	1	0xB6
4	Node ID	2	1…65535
6	Signal Number	1	0, 1
7	Signal Level	1	0, 1
8	Vehicle ID (if applicable)	2	0…65535
10	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Set Signal command.

Node ID – The ID of the Terminus Node for the signal. The ID must be a nonzero positive integer that references a Terminus Node that exists in the configuration.

Signal Number – Specifies the signal to set.

Value	Signal
0	ENTRY_REQUESTED
1	EXIT_ALLOWED

Signal Level – Specifies the level of the signal.

Value	Signal Level
0	Low (inactive)
1	High (active)

Vehicle ID – Only valid for a request for a vehicle to enter, this field is ignored for a request to exit. This field specifies the ID to use for the vehicle that is entering (when setting ENTRY_REQUESTED High).

- To specify a vehicle ID, the ID must be a nonzero positive integer that is not currently used by another vehicle.
- To have the HLC assign the vehicle ID, enter zero. The Node Status response contains the **Vehicle ID** assigned by the transport system when the host controller specifies a Vehicle ID of “0” in an entry request.

NOTE: Rockwell Automation recommends not specifying a Vehicle ID, which commands the system to assign it (Vehicle ID = `0000`).

Message CRC - A check value of the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller for execution.

See Also

Command Status on page 261
Node Status on page 275

Examples

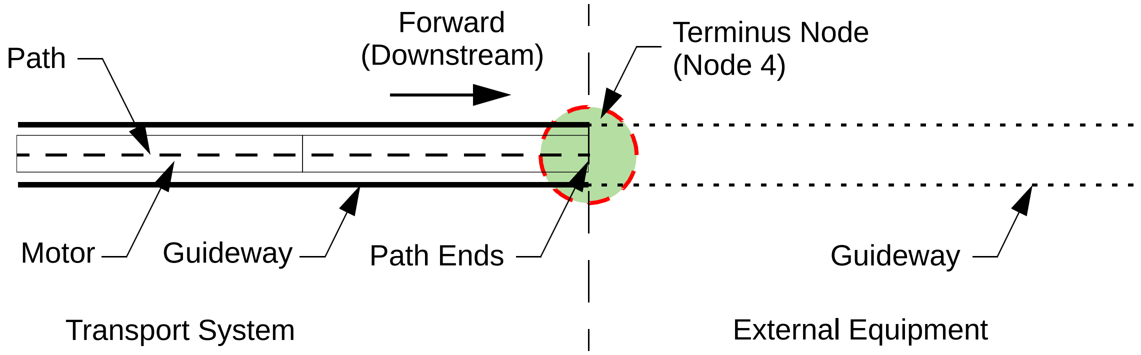


Figure 4-1: Terminus Node Example

The following example (see Figure 4-1 for the node configuration) shows a Set Signal command that is issued for a Terminus Node. The command allows a vehicle onto the transport system from the external equipment. The HLC response shows that the command was accepted and vehicle ID 1 was assigned to the incoming vehicle.

```
0xB6 0004 00 01 0000

0xD0 B6 00 0004 00 01 0001
```

The following example (see Figure 4-1 for the node configuration) shows a Set Signal command that is issued for a Terminus Node. The command allows vehicle 1 off the transport system and onto the external equipment. The HLC response shows that the command was accepted.

```
0xB6 0004 01 01 0001

0xD0 B6 00 0004 01 01
```

Vehicle Follow Order

Message: 0xB7

Source/Destination

Host Controller ➛ HLC

Purpose

Orders a vehicle to catch up to another vehicle using a specified profile and then maintain a specified following distance. Once the vehicle has caught up to the specified vehicle at the commanded following distance, the system maintains the following distance indefinitely, matching the motion profile of the followed vehicle. The vehicle continues to follow the specified vehicle at the specified distance until the follow order is overridden with a Move Vehicle To Position, a Move Vehicle To Station, or a new Vehicle Follow Order.

There is an auto-decouple feature, where a decouple path ID and position are defined. When the platoon with the following vehicle approaches that position, the vehicle with that position specified as part of its follow order decouples from the platoon, slows down, and stops at that position.

NOTE: The high-level controller, node controllers, and the path the vehicle resides on must be in the operational state. If the path where the vehicle is located is not in the operational state, the command is rejected. If any component required to execute the command is not in an operational state, the command does not complete.

Vehicles must be stopped to couple them into a platoon.

Platoons must be stopped to decouple vehicles unless the command coupling a vehicle to the platoon specified a decouple destination for that vehicle.

If the acceleration or velocity values in a motion command are higher than the limit that is defined in the Node Controller Configuration File, the command is rejected.

If the commanded velocity value is higher than the value for a specific motor, the motor overrides that value while the vehicle is on that motor.

If a move command is issued to a vehicle already in motion and the command has a lower acceleration than the previous command to that vehicle, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	31
3	Message Type	1	0xB7
4	Order Number	4	0x0…0xFFFFFFFF
8	Vehicle ID	2	1…65535
10	Direction	1	1, 2
11	Follow Distance	4	0…41.0 (m, floating point)
15	Followed Vehicle ID	2	1…65535
17	PID Set Index	1	0…15
18	Catchup or Decouple Acceleration	4	0.0…100.0 (m/s^2, floating point)
22	Catchup or Decouple Velocity	4	0.0…100.0 (m/s, floating point)
26	Decouple Destination Path ID (MM LITE only)	2	0…65535
28	Decouple Destination Position (MM LITE only)	4	0…41.0 (m, floating point)
32	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Vehicle Follow Order command.

Order Number – Host controller derived unique order number that is used for tracking, which is included in the command status response messages the high-level controller returns. Allows the host to track internally any specific vehicle follow order message that is sent to the HLC.

Vehicle ID – The ID number of the vehicle to execute this vehicle follow order. This vehicle follows the vehicle that **Followed Vehicle ID** specifies. The ID must be a nonzero positive integer that references a vehicle that exists in the transport system.

Direction – The direction on the track relative to this vehicle where the vehicle specified by **Followed Vehicle ID** is located.

Value	Description
0x01	Vehicle to follow is downstream of this vehicle (motion is downstream).
0x02	Vehicle to follow is upstream of this vehicle (motion is upstream).

Follow Distance – The distance (in meters) center-to-center that this vehicle maintains behind the vehicle being followed (specified by **Followed Vehicle ID**) in the **Direction** that is specified, once the pair of vehicles are coupled into a platoon. The distance is expressed as a 32-bit single-precision floating point number using little-endian format.

NOTE: When coupling to a platoon, the vehicle must be within 30 mm of this distance before sending the vehicle follow order.

Followed Vehicle ID – The ID number of the vehicle being followed by this vehicle (executing this vehicle follow order). The Followed Vehicle ID must be a nonzero positive integer.

PID Set Index – The PID set to use when executing the command.

Bit	Description
0…3	0…15 – Reserved
4…7	**PID Loop Set Index:** 0 – Use user-defined PID set 0 – Unloaded PID values. 1 – Use user-defined PID set 1 – Loaded PID values. 2 – Use user-defined PID set 2. 3 – Use user-defined PID set 3. 4 – Use user-defined PID set 4. 5 – Use user-defined PID set 5. 6 – Use user-defined PID set 6. 7 – Use user-defined PID set 7. 8 – Use user-defined PID set 8. 9 – Use user-defined PID set 9. 10 – Use user-defined PID set 10. 11 – Use user-defined PID set 11. 12 – Use user-defined PID set 12. 13 – Use user-defined PID set 13. 14 – Use user-defined PID set 14. 15 – Use user-defined PID set 15 – Startup PID values. This PID set is automatically used during startup. If it is not defined, PID set 0 is scaled by 25% and used for startup.

Catchup or Decouple Acceleration – The maximum acceleration/deceleration (system dependent) for the vehicle (in m/s^2) to use when catching up to the vehicle to follow or to use when auto-decoupling from the vehicle that it is following. This value is a positive number (0.001…60.0 m/s2, expressed as a 32-bit single-precision floating point number using little-endian format). This value is checked against the system limit that is defined in the Node Controller Configuration File, and if higher, the command is updated to the value from the Node Controller Configuration File. Only available in MagneMover LITE systems.

NOTE: Currently only used when decoupling as vehicles must not be moving when creating a platoon, but it must always be defined.

Catchup or Decouple Velocity – The maximum velocity (system dependent) for the vehicle (in m/s) to use when catching up to the vehicle to follow or to use when auto-decoupling from the vehicle that it is following. This value is a positive number that is expressed as a 32-bit single-precision floating point number using little-endian format. This value is checked against the system limit that is defined in the Node Controller Configuration File, and if higher, the command is updated to the value from the Node Controller Configuration File. Only available in MagneMover LITE systems.

NOTE: Currently only used when decoupling as vehicles must not be moving when creating a platoon, but it must always be defined.

Decouple Destination Path ID – The ID of the path where the decoupling position is located (where the vehicle stops after decoupling). The ID must be a nonzero positive integer that references a path that exists in the configuration. If the value is zero, the decoupling feature is disabled. Only available in MagneMover LITE systems.

Decouple Destination Position – The location of the decouple location in meters relative to the start of the specified path (where the vehicle stops after decoupling). The position is expressed as a 32-bit single-precision floating point number using little-endian format. Zero position is defined as the midpoint of the vehicle at the beginning of the path. This value is only used when **Decouple Path ID** is not zero. Only available in MagneMover LITE systems.

Message CRC - A check value of the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller for execution.

- **Vehicle Following –** If the auto-decouple feature is not configured in the vehicle follow order (Destination Path ID = 0), when the command completes the following events occur:

  A. When the vehicle is within the following distance from the specified vehicle, the Caught Up bit is set in the Extended Vehicle Status, Vehicle Flags.

  B. The HLC sends a Command Status message with a "Command Status –Vehicle has caught up" response (0x81).

  NOTE: If the vehicle never catches up to the specified vehicle, this command never completes.

- **Vehicle Auto-decouple –** If the auto-decouple feature is configured in the vehicle follow order (Destination Path ID > 0), when the vehicle begins to brake for arrival at the destination path and position that is supplied in the follow order the following events occur:

  A. The HLC sends a Command Status message with a "Vehicle Decoupling" response (0x82).

  B. The Vehicle Decoupled bit is set in the Extended Vehicle Status, Vehicle Flags).

  C. The order type for the command the vehicle is under changes from a Vehicle Follow Order (0xB7) to a Vehicle Position Order (0xB1).

  D. A "Command Complete" response (0x80) is sent for the Vehicle Position Order (0xB1) when the vehicle arrives at the defined decouple position. If the vehicle has followers during auto decoupling, it becomes the head of a new platoon for these followers.

  E. The Vehicle Decoupled bit is cleared in the Extended Vehicle Status, Vehicle Flags).

See Also

Move Vehicle To Station on page 136
Move Vehicle To Position on page 141
Delete Vehicle on page 170
Extended Vehicle Status, on page 307
Command Status on page 261
Vehicle Status on page 284

Example

The following example (see Figure 2-1 on page 27 for the system layout) shows a Vehicle Follow Order command. The command is order 7, which is issued to create a platoon with vehicle 3 following vehicle 4, which is downstream from it, at a distance of 0.3 m using unloaded PID values, a catchup acceleration of 1 m/s^2, a catchup velocity of 1 m/s, without automatic decoupling. The HLC responses show that the command was accepted and that the vehicle has caught up to the platoon.

```
0xB7 00000007 0003 01 9a99993e 0004 00 0000803f 0000803f
0000 00000000

0xD0 B7 00 00000007 0003 01 9A99993E 0004 00 0000803F
0000803F 0000 00000000
0xD0 B7 81 00000007 0003 01 8399993E 0004 00 5BE47F3F
A6FD7F3F 0000 00000000
```

The following example (see Figure 2-1 on page 27 for the system layout) shows a Vehicle Follow Order command. The command is order 8, which is issued to create a platoon with vehicle 3 following vehicle 4, which is downstream from it, at a distance of 0.3 m using unloaded PID values, a catchup acceleration of 1 m/s^2, a catchup velocity of 1 m/s, with automatic decoupling set for 0.5 m on path 2. The HLC responses show that the command was accepted and that the vehicle has caught up to the platoon.

```
0xB7 00000008 0003 01 9A99993E 0004 00 0000803F 0000803F
0002 0000003F

0xD0 B7 00 00000008 0003 01 9A99993E 0004 00 0000803F
0000803F 0002 0000003F
0xD0 B7 81 00000008 0003 01 8399993E 0004 00 5BE47F3F
A6FD7F3F 0002 EBFFFF3E
```

Reset

Message: 0xB8

Source/Destination

Host Controller ► HLC

Purpose

Performs a cold rest of all the motors on the specified path. The runtime software in each motor is restarted and all vehicle records for vehicles on the specified paths are deleted. If the Path ID specified is zero, all paths in the transport system are reset and all vehicle records are deleted. After any path is reset, a Startup command must be executed on it to return it to the operational state.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

A Reset command is valid at any time except when the high-level controller is in the initialization state.

All motion on the specified paths must be suspended using a Suspend Motion command before a Reset is issued.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xB8
4	Path ID	2	0…65535
6	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Reset command.

Path ID – The ID of the path to reset. The ID must be either a nonzero positive integer that references a path that exists in the configuration, or zero to reset all paths in the transport system.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a “Command Accepted” or a “Command Rejected” response as appropriate (see Command Status message) for each path that is specified in the command. If the command is accepted, it is forwarded to the appropriate node controllers for execution.

On command completion, the HLC sends a “Command Complete” or “Command Failed” response for each path that is specified in the command.

See Also

Startup on page 146
Suspend Motion on page 150
Warm Reset on page 181
Command Status on page 261
Path Status on page 281

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a Reset command that is issued for all paths. The HLC responses show that the command was accepted for each path and that it completed successfully for each path.

```
0xB8 0000

0xD0 B8 00 0001
0xD0 B8 00 0002
0xD0 B8 00 0003

0xD0 B8 80 0001
0xD0 B8 80 0002
0xD0 B8 80 0003
```

Delete Vehicle

Message: 0xB9

Source/Destination

Host Controller ► HLC

Purpose

Commands a node controller and all related motors to delete any vehicle records with a matching ID. The node controller deletes the record so the vehicle can't own any nodes and the motors delete their records if the vehicle has permissions for blocks on that motor.

This command must be used carefully. It is typically used only when a vehicle is physically removed from the guideway because it is inoperable, or when a vehicle entry is canceled. If a vehicle entry or exit is active for the vehicle, the entry/exit state machine is also cleared.

NOTE: The high-level controller, node controllers, and the path the vehicle resides on must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Once the command is issued, the area on the motor where the vehicle is located is defined as empty. Caution must be taken, as a collision can occur if another vehicle attempts to move into that area before the vehicle is physically removed. Therefore, when using the Delete Vehicle command, the vehicle must be removed before issuing the command.

Rockwell Automation recommends using a Terminus Node to add or remove vehicles on the transport system.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xB9
4	Vehicle ID	2	1…65535
6	Message CRC	2	1…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Delete Vehicle command.

Vehicle ID – The ID of the vehicle to delete. The ID must be a nonzero positive integer that references a vehicle that exists in the transport system.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller for execution.

On completion of the command, the HLC sends a "Command Complete" or "Command Failed" response.

See Also

Move Vehicle To Station on page 136
Move Vehicle To Position on page 141
Vehicle Follow Order on page 161
Command Status on page 261
Vehicle Status on page 284
Extended Vehicle Status, on page 307

Examples

The following example shows a Delete Vehicle command that is issued for vehicle 4. The HLC responses show that the command was accepted and that it completed successfully.

```
0xB9 0004

0xD0 B9 00 0004
0xD0 B9 80 0004
```

Set Control Loop Parameters

Message: 0xBA

Source/Destination

Host Controller ➛ HLC

Purpose

Modifies one or more PID Control Loop parameters for the specified PID set for all motors on the specified path for testing. Once the correct PID settings have been identified, the Node Controller Configuration File must be updated to make sure that those values are always used.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

These changes are not persistent. If the node controller is restarted or rebooted, it reverts to the Node Controller Configuration File settings.

Issuing this command when a vehicle is in motion could cause a loss of vehicle control.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	26
3	Message Type	1	0xBA
4	Path ID	2	0…65535
6	Proportional Loop Gain (P or K_p)	4	0…10,000.0 (floating-point)
10	Integral Loop Gain (I or K_i)	4	0…1,000.0 (floating-point)
14	Derivative Loop Gain (D or K_d)	4	0…1,000.0 (floating-point)
18	Vehicle Mass	4	0…10,000.0 (kg, floating-point)
22	PID Set Index	1	0…15
23	Scaling Factor (K_{ff})	4	0…400.0 (floating-point)
27	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Set Control Loop Parameters command.

Path ID – The ID of the path where the motors are located. The ID must be either a nonzero positive integer that references a path that exists in the configuration or zero to set all paths in the transport system.

Proportional Loop Gain (K_p) – Proportional Loop Gain parameter (in N/m/kg) to set in the PID Control Loop for all motors on the specified path. K_p is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

Integral Loop Gain (K_i) – Integral Loop Gain parameter (in N/(m-s)/kg) to set in the PID Control Loop for all motors on the specified path. K_i is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

Derivative Loop Gain (K_d) – Derivative Loop Gain parameter (in N/(m/s)/kg) to set in the PID Control Loop for all motors on the specified path. K_d is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

Vehicle Mass – Vehicle mass (in kg) for the specified PID set. The mass is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

PID Set Index – Indicates the PID control loop set to be updated.

Value	Description
0	Use user-defined PID set 0 – Unloaded PID values.
1	Use user-defined PID set 1 – Loaded PID values.
2	Use user-defined PID set 2.
3	Use user-defined PID set 3.
4	Use user-defined PID set 4.
5	Use user-defined PID set 5.
6	Use user-defined PID set 6.
7	Use user-defined PID set 7.
8	Use user-defined PID set 8.
9	Use user-defined PID set 9.
10	Use user-defined PID set 10.
11	Use user-defined PID set 11.
12	Use user-defined PID set 12.
13	Use user-defined PID set 13.
14	Use user-defined PID set 14.
15	Use user-defined PID set 15 – Startup PID values. This PID set is automatically used during startup. If it is not defined, PID set 0 is scaled by 25% and used for startup.

Scaling Factor (K_{ff}) – Scaling factor, expressed as a percentage from 0% to 400%, to apply to the PID Control Loop Feed-Forward parameter for all motors on the specified path. K_{ff} is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller for execution.

See Also

Command Status on page 261

Examples

The following example shows the PID settings for path 1 being changed to K_p = 1000, K_i = 11, K_d = 50, unloaded vehicle mass = 0.33 kg, and K_{ff} = 100. The HLC response shows that the command was accepted.

```
0xBA 0001 00007A44 00003041 00004842 C2F5A83E 00 0000C842

0xD0 BA 00 0001
```

Set Node Parameters

Message: 0xBB

Source/Destination

Host Controller ➤ HLC

Purpose

Sets the node parameters as specified by the arguments. This command is used for enabling and disabling Gateway Nodes.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

These changes are not persistent. If the node controller is restarted or rebooted, the Gateway Nodes default to disabled.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBB
4	Node ID	2	1…65535
6	Command Subtype	1	0
7	Command Data	1	0, 1
8	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Set Node Parameters command.

Node ID – The ID of the Gateway Node to which the parameter modification applies. The ID must be a nonzero positive integer that references a Gateway Node that exists in the configuration. Zero is not a valid Node ID for this command.

Command Subtype – The only Command Subtype supported in this software version is 0, which designates a node enable or disable command.

Command Data – Specifies the enable/disable action.

Value	Description
0	Disable the node, prevents vehicles from crossing the specified Gateway Node.
1	Enable the node, allows vehicles to cross the specified Gateway Node.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller for execution.

See Also

Command Status on page 261

Examples

The following example shows the status for node 1 being changed to allow vehicles to cross the node. The HLC response acknowledges receipt of the command.

```
0xBB 0001 00 01

0xD0 BB 00 0001
```

FastStop Motion

Message: 0xBC

Source/Destination

Host Controller ► HLC

Purpose

Suspends all motion on the specified path. If the Path ID specified is zero, motion is suspended on all paths in the transport system. Vehicles immediately decelerate with maximum thrust opposing motion. Previously commanded motion does not resume until a Resume Motion command is received. The control loop is still enabled while motion is suspended holding all vehicles in place. Additional vehicles are not allowed to enter the path while it is suspended.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

A request to FastStop motion on a path can be issued at any time.

Vehicles in motion within a keep-out area stop within the keep-out area.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xBC
4	Path ID	2	0…65535
5	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a FastStop Motion command.

Path ID – The ID of the path to suspend motion on. The ID must be either a nonzero positive integer that references a path that exists in the configuration or zero to suspend motion on all paths in the transport system.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a “Command Accepted” or a “Command Rejected” response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller for execution.

See Also

Resume Motion on page 148
Suspend Motion on page 150
Command Status on page 261

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a FastStop Motion command that is issued for all paths. The HLC responses show that the command was accepted for each path.

```
0xBC 0000

0xD0 BC 00 0001
0xD0 BC 00 0002
0xD0 BC 00 0003
```

Warm Reset

Message: 0xBD

Source/Destination

Host Controller ► HLC

Purpose

Performs a warm reset of all motors on the specified path. The runtime software in each motor is restarted and all vehicle records for vehicles on the specified paths are deleted. The network chip inside each motor is not reset, which allows the network link for the motor to be operational through the reset process. If the Path ID specified is zero, all paths in the transport system are reset and all vehicle records are deleted. After any path is reset, a Startup command must be executed on that path to return it to the operational state.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

A Warm Reset command is valid at any time except when the high-level controller is in the initialization state.

All motion on the specified paths must be suspended using a Suspend Motion command before a Warm Reset is issued.

A Warm Reset command reverts to a cold Reset when issued on a path that does not support warm resets.

Support

Warm Resets are supported in the latest software release for the following motors that use Ethernet for motor to motor and for motor to node controller communications:

- MagneMover LITE Gen4 Ethernet motors.
- QuickStick 150 motors.
- QuickStick HT 5700 motors.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xBD
4	Path ID	2	0…65535
6	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Warm Reset command.

Path ID – The ID of the path to reset. The ID must be either a nonzero positive integer that references a path that exists in the configuration, or zero to reset all paths in the transport system.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message) for each path that is specified in the command. If the command is accepted, it is forwarded to the appropriate node controller for execution.

On completion of the command the HLC sends a "Command Complete" or a "Command Failed" response for each path that is specified in the command.

See Also

Startup on page 146
Suspend Motion on page 150
Reset on page 167
Command Status on page 261
Path Status on page 281

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a Warm Reset command that is issued for all paths. The HLC responses show that the command was accepted for each path and that it completed successfully for each path.

```
0xBD 0000

0xD0 BD 00 0001
0xD0 BD 00 0002
0xD0 BD 00 0003

0xD0 BD 80 0001
0xD0 BD 80 0002
0xD0 BD 80 0003
```

Create Station

Message: 0xBF 0x01 0x01

Source/Destination

Host Controller ► HLC

Purpose

Used to dynamically create a station or modify an existing station at a specified position on the designated Path.

NOTE: The high-level controller, node controllers, and the path the station resides on must be in the operational state.

Attempting to modify a station that is in use by one or more vehicles is rejected.

Dynamically created stations or changes to existing stations are not maintained after an HLC restart or after loading and activating a Node Controller Configuration File.

NOTE: These stations are added in the node controller configuration file or dynamically created, and do not function the same as the station commands [raM_MM_Tec_Station, raM_MM_Tec_SupervisedStation} that are found in the MagneMotion libraries from the PCDC. It is highly recommended for new systems that stations are defined and controlled using the station AOIs (add-on instructions) in the MagneMotion library.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	13
3	Message Type	1	0xBF
4	Extension Type	1	0x01
5	Extension Subtype	1	0x01
6	Path ID	2	1…65535
8	Position	4	0…+41.0 (m, floating point)
12	Station ID	2	0…2048
14	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Create Station extension command.

Extension Type – Fixed extension type that identifies this command as a Station extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Create Station command.

Path ID – The ID of the path where the station is being added or changed. The ID must be a nonzero positive integer that references a path that exists in the configuration.

Position – The position of the station in meters relative to the start of the specified path. The position is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401). Zero position is defined as the beginning of the path.

Station ID – The ID of the station to create or change.

Value	Description
0	Create a new station. The Station ID is returned in the Command Status response.
1…2048	If the station does not exist, the new station is created with the requested station ID. If the station exists, the location of the station is changed to the specified position on the designated path.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a “Command Accepted” or a “Command Rejected” response as appropriate (see Command Status message). If the command is accepted, the HLC creates or changes the station as specified and the ID of the new/modified station is included in the Command Status response.

See Also

Delete Station on page 187
Get Station Status on page 190
Command Status on page 261
Station Status on page 303

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a command to create Station 2 at 1.25 m on Path 1. The HLC response shows that the command was accepted and that it completed successfully.

```
0xBF 01 01 0001 0000A03F 0002

0xD0 BF 00 01 01 0001 0000A03F 0002
```

The following example (see Figure 2-1 on page 27 for the system layout) shows the existing station at 1.25 m on Path 1 being moved to 2.50 m on Path 1. The HLC response shows that the command was accepted and that it completed successfully.

```
0xBF 01 01 0001 00002040 0002

0xD0 BF 00 01 01 0001 00002040 0002
```

Delete Station

Message: 0xBF 0x01 0x02

Source/Destination

Host Controller ► HLC

Purpose

Used to delete a specific station or all stations that are defined in the transport system. Any station, whether defined statically by the Node Controller Configuration File or created dynamically using the Create Station command, can be deleted.

NOTE: The high-level controller, node controllers, and the path the station resides on must be in the operational state.

Attempting to delete a station that is in use by one or more vehicles is rejected.

Dynamically created stations are not persistent after an HLC restart or after loading a Node Controller Configuration File.

NOTE: These stations are added in the node controller configuration file or dynamically created, and do not function the same as the station commands [raM_MM_Tec_Station, raM_MM_Tec_SupervisedStation} that are found in the MagneMotion libraries from the PCDC. It is highly recommended for new systems that stations are defined and controlled using the station AOIs (add-on instructions) in the MagneMotion library.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBF
4	Extension Type	1	0x01
5	Extension Subtype	1	0x02
6	Station ID	2	0…2048
7	Message CRC	3	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Station extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Delete Station command.

Station ID – The ID of the station to delete.

Value	Description
0	Deletes all currently defined stations.
1…2048	Deletes the specified station.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC send on or more Command Status messages to the host controller as follows:

- If the station ID field is 0, the HLC deletes all currently defined stations. A Command Status message is returned for each station with its success or failure status. If no stations exist, the command is accepted by the HLC and a station ID of 0 and a command status of "Command Accepted" is returned in the Command Status message.
- If the station ID field is nonzero, only the station with the matching station ID is deleted. If the specified station does not exist, a Command Status message is returned with the failing station ID and the command status set to indicate that no such station exists in the transport system.
- If the command is rejected, the ID of the station and the failure status is included in the Command Status message.

See Also

Create Station on page 184
Get Station Status on page 190
Command Status on page 261
Station Status on page 303

Example

The following example (see Figure 2-1 on page 27 for the system layout) shows Station 2 being deleted. The HLC response shows that the command was accepted and that it completed successfully.

```
0xBF 01 02 0002

0xD0 BF 00 01 02 0002
```

Get Station Status

Message: 0xBF 0x01 0x03

Source/Destination

Host Controller ► HLC

Purpose

Requests the status for a specific station or for all stations in a transport system.

NOTE: The high-level controller, node controllers, and the path the station resides on must be in the operational state.

When status is requested for all stations, a separate status response is sent for each station.

NOTE: These stations are added in the node controller configuration file or dynamically created, and do not function the same as the station commands [raM_MM_Tec_Station, raM_MM_Tec_SupervisedStation} that are found in the MagneMotion libraries from the PCDC. It is highly recommended for new systems that stations are defined and controlled using the station AOIs (add-on instructions) in the MagneMotion library.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBF
4	Extension Type	1	0x01
5	Extension Subtype	1	0x03
6	Station ID	2	0…2048
8	Message CRC	3	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Station extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Get Station Status command.

Station ID – The ID of the station that status is requested for.

Value	Description
0	Returns the status for all currently defined stations.
1…2048	Returns the status for the specified station.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends one or more Station Status messages to the host controller as follows:

- If the station ID field is 0, the HLC gets status for all currently defined stations. A Station Status message is returned for each station with its success or failure status. If no stations exist, the command is accepted by the HLC and a station ID of 0 and a command status of "Command Accepted" is returned in the Station Status message.
- If the station ID field is nonzero, the HC gets status for only the station with the matching station ID. If the specified station does not exist, a Station Status message is returned with the failing station ID and the command status set to indicate that no such station exists in the transport system.
- If the command is rejected, the ID of the station and the failure status is included in the Station Status message.

See Also

Station Status on page 303

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a Get Station Status command that is issued for all stations. The HLC responses provide the status information for each station, which includes the station ID and location.

```
0xBF 01 03 0000

0xDF 01 01 00 0001 0003 00002040
0xDF 01 01 00 0002 0001 0000A03F
```

Create Traffic Light

Message: 0xBF 0x02 0x01

Source/Destination

Host Controller ► HLC

Purpose

Creates a traffic light at the specified position on the designated path. When a new traffic light is created, it is assigned the next available Traffic Light ID and set to green to allow vehicles to move beyond the traffic light position.

There is a limit of one traffic light per motor block and 32 traffic lights per path. The HLC rejects commands that attempt to create a traffic light on a motor block where a traffic light is already located or create more than 32 traffic lights on a path. The motor block lengths for each motor type are provided in Table 3-1 on page 80 for reference.

NOTE: The high-level controller, node controllers, and the path the traffic light resides on must be in the operational state.

These changes are not persistent. Any traffic lights created using this command must be recreated in the following situations; whenever the node controller is restarted or rebooted, or whenever any paths where traffic lights are located are reset.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	15
3	Message Type	1	0xBF
4	Extension Type	1	0x02
5	Extension Subtype	1	0x01
6	Path ID	2	1…65535
8	Position	4	0…+41.0 (m, floating point)
12	Host Command Count	4	0x0…0xFFFFFFFF
16	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Traffic Light extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Create Traffic Light extension command.

Path ID – The ID of the path where the traffic light is to be placed.

Position – The position (in meters) relative to the start of the specified path where the traffic light is to be placed. Expressed as a 32-bit single-precision floating-point number.

Host Command Count – A unique value is supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in Command Status messages.

The host controller maintains the Host Command Count. It is initialized to 1 when the host controller restarts, it is then incremented for each new command that is issued, and continues from 1 when it rolls over.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC creates the traffic light and sets it to green on the targeted motor block.

On command completion, the HLC sends a "Command Complete" or "Command Failed" response. The ID of the new traffic light is included in the Command Status message.

See Also

Set Traffic Light on page 196
Get Traffic Light Status on page 199
Delete Traffic Light on page 202
Command Status on page 261
Traffic Light Status on page 305

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a command to create a traffic light at a position of 1.5 m from the beginning of path 1. The HLC responses show that the command was accepted, that it completed successfully, and includes the ID of the new traffic light.

```
0xBF 02 01 0001 0000C03F 00000001

0xD0 BF 00 02 01 0001 0000C03F 00000001 0001
0xD0 BF 80 02 01 0001 0000C03F 00000001 0001
```

Set Traffic Light

Message: 0xBF 0x02 0x02

Source/Destination

Host Controller ► HLC

Purpose

Used to set the color of a traffic light to either green or red. Depending on color, a traffic light controls vehicle motion as follows:

- When set to green, vehicles are granted permission to move beyond the position of the traffic light.
- When set to red, vehicles are not granted permission to move into or beyond the motor block at the position of the traffic light. The obstructed status bit for the vehicle is set if the vehicle is stopped at a traffic light is waiting in a queue of vehicles, or is waiting at a switch.
- Vehicles that have permission to move beyond the motor block at the position of the traffic light do not stop if the traffic light is commanded to turn red.

NOTE: The high-level controller, node controllers, and the path the traffic light resides on must be in the operational state.

These changes are not persistent. Any traffic lights set using this command must be reset in the following situations; whenever the node controller is restarted or rebooted, or whenever any paths where traffic lights are located are reset.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	12
3	Message Type	1	0xBF
4	Extension Type	1	0x02
5	Extension Subtype	1	0x02
6	Traffic Light ID	2	1…4096
8	Color	1	0, 1
9	Host Command Count	4	0x0…0xFFFFFFFF
13	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Traffic Light extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Set Traffic Light command.

Traffic Light ID – The ID of the traffic light to change. The ID must be a nonzero positive integer that references a traffic light that exists in the transport system.

Color – The color to set for the specified traffic light.

Value	Description
0	Green (allows traffic to pass)
1	Red (stops traffic)

Host Command Count – - A unique value supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in Command Status messages

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC sets the traffic light to the specified color.

On command completion, the HLC sends a "Command Complete" or a "Command Failed" response.

See Also

Create Traffic Light on page 193
Get Traffic Light Status on page 199
Delete Traffic Light on page 202
Command Status on page 261
Traffic Light Status on page 305

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a command to set traffic light 1 to red. The HLC response shows that the command was accepted, that it completed successfully, and the color changed to red.

```
0xBF 02 02 0001 01 0000000d

0xD0 BF 00 02 02 0001 01 0000000d
0xD0 BF 80 02 02 0001 01 0000000d
```

Get Traffic Light Status

Message: 0xBF 0x02 0x03

Source/Destination

Host Controller ► HLC

Purpose

Requests the status for a specific traffic light or for all traffic lights in a transport system.

NOTE: The high-level controller, node controllers, and the path the traffic light resides on must be in the operational state.

When status is requested for all traffic lights, a separate status response is sent for each traffic light.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	12
3	Message Type	1	0xBF
4	Extension Type	1	0x02
5	Extension Subtype	1	0x03
6	Traffic Light ID	2	0…4096
8	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Traffic Light extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Get Traffic Light Status command.

Traffic Light ID – The ID of the traffic light that status is requested for.

Value	Description
0	Get the status for all currently defined traffic lights.
1…4096	Get the status for the specified traffic light.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends one or more Traffic Light Status messages to the host controller as follows:

- If the Traffic Light ID field is 0, the HLC gets status for all currently defined traffic lights. A Traffic Light Status response message is returned for each traffic light with its success or failure status. If no traffic lights exist, the command is accepted by the HLC and a Traffic Light ID of 0 and a command status of "Command Accepted" is returned in the Traffic Light Status response message.
- If the Traffic Light ID field is nonzero, the HLC gets status only for the specified traffic light. If the specified traffic light does not exist, a Traffic Light Status response message is returned with the failing Traffic Light ID and the command status set to indicate that no such traffic light exists in the transport system.
- If the command is rejected, the ID of the traffic light and the failure status is included in the Traffic Light Status response message.

See Also

Create Traffic Light on page 193
Set Traffic Light on page 196
Delete Traffic Light on page 202
Traffic Light Status on page 305

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a Get Traffic Light Status command that is issued for all traffic lights. The HLC responses messages provide the status information for each traffic light, which includes the traffic light ID, position, and color.

```
0xBF 02 03 0000

0xDF 02 01 00 0001 0001 0000C03F 01
0xDF 02 01 00 0002 0001 0000803F 00
```

Delete Traffic Light

Message: 0xBF 0x02 0x04

Source/Destination

Host Controller ➛ HLC

Purpose

Deletes the specified traffic light. When a traffic light is deleted, the associated motor block is set to green to allow vehicles to move beyond the traffic light position.

NOTE: The high-level controller, node controllers, and the path the traffic light resides on must be in the operational state.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	11
3	Message Type	1	0xBF
4	Extension Type	1	0x02
5	Extension Subtype	1	0x04
6	Traffic Light ID	2	1…4096
8	Host Command Count	4	0x0…0xFFFFFFFF
12	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Traffic Light extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Delete Traffic Light extension command.

Traffic Light ID – The ID of the specified traffic light to delete.

Host Command Count – A unique value supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in Command Status messages.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC deletes the specified traffic light.

On command completion, the HLC sends a "Command Complete" or a "Command Failed" response

See Also

Create Traffic Light on page 193
Set Traffic Light on page 196
Get Traffic Light Status on page 199
Command Status on page 261
Traffic Light Status on page 305

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows Traffic Light 1 being deleted. The HLC response shows that the command was accepted and completed successfully.

```
0xBF 02 04 0001 00000016

0xD0 BF 00 02 04 0001 00000016
0xD0 BF 80 02 04 0001 00000016
```

Lock Vehicle

Message: 0xBF 0x03 0x01

Source/Destination

Host Controller ► HLC

Purpose

Used to lock or unlock a vehicle. While a vehicle is locked, any motion commands are rejected. The vehicle lock status is displayed in the Extended Vehicle Status, message.

NOTE: The high-level controller, node controllers, and the path the vehicle resides on must be in the operational state.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	8
3	Message Type	1	0xBF
4	Extension Type	1	0x03
5	Extension Subtype	1	0x01
6	Vehicle ID	2	1…65535
8	Subcommand	1	0x00, 01
9	Message CRC	2	0…65535

Data Fields

Command Header – Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Vehicle extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Vehicle Lock/Unlock command.

Vehicle ID – The ID of the vehicle targeted by this command. The ID must be a nonzero positive integer that references a vehicle that exists in the transport system.

Subcommand – Fixed value that identifies the vehicle subcommand to execute.

Value	Description
0x00	Lock
0x01	Unlock

Response

After receiving the command and verifying the command parameters, the HLC sends a Command Status message with either a “Command Accepted” response (0x00) or a “Command Rejected” response as appropriate (see Table 4-10). If the command is accepted, the HLC routes the command to the motor responsible for the vehicle. If the command is rejected, the Command Status response includes a rejection code that shows the reason for rejection.

See Also

Command Status on page 261
Extended Vehicle Status, on page 307

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a Lock/Unlock Vehicle command that is issued to lock vehicle 1.

```
0xBF 03 01 0001 00

0xD0 BF 00 03 01 0001 00
```

The following example (see Figure 2-1 on page 27 for the system layout) shows a Lock/Unlock Vehicle command that is issued to unlock vehicle 1.

```
0xBF 03 01 0001 01

0xD0 BF 00 03 01 0001 01
```

Vehicle Command

Message: 0xBF 0x03 0x04

Source/Destination

Host Controller ► HLC

Purpose

Used to send vehicle-specific subcommands to the motor controllers, such as clearing the vehicle suspect bit.

NOTE: The high-level controller, node controllers, and the path the vehicle resides on must be in the operational state.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	8
3	Message Type	1	0xBF
4	Extension Type	1	0x03
5	Extension Subtype	1	0x04
6	Vehicle ID	2	1…65535
8	Subcommand	1	0x00
9	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Vehicle extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Vehicle Command extension command.

Vehicle ID – The ID of the vehicle targeted by this command. The ID must be a nonzero positive integer that references a vehicle that exists in the transport system.

Subcommand – Fixed value that identifies the vehicle subcommand to execute.

Value	Description
0x00	**Clear Vehicle Suspect Bit** Commands the master controller responsible for the specified vehicle to clear the suspect bit in the vehicle record for the vehicle.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC handles locking/unlocking the specified vehicle.

On command completion, the HLC sends a "Command Complete" or a "Command Failed" response.

See Also

Command Status on page 261

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows a Vehicle Command that is issued to clear the suspect bit for vehicle 1 and the response showing that the command completed successfully.

```
0xBF 03 04 0001 00

0xD0 BF 00 03 04 0001 00
```

Get Extended Vehicle Status

Message: 0xBF 0x03 0x07 or 0xBF 0x03 0xFF

Source/Destination

Host Controller ► HLC

Purpose

Requests the extended vehicle status for the specified vehicle. The extended vehicle status returns all status information that the Vehicle Status message returns and includes additional status information. If the Vehicle ID specified is zero, the extended vehicle status for all vehicles is returned.

NOTE: The high-level controller, node controllers, and paths must be in the operational state.

When a status request is received for all vehicles, a separate status response is sent for each vehicle.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBF
4	Extension Type	1	0x03
5	Extension Subtype	1	0x07, 0xFF
6	Vehicle ID	2	0…65535
8	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Vehicle extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Get Extended Vehicle Status extension command

The extension subtype also selects the version of the extended status to return for the specified vehicle.

Value	Description
0x02	Get version 1 of the extended status for the specified vehicle(s).
0x03	Get version 2 of the extended status for the specified vehicle(s).
0x06	Get version 3 of the extended status for the specified vehicle(s).
0x07	Get version 4 of the extended status for the specified vehicle(s).
0xFF	Get the current version of the extended status for the specified vehicle(s).

Vehicle ID – The ID of the vehicle that status is requested for.

Value	Description
0	Get the extended status for all vehicles.
1…65535	Get extended status for the specified vehicle.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends one or more Extended Vehicle Status, messages to the host controller as follows:

- If the Vehicle ID field is 0, the HLC gets extended vehicle status for all currently defined vehicles. An Extended Vehicle Status,) response message is returned for each vehicle.
- If the Vehicle ID field is nonzero, the HLC gets extended status only for the specified vehicle. If status is requested for a vehicle that does not exist, an Extended Vehicle Status, response message is returned with the Vehicle Present field set to zero to indicate that no such vehicle exists.

See Also

Status Request on page 152
Vehicle Status on page 284
Extended Vehicle Status, on page 307

Examples

The following example (see Figure 2-1 on page 27 for the system layout) shows an Extended Vehicle Status Request command that is issued for vehicle 1. The HLC response provides the status information for the vehicle, which shows the path it is on, the current position and motion commands. The response includes the state of all status flags and additional motion information.

```
0xBF 03 07 0001
```

```
0xDF 03 07 0001 01 0003 0003 D2FFFF3E 00000000 00 0001
00000000 D2FFFF3E 0000 E23E323D 0000 00 00 5BE47F3F
A6FDFF3E 00000000
```

The following example (see Figure 2-1 for the system layout) shows an Extended Vehicle Status Request command that is issued for all vehicles. The HLC responses provide the status information for each vehicle, which shows the path it is on and its current position and motion commands. The response includes the state of all status flags and additional motion information. The response for vehicle 3 shows it is following vehicle 4.

```
0xBF 03 07 0000
```

```
0xDF 03 07 0001 01 0003 0003 D2FFFF3E 00000000 00 0001
00000000 D2FFFF3E 0000 E23E323D 0000 00 00 5BE47F3F
A6FDFF3E 00000000
0xDF 03 07 0002 01 0001 0000 FDFFBF3F 00000000 00 0001
00000000 FDFFBF3F 0000 E23E323D 0000 00 00 00000000
00000000 00000000
0xDF 03 07 0003 01 0003 0000 FDFFBF3F 00000000 B7 0601
8399993E FDFFBF3F 0004 E23E323D 0000 00 00 5BE47F3F
00000000 00000000
0xDF 03 07 0004 01 0001 0001 5E66E63F 00000000 00 0001
00000000 5E66E63F 0000 E23E323D 0000 00 00 5BE47F3F
00000000 00000000
```

Get Node Controller Digital I/O Status

Message: 0xBF 0x07 0x01

Source/Destination

Host Controller ► HLC

Purpose

Requests the status of the digital I/O inputs and outputs for the specified node controller.

NOTE: The high-level controller and node controllers must be in the operational state.

The node controller that is specified must support digital I/O (for example, NC-E).

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBF
4	Extension Type	1	0x07
5	Extension Subtype	1	0x01
6	Node Controller ID	2	0…96
8	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Vehicle extension command.

Extension Type – Fixed extension type that identifies this command as a Node Controller Digital I/O extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Get Node Controller Digital I/O Status extension command.

Node Controller ID – The ID of the node controller that digital I/O status is being requested from.

Value	Description
0	Get the digital I/O status for all node controllers.
1…65535	Get the digital I/O status for the specified node controller.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends one or more Node Controller Digital I/O Status messages to the host controller as follows:

- If the Node Controller ID field is 0, the HLC gets digital I/O status for each connected node controller. A Node Controller Digital I/O Status response message is returned for each node controller with its associated node controller ID and digital I/O status.
- If the Node Controller ID field is nonzero, the HLC gets status only for the specified node controller. If the specified node controller does not exist, a Node Controller Digital I/O Status response message is returned with the failing node controller ID and the command status set to indicate that no such node controller exists in the transport system.
- If the command is rejected, the node controller ID and the failure status is included in the Node Controller Digital I/O Status response message.

See Also

Set Node Controller Digital I/O Outputs on page 214
Set Node Controller Digital I/O Notification Mask on page 217
Node Controller Digital I/O Status on page 315

Examples

The following example shows a Get Node Controller Digital I/O Status command that is issued for NC1 (where NC1 is an NC-12 node controller). The HLC response provides the digital I/O status information, which shows that all input and output bits are high and the notification mask specifies all inputs.

```
0xBF 07 01 0001

0xDF 07 01 00 0001 10 10 0000FFFF 0000FFFF 0000FFFF
```

The following example shows a Get Node Controller Digital I/O Status command that is issued for all node controllers (where NC1 is an NC-12 and NC2 is an NC LITE). The HLC responses provide the digital I/O status information for each node controller. Since the NC LITE does not have digital I/O, the number of inputs and outputs is 0 and the remaining data is undetermined.

```
0xBF 07 01 0000

0xDF 07 01 00 0001 10 10 0000FFFF 0000FFFF 0000FFFF
0xDF 07 01 00 0002 00 00 00000000 00000000 00000000
```

Set Node Controller Digital I/O Outputs

Message: 0xBF 0x07 0x02

Source/Destination

Host Controller ► HLC

Purpose

Sets the digital I/O outputs for the specified node controller.

NOTE: The high-level controller and node controllers must be in the operational state.

The node controller that is specified must support digital I/O (for example, NC-E).

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport system.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	15
3	Message Type	1	0xBF
4	Extension Type	1	0x07
5	Extension Subtype	1	0x02
6	Node Controller ID	2	1…96
8	Output Data Mask	4	0x0…0xFFFFFFFF
12	Output Data	4	0x0…0xFFFFFFFF
16	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Node Controller Digital I/O extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Set Node Controller Digital I/O Outputs extension command.

Node Controller ID – The ID of the node controller for which the digital I/O outputs are being set. The ID must be a nonzero positive integer that references a node controller that exists in the configuration.

Output Data Mask – Specifies which bits of the output data field to write into the digital I/O outputs in the node controller. For each bit set to 1 in the output data mask field, the corresponding bit in the **Output Data** field is written into the digital I/O outputs. For each bit set to 0 in this field, the corresponding bit in the digital I/O outputs is not changed.

The **Output Data Mask** field must only specify digital I/O outputs implemented by the specified node controller (see Table 4-2).

Output Data – The digital I/O output data to set, consisting of 1 to 32 individual outputs. For each bit set to 1 in the **Output Data Mask** field, the corresponding bit in this field is written into the digital I/O outputs.

Arrange output data from the low-order bits of the **Output Data** field to the high-order bits. Table 4-2 lists the available digital I/O outputs for each node controller type.

Table 4-2: Node Controller Digital Outputs

Node Controller Type	Available Digital I/O Outputs
NC LITE	none
NC-12	0x0000FFFF
NC-E Series A	0x0000000F
NC-E Series B	0x000000FF
NC-S	none

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller to set its Digital I/O outputs.

On command completion, the HLC sends a "Command Complete" or a "Command Failed" response

See Also

Get Node Controller Digital I/O Status on page 211
Set Node Controller Digital I/O Notification Mask on page 217
Command Status on page 261
Node Controller Digital I/O Status on page 315

Examples

The following example shows a Set Node Controller Digital I/O Outputs command that is issued for NC1 (where NC1 is an NC-12 node controller). The command specifies bits 0…7 to allow change, bits 8…15 to not allow change, bits 16…31 are not used (not supported by NC-12), and bit 1 is being set. The HLC response shows that the command was accepted.

```
0xBF 07 02 0001 000000FF 00000001

0xD0 BF 00 07 02 0001 000000FF 00000001
```

Set Node Controller Digital I/O Notification Mask

Message: 0xBF 0x07 0x03

Source/Destination

Host Controller ► HLC

Purpose

Sets the digital I/O notification mask for the specified node controller. The notification mask selects which digital I/O inputs are monitored for state changes.

NOTE: The high-level controller and node controllers must be in the operational state.

The node controller that is specified must support digital I/O (for example, NC-E).

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport system.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	11
3	Message Type	1	0xBF
4	Extension Type	1	0x07
5	Extension Subtype	1	0x03
6	Node Controller ID	2	1…96
8	Notification Mask	4	0x0…0xFFFFFFFF
12	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Node Controller Digital I/O extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Set Node Controller Digital I/O Notification Mask extension command.

Node Controller ID – The ID of the node controller for which the digital I/O notification mask is being set. The ID must be a nonzero positive integer that references a node controller that exists in the configuration.

Notification Mask – Specifies the digital I/O inputs that the HLC monitors for state changes. For each bit set to 1 in the Notification Mask field, the corresponding bit in the digital I/O inputs in the specified node controller is monitored for state changes.

Whenever any digital I/O input changes state and its corresponding Notification Mask bit is set, the HLC sends a Node Controller Digital I/O Status message to the host controller.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends a “Command Accepted” or “Command Rejected” response as appropriate (see Command Status message). If the command is accepted, it is forwarded to the appropriate node controller for execution.

See Also

Get Node Controller Digital I/O Status on page 211
Set Node Controller Digital I/O Outputs on page 214
Command Status on page 261
Node Controller Digital I/O Status on page 315

Examples

The following example shows a Set Node Controller Digital I/O Notification Mask command that is issued for NC1 (where NC1 is an NC-12 node controller). The command specifies bits 0…3 to be monitored, bits 4…15 not to be monitored, and bits 16…31 are not used (not supported by NC-12). The HLC response shows that the command was accepted.

```
0xBF 07 03 0001 0000000F

0xD0 BF 00 07 03 0001 0000000F
```

MP Get Path End Status

Message: 0xBF 0x09 0x01

Source/Destination

Host Controller ► HLC

Purpose

Request the status of all path ends in a specific Moving Path node or for all Moving Path nodes in a transport system.

NOTE: The high-level controller, node controllers, and the paths that are specified must be in the operational state.

A separate status response is sent for each path end in the specified Moving Path nodes.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBF
4	Extension Type	1	0x09
5	Extension Subtype	1	0x01
6	Node ID	2	0…65535
8	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Node extension command.

Extension Subtype – Fixed extension subtype that identifies this command as an MP Get Path End Status extension command.

Node ID – The ID of the Moving Path node that path end status is being requested for.

Value	Description
0	Get the path end status for all currently defined Moving Path nodes.
1…65535	Get the path end status for the specified Moving Path node.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends one or more MP Path End Status Report messages to the host controller as follows:

- If the Node ID field is 0, the HLC gets path end status for all Moving Path nodes. A MP Path End Status Report message is returned for each path end with its associated node ID and command status of "Command Accepted". If no Moving Path nodes exist, the command is rejected by the HLC and a single MP Path End Status Report message is returned with a Node ID of 0 and a command status of "No Record Available".
- If the Node ID field is nonzero, the HLC gets path end status only for the Moving Path node with a matching Node ID. If the specified Moving Path node does not exist, a single MP Path End Status Report message is returned with the failing Node ID and a command status of "No Record Available".
- If the command is rejected, the Node ID and the failure status is included in the MP Path End Status Report message.

See Also

MP Path End Status Report on page 317

Examples

The following example (see Figure 3-45 on page 124 for the system layout) shows an MP Get Path End Status command that is issued for the Moving Path node that is node 3.

```
0xBF 09 01 0003
```

A MP Path End Status Report is returned for each path in the node. The responses show that the command completed successfully and provide the status for each path in the node. Path 1 is unlinked and has vehicle 2 present.

```
DF 09 01 00 0003 0001 00000000 07 01 01 0000 0000 0000
0000 0000 0002 0002 0000 0000
```

Path 2 is the Control Path that is linked to path 4, the link automatically unlinks after vehicle 8 goes through, has vehicle 1 present, and node 4 is a peer node.

```
DF 09 01 00 0003 0002 00000001 04 02 01 0000 0000 0004
0008 0000 0001 0001 0000 0000
```

Path 3 is unlinked.

```
DF 09 01 00 0003 0003 00000000 01 01 01 0000 0000 0000
0000 0000 0000 0000 0000 0000
```

Path 4 is the Peer Path that is linked to path 2.

```
DF 09 01 00 0003 0004 00000000 06 03 01 0000 0000 0002
0000 0000 0000 0000 0000 0000
```

MP Link Command

Message: 0xBF 0x09 0x02

Source/Destination

Host Controller ► HLC

Purpose

The host controller sends an MP Link Command to the HLC when the Control and Peer Paths are physically aligned. This message notifies the HLC that the path ends are aligned and can be linked to allow vehicles to navigate the path junction.

The initial MP Link Command, sent to form a junction between two unlinked path ends, establishes the Control Path and Peer Path. Subsequent link commands can be issued to modify the Last Allowed Vehicle ID but must specify the same Control and Peer Path IDs used initially to establish the junction.

This command can be sent asynchronously or in response to a previous MP Alignment Request message.

NOTE: The high-level controller, node controllers, and the paths that are specified must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	19
3	Message Type	1	0xBF
4	Extension Type	1	0x09
5	Extension Subtype	1	0x02
6	Node ID	2	1…65535
8	Control Path ID	2	1…65535
10	Peer Path ID	2	1…65535
12	Last Allowed Vehicle ID	2	0…65535
14	Alignment Request Count	2	1…65535
16	Host Command Count	4	0x0…0xFFFFFFFF
20	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Node extension command.

Extension Subtype – Fixed extension subtype that identifies this command as an MP Link Command extension.

Node ID – The ID of the Moving Path node that this link command operates on. The ID must reference a node that exists in the transport system.

Control Path ID – The ID of the Control Path end to link to the specified Peer Path end to form a path junction. The ID must reference a path that exists in the transport system.

Peer Path ID – The ID of the Peer Path end to link to the specified Control Path end to form a path junction. The ID must reference a path that exists in the transport system.

NOTE: New path junctions transition to the Linked state.

Last Allowed Vehicle ID – Used to set the conditions for unlinking the path junction.

Value	Description
0	The HLC keeps the Control and Peer Paths linked until the host controller sends an MP Unlink Command. Vehicles approaching the junction are granted permission to navigate the junction if the Control Path offers an equivalent route to the vehicle's destination. If the approaching vehicle requires another path alignment, the junction is unlinked as soon as there are no vehicles navigating the node and a new alignment request is sent to the host controller.
1…65535	Once the last allowed vehicle begins to navigate the path junction, the Control Path end transitions to the Linked Unlink Pending state and no other vehicles are granted permission to navigate the junction. The HLC unlinks the path junction as soon as all navigating vehicles are clear of the path junction.

Alignment Request Count – A sequence count, unique to the most recent alignment request for a path end, which the HLC increments whenever alignment for a path end is requested using an MP Alignment Request.

The host controller can use this field to track which alignment request a link command is responding to by including the *Alignment Request Count* from an MP Path End Status Report in the link command.

Host Command Count – A sequence count, unique to the specified Control Path end, which the host controller increments when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in MP Path End Status Report messages.

The host controller maintains the Host Command Count. It is initialized to 1 when the host controller restarts, it is then incremented for each new link/unlink command issued, and continues from 1 when it rolls over.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC handles linking the specified path junctions for the selected Moving Path node.

On completion of the command, the HLC sends a "Command Complete" or "Command Failed" response.

If the HLC is configured to report path end status changes asynchronously, the HLC responds with one or more MP Path End Status Report messages reflecting changes resulting from executing this link command.

See Also

MP Get Path End Status on page 219
MP Unlink Command on page 226
Command Status on page 261
MP Path End Status Report on page 317

Examples

The following example (see Figure 3-44 on page 123 for the system layout) shows an MP Link Command that is issued for the Moving Path node that is Node 3. The command links path 2 (control path) and path 4 (peer path) in node 3 with vehicle ID 8 set as the last vehicle. The HLC responses show that the command was accepted and that it completed successfully.

```
0xBF 09 02 0003 0002 0004 0008 0000 00000001

0xD0 BF 00 09 02 0003 0002 0004 0008 0000 00000001
0xD0 BF 80 09 02 0003 0002 0004 0008 0000 00000001
```

MP Unlink Command

Message: 0xBF 0x09 0x03

Source/Destination

Host Controller ► HLC

Purpose

The host controller sends an MP Unlink command to the HLC to unlink one or more path junctions that are associated with the specified Moving Path node.

NOTE: The high-level controller, node controllers, and the paths that are specified must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	13
3	Message Type	1	0xBF
4	Extension Type	1	0x09
5	Extension Subtype	1	0x03
6	Node ID	2	1…65535
8	Control Path ID	2	0…65535
10	Host Command Count	4	0x0…0xFFFFFFFF
14	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Node extension command.

Extension Subtype – Fixed extension subtype that identifies this command as an MP Unlink Command extension.

Node ID – The ID of the Moving Path node that this unlink command operates on. The ID must reference a node that exists in the transport system.

Control Path ID – The ID of the path ends to unlink from path junctions that are associated with the specified Moving Path node. The ID must reference a path that exists in the transport system.

Value	Description
0	All member path ends are unlinked for the specified Moving Path node.
1…65535	Only the specified Control Path end is unlinked. Additionally, the Control Path ID specified must be the same Control Path that was used initially to establish the junction.

If a vehicle is navigating a path junction, the path junction transitions to the Linked Unlink Pending state; otherwise, the path junction transitions to the Unlinked state.

See Table 4-17, *Moving Path Node Path End States, on page 319* for detailed descriptions of the path end states.

Host Command Count – A sequence count, unique to the specified Control Path end, which the host controller increments when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in MP Path End Status Report messages.

The host controller maintains the Host Command Count. It is initialized to 1 when the host controller restarts, it is then incremented for each new link/unlink command that is issued, and continues from 1 when it rolls over.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends a "Command Accepted" or "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC handles unlinking the specified path junctions for the selected Moving Path node.

On completion of the command, the HLC sends a "Command Complete" (path junctions transitions to the Linked Unlinked pending state) or "Command Failed" response.

If the HLC is configured to report path end status changes asynchronously, the HLC responds with one or more MP Path End Status Report messages reflecting changes resulting from executing this unlink command.

See Also

MP Get Path End Status on page 219
MP Link Command on page 222
Command Status on page 261
MP Path End Status Report on page 317

Examples

The following example (see Figure 3-44 on page 123 for the system layout) shows an MP Unlink Command that is issued for path 2 (control path) in the Moving Path node that is Node 3. The HLC responses show that the command was accepted and that it completed successfully.

```
0xBF 09 03 0003 0002 00000002

D0 BF 00 09 03 0003 0002 00000002
D0 BF 80 09 03 0003 0002 00000002
```

SM Subscription Command

Message: 0xBF 0x0A 0x01

Source/Destination

Host Controller ➛ HLC

Purpose

Provides monitoring of system metrics. The host controller sends a “subscribe” SM Subscription command message to the HLC to register for metric data updates on the specified component type. Once registered, metric updates are reported continuously to the host controller at the specified interval until stopped with an “unsubscribe” SM Subscription Command.

NOTE: The high-level controller, node controllers, and the components that are specified must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	23
3	Message Type	1	0xBF
4	Extension Type	1	0x0A
5	Extension Subtype	1	0x01
6	Component Type	8	Varies with each type
14	Host Context	2	0x0000…0xFFFF
16	Command Options	2	0x0000, 0x0001
18	Subscription Interval	2	0…65535
20	Metric ID	2	0x0000…0xFFFF
22	Metric Instance	2	0…65535
24	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as an Info Serv extension command.

Extension Subtype – Fixed extension subtype that identifies this command as an SM Subscription extension command.

Component Type – The ID of the component targeted for system monitoring.

Table 4-3: System Monitoring Component Types

Value	Description
1	See HLC Component Type
2	See Node Controller Component Type
3	See Node Component Type
4	See Path Component Type
5	See Vehicle Component Type
6	See Station Component Type
7	See Motor Component Type

HLC Component Type – Component type that selects collecting system metrics that are associated with the high-level controller. Currently, high-level controller component monitoring is not supported.

Offset	Item	Size	Range
0	Component Type	1	1 – High-Level Controller
1	Reserved	7	Must be zero

Node Controller Component Type – Component type that selects collecting system metrics that are associated with a node controller. The ID of the Node Controller targeted for system monitoring or polling. The ID must be a non-zero 16-bit integer that exists in the configuration. Zero is not a valid Node Controller ID for this command. Currently, node controller component monitoring is not supported.

Offset	Item	Size	Range
0	Component Type	1	2 – Node Controller
1	Reserved	1	Must be zero
2	Node Controller ID	2	1…128 – The ID of the node controller that is targeted for system monitoring or polling.
4	Reserved	4	Must be zero

Node Component Type – A fixed component type that selects collecting system metrics that are associated with a node. The ID of the Node Controller targeted for system monitoring or polling. The ID must be a non-zero 16-bit integer that exists in the configuration. Zero is not a valid Node Controller ID for this command. Currently, node component monitoring is not supported.

Offset	Item	Size	Range
0	Component Type	1	3 – Node
1	Reserved	1	Must be zero
2	Node ID	2	1…256 – The ID of the node that is targeted for system monitoring or polling.
4	Reserved	4	Must be zero

Path Component Type – Component type that selects collecting system metrics that are associated with a path. The ID of the Node Controller targeted for system monitoring or polling. The ID must be a non-zero 16-bit integer that exists in the configuration. Zero is not a valid Path ID for this command. Currently, path component monitoring is not supported.

Offset	Item	Size	Range
0	Component Type	1	4 – Path
1	Reserved	1	Must be zero
2	Path ID	2	1…65535 – The ID of the path that is targeted for system monitoring or polling.
4	Reserved	4	Must be zero

Vehicle Component Type – Component type that selects collecting system metrics that are associated with a vehicle. The ID of the Node Controller targeted for system monitoring or polling. The ID must be a non-zero 16-bit integer that exists in the configuration. Zero is not a valid Vehicle ID for this command. Currently, vehicle component monitoring is not supported.

Offset	Item	Size	Range
0	Component Type	1	5 – Vehicle
1	Reserved	1	Must be zero
2	Vehicle ID	2	1…65535 – The ID of the vehicle that is targeted for system monitoring or polling.
4	Reserved	4	Must be zero

Station Component Type – Component type that selects collecting system metrics that are associated with a station. The ID of the station targeted for system monitoring or polling. The ID must be a nonzero 16-bit integer that exists in the configuration. Zero is not a valid Station ID for this command. Currently, station component monitoring is not supported.

Offset	Item	Size	Range
0	Component Type	1	6 – Station
1	Reserved	1	Must be zero
2	Station ID	2	1…2048 – The ID of the station that is targeted for system monitoring or polling.
4	Reserved	4	Must be zero

Motor Component Type – Component type that selects collecting system metrics that are associated with a motor. The ID of the motor targeted for system monitoring or polling. The ID must be a nonzero 16-bit integer that exists in the configuration. Zero is not a valid Motor ID for this command. Currently, slave controller monitoring is not supported and the Slave ID field must be specified as all ones.

Offset	Item	Size	Range
0	Component Type	1	7 – Motor
1	Reserved	1	Must be zero
2	Path ID	2	1…65535 – The ID of the path where the targeted motor is located.
4	Motor ID	2	1…65535 – The ID of the motor that is targeted for system monitoring or polling.
6	Slave ID	2	Must be 0xFFFF – The ID of the slave controller in the motor that is targeted for system monitoring or polling.

Host Context – Unique value supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command.

The Host Context is always returned in SM Subscription Response and SM Subscription Data Response messages.

Command Options – Permits the selection of one or more processing options.

Option Bitmask	Command Option Description
0x0000	None
0x0001	**Clear Metric Data on Sampling Start** – Instructs the component to clear metric data at the beginning of each sampling interval. Ignored by “unsubscribe” SM Subscription commands.

Subscription Interval – Specifies the sampling interval (in seconds) that must elapse before a report of metric data for the specified metric is returned to the host controller. Specify zero to unsubscribe (that is, stop) the current subscription for the specified metric. When nonzero, it specifies a sampling interval for the specified metric from 1 second to over 18 hours (see Table 4-4 for common system monitoring intervals).

Table 4-4: Common System Monitoring Intervals

Time Interval (in seconds)	Time Duration
0	None – Unsubscribe from the specified metric.
1	1 second (minimum interval)
30	30 seconds
60	1 minute
300	5 minutes
900	15 minutes
1800	30 minutes
3600	1 hour
14400	4 hours
28800	8 hours
43200	12 hours
64800	18 hours
65535	18 hours, 12 minutes, 15 seconds (maximum interval)

Metric ID – For “subscribe” commands, the unique nonzero ID of a metric to report to the host controller, see Table 4-5.

For “unsubscribe” commands (**Subscription Interval** = 0), the ID of one or more metrics to stop reporting to the host controller. If the Metric ID field is zero, all subscriptions for the targeted component are unsubscribed. If the Metric ID field is nonzero, only the subscription for the specified metric is unsubscribed.

Table 4-5: System Monitoring, Hardware Metrics

Name	ID	Instance	Syntax*	Units	Conversion	Min	Max	Description
Propulsion Soft Start Input Voltage†	0x1015	0	GaugeU16	Volts	(Value * 128) / 65535	0V	+128V	Propulsion voltage present at the input connector after the input fuse.
Propulsion Soft Start Output Voltage‡	0x1016	0	GaugeU16	Volts	(Value * 128) / 65535	0V	+128V	Propulsion Voltage at input of inverters.
Raw Board Temperature§	0x1020	0	GaugeU16	Celsius	(Value * 120) / 65535	0° C	120 °C	Raw hardware board temperature in degrees C. The value is represented as an unsigned 16-bit number as temperature ranging from 0…120 °C [32…248 °F]
External Board Temperature(†)	0x1021	0	GaugeU16	°C	(Value * 120) / 65535	0° C	120° C	The external hardware board temperature in degrees C. The value is represented as an unsigned 16-bit number as temperature ranging from 0…120° C [32…248° F]

* See Table 4-6
† Supported on QuickStick 100 motors.
‡ Supported on QuickStick 100 and QuickStick 150 (QuickStick) motors.
§ Supported on QuickStick 150 motors.

Table 4-6: System Monitoring Syntax Types

Syntax	Description
Integer16	Single signed 16-bit integer. Values range between -32768 and 32767. Returns only the last updated value when read.
Integer32	Single signed 32-bit integer with values range between -2147483648 and 2147483647. Returns only the last updated value when read.
Unsigned16	Single unsigned 16-bit integer with values ranging between 0 and 65535. Returns only the last updated value when read.
Unsigned32	Single unsigned 32-bit integer with values ranging between 0 and 4294967295. Returns only the last updated value when read.

Table 4-6: System Monitoring Syntax Types (Continued)

Syntax	Description
Counter16	Single unsigned 16-bit integer that monotonically increases until it reaches its maximum value, when it wraps around and starts increasing again from zero. Counters only return the last updated value when read.
Counter32	Single unsigned 32-bit integer that monotonically increases until it reaches its maximum value, when it wraps around and starts increasing again from zero. Counters only return the last updated value when read.
Counter64	Single unsigned 64-bit integer that monotonically increases until it reaches its maximum value, when it wraps around and starts increasing again from zero. Counters only return the last updated value when read.
Gauge16	Four signed 16-bit values that can increase or decrease, which returns the following properties (in order): last value, minimum value, maximum value, and average value. When the internal software updates a Gauge metric, it updates all four properties to maintain statistical consistency. When this pattern is read, all four values are returned.
GaugeU16	Four unsigned 16-bit values that can increase or decrease, which returns the following properties (in order): last value, minimum value, maximum value, and average value. When the internal software updates a Gauge metric, it updates all four properties to maintain statistical consistency. When this pattern is read, all four values are returned.

Metric Instance – Identifies unique instances of a metric.

Value	Description
0	For metrics that have only one instance, this field must always be zero. To unsubscribe from all instances of a metric, this field must be zero.
1…65535	For metrics that have multiple unique instances, this field selects the specific metric instance to report to the host controller (see Table 4-5).

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC responds to a SM Subscription Command as follows:

- If a "subscribe" SM Subscription Command is accepted, the HLC returns a SM Subscription Response message with associated command parameters and success status.

  After processing "subscribe" SM Subscription commands and registering new metric data updates, the HLC forwards SM Subscription Data Response SM Subscription Data Response messages received from the specified component to the host controller at the specified interval.
- If an "unsubscribe" SM Subscription Command is accepted, the HLC terminates one or more metric updates and returns a SM Subscription Response message with associated command parameters and success status.
- If the SM Subscription Command is rejected, the HLC returns a SM Subscription Response message with associated command parameters and failure status.

See Also

SM Poll Command on page 238
SM Subscription Response on page 326
SM Subscription Data Response on page 328

SM Poll Command

Message: 0xBF 0x0A 0x03

Source/Destination

Host Controller ► HLC

Purpose

Provides polling of system metrics. The host controller sends an SM Poll Command to the high-level controller to poll for metric data on the specified component.

NOTE: The high-level controller, node controllers, and the components that are specified must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	23
3	Message Type	1	0xBF
4	Extension Type	1	0x0A
5	Extension Subtype	1	0x03
6	Component Type	8	Varies with each type
14	Host Context	2	0x0000…0xFFFF
16	Command Options	2	0x0000, 0x0001
18	Reserved	2	Must be zero
20	Metric ID	2	0x0001…0xFFFF
22	Metric Instance	2	0…65535
24	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as an Info Serv extension command.

Extension Subtype – Fixed extension subtype that identifies this command as an SM Poll Command extension command.

Component Type – The ID of the component targeted for system monitoring (see Table 4-3 on page 230).

Host Context – Unique value supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command.

The Host Context is always returned in SM Poll Data Response messages.

Command Options – Permits the selection of one or more processing options.

Option Bitmask	Command Option Description
0x0000	None
0x0001	**Clear Metric Data on Sampling Start** – Instructs the component to clear metric data after returning it to the host controller.

Metric ID – The unique nonzero ID of a metric to report to the host controller, see Table 4-5 on page 235.

Metric Instance – Specific metric instance to poll.

Value	Description
0	For metrics that have only a single instance, this field must always be zero.
1…65535	For metrics that have multiple unique instances, this field selects the specific metric instance to report to the host controller (see Table 4-5).

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC responds to a SM Poll Command as follows:

- If the command is accepted, the HLC returns an SM Poll Data Response message with command parameters, the requested metric data, and success status.
- If the command is rejected, the HLC returns an SM Poll Data Response message with command parameters, no metric data, and failure status

See Also

SM Subscription Command on page 229
SM Poll Data Response on page 330

Motor Inverter Command

Message: 0xBF 0x0B 0x01

Source/Destination

Host Controller ► HLC

Purpose

Used to enable/disable one or more inverters associated with the QuickStick HT motor.

Motor Inverter Control provides a mechanism for controlling the individual inverters in an LSM motor to control power to the coils in a motor block. This control can be used to disable the propulsion power to the motor, preventing any commanded motion until the inverters are enabled. Use the Status Request command and the Motor Status response to determine the status of the motor inverters.

- In a 1 m QSHT stator, the upstream block is Block 1 and the downstream block is Block 2.
- In a 0.5 m QSHT stator, there is only one block, which is Block 1.
- When two 0.5 m QSHT stators are configured as a QSHT Dual Half motor, the upstream stator is configured as Block 1 and the downstream stator is configured as Block 2.
- In a 0.5 m double-wide QSHT stator, there is only one block, which is Block 1.

NOTE: The high-level controller, node controllers, path, and the motors must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

These changes are not persistent. If the motor is restarted or rebooted, the inverters default to enabled.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	16
3	Message Type	1	0xBF
4	Extension Type	1	0x0B
5	Extension Subtype	1	0x01
6	Path ID	2	1…65535
8	Motor ID	2	0…22
10	Inverter ID	2	0…2
12	Inverter Control	1	0, 1
13	Host Command Count	4	0x0…0xFFFFFFFF
17	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Motor extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Motor Inverter Command extension command.

Path ID – The ID of the path where the specified motors are located. The ID must be a nonzero positive integer referencing a path that exists in the configuration.

Motor ID – The ID of the motors selected by this command. The ID must reference a motor that exists on the specified path.

Value	Description
0	All motors on the specified path are selected.
1…22	Only the specified motor is selected.

Inverter ID – The ID of the inverters that this command specifies. The ID must reference an inverter that exists in the specified motor.

Value	Description
0	All inverters on the specified motor are selected.
1, 2	Only the specified inverter is selected.

Inverter Control – Controls whether to enable or disable inverter(s) targeted by this command.

Value	Description
0	The specified inverter is enabled.
1	The specified inverter is disabled.

Host Command Count – A unique value that is supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in Command Status messages.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC handles enabling/disabling the inverters on the targeted motors.

On command completion the HLC sends a "Command Complete" or a Command Failed response.

See Also

Command Status on page 261
Motor Status on page 289

Examples

The following example shows a Motor Inverter Command that is issued to disable inverter 1 in motor 1 on path 1. The HLC responses show that the command was accepted and that it completed successfully.

```
0xBF 0B 01 0001 0001 0001 01 000000F5

0xD0 BF 00 0B 01 0001 0001 0001 01 000000F5
0xD0 BF 80 0B 01 0001 0001 0001 01 000000F5
```

The following example shows a Motor Inverter Command that is issued to enable inverter 1 in motor 1 on path 1. The HLC responses show that the command was accepted and that it completed successfully.

```
0xBF 0B 01 0001 0001 0001 00 000000F6

0xD0 BF 00 0B 01 0001 0001 0001 00 000000F6
0xD0 BF 80 0B 01 0001 0001 0001 00 000000F6
```

Set Node Controller Configuration

Message: 0xBF 0x0C 0x01

Source/Destination

Host Controller ► HLC

Purpose

Selects the specified Managed Configuration File as active on all node controllers in a transport system.

Using Managed Node Controller Configuration Files allows the creation of up to 20 custom files with different system behavior (for example different PID settings). Each file is assigned a unique Configuration ID when it is uploaded to the HLC.

NOTE: The high-level controller and node controllers must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	10
3	Message Type	1	0xBF
4	Extension Type	1	0x0C
5	Extension Subtype	1	0x01
6	Configuration ID	1	1…20
7	Host Command Count	4	0x0…0xFFFFFFFF
11	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Remote Management extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Set Node Controller Configuration extension command.

Configuration ID – The ID of the managed Node Controller Configuration File to set as active on all node controllers in a transport system.

Host Command Count – A unique value that is supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in Command Status messages.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a “Command Accepted” or “Command Rejected” response as appropriate (see Command Status message). If the command is accepted, the HLC handles setting the specified Node Controller Configuration File as active on all node controllers in a transport system.

On command completion, the HLC sends a “Command Complete” or a “Command Failed” response.

See Also

Restart Node Controllers on page 247
Get Extended Node Controller Status on page 249
Get Extended High-Level Controller Status on page 252
Command Status on page 261

Examples

The following example shows a Set Node Controller Configuration command that is issued to select Managed Node Controller Configuration File #6. The HLC responses show that the command was accepted and that it completed successfully.

```
0xBF 0C 01 06 0000000A

0xD0 BF 00 0C 01 06 0000000A
0xD0 BF 80 0C 01 06 0000000A
```

Restart Node Controllers

Message: 0xBF 0x0C 0x02

Source/Destination

Host Controller ► HLC

Purpose

Restarts all node controllers in the transport system.

NOTE: The high-level controller and node controllers must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

All paths must be in the Init state (Reset complete, Startup not started).

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	9
3	Message Type	1	0xBF
4	Extension Type	1	0x0C
5	Extension Subtype	1	0x02
6	Host Command Count	4	0x0…0xFFFFFFFF
10	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Remote Management extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Restart Node Controllers extension command.

Host Command Count – A unique value that is supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in Command Status messages.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or a "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC handles restarting all node controllers in the transport system.

On command completion, the HLC sends a "Command Complete" or a "Command Failed" response.

See Also

Set Node Controller Configuration on page 245
Get Extended Node Controller Status on page 249
Get Extended High-Level Controller Status on page 252
Command Status on page 261
Path Status on page 281

Examples

The following example shows a Restart Node Controllers command. The HLC responses show that the command was accepted and that it completed successfully.

```
0xBF 0C 02 00000008

0xD0 BF 00 0C 02 00000008
0xD0 BF 80 0C 02 00000008
```

Get Extended Node Controller Status

Message: 0xBF 0x0C 0x03

Source/Destination

Host Controller ► HLC

Purpose

Requests extended node controller status for a specific node controller or for all node controllers in a transport system.

NOTE: The high-level controller and specified node controllers must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

When status is requested for all node controllers, a separate status response is sent for each node controller.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBF
4	Extension Type	1	0x0C
5	Extension Subtype	1	0x03
6	Node Controller ID	2	0…65535
8	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Remote Management extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Get Extended Node Controller Status extension command.

Node Controller ID – The ID of the node controller that status is being requested for.

Value	Description
0	Gets the status for all configured node controllers.
1…65535	Gets the status for the specified node controller.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends one or more Extended Node Controller Status response messages to the host controller as follows:

- If the node controller ID field is 0, the HLC gets status for all configured node controllers. An Extended Node Controller Status response message is returned for each node controller with its success or failure status.
- If the node controller ID field is nonzero, the HLC gets status only for the node controller with the matching ID. If the specified node controller does not exist, an Extended Node Controller Status response message is returned with the failing node controller ID and the command status set to indicate that no such node controller exists in the transport system.
- If the command is rejected, the ID of the node controller and the failure status is included in the Extended Node Controller Status response message.

See Also

[Status Request on page 152]
[Set Node Controller Configuration on page 245]
[Restart Node Controllers on page 247]
[Get Extended High-Level Controller Status on page 252]
[High-Level Controller Status on page 271]
[Node Controller Status on page 273]
[Extended Node Controller Status on page 332]
[Extended High-Level Controller Status on page 335]

Examples

The following example shows a Get Extended Node Controller Status command that is issued for all node controllers. The HLC responses show that the command was accepted and provides the extended status for all node controllers.

```
0xBF 0C 03 0000

0xDF 0C 03 00 0001 03 0F 0B 0B 00 01
0xDF 0C 03 00 0002 03 0F 0B 0B 00 01
```

Get Extended High-Level Controller Status

Message: 0xBF 0x0C 0x04

Source/Destination

Host Controller ➛ HLC

Purpose

Requests extended status for the high-level controller of the transport system.

NOTE: The high-level controller must be in the operational state.

Support

This command is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xBF
4	Extension Type	1	0x0C
5	Extension Subtype	1	0x04
6	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Remote Management extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Get Extended High-Level Controller Status extension command.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends an Extended High-Level Controller Status response message to the host controller.

See Also

Status Request on page 152
Set Node Controller Configuration on page 245
Restart Node Controllers on page 247
Get Extended Node Controller Status on page 249
High-Level Controller Status on page 271
Node Controller Status on page 273
Extended Node Controller Status on page 332
Extended High-Level Controller Status on page 335

Examples

The following example shows a Get Extended High-Level Controller Status command. The HLC response shows that the command was accepted and provides the extended status for the high-level controller.

```
0xBF 0C 04

0xDF 0C 04 03 0F 0B 0B 00 01 00
```

Get Propulsion Power Status

Message: 0xBF 0x0D 0x01

Source/Destination

Host Controller ➛ HLC

Purpose

Requests the propulsion power supply status for a specific propulsion power supply or for all propulsion power supplies in a transport system.

NOTE: The high-level controller and the propulsion power supply specified must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick HT 5700 transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xBF
4	Extension Type	1	0x0D
5	Extension Subtype	1	0x01
6	Supply ID	2	0…32
8	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Propulsion Power extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Get Propulsion Power Status extension command.

Supply ID – The ID of the power supply that status is being requested for.

Value	Description
0	Gets the status for all currently defined propulsion power supplies.
1…32	Gets the status for the specified propulsion power supply (see Table 4-7 on page 257).

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends one or more Propulsion Power Supply Status response messages to the host controller as follows:

- If the supply ID field is 0, the HLC gets status for all configured propulsion power supplies. An Propulsion Power Supply Status response message is returned for each propulsion power supply with its success or failure status. If no propulsion power supplies exist, the command is accepted by the HLC and a supply ID of 0 and a command status of "Command Accepted" is returned in the Propulsion Power Supply Status response message.
- If the supply ID field is nonzero, the HLC gets status only for the propulsion power supply with the matching supply ID. If the specified power supply does not exist, a Propulsion Power Supply Status response message is returned with the failing supply ID and the command status set to indicate that no such propulsion power supply exists in the transport system.
- If the command is rejected, the ID of the power supply and the failure status is included in the Propulsion Power Supply Status response message.

See Also

Set Propulsion Power State on page 256
Propulsion Power Supply Status on page 338

Examples

The following example shows a Get Propulsion Power Status command that is issued for all power supplies. The HLC response shows that there is one power supply, it is a diode front end (DFE) type, and it is in the precharge state.

```
0xBF 0D 01 0000

0xDF 0D 01 00 0001 01 01
```

Set Propulsion Power State

Message: 0xBF 0x0D 0x02

Source/Destination

Host Controller ► HLC

Purpose

Processes state changes in the propulsion power supply. Each propulsion power supply is affiliated with one or more QuickStick HT 5700 motors. Whenever a state change occurs in a propulsion power supply, the new state is propagated to its affiliated motor.

NOTE: This feature is only available on QSHT transport systems using QuickStick HT 5700 motor controllers with Kinetix 2198-P*xxx* DC-bus power supplies.

The high-level controller and the propulsion power supplies specified must be in the operational state. If any component required to execute the command is not in the operational state, the command is rejected.

Support

This command is supported in the latest software release for the following product lines:

- QuickStick HT 5700 transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	13
3	Message Type	1	0xBF
4	Extension Type	1	0x0D
5	Extension Subtype	1	0x02
6	Supply ID	2	1…32
8	Type	1	1
9	State	1	0…255
10	Host Command Count	4	0x0…0xFFFFFFFF
14	Message CRC	2	0…65535

Data Fields

Message Header - Constant byte pattern designating the start of a message.

Message Length - Length of this message, given as the number of bytes following the Message Length field.

Message Type - Fixed message type that identifies this message as a Host Extension command.

Extension Type – Fixed extension type that identifies this command as a Propulsion Power extension command.

Extension Subtype – Fixed extension subtype that identifies this command as a Set Propulsion Power State extension command.

Supply ID – The ID of the propulsion power supply undergoing a state change.

Type – The type of the specified propulsion power supply.

Value	Description
1	Diode Front End (DFE) propulsion power supply

State – The new state for the specified propulsion power supply.

Table 4-7: QSHT 5700 Propulsion Power Supply State

Value	State	Description
0x00	Unconnected	The controller is trying to establish communication with the EtherNet/IP controller.
0x01	Precharge	The controller is ready for mains input power.
0x02	Stopped	The controller has DC bus ready, but the control loops are not enabled.
0x03	Starting	The controller is enabled and checking various conditions before entering the RUNNING or TESTING state. For example, the controller checks the Brake Release delay time during the STARTING state.
0x04	Running	• The controller is enabled, configured with an active control mode, and actively tracking a command. • The controller is configured for No Control and is fully operational.
0x05	Testing	The controller is actively executing a test procedure, for example, a hookup test.

Table 4-7: QSHT 5700 Propulsion Power Supply State

Value	State	Description
0x06	Stopping	The controller is decelerating to a stop as the result of a disable.
0x07	Aborting	The controller is decelerating to a stop as the result of a fault or an abort request.
0x08	Faulted	The controller is faulted due to an existing or past fault condition.
0x09	Start Inhibited	The controller has an active condition that inhibits it from being enabled.
0x0A	Shutdown	The controller has been shut down.
0x0B	Axis Inhibited	The axis is inhibited. If this is the only instance that is supported by the CIP Motion connection, the connection is closed
0x0C	Not Grouped	A CIP Motion axis instance exists but is not associated with a Motion Group.
0x0D	No Device	A CIP Motion axis instance exists but is not associated with a CIP Motion device.
0x0E	Configuring	The controller is receiving configuration information from the controller.
0x0F	Synchronizing	The controller is waiting for a successful Group Sync service.
0x10	Waiting for Group	There are other axes in the Motion Group that are still being configured or synchronized.
0x11…0xFF	Reserved	Reserved

Host Command Count – A unique value supplied by the host controller when this message is sent. It aids the host controller in tracking the execution of this command. It is always returned in Command Status messages.

Message CRC - A check value on the entire message (excluding the CRC) to verify integrity.

Response

After receiving the command and verifying the command parameters, the HLC sends either a "Command Accepted" or "Command Rejected" response as appropriate (see Command Status message). If the command is accepted, the HLC handles propagating the new state to the motors affiliated with the specified propulsion power supply.

On command completion, the HLC sends a "Command Complete" or a "Command Failed" response.

See Also

Get Propulsion Power Status on page 254
Command Status on page 261
Propulsion Power Supply Status on page 338

Examples

The following example shows a Set Propulsion Power State command that is issued for power supply 1 to set the state to running. The HLC responses show that the command was accepted and that it completed successfully.

```
0xBF 0D 02 0001 01 04 00000001

0xD0 BF 00 0D 02 0001 01 04 00000001
0xD0 BF 80 0D 02 0001 01 04 00000001
```

HLC to Host Controller Communications

The high-level controller responds to commands from the host controller by issuing the following responses (shown in Table 4-8) to the host controller.

Table 4-8: HLC to Host Controller Status Responses

Description	Message Type	Extension Type / Subtype	Page
Command Status	0xD0	—	261
High-Level Controller Status	0xD1	—	271
Node Controller Status	0xD2	—	273
Node Status	0xD3	—	275
Path Status	0xD4	—	281
Vehicle Status	0xD5	—	284
Motor Status	0xD7	—	289
Station Status	0xDF	0x01 / 0x/01	303
Traffic Light Status	0xDF	0x02 / 0x/01	305
Extended Vehicle Status,	0xDF	0x03 / 0x/07	307
Node Controller Digital I/O Status	0xDF	0x07 / 0x/01	315
MP Path End Status Report	0xDF	0x09 / 0x/01	317
MP Alignment Request	0xDF	0x09 / 0x/02	324
SM Subscription Response	0xDF	0x0A / 0x/01	326
SM Subscription Data Response	0xDF	0x0A / 0x/02	328
SM Poll Data Response	0xDF	0x0A / 0x/03	330
Extended Node Controller Status	0xDF	0x0C / 0x/03	332
Extended High-Level Controller Status	0xDF	0x0C / 0x/04	335
Propulsion Power Supply Status	0xDF	0x0D / 0x/01	338

NOTE: Additional fields may be added to these responses as additional functionality is added to the command set. When parsing any message from the HLC, use the message length that is provided in the message header (see Command Message Framing on page 399) to determine the location of the CRC.

Command Status

Message: 0xD0

Source/Destination

HLC ➤ Host Controller

Purpose

Acknowledges the reception or rejection of a command, signals command execution failure, or signals command completion to the host controller.

NOTE: This response is sent as an asynchronous message with the appropriate command data when certain commands complete or fail.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	Varies
3	Message Type	1	0xD0
4	Command	1	0xB0…0xB4, 0xB6…0xBD, 0xBF
5	Command Status	1	0x00…0x1E, 0x20…0x29, 0x30, 0x40…0x46, 0x80…0x83
6	Command Data Detail	Varies	Varies
Varies	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a high-level controller Status response.

Command – Fixed codes identifying the command that this message acknowledges as described in Table 4-9.

Table 4-9: Command Message Codes

Cmd	Ext	Sub	Command	Valid Command Status Values
0xB0			Move Vehicle To Station	0x00, 0x01, 0x02, 0x04, 0x05, 0x06, 0x07, 0x0A, 0x0B, 0x0C, 0x0D, 0x10, 0x13, 0x41, 0x80
0xB1			Move Vehicle To Position	0x00, 0x01, 0x03,0x04, 0x05, 0x06, 0x07, 0x0A, 0x0B, 0x0C, 0x0D, 0x10, 0x13, 0x41, 0x80
0xB2			Startup	0x00, 0x03, 0x05, 0x06, 0x08, 0x09, 0x0B, 0x0C, 0x0D, 0x24, 0x41, 0x42, 0x44, 0x80
0xB3			Resume Motion	0x00, 0x03, 0x05, 0x06, 0x0A, 0x0B, 0x0C, 0x0D, 0x10, 0x41
0xB4			Suspend Motion	0x00, 0x03, 0x05, 0x06, 0x0B, 0x0C, 0x0D, 0x10, 0x41
0xB6			Set Signal	0x00, 0x01, 0x05, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10, 0x41
0xB7			Vehicle Follow Order	0x00, 0x01, 0x04, 0x0A, 0x0B, 0x0C, 0x0D, 0x10, 0x13, 0x1D, 0x1E, 0x41, 0x81, 0x82, 0x83
0xB8			Reset	0x00, 0x03, 0x0C, 0x0D, 0x10, 0x41, 0x80
0xB9			Delete Vehicle	0x00, 0x01, 0x0C, 0x41, 0x80
0xBA			Set Control Loop Parameters	0x00, 0x03, 0x41
0xBB			Set Node Parameters	0x00, 0x0B, 0x41
0xBC			FastStop Motion	0x00, 0x03, 0x05, 0x06, 0x0B, 0x0C, 0x0D, 0x10, 0x41
0xBD			Warm Reset	0x00, 0x03, 0x0C, 0x0D, 0x10, 0x41, 0x80
0xBF	0x01	0x01	Create Station	0x00, 0x02, 0x03, 0x04, 0x0C, 0x0D, 0x0E, 0x15
0xBF	0x01	0x02	Delete Station	0x00, 0x02, 0x0C, 0x0E, 0x15

Table 4-9: Command Message Codes (Continued)

Cmd	Ext	Sub	Command	Valid Command Status Values
0xBF	0x02	0x01	Create Traffic Light	0x00, 0x03, 0x04, 0x09, 0x0A, 0x0C, 0x0D, 0x0E, 0x10, 0x14, 0x41, 0x42, 0x80
0xBF	0x02	0x02	Set Traffic Light	0x00, 0x0B, 0x0C, 0x0E, 0x11, 0x41, 0x42, 0x80
0xBF	0x02	0x04	Delete Traffic Light	0x00, 0x0C, 0x0E, 0x11, 0x41, 0x42, 0x80
0xBF	0x03	0x01	Lock/Unlock Vehicle	0x00, 0x01, 0x05, 0x0C, 0x0D
0xBF	0x03	0x04	Vehicle Command	0x00, 0x01, 0x05, 0x0C, 0x0D, 0x12, 0x41, 0x42, 0x80
0xBF	0x07	0x02	Set NC Digital I/O Outputs	0x00, 0x0B, 0x0C, 0x25, 0x41
0xBF	0x07	0x03	Set NC Digital I/O Notification Mask	0x00, 0x0B, 0x0C, 0x25, 0x41
0xBF	0x09	0x02	MP Link Command	0x00, 0x03, 0x09, 0x0C, 0x0D, 0x10, 0x19, 0x20, 0x41, 0x42, 0x80
0xBF	0x09	0x03	MP Unlink Command	0x00, 0x03, 0x0C, 0x19, 0x20, 0x41, 0x42, 0x80
0xBF	0x0B	0x01	Motor Inverter Command	0x00, 0x03, 0x09, 0x0B, 0x0C, 0x0D, 0x0E, 0x10, 0x16, 0x1B, 0x23, 0x41, 0x42, 0x80
0xBF	0x0C	0x01	Set Node Controller Configuration	0x00, 0x0B, 0x0C, 0x0E, 0x41, 0x80
0xBF	0x0C	0x02	Restart Node Controllers	0x00, 0x0C, 0x0E, 0x41, 0x80
0xBF	0x0D	0x02	Set Propulsion Power Supply State	0x00, 0x0B, 0x0C, 0x0E, 0x1C, 0x41, 0x42, 0x80

Command Status – The status of the command that this message acknowledges as described in Table 4-10.

NOTE: Some of these status codes are only returned by specific builds of the Node Controller Image File.

Table 4-10: HLC Command Status Codes

Value	Status Description	Possible Cause
0x00	Command Accepted	—
0x01	Command Rejected – Invalid vehicle ID	Specified vehicle does not exist.
0x02	Command Rejected – Invalid station ID	Specified station does not exist.

Table 4-10: HLC Command Status Codes (Continued)

Value	Status Description	Possible Cause
0x03	Command Rejected – Invalid path ID	Specified path does not exist.
0x04	Command Rejected – Invalid position	Commanded position is off the path that is specified.
0x05	Command Rejected – E-stop signal active	E-stop is on. E-stop circuit is not powered.
0x06	Command Rejected – Interlock signal active	Interlock is on. Interlock circuit is not powered.
0x07	Command Rejected – Motion suspended, Stop active	Path is in Suspended state.
0x08	Command Rejected – Startup sequence already completed	Path has already been started.
0x09	Command Rejected – Startup sequence already started	Path is in process of starting up.
0x0A	Command Rejected – Startup sequence not initiated/not completed	Attempted to move a vehicle before startup completed.
0x0B	Command Rejected – Invalid parameter	Incorrectly formatted command (acceleration, velocity, direction, node, set signal, signal level, and so on) not within the correct range.
0x0C	Command Rejected – Initialization has not completed	Sent a command while the system is initializing.
0x0D	Command Rejected – Reset active	Sent a path command while path is resetting.
0x0E	Command Rejected – No record available	Attempted to access an item (for example, station or vehicle) that does not exist.
0x0F	Command Rejected – Terminus node busy	A vehicle is already entering or exiting the Terminus Node.
0x10	Command Rejected – Programming active	Sent command while programming motors.
0x11	Command Rejected – Invalid traffic light index	—
0x12	Command Rejected – Unrecognized command	Command was not formatted correctly.
0x13	Command Rejected – Vehicle lock active	Attempted to move a locked vehicle.
0x14	Command Rejected – Duplicate record	—

Table 4-10: HLC Command Status Codes (Continued)

Value	Status Description	Possible Cause
0x15	Command Rejected – Station in use	—
0x16	Command Rejected – Invalid motor index	Sent a command to a motor that does not exist.
0x17	Command Rejected – Motor busy	—
0x18	Command Rejected – Invalid coil board index	—
0x19	Command Rejected – Invalid node ID	Sent a command to a node that does not exist.
0x1A	Command Rejected – Invalid controller type	—
0x1B	Command Rejected – Invalid slave index	—
0x1C	Command Rejected – Resource busy	—
0x1D	Command Rejected – Vehicle in motion	Sent a command to a vehicle that requires the vehicle to be stopped.
0x1E	Command Rejected – Decouple not complete	A platoon follower has started to decouple, but auto-decouple hasn’t completed.
0x20	Command Rejected – Invalid node type	Sent a command to a node, but specified the wrong node type.
0x21	Command Rejected – Invalid item type	—
0x22	Command Rejected – Invalid item index	—
0x23	Command Rejected – Invalid motor type	—
0x24	Command Rejected – FastStop active	—
0x25	Command Rejected – Invalid node controller ID	Sent a command to a node controller that does not exist.
0x26	Command Rejected – Invalid System Monitoring metric index	—
0x27	Command Rejected – Item in use	—
0x28	Command Rejected – No such item	—
0x29	Command Rejected – Scrubbing not complete	—
0x30	Command Rejected – Gateway entry incomplete	—
0x40	Command Failed – Unable to acquire status from motor	Communication to motor disconnected. Logic power is off.

Table 4-10: HLC Command Status Codes (Continued)

Value	Status Description	Possible Cause
0x41	Command Failed – Unable to complete	Vehicle commanded, but no route to destination available (consult NC/HLC logs for specifics). Path is in incorrect state.
0x42	Command Failed – Timed out	Completion response not received in time. • If executing a Reset command, a motor failed to reset. • If executing a vehicle subcommand, the command completed but the suspect bit did not clear.
0x43	Command Failed – Soft Start active	—
0x44	Command Failed – FastStop active	—
0x45	Command Failed – Downstream neighbor not in startup	—
0x46	Command Failed – Upstream neighbor not in startup	—
0x80	Command Status – Command completed successfully	—
0x81	Command Status – Vehicle has caught up to the vehicle in a platoon it is following	—
0x82	Command Status – Vehicle in a platoon is in the process of decoupling from the vehicle that it is following	Auto-decouple start
0x83	Command Status – Vehicle decoupling from a platoon has arrived at its destination before completing the decouple process	Arrival position tolerance permits the decoupling vehicle to be identified as having arrived at its destination in the following cases: • Waiting for decoupling to start. • Waiting for decoupling to complete.

Command Data Detail – Variable amount of data that provides the details of the command that is specified in the **Command** field as described in Table 4-11.

Table 4-11: Command Data Detail

Cmd	Ext	Sub	Command Description	Command Data
0xB0			Move Vehicle To Station	Order Number (4 bytes) Vehicle ID (2 bytes) Station ID (2 bytes) Path ID (2 bytes) Acceleration Limit* (4 bytes) Velocity Limit(*) (4 bytes) Flags and Direction (1 byte) Offset (4 bytes, if applicable)
0xB1			Move Vehicle To Position	Order Number (4 bytes) Vehicle ID (2 bytes) Position(*) (4 bytes) Path ID (2 bytes) Acceleration Limit(*) (4 bytes) Velocity Limit(*) (4 bytes) Flags and Direction (1 byte)
0xB2			Startup	Path ID (2 bytes)
0xB3			Resume Motion	Path ID (2 bytes)
0xB4			Suspend Motion	Path ID (2 bytes)
0xB6			Set Signal	Node ID (2 bytes) Signal Number (1 byte) Signal Level (1 byte) Vehicle ID (2 bytes, if applicable)
0xB7			Vehicle Follow Order	Order Number (4 bytes) Vehicle ID (2 bytes) Direction (1 byte) Follow Distance (4 bytes) Followed Vehicle ID (2 bytes) PID Set Index (1 byte) Catchup or Decouple Acceleration (4 bytes) Catchup or Decouple Velocity (4 bytes) Destination Path ID (2 bytes) Destination Position (4 bytes)
0xB8			Reset	Path ID (2 bytes)

Table 4-11: Command Data Detail (Continued)

Cmd	Ext	Sub	Command Description	Command Data
0xB9			Delete Vehicle	Vehicle ID (2 bytes)
0xBA			Set Control Loop Parameters	Path ID (2 bytes)
0xBB			Set Node Parameters	Node ID (2 bytes)
0xBC			FastStop Motion	Path ID (2 bytes)
0xBD			Warm Reset	Path ID (2 bytes)
0xBF	01	01	Create Station	Extension Type (1 byte) Extension Subtype (1 byte) Path ID (2 bytes) Position (4 bytes) Station ID (2 bytes)
0xBF	01	02	Delete Station	Extension Type (1 byte) Extension Subtype (1 byte) Station ID (2 bytes)
0xBF	02	01	Create Traffic Light	Extension Type (1 byte) Extension Subtype (1 byte) Path ID (2 bytes) Position (4 bytes) Host Command Count (4 bytes) Traffic Light ID (2 bytes)
0xBF	02	02	Set Traffic Light	Extension Type (1 byte) Extension Subtype (1 byte) Traffic Light ID (2 bytes) Color (1 byte) Host Command Count (4 bytes)
0xBF	02	04	Delete Traffic Light	Extension Type (1 byte) Extension Subtype (1 byte) Traffic Light ID (2 bytes) Host Command Count (4 bytes)
0xBF	03	01	Lock/Unlock Vehicle	Extension Type (1 byte) Extension Subtype (1 byte) Vehicle ID (2 bytes) Lock Request (1 byte)

Table 4-11: Command Data Detail (Continued)

Cmd	Ext	Sub	Command Description	Command Data
0xBF	03	04	Vehicle Command	Extension Type (1 byte) Extension Subtype (1 byte) Vehicle ID (2 bytes) Subcommand (1 byte)
0xBF	07	02	Set NC Digital I/O Outputs	Extension Type (1 byte) Extension Subtype (1 byte) Node Controller ID (2 bytes) Output Data Mask (4 bytes) Output Data (4 bytes)
0xBF	07	03	Set NC Digital I/O Notification Mask	Extension Type (1 byte) Extension Subtype (1 byte) Node Controller ID (2 bytes) Notification Mask (4 bytes)
0xBF	09	02	MP Link Command	Extension Type (1 byte) Extension Subtype (1 byte) Node ID (2 bytes) Control Path ID (2 bytes) Peer Path ID (2 bytes) Last Allowed Vehicle ID (2 bytes) Alignment Request Count (2 bytes) Host Command Count (4 bytes)
0xBF	09	03	MP Unlink Command	Extension Type (1 byte) Extension Subtype (1 byte) Node ID (2 bytes) Control Path ID (2 bytes) Host Command Count (4 bytes)
0xBF	0B	01	Motor Inverter Command	Extension Type (1 byte) Extension Subtype (1 byte) Path ID (2 bytes) Motor ID (2 bytes) Inverter ID (2 bytes) Inverter Control (1 byte) Host Command Count (4 bytes)

Table 4-11: Command Data Detail (Continued)

Cmd	Ext	Sub	Command Description	Command Data
0xBF	0C	01	Set Node Controller Configuration	Extension Type (1 byte) Extension Subtype (1 byte) Configuration ID (1 byte) Host Command Count (4 bytes)
0xBF	0C	02	Restart Node Controllers	Extension Type (1 byte) Extension Subtype (1 byte) Host Command Count (4 bytes)
0xBF	0D	02	Set Propulsion Power State	Extension Type (1 byte) Extension Subtype (1 byte) Supply ID (2 bytes) Type (1 byte) State (1 byte) Host Command Count (4 bytes)

* For a Command Accepted or a Command Rejected response, the value that is returned is the value from the host controller. For a Command Completed response, the value that is returned is a floating-point value that is converted from the fixed-point value from the motor.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

Examples

For examples, see the specific commands.

High-Level Controller Status

Message: 0xD1

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the high-level controller that is specified in a Status Request command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	5
3	Message Type	1	0xD1
4	High-Level Controller State	1	0x01…0x03
5	EtherNet/IP Communication Status	1	0x00…0x02
6	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a high-level controller Status response.

High-Level Controller State – This field indicates the state of the high-level controller.

Value	Description
0x01	**Initialization** – Loading the Node Controller Configuration File or an error was detected in the configuration preventing the node controller from exiting this state. Consult the high-level controller log for additional details when the HLC does not exit this state.
0x02	**Degraded** – The high-level controller is unable to communicate with one or more node controllers in the transport system. See the high-level controller log for additional details.
0x03	**Operational** – The high-level controller configuration is valid and successfully communicating with all node controllers that are configured in the transport system.

EtherNet/IP Communication Status – This field shows the status of communication via Ethernet Industrial Protocol (EtherNet/IP) with a PLC if the high-level controller is configured for communication with a PLC using EtherNet/IP.

Value	Description
0x00	**Not configured** – EtherNet/IP control is not configured.
0x01	**Link up** – EtherNet/IP communication link is UP, which indicates the high-level controller is connected to the PLC at the configured IP address.
0x02	**Link down** – EtherNet/IP communication link is DOWN, which indicates the high-level controller is unable to contact the PLC at the configured IP address or connectivity has been lost.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Status Request on page 152
Set Node Controller Configuration on page 245
Restart Node Controllers on page 247
Get Extended Node Controller Status on page 249
Get Extended High-Level Controller Status on page 252
Node Controller Status on page 273
Extended Node Controller Status on page 332
Extended High-Level Controller Status on page 335

Examples

See Status Request on page 152 for examples.

Node Controller Status

Message: 0xD2

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the node controllers that are specified in a Status Request command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Status Request is received for all node controllers, a separate status response is sent for each node controller.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	7
3	Message Type	1	0xD2
4	Node Controller ID	2	1…96
6	Node Controller Present	1	0, 1
7	Node Controller State	1	0x01…0x03
8	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Node Controller Status response.

Node Controller ID – The ID of the node controller for which status is reported.

Node Controller Present – Equal to 0 if the node controller that is specified is not present (and the remainder of message is undefined). Equal to 1 if the node controller that is specified is present based on the system configuration.

Node Controller State – This field indicates the state of the node controller.

Value	Description
0x01	**Initialization** – Loading the Node Controller Configuration File or an error was detected in the configuration preventing the node controller from exiting this state. Consult the node controller log for additional details when a node controller does not exit the Initialization state.
0x02	**Disconnected** – The TCP/IP connection from the high-level controller to this node controller is down.
0x03	**Operational** – The high-level controller connection to this node controller is established and the node controller is operational.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Status Request on page 152
Set Node Controller Configuration on page 245
Restart Node Controllers on page 247
Get Extended Node Controller Status on page 249
Get Extended High-Level Controller Status on page 252
High-Level Controller Status on page 271
Extended Node Controller Status on page 332
Extended High-Level Controller Status on page 335

Examples

See Status Request on page 152 for examples.

Node Status

Message: 0xD3

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the nodes that are specified in a Status Request command, including any node switch information. See MP Get Path End Status on page 219 for Moving Path status information. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Status Request is received for all nodes, a separate status response is sent for each node.

This response is sent as an asynchronous message when **Send Node Status Asynchronously** is selected in the Node Controller Configuration File and any node changes state, such as a vehicle entering/exiting. Enabling **Send Node Status Asynchronously** can result in many node status messages being sent.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	12
3	Message Type	1	0xD3
4	Node ID	2	1…65535
6	Node Present	1	0, 1
7	Node Type	1	0…12
8	Vehicle ID	2	0…65535
10	Requested Position or Terminus Signals	1	Node type 0: 0 Node type 1: 0…8 Node type 2: 0…8 Node type 3: — Node type 4: Bits 0…7 are Signal States Node type 5: 0 Node type 6: — Node type 7: — Node type 8: 0…8 Node type 9: 0 Node type 10: 0 Node type 11: 0 Node type 12: 0
11	Reported Position	1	0…8
12	Flags and Device Status	1	0x0…0x1F
13	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Node Status message.

Node ID – The ID of the node for which status is provided.

Node Present – Equal to 0 if the node specified is not present (remainder of data in the message is undefined). Equal to 1 if the node specified is present based on the system configuration.

Node Type – The type of the node for which status is provided (see Node Type Descriptions and Usage on page 90).

Value	Node Type	Supported Transport Systems
0	Relay Node	MagneMover LITE, QuickStick, QSHT
1	Merge Node	MagneMover LITE
2	Diverge Node	MagneMover LITE
3	Reserved	—
4	Terminus Node	MagneMover LITE, QuickStick, QSHT
5	Simple Node	MagneMover LITE, QuickStick, QSHT
6	Reserved	—
7	Reserved	—
8	Merge-Diverge Node	MagneMover LITE
9	Gateway Node	MagneMover LITE, QuickStick, QSHT
10	Overtravel Node	QuickStick, QSHT
11	Moving Path Node	QuickStick, QSHT
12	Reserved	—

Vehicle ID – The ID of the vehicle currently navigating this node or zero when the node is idle and no vehicle is navigating the node.

A vehicle is said to be navigating the node (or owning the node) when the node is within the motor permissions that are required for brick-wall headway for the vehicle. Vehicles are considered to have left the node once the center of the vehicle is 1/2 vehicle length plus 1/4 motor cycle past the node in the direction of travel for the vehicle.

Requested Position or Terminus Signals – The position the node apparatus was last commanded to move to or zero if no position previously commanded.

- Valid for a **Node Type** of 1, 2, or 8 (Merge, Diverge, or Merge-Diverge).
- Not used for a **Node Type** of 0, 5, 9, 10, or 11 (Relay, Simple, Gateway, Overtravel, or Moving Path), this value is zero.

Value	Description
0	No switch present or position has not yet been commanded
1	Switch position 1 (straight if merge or diverge)
2	Switch position 2 (curve if merge or diverge)
3	Switch position 3
4	Switch position 4
5	Switch position 5
6	Switch position 6
7	Switch position 7
8	Switch position 8

- For a **Node Type** of 4 (Terminus), these bits are the handshake signals for the Terminus Node where the bits have the following meanings.

Bit	Description
0	ENTRY_REQUESTED (last input state)
1	EXIT_ALLOWED (last input state)
2, 3	Reserved
4	ENTRY_CLEAR (output) **NOTE:** This bit is masked to be always Off for MM LITE systems.
5	ENTRY_ALLOWED (output)
6	EXITING (output)
7	Reserved

Reported Position – The position the node apparatus was last reported to be in or zero if no position previously reported.

- Valid for a **Node Type** of 1, 2, or 8 (Merge, Diverge, or Merge-Diverge).
- Not used for a **Node Type** of 0, 4, 5, 9, 10, or 11 (Relay, Terminus, Simple, Gateway, Overtravel, or Moving Path), this value is zero.

Value	Description
0	No switch present
1	Switch position 1 (straight if merge or diverge)
2	Switch position 2 (curve if merge or diverge)
3	Switch position 3
4	Switch position 4
5	Switch position 5
6	Switch position 6
7	Switch position 7
8	Switch position 8

Flags and Device Status – Bits 0…4 provide the status of the node.

- Valid for a **Node Type** of 1, 2, 8, or 11 (Merge, Diverge, Merge-Diverge, or Moving Path). Bits 0…3 provide the status of the device affiliated with the node (these are generic to cover any type of switch apparatus including generic digital I/O switches). Bit 4 is not used.
- Not used for a **Node Type** of 0, 4, 5, or 10 (Relay, Terminus, Simple, or Overtravel), bits 0…4 are zero.
- Valid for a **Node Type** of 9 (Gateway), Bits 0…3 are not used, Bit 4 is the status of the node.

Bit	Description
0…3	**Node Status:** 0 – No Device Present 1 – Initializing 2 – Faulted 3 – Operational 4 – Junction/Startup Fault 5 – Reserved 6 – Reserved
4	**Node Flags:** 0 – Enabled 1 – Disabled
5…7	Reserved

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Status Request on page 152
MP Get Path End Status on page 219
MP Path End Status Report on page 317

Examples

See Status Request on page 152 for examples.

Path Status

Message: 0xD4

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the paths that are specified in a Status Request command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Status Request is received for all paths, a separate status response is sent for each path.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	11
3	Message Type	1	0xD4
4	Path ID	2	1…65535
6	Path Present	1	0, 1
7	Number of Vehicles	1	0…384
8	Path State	1	0…4
9	Path Motion Status	1	Bits 0…7 are flags
10	Upstream Communication Link Status	1	0…2
11	Downstream Communication Link Status	1	0…2
12	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Path Status message.

Path ID – The ID of the path reporting status.

Path Present – Equal to 0 if the path is not present (remainder of data in the message is undefined) or 1 if the path that is specified is present based on the system configuration.

Number Of Vehicles – The number of vehicles on the specified path.

Path State – The current state of the path.

Value	Description
0	Initialization
1	Startup (locating vehicles)
2	Operational
3	Reset in progress
4	Programming in progress

Path Motion Status – Bit field. When all bits are set to 0, motion on the path is enabled. When any bit is set to 1, motion on the path is suspended and the bits that are set to 1 describe the reason for motion suspension.

Bit	Description
0	Suspended by command
1	E-stop active
2	Interlock active
3	FastStop active (QuickStick and QSHT only)
4…7	Reserved

Upstream Communication Link Status – The state of the communication link at the upstream end of this path.

Value	Description
0	Link OK
1	Link FAILED
2	Link Not Configured (no physical connection is configured or no connection is defined in the Node Controller Configuration File).

Downstream Communication Link Status – The state of the communication link at the downstream end of this path.

Value	Description
0	Link OK
1	Link FAILED
2	Link Not Configured (no physical connection is configured or no connection is defined in the Node Controller Configuration File).

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Status Request on page 152

Examples

See Status Request on page 152 for examples.

Vehicle Status

Message: 0xD5

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the vehicles that are specified in a Status Request command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Status Request is received for all vehicles, a separate status response is sent for each vehicle.

This response is sent as an asynchronous message when vehicles become obstructed and **Obstructed Status Notification** is selected in the Node Controller Configuration File. Enabling **Obstructed Status Notification** can result in many Vehicle Status messages being sent when vehicles are waiting in a queue.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	22 - Move Vehicle to Station 24 - Move Vehicle to Position 24 - Vehicle Follow Order 24 - No command in progress
3	Message Type	1	0xD5
4	Vehicle ID	2	1…65535
6	Vehicle Present	1	0, 1
7	Path ID	2	1…65535
9	Destination Path ID	2	0…65535
11	Vehicle Position	4	-41.0…+41.0 (m, floating-point)*
15	Vehicle Velocity	4	-41.0…+41.0 (m/s, floating-point)(*)
19	Vehicle Command Type	1	0x00, 0xB0, 0xB1, 0xB7
20	Vehicle Flags	1	Bits 0…7 are flags
21	Commanded Position or Station ID	4	0…41.0 (m, floating-point) if position†
		2	1…2048 if Station ID
Varies	Message CRC	2	0…65535

* The fixed-point value from the motor that is converted to floating-point.
† The floating-point value from the host controller.

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Vehicle Status message.

Vehicle ID – The ID of the vehicle for which status is being reported.

Vehicle Present – Equal to 0 if the vehicle is not present (remainder of data in the message is undefined) or 1 if the vehicle is present.

Path ID – The ID of the path where the vehicle is located.

Destination Path ID – The ID number of the path where the vehicle is headed.

Value	Description
0	Equal to 0 if the vehicle has never moved (no command in progress (0x00)) or if the vehicle is under a Vehicle Follow Order (0xB7).
1…65535	Equal to the ID of the path where the vehicle is headed or the ID of the path where the vehicle is located if motion has completed.

Vehicle Position – The last reported position of the vehicle in meters relative to the start of the specified path. Vehicle position is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401). Zero position is defined as the midpoint of the vehicle at the beginning of the path.

Vehicle Velocity – The last reported velocity (in m/s) of the vehicle on the specified path. Vehicle velocity is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

Vehicle Command Type – Indicates the type of command currently active for the vehicle.

Value	Description
0x00	No command is in progress.
0xB0	Move vehicle to station.
0xB1	Move vehicle to position.
0xB7	Vehicle Follow Order.

Vehicle Flags – These bits are the vehicle status flags.

Bit	Description
0	**Vehicle Signal:** 0 – The motor does not detect the vehicle. 1 – The motor currently in charge of the vehicle detects the vehicle.
1	**Obstructed Status:** 0 – The vehicle is not obstructed and able to acquire permission to move further. 1 – The vehicle is obstructed and unable to acquire permission to move further. Obstructions are due to either a vehicle in the way, a hardware fault, or motion is suspended. Vehicles in the way occur during normal operation when vehicles are in a queue or when a vehicle is in a switch, which keeps another vehicle from entering the switch. If after three consecutive reports the obstructed bit is still set, check for an actual obstruction.

Bit	Description
2	**Hindered Status:** 0 – The vehicle is making progress on its current move profile. 1 – The vehicle is not making progress towards the position that it has most recently been granted permission to move to. The motor continues to apply force on the vehicle to try to move it indefinitely. Lack of progress can happen in the following situations: • A vehicle is held back by some external force including a foreign object on the guideway that prevents vehicle motion. • A vehicle is not in motion while under sync control. • A vehicle command uses a velocity of zero. • A vehicle command uses a PID set equal to zero. • The Control Off Position Tolerance or the Integrator Distance Threshold positions are set outside of the arrival tolerance. • The motor does not have propulsion power.
3…6	**PID Loop Set Index:** 0 – Use user-defined PID set 0 – Unloaded PID values. 1 – Use user-defined PID set 1 – Loaded PID values. 2 – Use user-defined PID set 2. 3 – Use user-defined PID set 3. 4 – Use user-defined PID set 4. 5 – Use user-defined PID set 5. 6 – Use user-defined PID set 6. 7 – Use user-defined PID set 7. 8 – Use user-defined PID set 8. 9 – Use user-defined PID set 9. 10 – Use user-defined PID set 10. 11 – Use user-defined PID set 11. 12 – Use user-defined PID set 12. 13 – Use user-defined PID set 13. 14 – Use user-defined PID set 14. 15 – Use user-defined PID set 15 – Startup PID values. This PID set is automatically used during startup. If it is not defined, PID set 0 is scaled by 25% and used for startup.
7	**Suspect Status (MM LITE and QuickStick only):** 0 – The vehicle is not suspect and is free to move. 1 – The vehicle is suspect. The motor has detected that the vehicle has been manually moved out of the control region. Typically, control of the vehicle cannot be regained. Even if control is regained, the vehicle continues to be suspect until a Vehicle Command is issued to clear the Suspect Bit. Or, the vehicle must be deleted using a Delete Vehicle command and each path that is reporting an unlocated vehicle fault must be restarted.

Commanded Position or Station ID – The destination for the vehicle.

Cmd	Command Type	Description
0x00	No command or command complete	4-byte value that is set to zero
0xB0	Vehicle station order	2-byte value that is the commanded Station (1…2048).
0xB1	Vehicle position order	4-byte value that is the commanded destination on the specified path (-41.0…+41.0 m) expressed as a 32-bit single-precision floating point number using little-endian format (see Data Encoding on page 401)).
0xB7	Vehicle follow order	4-byte value that is the distance to the vehicle being followed (0…41.0 m) expressed as a 32-bit single-precision floating point number using little-endian format (see Data Encoding on page 401)).

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Move Vehicle To Station on page 136
Move Vehicle To Position on page 141
Status Request on page 152
Vehicle Follow Order on page 161
Get Extended Vehicle Status on page 208

Examples

See Move Vehicle To Station on page 136, Move Vehicle To Position on page 141, and Vehicle Follow Order on page 161 for examples.

Motor Status

Message: 0xD7

Source/Destination

HLC ► Host Controller

Purpose

Reports the status and fault information of the motors that are specified in a Status Request command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Status Request command is received for all motors, a separate status response is sent for each motor.

Intermittent, temporary propulsion under-voltage and over-voltage warnings (not faults) can occur as a result of power distribution anomalies, dissipating regenerative braking energy when vehicles decelerate, or power consumption when accelerating vehicles. These conditions are not reported to the host controller. To check for their occurrence, set the HLC or NC Fault module logging level in the Node Controller Web Interface to Warning (see the MagneMotion Node Controller Interface User Manual, publication MMI-UM001). Log messages are displayed for the motors when the motor reports the condition. The warnings clear when the condition is removed.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	Varies
3	Message Type	1	0xD7
4	Path ID	2	1…65535
6	Path Present	1	0, 1
7	Motor Index	1	1…40 or 255
8	Motor Present	1	0, 1
9	Motor Type	1	0 – Motor not present 1 – Motor is QuickStick 100 2 – Motor is MagneMover LITE 3 – Motor is QuickStick HT Gen 2 4 – Motor is QuickStick HT 5700 5 – Motor is QuickStick 150
10	Motor Fault Data	Varies	Motor fault data varies based on the motor type that is specified in **Motor Type**.
Varies	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Motor Status message.

Path ID – The ID of the path where the motor is located.

Path Present – Equal to 0 if the path is not present (remainder of data in the message is undefined) or 1 if the path is present.

Motor Index – The position of the motor along the path, 1…40 or 255 to indicate the host controller queried for all motors with active faults along the given path, but no motors have any faults active. A single motor status response with Motor Index set to 255 is sent by the HLC when no motors have any faults to report, the remainder of the message is zeroed out and can be ignored.

Motor Present – Equal to 0 when the motor is not present (remainder of data in the message is undefined). Equal to 1 if the motor specified by **Motor Index** is present within the specified path. If the host controller requests motor status for a non-existent Motor Index, this field is set to 0 to indicate that the motor does not exist.

Motor Type – The type of motor returning the status information. When the host controller requests motor status for a non-existent path, Motor ID, or both, this field is set to 0.

Value	Description
0	**Invalid** – This field is zero when the path or motor being reported doesn't exist (an error in the request from the host controller).
1	**Motor type is QuickStick 100** – The fault data for the motor is provided after the Motor Type field and varies based on the motor.
2	**Motor type is MagneMover LITE** – The fault data for the motor is provided after the Motor Type field and varies based on the motor.
3	**Motor type is QuickStick HT Gen 2** – The fault data for the motor is provided after the Motor Type field and varies based on the motor.
4	**Motor type is QuickStick HT 5700** – The fault data for the motor is provided after the Motor Type field and varies based on the motor.
5	**Motor type is QuickStick 150** – The fault data for the motor is provided after the Motor Type field and varies based on the motor.

Motor Fault Data, MagneMover LITE – Status and fault data for the MM LITE motors. See Table 5-3, *MagneMover LITE Motor Fault Troubleshooting, on page 347* for detailed fault definitions, set and clear conditions, and user actions for these faults. For each fault bit, when the bit is low, the fault is inactive. When the bit is high (1), the fault is active. A normally operational transport system has no fault bits active for any motor along any path

Some MagneMover LITE motors do not populate every driver board fault slot in the fault data. For such motors, the **Number of Driver Boards** field in the fault data indicates the number of driver board faults the motor populates and the length of the message is shortened appropriately.

Table 4-12: MagneMover LITE Motor Fault Data

Offset	Item	Size	Range
10	Number of Driver Boards	1	Number of driver boards used in a motor.
11	Scheduler Errors	1	Bit 0: Scheduler not initialized Bit 1: Scheduler event queue full Bit 2…7: Reserved
12	Upstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Miscellaneous comm warning Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4…7: Reserved

Table 4-12: MagneMover LITE Motor Fault Data (Continued)

Offset	Item	Size	Range
13	Downstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Miscellaneous comm warning Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4…7: Reserved
14	State Errors	1	Bit 0: Motor not in operational mode Bit 1: Motor in configuration mode Bit 2: Motor in diagnostic mode Bit 3: Motion suspended by node controller Bit 4…6: Reserved Bit 7: Motor not responding
15	Master Board Faults A	1	Bit 0: Unlocated vehicle fault Bit 1: Switch position fault Bit 2: Propulsion power not ready Bit 3: Under-voltage fault Bit 4: Over-voltage fault Bit 5: Over-temperature fault Bit 6: Over-current fault Bit 7: Configuration fault
16	Master Board Faults B	1	Bit 0…7: Reserved
17	Driver Board 1 Faults	1	Bit 0: Estimated coil temperature too high Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Over-temperature fault Bit 4: Not responding Bit 5: Reserved Bit 6: Switch unable to complete move Bit 7: Switch mechanism out of position
. . .	Driver board errors can repeat for a total of eight driver boards	. . .	
24	Driver Board 8 Faults	1	Bit 0: Estimated coil temperature too high Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Over-temperature fault Bit 4: Not responding Bit 5:Reserved Bit 6: Switch unable to complete move Bit 7: Switch mechanism out of position

Motor Fault Data, QuickStick 100 – Status and fault data for the QS 100 motors. See Table 5-4, *QuickStick 100 Motor Fault Troubleshooting, on page 355* for detailed fault definitions, set and clear conditions, and user actions for these faults. For each fault bit, when the bit is low, the fault is inactive. When the bit is high (1), the fault is active. A normally operational transport system has no fault bits active for any motor along any path

Some QuickStick 100 motors do not populate every block fault slot in the fault data. For those motors, the **Number of Blocks** field in the fault data indicates the number of blocks the motor populates and the length of the message is shortened appropriately.

Table 4-13: QuickStick 100 Motor Fault Data

Offset	Item	Size	Range
10	Number of Blocks	1	Number of blocks that are used in a motor.
11	Scheduler Errors	1	Bit 0: Scheduler not initialized Bit 1: Scheduler event queue full Bit 2…7: Reserved
12	Upstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Miscellaneous comm warning Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4…7: Reserved
13	Downstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Miscellaneous comm warning Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4…7: Reserved
14	State Errors	1	Bit 0: Motor not in operational mode Bit 1: Motor in configuration mode Bit 2: Motor in diagnostic mode Bit 3: Motion suspended Bit 4: Motion suspended – FastStop active Bit 5, 6: Reserved Bit 7: Motor not responding
15	Block 1 Faults A	1	Bit 0: Inverter Disabled Bit 1: Motor under-voltage warning (Block 1 only) Bit 2: Motor over-voltage warning (Block 1 only) Bit 3: Reserved Bit 4: In bootload mode Bit 5: Slave module is not responding Bit 6: Soft Start not complete (Block 1 only) Bit 7: Slave processor reset initiated

Table 4-13: QuickStick 100 Motor Fault Data (Continued)

Offset	Item	Size	Range
16	Block 1 Faults B	1	Bit 0: Over-current fault Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Motor stall detected Bit 4: Slave module not configured Bit 5: Over-temperature fault Bit 6: Hall Effect sensor fault Bit 7: Slave Communication fault
17	Block 2 Faults A	1	Bit 0: Inverted disabled Bit 1: Vehicle not located Bit 2: Unexpected slave module reset Bit 3: Reserved Bit 4: In bootload mode Bit 5: Slave module is not responding Bit 6: Reserved Bit 7: Slave processor reset initiated
18	Block 2 Faults B	1	Bit 0: Over-current fault Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Motor stall detected Bit 4: Slave module not configured Bit 5: Over-temperature fault Bit 6: Hall Effect sensor fault Bit 7: Slave Communication fault
. . .	Block errors can repeat for a total of 10 blocks	. . .	
33	Block 10 Faults A	1	Bit 0: Inverted disabled Bit 1: Vehicle not located Bit 2: Unexpected slave module reset Bit 3: Reserved Bit 4: In bootload mode Bit 5: Slave module is not responding Bit 6: Reserved Bit 7: Slave processor reset initiated
34	Block 10 Faults B	1	Bit 0: Over-current fault Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Motor stall detected Bit 4: Slave module not configured Bit 5: Over-temperature fault Bit 6: Hall Effect sensor fault Bit 7: Slave Communication fault

Motor Fault Data, QuickStick 150 - Status and fault data collected for QuickStick 150 motors. See Table 5-5, *QuickStick 150 Motor Fault Troubleshooting, on page 363* for detailed fault definitions, set and clear conditions, and user actions for these faults. For each fault bit, when the bit is low, the fault is inactive. When the bit is high (1), the fault is active. A normally operational transport system has no fault bits active for any motor along any path.

Some QuickStick 150 motors do not populate every block fault slot in the fault data. For such motors, the **Number of Blocks** field in the fault data indicates the number of blocks the motor populates and the length of the message is shortened appropriately.

Table 4-14: QuickStick 150 Motor Fault Data

Offset	Item	Size	Range
10	Number of Blocks	1	Number of blocks that are used in a motor.
11	Scheduler Errors	1	Bit 0: Scheduler not initialized Bit 1: Scheduler event queue full Bit 2…7: Reserved
12	Upstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Reserved Bit 2: Transmit buffer full Bit 3: Reserved Bit 4: Reserved Bit 5: Link down Bit 6, 7: Reserved
13	Downstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Reserved Bit 2: Transmit buffer full Bit 3: Reserved Bit 4: Reserved Bit 5: Link down Bit 6, 7: Reserved
14	State Errors	1	Bit 0: Motor not in operational mode Bit 1: Motor in configuration mode Bit 2: Motor in diagnostic mode Bit 3: Motion suspended Bit 4: Motion suspended – FastStop active Bit 5, 6: Reserved Bit 7: Motor not responding
15	Block 1 Faults A	1	Bit 0: Reserved Bit 1: Reserved Bit 2: Gate driver not responding Bit 3…7: Reserved

Table 4-14: QuickStick 150 Motor Fault Data (Continued)

Offset	Item	Size	Range
16	Block 1 Faults B	1	Bit 0: Gate driver over-current fault Bit 1: Gate driver under-voltage fault Bit 2: Gate driver over-temperature fault Bit 3: Motor stall detected Bit 4: Reserved Bit 5: Stator over-temperature fault Bit 6: Hall Effect sensor fault Bit 7: Hall Effect sensor not responding
17	Block 2 Faults A	1	Bit 0: Reserved Bit 1: Reserved Bit 2: Gate driver not responding Bit 3…7: Reserved
18	Block 2 Faults B	1	Bit 0: Gate driver over-current fault Bit 1: Gate driver under-voltage fault Bit 2: Gate driver over-temperature fault Bit 3: Motor stall detected Bit 4: Slave module not configured Bit 5: Stator over-temperature fault Bit 6: Hall Effect sensor fault Bit 7: Hall Effect sensor not responding
. . .	Block errors can repeat for a total of 10 blocks	. . .	
33	Block 10 Faults A	1	Bit 0: Reserved Bit 1: Reserved Bit 2: Gate driver not responding Bit 3…7: Reserved
34	Block 10 Faults B	1	Bit 0: Gate driver over-current fault Bit 1: Gate driver under-voltage fault Bit 2: Gate driver over-temperature fault Bit 3: Motor stall detected Bit 4: Reserved Bit 5: Stator over-temperature fault Bit 6: Hall Effect sensor fault Bit 7: Hall Effect sensor not responding
35	Master Board Faults	1	Bit 0: Aux Core not responding Bit 1: Unexpected Aux Core reset detected Bit 2: Reserved Bit 3: Reserved Bit 4: Manufacturing Data CRC error Bit 5: Calibration Data CRC error Bit 6, 7: Reserved

Table 4-14: QuickStick 150 Motor Fault Data (Continued)

Offset	Item	Size	Range
36	Secondary Core Faults	1	Bit 0: Aux Core not responding Bit 1: Unexpected Aux Core reset detected Bit 2: Reserved Bit 3: Reserved Bit 4: Manufacturing Data CRC error Bit 5: Calibration Data CRC error Bit 6, 7: Reserved
37	Drive Core 1 Faults	1	Bit 0: Bus under-voltage fault Bit 1: Bus over-voltage fault Bit 2: Fuse open Bit 3: Soft start not complete Bit 4: Bus under-voltage warning Bit 5: Bus over-voltage warning Bit 6, 7: Reserved
38	Drive Core 2 Faults	1	Bit 0: Bus under-voltage fault Bit 1: Bus over-voltage fault Bit 2: Fuse open Bit 3: Soft start not complete Bit 4: Bus under-voltage warning Bit 5: Bus over-voltage warning Bit 6, 7: Reserved
39	Ethernet Comm Errors	4	Bit 0: Duplicate IP Address detected Bit 1: IP Address not configured Bit 2…31: Reserved

Motor Fault Data, QuickStick High Thrust – Status and fault data for the QSHT motors. See Table 5-6, *QSHT and QSHT 5700 Motor Fault Troubleshooting, on page 374* for detailed fault definitions, set and clear conditions, and user actions for these faults. For each fault bit, when the bit is low, the fault is inactive. When the bit is high (1), the fault is active. A normally operational transport system has no fault bits active for any motor along any path.

Some QuickStick HT motors do not populate every block fault slot in the fault data. For such motors, the unpopulated blocks in the fault data contain 0. The **Number of Blocks** filed in the fault data indicates the number of blocks the motor populates.

Table 4-15: QuickStick HT Gen 2 Motor Fault Data

Offset	Item	Size	Range
10	Number of Blocks	1	Number of blocks used in a motor.
11	Scheduler Errors	1	Bit 0: Scheduler not initialized Bit 1: Scheduler event queue full Bit 2…7: Reserved

Table 4-15: QuickStick HT Gen 2 Motor Fault Data (Continued)

Offset	Item	Size	Range
12	Upstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Reserved Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4: Reserved Bit 5: Link down Bit 6, 7: Reserved
13	Downstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Reserved Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4: Reserved Bit 5: Link down Bit 6, 7: Reserved
14	State Errors	1	Bit 0: Motor not in operational mode Bit 1: Motor in configuration mode Bit 2: Motor in diagnostic mode Bit 3: Motion suspended Bit 4: Motion suspended – FastStop active Bit 5, 6: Reserved Bit 7: Motor not responding
15	Master Board Faults A	1	Bit 0…7: Reserved
16	Master Board Faults B	1	Bit 0: Logic under-voltage fault Bit 1: Logic over-voltage fault Bit 2: Logic over-temperature fault Bit 3…7: Reserved
17	Block 1 HES Faults A	1	Bit 0: Reserved Bit 1: Stator mismatch Bit 2: Reserved Bit 3: In diagnostic mode Bit 4: Reserved Bit 5: In bootload mode Bit 6, 7: Reserved
18	Block 1 HES Faults B	1	Bit 0…4: Reserved Bit 5: Stator over-temperature fault Bit 6: Not responding Bit 7: Reserved

Table 4-15: QuickStick HT Gen 2 Motor Fault Data (Continued)

Offset	Item	Size	Range
19	Block 2 HES Faults A	1	Bit 0: Reserved Bit 1: Stator mismatch Bit 2: Reserved Bit 3: In diagnostic mode Bit 4: Reserved Bit 5: In bootload mode Bit 6, 7: Reserved
20	Block 2 HES Faults B	1	Bit 0…4: Reserved Bit 5: Stator over-temperature fault Bit 6: Not responding Bit 7: Reserved
21	Block 1 Inverter Faults A	1	Bit 0: Motor controller disabled by command Bit 1: Soft Start switch Off Bit 2: Reserved Bit 3: Hardware over-current warning Bit 4: Software over-current fault Bit 5: In bootload mode Bit 6: Fan fault Bit 7: Motor controller fault
22	Block 1 Inverter Faults B	1	Bit 0: Hardware over-current fault Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Motor stall detected Bit 4: Reserved Bit 5: Over-temperature fault Bit 6: Not responding Bit 7: Reserved
23	Block 2 Inverter Faults A	1	Bit 0: Motor controller disabled by command Bit 1: Soft Start switch Off Bit 2: Reserved Bit 3: Hardware over-current warning Bit 4: Software over-current fault Bit 5: In bootload mode Bit 6: Fan fault Bit 7: Motor controller fault
24	Block 2 Inverter Faults B	1	Bit 0: Hardware over-current fault Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Motor stall detected Bit 4: Reserved Bit 5: Over-temperature fault Bit 6: Not responding Bit 7: Reserved

Table 4-15: QuickStick HT Gen 2 Motor Fault Data (Continued)

Offset	Item	Size	Range
25	Ethernet Comm Faults	4	Bit 0…31: Reserved
26	Block 1 Safety Faults	1	Bit 0…7: Reserved

Motor Fault Data, QuickStick HT 5700 - Status and fault data collected for QuickStick HT 5700 motors. See *Table 5-6, QSHT and QSHT 5700 Motor Fault Troubleshooting, on page 374* for detailed fault definitions, set and clear conditions, and user actions for these faults. For each fault bit, when the bit is low, the fault is inactive. When the bit is high (1), the fault is active. A normally operational transport system has no fault bits active for any motor along any path.

Some QuickStick HT 5700 motors do not populate every block fault slot in the fault data. For such motors, the unpopulated blocks in the fault data contain 0. The **Number of Blocks** field in the fault data indicates the number of blocks the motor populates

Table 4-16: QuickStick HT 5700 Motor Fault Data

Offset	Item	Size	Range
10	Number of Blocks	1	Number of blocks used in a motor.
11	Scheduler Errors	1	Bit 0: Scheduler not initialized Bit 1: Scheduler event queue full Bit 2…7: Reserved
12	Upstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Reserved Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4: Reserved Bit 5: Link down Bit 6, 7: Reserved
13	Downstream Comm Errors	1	Bit 0: Connection inoperative Bit 1: Reserved Bit 2: Transmit buffer full Bit 3: UART settings corrected Bit 4: Reserved Bit 5: Link down Bit 6, 7: Reserved
14	State Errors	1	Bit 0: Motor not in operational mode Bit 1: Motor in configuration mode Bit 2: Motor in diagnostic mode Bit 3: Motion suspended Bit 4: Motion suspended – FastStop active Bit 5, 6: Reserved Bit 7: Motor not responding
15	Master Board Faults A	1	Bit 0…7: Reserved

Table 4-16: QuickStick HT 5700 Motor Fault Data (Continued)

Offset	Item	Size	Range
16	Master Board Faults B	1	Bit 0: Logic under-voltage fault Bit 1: Logic over-voltage fault Bit 2: Over-temperature fault Bit 3: Safety Core not responding Bit 4: Aux Core not responding Bit 5: Safety Core invalid software Bit 6, 7: Reserved
17	Block 1 HES Faults A	1	Bit 0: Reserved Bit 1: Stator mismatch Bit 2: Reserved Bit 3: In diagnostic mode Bit 4: Reserved Bit 5: In bootload mode Bit 6, 7: Reserved
18	Block 1 HES Faults B	1	Bit 0…4: Reserved Bit 5: Stator over-temperature fault Bit 6: Not responding Bit 7: Reserved
19	Block 2 HES Faults A	1	Bit 0: Reserved Bit 1: Stator mismatch Bit 2: Reserved Bit 3: In diagnostic mode Bit 4: Reserved Bit 5: In bootload mode Bit 6, 7: Reserved
20	Block 2 HES Faults B	1	Bit 0…4: Reserved Bit 5: Stator over-temperature fault Bit 6: Not responding Bit 7: Reserved
21	Block 1 Inverter Faults A	1	Bit 0: Inverter disabled by command Bit 1: Inverter disabled by Power Supply Not Ready Bit 2: Fuse open Bit 3: Hardware over-current warning Bit 4: Reserved Bit 5: In bootload mode Bit 6: Reserved Bit 7: Inverter fault
22	Block 1 Inverter Faults B	1	Bit 0: Hardware over-current fault Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Motor stall detected Bit 4: Guard stop request status Bit 5: Over-temperature fault Bit 6: Not responding Bit 7: Gate drive under-voltage lockout

Table 4-16: QuickStick HT 5700 Motor Fault Data (Continued)

Offset	Item	Size	Range
23	Block 2 Inverter Faults A	1	Bit 0: Inverter disabled by command Bit 1: Inverter disabled by Power supply Not Ready Bit 2: Fuse open Bit 3: Hardware over-current warning Bit 4: Reserved Bit 5: In bootload mode Bit 6: Reserved Bit 7: Inverter fault
24	Block 2 Inverter Faults B	1	Bit 0: Hardware over-current fault Bit 1: Under-voltage fault Bit 2: Over-voltage fault Bit 3: Motor stall detected Bit 4: Guard stop request status Bit 5: Over-temperature fault Bit 6: Not responding Bit 7: Gate driver under-voltage lockout
25	Ethernet Comm Faults	4	Bit 0…31: Reserved
29	Block 1 Safety Core Faults	1	Bit 0: Safety Core Fault Bit 1: Safe Torque Off Fault Bit 2: Guard Stop Input Fault Bit 3…7: Reserved
30	Block 2 Safety Core Faults	1	Bit 0: Safety Core Fault Bit 1: Safe Torque Off Fault Bit 2: Guard Stop Input Fault Bit 3…7: Reserved
31	Digital Input Status	1	Bit 0: Digital Input 1 Bit 1: Digital Input 2 Bit 2: Digital Input 3 Bit 3: Digital Input 4 Bit 4…7: Reserved

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Status Request on page 152

Examples

See Status Request on page 152 for examples.

Station Status

Message: 0xDF 0x01 0x01

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the stations that are specified in a Get Station Status command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Get Station Status is received for all stations, a separate status response is sent for each station.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	14
3	Message Type	1	0xDF
4	Extension Type	1	0x01
5	Extension Subtype	1	0x01
6	Command Status	1	0x00, 0x02, 0x0C, 0x0E
7	Station ID	2	1…2048
9	Path ID	2	1…65535
11	Position	4	0…41.0 (m, floating-point)
15	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Station extension message.

Extension Subtype – Fixed extension subtype that identifies this message as a Station Status message.

Command Status – The status of the command that this Station Status message is acknowledging (see HLC Command Status Codes).

Station ID – The ID of the station for which status is being reported.

Path ID – The ID of the path where the station is located.

Position – The position of the station in meters relative to the start of the specified path. Station position is expressed as a 32-bit single-precision floating point number using little-endian format (see Data Encoding on page 401). Zero position is defined as the beginning of the path.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Get Station Status on page 190

Examples

See Get Station Status on page 190 for examples.

Traffic Light Status

Message: 0xDF 0x02 0x01

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the traffic lights that are specified in a Get Traffic Light Status command.

NOTE: When an Get Traffic Light Status is received for all traffic lights, a separate status response is sent for each traffic light.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	15
3	Message Type	1	0xDF
4	Extension Type	1	0x02
5	Extension Subtype	1	0x01
6	Command Status	1	0x00, 0x0C, 0x0E, 0x11
7	Traffic Light ID	2	0…4096
9	Path ID	2	1…65535
11	Position	4	0…41.0 (m, floating-point)
15	Color	1	0, 1
16	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Traffic Light extension message.

Extension Subtype – Fixed extension subtype that identifies this message as a Traffic Light Status extension message.

Command Status – The status of the command that this Traffic Light Status response message is acknowledging (see HLC Command Status Codes).

Traffic Light ID – The ID of the traffic light for which status is being reported.

Path ID – The ID of the path where the traffic light is located.

Position – The position of the traffic light in meters relative to the start of the specified path. Position is expressed as a 32-bit single-precision floating point number using little-endian format (see Data Encoding on page 401). Zero position is defined as the beginning of the path.

Color – The color of the traffic light for which status is being reported.

Value	Description
0	Green (allows traffic to pass)
1	Red (stops traffic)

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Get Traffic Light Status on page 199

Examples

See Get Traffic Light Status on page 199 for examples.

Extended Vehicle Status,

Message: 0xDF 0x03 0x07

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the vehicles that are specified in a Get Extended Vehicle Status command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When an Get Extended Vehicle Status is received for all vehicles, a separate status response is sent for each vehicle.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	53
3	Message Type	1	0xDF
4	Extension Type	1	0x03
5	Extension Subtype	1	0x07
6	Vehicle ID	2	1…65535
8	Vehicle Present	1	0…1
9	Path ID	2	1…65535
11	Destination Path ID	2	0…65535
13	Vehicle Position	4	-41.0…+41.0 (m, floating-point)*
17	Vehicle Velocity	4	-41.0…+41.0 (m/s, floating point)(*)
21	Vehicle Command Type	1	0x00, 0xB0, 0xB1, 0xB7
22	Vehicle Flags	2	Bits 0…15 are status flags

Offset	Item	Bytes	Range
24	Commanded Position	4	-41.0…+41.0 (m, floating-point)†
28	Current Target	4	-41.0…+41.0 (m, floating-point)
32	Followed Vehicle ID	2	Command types 0x00, 0xB0, 0xB1 – 0 Command type 0xB7 – 1…65535
34	Time Since Last Report	4	Seconds (floating-point)
38	Destination Station ID	2	0…2048
40	Reported PID Set Index	1	0…15
41	Ordered PID Set Index	1	0…15
42	Ordered Acceleration Limit	4	Float in m/s^2 (limits system dependent)
46	Ordered Velocity Limit	4	Float in m/s (limits system dependent)
50	Active Move to Station Offset	4	-41.0…+41.0 (m, floating point)
54	Message CRC	2	0…65535

* The fixed-point value from the motor that is converted to floating-point.
† The floating-point value from the host controller.

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Vehicle extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an Extended Vehicle Status extension message.

Vehicle ID – The ID of the vehicle for which status is being reported.

Vehicle Present – Equal to 0 if the vehicle is not present (remainder of data in the message is undefined) or 1 if the vehicle is present.

Path ID – The ID of the path where the vehicle is located.

Destination Path ID – The ID of the path where the vehicle is headed.

Value	Description
0	Equal to 0 if the vehicle has never moved (no command in progress (0x00) or if the vehicle is under a Vehicle Follow Order (0xB7).
1…65535	Equal to the ID of the path where the vehicle is headed or the ID of the path where the vehicle is located if motion has completed.

Vehicle Position – The last reported position of the vehicle in meters relative to the start of the specified path. Vehicle position is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401). Zero position is defined as the midpoint of the vehicle at the beginning of the path.

Vehicle Velocity – The last reported velocity (in m/s) of the vehicle on the specified path. Vehicle velocity is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

Vehicle Command Type – Indicates the type of command currently active for the vehicle.

Value	Description
0x00	No command is in progress.
0xB0	Move vehicle to station.
0xB1	Move vehicle to position.
0xB7	Vehicle Follow Order.

Vehicle Flags – The vehicle status flags.

Bit	Description
0	**Vehicle Signal:** 0 – The motor does not detect the vehicle. 1 – The motor currently in charge of the vehicle detects the vehicle.
1	**Obstructed Status:** 0 – The vehicle is not obstructed and able to acquire permission to move further. 1 – The vehicle is obstructed and unable to acquire permission to move further. Obstructions are due to either a vehicle in the way, a hardware fault, or motion is suspended. Vehicles in the way occur during normal operation when vehicles are in a queue or when a vehicle is in a switch, which keeps another vehicle from entering the switch. If after three consecutive reports the obstructed status bit is still set, check for an actual obstruction.

Bit	Description
2	**Hindered Status:** 0 – The vehicle is making progress on its current move profile. 1 – The vehicle is not making progress towards the position that it has most recently been granted permission to move to. The motor continues to apply force on the vehicle to try to move it indefinitely. Lack of progress can happen in the following situations: • A vehicle is held back by some external force including a foreign object on the guideway that prevents vehicle motion. • A vehicle is not in motion while under sync control. • A vehicle command uses a velocity of zero. • A vehicle command uses a PID set equal to zero. • The Control Off Position Tolerance or the Integrator Distance Threshold positions are set outside of the arrival tolerance. • The motor does not have propulsion power.
3	**Stall Status:** 0 – The vehicle is not stalled. 1 – The vehicle has stalled. A stalled vehicle is defined as the motor in control of a vehicle has had to reduce power to keep from overheating.
4	**Lock Status:** 0 – The vehicle is not locked. 1 – The vehicle is locked. Motion commands are rejected (vehicle must be unlocked).
5	**Locate Status:** 0 – Vehicle locate is not completed (startup is in progress). 1 – Vehicle locate has completed.
6	Reserved
7	**Suspect Status (MM LITE and QuickStick only):** 0 – The vehicle is not suspect and is free to move. 1 – The vehicle is suspect. The motor has detected that the vehicle has been manually moved out of the control region. Typically, control of the vehicle cannot be regained. Even if control is regained, the vehicle continues to be suspect until a Vehicle Command is issued to clear the Suspect Bit. Or, the vehicle must be deleted using a Delete Vehicle command and each path that is reporting an unlocated vehicle fault must be restarted.
8	**Following Upstream:** 0 – This vehicle is not following another vehicle. 1 – This vehicle is following the vehicle that is specified in the **Followed Vehicle ID** field and that vehicle is upstream from this vehicle.

Bit	Description
9	**Following Downstream:** 0 – This vehicle is not following another vehicle. 1 – This vehicle is following the vehicle that is specified in the **Followed Vehicle ID** field and that vehicle is downstream from this vehicle.
10	**Caught Up:** 0 – This vehicle is following another vehicle and has not yet caught up to it. Or, this vehicle is not following another vehicle. 1 – The motor controller is reporting that this following vehicle has caught up to the vehicle that it is following and is at the requested following distance.
11	**Profile Stale Error:** 0 – This vehicle is not following another vehicle. 1 – This vehicle is following another vehicle and the motor controller is reporting that this following vehicle has not received profile data from the vehicle that it is following, in the expected time (~10 ms). The vehicle slows to a stop. If the profile data is restored, the vehicle catches up to its leader using the catchup acceleration and velocity.
12	**Excessive Following Error:** 0 – This vehicle is not following another vehicle. 1 – This vehicle is following another vehicle and the motor controller is reporting that this following vehicle is too far from where it is expected to be (possibly stopped receiving following profile data).
13	**Vehicle Decoupled:** 0 – This vehicle is a follower in a platoon and is not in the decoupling state, or this vehicle is not in a platoon. 1 – This vehicle is a platoon follower vehicle that is decoupling from the vehicle that it is following. This bit stays set from when the motor controller reports decoupling, until the decoupled vehicle reaches its new destination under the 0xB1 position order that was converted from the 0xB7 follow order.
14, 15	Reserved

Commanded Position – The destination for the vehicle.

Value	Command Type	Description
0x00	No command or command complete	4-byte value that is set to zero.

Value	Command Type	Description
0xB0	Vehicle station order	4-byte value that is the commanded destination on the specified path (0…41.0 m expressed as a 32-bit single-precision floating point number using little-endian format (see Data Encoding on page 401)). For a 0xB0 command, this destination is the location of the station.
0xB1	Vehicle position order	
0xB7	Vehicle follow order	4-byte value that is the distance to the vehicle being followed (0…41.0 m expressed as a 32-bit single-precision floating point number using little-endian format (see Data Encoding on page 401)).

Current Target – The position that the vehicle has permission to move to (in meters) as measured from the upstream end of the current path. The target is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401). If the vehicle is stopped, it is equal to the position of the vehicle. If the vehicle is under a Vehicle Follow Order, this value is the Follow Distance (the distance (in meters) center-to-center that this vehicle maintains behind the vehicle being followed.

Followed Vehicle ID – The ID number of the vehicle that this vehicle is following when this vehicle is under a vehicle follow order and is part of a platoon (vehicle command type is 0xB7). Equal to 0 if this vehicle is not under a vehicle follow order (vehicle command type is 0x00, 0xB0, or 0xB1).

Time Since Last Report – The time in seconds since the last vehicle status for this vehicle was received from the motor. The time is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401).

Destination Station ID – The ID of the station the vehicle is commanded to when under a Move to Station command. Equal to 0 if the vehicle command type is a 0x00, 0xB1, or 0xB7. This field is only valid when the **Vehicle Command Type** field is 0xB0.

PID Set Index – The PID control loop set being used by the motor in control of the vehicle.

Bit	Description
0…3	**PID Loop Set Index:** 0 – Use user-defined PID set 0 – Unloaded PID values. 1 – Use user-defined PID set 1 – Loaded PID values. 2 – Use user-defined PID set 2. 3 – Use user-defined PID set 3. 4 – Use user-defined PID set 4. 5 – Use user-defined PID set 5. 6 – Use user-defined PID set 6. 7 – Use user-defined PID set 7. 8 – Use user-defined PID set 8. 9 – Use user-defined PID set 9. 10 – Use user-defined PID set 10. 11 – Use user-defined PID set 11. 12 – Use user-defined PID set 12. 13 – Use user-defined PID set 13. 14 – Use user-defined PID set 14. 15 – Use user-defined PID set 15 – Startup PID values. This PID set is automatically used during startup. If it is not defined, PID set 0 is scaled by 25% and used for startup.
4…7	Reserved

Ordered PID Set Index – The commanded PID Control Loop set for vehicle movement during the movement order when a movement order is in progress. This field is zero if the vehicle has not been given an order since startup.

Ordered Acceleration Limit – The last acceleration order from the host (in m/s^2) for the vehicle on the specified path. The acceleration is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401) when a movement order is in progress. This field is zero if the vehicle has not been given an order since startup.

Ordered Velocity Limit – The last velocity order from the host (in m/s) for the vehicle on the specified path. The velocity is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401) when a movement order is in progress. This field is zero if the vehicle has not been given an order since startup.

Active Move to Station Offset – Commanded offset from the destination station relative to the position of the station. The offset is expressed as a 32-bit single-precision floating-point number using little-endian format (see Data Encoding on page 401). Offsets downstream from a station are positive values, offsets upstream from a station are negative values.

NOTE: If the move to station was commanded using a Move Vehicle To Station (0xB0) command, this field returns the configured offset. If the move to station was commanded using an MMI_vehicle_station_order tag, this field always returns 0.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Get Extended Vehicle Status on page 208

Examples

See Get Extended Vehicle Status on page 208 for examples.

Node Controller Digital I/O Status

Message: 0xDF 0x07 0x01

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the digital I/O inputs and outputs for the node controllers that are specified in a Get Node Controller Digital I/O Status command.

NOTE: This response is also sent as an asynchronous message when there is a change in the digital I/O inputs for the node controller as specified by the Notification Mask.

When a Get Node Controller Digital I/O Status command is received for all node controllers, a separate status response is sent for each node controller.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	22
3	Message Type	1	0xDF
4	Extension Type	1	0x07
5	Extension Subtype	1	0x01
6	Command Status	1	0x00, 0x0B, 0x0C, 0x25, 0x41
7	Node Controller ID	2	1…96
9	Number of Digital I/O Inputs	1	0…32
10	Number of Digital I/O Outputs	1	0…32
11	Input Data	4	0x0…0xFFFFFFFF
15	Output Data	4	0x0…0xFFFFFFFF
19	Notification Mask	4	0x0…0xFFFFFFFF
23	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Node Controller Digital I/O extension message.

Extension Subtype – Fixed extension subtype that identifies this message as a Node Controller Digital I/O Status message.

Command Status – The status of the command that this Node Controller Digital I/O Status message is acknowledging (see HLC Command Status Codes).

Node Controller ID – The ID of the node controller for which digital I/O status is reported.

Number of Digital I/O Inputs – The number of digital I/O inputs available on the specified node controller.

Number of Digital I/O Outputs – The number of digital I/O outputs available on the specified node controller.

Input Data – The digital I/O input data for the specified node controller.

Output Data – The digital I/O output data for the specified node controller.

Notification Mask – The digital I/O input notification mask for the specified node controller.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Get Node Controller Digital I/O Status on page 211

Examples

See Get Node Controller Digital I/O Status on page 211 for examples.

MP Path End Status Report

Message: 0xDF 0x09 0x01

Source/Destination

HLC ► Host Controller

Purpose

Reports the path end status in a Moving Path node in response to an MP Get Path End Status command or to report a path end status change in an Entry or Exit Path of a Moving Path node. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When an MP Get Path End Status command is received, a separate status response is sent for each path end in the specified Moving Path node.

This response is also sent as an asynchronous message when **Send Node Status Asynchronously** is selected in the Node Controller Configuration File and any path end status change occurs in an Entry or Exit Path of a Moving Path. Enabling **Send Node Status Asynchronously** can result in many MP Node status messages being sent.

Support

This response is supported in the latest software release for the following product lines:

- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	35
3	Message Type	1	0xDF
4	Extension Type	1	0x09
5	Extension Subtype	1	0x01
6	Command Status	1	0x00, 0x0C, 0x0E, 0x19, 0x20
7	Node ID	2	1…65535
9	Path ID	2	1…65535
11	Host Command Count	4	0x0…0xFFFFFFFF

Offset	Item	Bytes	Range
15	Path End State	1	1…7
16	Path End Role	1	1…3
17	Path End Type	1	1, 2
18	Peer Node ID	2	0…65535
20	Requested Path ID	2	0…65535
22	Linked Path ID	2	0…65535
24	Last Allowed Vehicle ID	2	0…65535
26	Requesting Vehicle ID	2	0…65535
28	Last Entered Vehicle ID	2	0…65535
30	Owner Vehicle ID	2	0…65535
32	Last Exited Vehicle ID	2	0…65535
34	Alignment Request Count	2	0…65535
36	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Node extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an MP Path End Status Report extension message.

Command Status – The status of the command that this MP Path End Status Report message is acknowledging (see HLC Command Status Codes).

Node ID – The ID of the Moving Path node for which status is reported for this message.

Path ID – The ID of the path for which status is provided.

Host Command Count – A sequence count unique to the specified path end, last received in a MP Link Command or MP Unlink Command. The host controller maintains the Host Command Count.

Path End State – The state for this path end.

Table 4-17: Moving Path Node Path End States

Value	Name	Description
1	Unlinked	The path end is not linked. • This is the path end state for all paths in the node following the reset of any path that is connected to the node or a node controller restart. • Motor-to-motor message forwarding is disabled in this state. • Vehicles cannot navigate the node when a path end is in this state. • When the headway for a vehicle approaches a path end, the motor sends an entry request to the NC responsible for the node requesting permission to move the vehicle beyond the entry gate and navigate the node. When the node controller receives the entry request from a path end in the Unlinked state, it checks the routing that is needed for the vehicle to reach its destination, updates the *Requested Path ID* and *Requesting Vehicle ID* to notify the host controller of the alignment that is required, and changes the path end state to the Unlinked Alignment Requested state. The HLC sends the updated path end status to the host controller. The host controller responds with an MP Link Command when a path that provides a route to the vehicle's destination is aligned.
2	Linked Unlink Pending	The path end is linked with an unlink pending. • A linked junction remains linked allowing vehicles already navigating the junction to proceed. • Vehicles that request entry to the junction are not granted permission to breach the entry gates. • When all navigating vehicles are clear of the junction, the path end transitions to the Unlinked state.

Table 4-17: Moving Path Node Path End States (Continued)

Value	Name	Description
3	Unlinked Alignment Requested	The path end is not linked and a request to align it with another path is pending. • Motor-to-motor message forwarding i4s disabled in this state. • Vehicles cannot navigate the node when a path end is in this state. • A transition to this state indicates that a vehicle is requesting permission to navigate the node. • Entry requests are processed in this state to check that the requested alignment is still a valid route for the entering vehicle. • When an MP Link Command is received from the host controller the path end transitions to the Linked state.
4	Linked	The Control Path is linked to a Peer Path to form a path junction. • Motor-to-motor message forwarding is enabled in this state. • Vehicles can navigate the node from this path end. • In the Linked state, the node controller grants entry requests if the path junction offers an equivalent route to the requesting vehicle's destination. • If *Last Allowed Vehicle ID* is zero, entry requests are granted as long as the path junction offers a route to the vehicle's destination. • If *Last Allowed Vehicle ID* is set, the path junction continues to let vehicles in until the Last Allowed Vehicle ID enters. When the last vehicle enters, the path junction transitions to the Linked Unlink Pending state. • If the host controller sends an unlink command when the junction is in the Linked state and there are no vehicles navigating the junction, the junction is unlinked and both the Control Path and Peer Path ends transition to the Unlinked state. • If there is a vehicle navigating the junction, the path junction transitions to the Linked Unlink Pending state.
5	Linked Comm Loss	Communication to a linked path end is lost. • Path ends in the Linked or Linked Unlink Pending states enter this state when communication between the host controller and the HLC, or communication between the HLC and an NC is lost. • Motor-to-motor message forwarding is enabled in this state. • No additional vehicles are granted permission to navigate. • A link command or an unlink command is required once communication is restored to transition from this state.

Table 4-17: Moving Path Node Path End States (Continued)

Value	Name	Description
6	Linked Peer	The Peer Path is linked to a Control Path to form a path junction. • The Peer Path remains in this state until the path junction is unlinked.
7	Unlinked Vehicle Present	There is a vehicle that is located in an unlinked junction. • A vehicle was located during startup and extends past the entry gate on a path end in the Moving Path node. The host controller must either clear the vehicle by moving it away from the path end or link a path to move it through the junction. **NOTE:** When any Moving Path node path end is in this state, the Device Status field (bits 0-3 in the Node Status message) is set to "Junction Fault".

Path End Role – The role for this path end.

Value	Description
1	**Unlinked Path** – The path is not linked to another path.
2	**Control Path** – The path that is specified in an MP Link command to link to a Peer Path.
3	**Peer Path** – The path that is specified in an MP Link command to link from a Control Path.

Path End Type – The type function of path end.

Value	Description
1	**Fixed Path End** – A path end that is configured as a specific-route path.
2	**Moving Path End** – A path end that is configured as an equivalent-route path.

Peer Node ID – The ID of the node at the far end of this Moving Path node member path.

Value	Description
0	There is no Moving Path node at the far end of a moving path or if the member path is a fixed path (that is, configured as a specific-route path).
1…65535	The ID of the node at the far end of a moving path if the member path is a moving path (that is, configured as an equivalent-route path).

Requested Path ID – The Requested Path ID field is written when a path end transitions to the Unlinked Alignment Requested state. This state is a signal to the host controller that a vehicle is requesting permission to navigate a Moving Path node. The Requested Path ID persists on the Control Path from the time the Control Path enters the Unlinked Alignment Requested state until the Control Path is unlinked. The Requested Path ID field is cleared on the Peer Path when linked with a Control Path to form a junction. See Table 4-17 for detailed descriptions of the path end states.

Value	Description
0	The path end is not linked and the host controller can align any equivalent moving path to provide a route to the vehicle's destination.
1…65535	The requested path must be aligned to provide a route to the vehicle's destination.

Linked Path ID – The ID of the Peer Path that is linked to this Control Path.

Value	Description
0	No Peer Path is linked.
1…65535	The Control Path and specified Peer Path are linked to form a path junction.

Last Allowed Vehicle ID – The ID of the last vehicle that is allowed through the node. The HLC updates the Last Allowed Vehicle ID field only on the Control Path of a linked junction.

Value	Description
0	The HLC keeps the Control and Peer Paths linked until the host controller sends an MP Unlink Command. Vehicles approaching the junction are granted permission to navigate the junction if the Control Path offers an equivalent route to the vehicle's destination.
1…65535	Once the last allowed vehicle begins to navigate the path junction, the Control Path end status transitions to the Linked Unlink Pending state and no other vehicles are granted permission to navigate the junction. The HLC unlinks the path junction as soon as all navigating vehicles are clear of the path junction.

Requesting Vehicle ID – The ID of the vehicle that is requesting permission to navigate the specified Moving Path node.

When in the Unlinked Alignment Requested state, the Requesting Vehicle ID field identifies the vehicle that is requesting permission to navigate the specified Moving Path node. The Requesting Vehicle ID field is cleared once the path end is linked and the vehicle is granted permission to navigate the junction.

Last Entered Vehicle ID – The ID of the last vehicle that was granted permission to navigate the junction. Updated by the HLC only on the Control Path of a linked junction.

Value	Description
0	No active vehicle. Set to zero before any vehicle is granted permission and is cleared when the last entered vehicle clears the junction.
1…65535	ID of vehicle requesting entry or the ID of the last vehicle through the entry gate. Updated by the HLC only on the Control Path to the entering vehicle ID when that vehicle is granted permission to enter.

Owner Vehicle ID – The ID of the vehicle that currently owns the path junction. Updated by the HLC only on the Control Path of a linked junction.

Value	Description
0	There are no vehicles navigating the path junction.
1…65535	The ID of the vehicle that currently owns the path junction.

Last Exited Vehicle ID – The ID of the most recent vehicle to clear the path junction. Updated by the HLC only on the Control Path of a linked junction.

Value	Description
0	No vehicle has cleared the junction since it was linked
1…65535	ID of the most recent vehicle to clear the junction

Alignment Request Count – A sequence count, unique to the most recent alignment request, which the HLC increments when an MP Alignment Request is sent.

The HLC maintains the Alignment Request Count for each path end. It is initialized to 0 when the HLC restarts and no alignment requests have been issued, it is incremented for each new alignment request, and continues from 1 when it rolls over.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

MP Get Path End Status on page 219
Node Status on page 275

Examples

See MP Get Path End Status on page 219 for examples.

MP Alignment Request

Message: 0xDF 0x09 0x02

Source/Destination

HLC ► Host Controller

Purpose

The HLC sends an MP Alignment Request message to the host controller to align a path for a vehicle that is requesting a route to a destination beyond the specified path junction.

The host controller responds with an MP Link Command once the requested path ends are properly aligned.

Support

This response is supported in the latest software release for the following product lines:

- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	15
3	Message Type	1	0xDF
4	Extension Type	1	0x09
5	Extension Subtype	1	0x02
6	Node ID	2	1…65535
8	Control Path ID	2	1…65535
10	Requested Path ID	2	0…65535
12	Vehicle ID	2	1…65535
14	Alignment Request Count	2	1…65535
16	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Node extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an MP Alignment Request extension message.

Node ID – The ID of the Moving Path node for which this path end alignment request applies.

Control Path ID – The ID of the path where a vehicle is requesting permission to navigate the specified Moving Path node.

Requested Path ID – The Requested Path ID field is a signal to the host controller that a vehicle is requesting permission to navigate a Moving Path node.

Value	Description
0	The host controller can align any path that offers an equivalent path route to the vehicle's destination.
1…65535	The host controller must align the requested path to the Control Path to satisfy the vehicle's route to its destination.

The Requested Path ID persists on a Control Path from the time the Control Path enters the Unlinked Alignment Requested state until the Control Path is unlinked. The Requested Path ID field is cleared on the Peer Path when linked with a Control Path to form a junction.

See Table 4-17, *Moving Path Node Path End States, on page 319* for detailed descriptions of the path end states.

Vehicle ID – The ID of the vehicle to navigate the path junction once it is properly aligned.

Alignment Request Count – A sequence count, unique to the most recent alignment request, which the HLC increments when an alignment request is sent.

The HLC maintains the Alignment Request Count for each path end. It is initialized to 0 when the HLC restarts and no alignment requests have been issued, it is incremented for each new alignment request, and continues from 1 when it rolls over.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

MP Link Command on page 222

Examples

See MP Path End Status Report on page 317 for examples.

SM Subscription Response

Message: 0xDF 0x0A 0x01

Source/Destination

HLC ➛ Host Controller

Purpose

Reports the status of an SM Subscription Command command to signal command completion or command failure.

Support

This response is supported in the latest software release for the following product lines:

- QuickStick transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	24
3	Message Type	1	0xDF
4	Extension Type	1	0x0A
5	Extension Subtype	1	0x01
6	Command Status	1	0x00, 0x03, 0x09, 0x0B,0x0C, 0x0D, 0x0E, 0x10, 0x16, 0x26, 0x27, 0x28, 0x29, 0x41, 0x42
7	Component Type	8	Varies with each type
15	Host Context	2	0x0000…0xFFFF
17	Command Options	2	0x0000…0x0001
19	Subscription Interval	2	0…65535
21	Metric ID	2	0x0000…0xFFFF
23	Metric Instance	2	0…65535
25	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as an Information Service extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an SM Subscription Response extension message.

Command Status – The status of the command that this SM Subscription Response message is acknowledging (see Table 4-10, *HLC Command Status Codes, on page 263*).

Component Type – The ID of the component targeted for system monitoring (see Table 4-3 on page 230).

Host Context – Unique value supplied by the host controller when the SM Subscription Command message was sent. It aids the host controller in tracking the execution of this command.

Command Options – The **Command Option**s specified by the host controller when the SM Subscription Command message was sent.

Subscription Interval – The **Subscription Interval s**pecified when the SM Subscription Command message was sent (see Table 4-4 on page 234 for common system monitoring intervals).

Metric ID – The **Metric ID** specified when the SM Subscription Command message was sent (see Table 4-5).

Metric Instance – The metric instance specified when the SM Subscription Command message was sent.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

SM Subscription Command on page 229

Examples

See SM Subscription Command on page 229 for examples.

SM Subscription Data Response

Message: 0xDF 0x0A 0x02

Source/Destination

HLC ► Host Controller

Purpose

Reports the data that is requested in an SM Subscription Command message.

Support

This response is supported in the latest software release for the following product lines:

- QuickStick transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	Varies
3	Message Type	1	0xDF
4	Extension Type	1	0x0A
5	Extension Subtype	1	0x02
6	Status	1	0x00, 0x03, 0x0E, 0x26, 0x41
7	Component Type	8	Varies with each type
15	Host Context	2	0x0000…0xFFFF
17	Command Options	2	0x0000, 0x0001
19	Sequence Number	2	0…65535
21	Metric ID	2	0x0001…0xFFFF
23	Metric Instance	2	0…65535
25	Metric Data Element Size	1	0, 1, 2, 4, 8
26	Metric Data Element Count	1	0, 1, 4
27	Metric Data	Varies	Depends on targeted metric data instance
Varies	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as an Info Serv extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an SM Subscription Data Response extension message.

Status – The status for this SM Subscription Data Response message (see Table 4-10, *HLC Command Status Codes, on page 263*).

Component Type – The ID of the component targeted for system monitoring (see Table 4-3 on page 230).

Host Context – Unique value supplied by the host controller when the SM Subscription Command message was sent. It aids the host controller in tracking the execution of this command.

Command Options – The **Command Options** specified by the host controller when the SM Subscription Command message was sent.

Sequence Number – A sequence number, unique to each subscription, which is incremented by the targeted component whenever new metric data is reported to the HLC. It is initialized to zero when a subscription for a metric is established, is incremented for each new metric update, and continues from zero when it rolls over.

Metric ID – The **Metric ID** specified when the SM Subscription Command message was sent (see Table 4-5).

Metric Instance – The metric instance specified when the SM Subscription Command message was sent

Metric Data Element Size – The size of the data elements in the metric instance data reported by the targeted component.

Metric Data Element Count – The count of the data elements in the metric instance data reported by the targeted component.

Metric Data – Metric instance data reported from the specified component (see Table 4-5).

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

SM Subscription Command on page 229

Examples

See SM Subscription Command on page 229 for examples.

SM Poll Data Response

Message: 0xDF 0x0A 0x03

Source/Destination

HLC ► Host Controller

Purpose

Reports metric data that is requested in an SM Poll Command.

Support

This response is supported in the latest software release for the following product lines:

- QuickStick transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	Varies
3	Message Type	1	0xDF
4	Extension Type	1	0x0A
5	Extension Subtype	1	0x31
6	Command Status	1	0x00, 0x03, 0x09, 0x0B, 0x0C, 0x0D, 0x0E, 0x10, 0x16, 0x26, 0x28, 0x41, 0x42
7	Component Type	8	Varies with each type
15	Host Context	2	0x0001…0xFFFF
17	Command Options	2	0x0000, 0x0001
19	Reserved	2	Must be zero
21	Metric ID	2	0x0001…0xFFFF
23	Metric Instance	2	0…65535
25	Metric Data Element Size	1	0, 1, 2, 4, 8
26	Metric Data Element Count	1	0, 1, 4
27	Metric Data	Varies	Depends on targeted metric data instance
Varies	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as an Info Serv extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an SM Poll Data Response extension message.

Command Status – The status of the command that this SM Poll Data Response message is acknowledging (see Table 4-10, *HLC Command Status Codes, on page 263*).

Component Type – The ID of the component targeted for system monitoring (see Table 4-3 on page 230).

Host Context – Unique value supplied by the host controller when the SM Poll Command message was sent. It aids the host controller in tracking the execution of this command.

Command Options – The **Command Options** specified by the host controller when the SM Poll Command message was sent.

Metric ID – The **Metric ID** specified when the SM Poll Command message was sent (see Table 4-5).

Metric Instance – Unique metric instance specified when the SM Poll Command message was sent.

Metric Data Element Size – The size of the data elements in the metric instance data reported by the targeted component.

Metric Data Element Count – The count of the data elements in the metric instance data reported by the targeted component.

Metric Data – Metric instance data reported from the specified component (see Table 4-5).

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

SM Poll Command on page 238

Examples

See SM Poll Command on page 238 for examples.

Extended Node Controller Status

Message: 0xDF 0x0C 0x03

Source/Destination

HLC ► Host Controller

Purpose

Reports the extended status of the node controllers that are specified in a Get Extended Node Controller Status command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Get Extended Node Controller Status command is received for all node controllers, a separate status response is sent for each node controller.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	14
3	Message Type	1	0xDF
4	Extension Type	1	0x0C
5	Extension Subtype	1	0x03
6	Command Status	1	0x00, 0x0C, 0x0E
7	Node Controller ID	2	1…65535
9	State	1	1…3
10	Software Major Version	1	0…255
11	Software Minor Version	1	0…255
12	Software Patch Version	1	0…255
13	Configuration ID	1	0…20
14	Configuration Valid	1	0, 1
15	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Remote Management extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an Extended Node Controller Status extension message.

Command Status – The status of the command that this Extended Node Controller Status message is acknowledging (see HLC Command Status Codes).

Node Controller ID – The ID of the node controller that status is provided for.

State – Indicates the operational state of the node controller.

Value	Description
0x01	**Initialization** – Loading the Node Controller Configuration File or an error was detected in the configuration preventing the node controller from exiting this state. Consult the node controller log for additional details when a node controller does not exit this state.
0x02	**Disconnected** – The TCP/IP connection from the high-level controller to this node controller is down.
0x03	**Operational** – The high-level controller connection to this node controller is established and the node controller is operational.

Software Major Version – The node controller software major version number.

Software Minor Version – The node controller software minor version number.

Software Patch Version – The node controller software patch version number.

Configuration ID – The ID of the active Node Controller Configuration File for this node controller.

Value	Description
0	Using a static Node Controller Configuration File.
1…20	Using the specified managed Node Controller Configuration File.

Configuration Valid – This field indicates if the node controller's configuration is valid.

Value	Description
0	**Configuration not valid** – There are errors in the Node Controller Configuration File.
1	**Configuration valid** – The Node Controller Configuration File is defined correctly.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Status Request on page 152
Set Node Controller Configuration on page 245
Restart Node Controllers on page 247
Get Extended Node Controller Status on page 249
Get Extended High-Level Controller Status on page 252
High-Level Controller Status on page 271
Node Controller Status on page 273
Extended High-Level Controller Status on page 335

Examples

See Get Extended Node Controller Status on page 249 for examples.

Extended High-Level Controller Status

Message: 0xDF 0x0C 0x04

Source/Destination

HLC ► Host Controller

Purpose

Reports the extended status of the high-level controller as specified in a Get Extended High-Level Controller Status command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

Support

This response is supported in the latest software release for the following product lines:

- MagneMover LITE transport systems.
- QuickStick transport systems.
- QuickStick HT transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	12
3	Message Type	1	0xDF
4	Extension Type	1	0x0C
5	Extension Subtype	1	0x04
6	State	1	1…3
7	Software Major Version	1	0…255
8	Software Minor Version	1	0…255
9	Software Patch Version	1	0…255
10	Configuration ID	1	0…20
11	Configuration Valid	1	0, 1
12	EtherNet/IP Communication Status	1	0…2
13	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Remote Management extension message.

Extension Subtype – Fixed extension subtype that identifies this message as an Extended High-Level Controller Status extension message.

State – This field indicates the operational state of the high-level controller.

Value	Description
1	**Initialization** – Loading the Node Controller Configuration File or an error was detected in the configuration preventing the node controller from exiting this state. Consult the high-level controller log for additional details when the HLC does not exit this state.
2	**Degraded** – The high-level controller is unable to communicate with one or more node controllers in the transport system. See the high-level controller log for additional details.
3	**Operational** – The high-level controller configuration is valid and successfully communicating with all node controllers that are configured in the transport system.

Software Major Version – The HLC software major version number.

Software Minor Version – The HLC software minor version number.

Software Patch Version – The HLC software patch version number.

Configuration ID – The ID of the active Node Controller Configuration File for this node controller.

Value	Description
0	Using a static Node Controller Configuration File.
1…20	Using the specified managed Node Controller Configuration File.

Configuration Valid – This field indicates if the node controller's configuration is valid.

Value	Description
0	**Configuration not valid** – There are errors in the Node Controller Configuration File.
1	**Configuration valid** – The Node Controller Configuration File is defined correctly.

EtherNet/IP Communication Status – Indicates the status of communication via Ethernet Industrial Protocol (EtherNet/IP) with a PLC if the high-level controller is configured for communication with a PLC using EtherNet/IP.

Value	Description
0	**Not configured** – EtherNet/IP control is not configured.
1	**Link up** – EtherNet/IP communication link is UP, which indicates the high-level controller is connected to the PLC at the configured IP address.
2	**Link down** – EtherNet/IP communication link is DOWN, which indicates the high-level controller is unable to contact the PLC at the configured IP address or connectivity has been lost.

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Status Request on page 152
Set Node Controller Configuration on page 245
Restart Node Controllers on page 247
Get Extended Node Controller Status on page 249
Get Extended High-Level Controller Status on page 252
High-Level Controller Status on page 271
Node Controller Status on page 273
Extended Node Controller Status on page 332

Examples

See Get Extended High-Level Controller Status on page 252 for examples.

Propulsion Power Supply Status

Message: 0xDF 0x0D 0x01

Source/Destination

HLC ► Host Controller

Purpose

Reports the status of the propulsion power supplies that are specified in a Get Propulsion Power Status command. See Chapter 5, Troubleshooting for details on troubleshooting the status messages.

NOTE: When a Get Propulsion Power Status command is received for all propulsion power supplies, a separate status response is sent for each power supply.

Support

This response is supported in the latest software release for the following product lines:

- QuickStick HT 5700 transport systems.

Format

Offset	Item	Bytes	Range
0	Message Header	2	0xABBA
2	Message Length	1	10
3	Message Type	1	0xDF
4	Extension Type	1	0x0D
5	Extension Subtype	1	0x01
6	Command Status	1	0x00, 0x0B, 0x0C, 0x0E
7	Supply ID	2	1…32
9	Type	1	1
10	State	1	0x00…0xFF
11	Message CRC	2	0…65535

Data Fields

Message Header – Constant byte pattern designating the start of a message.

Message Length – Length of this message, given as the number of bytes following the Message Length field.

Message Type – Fixed message type that identifies this message as a Host Extension message.

Extension Type – Fixed extension type that identifies this message as a Propulsion Power extension message.

Extension Subtype – Fixed extension subtype that identifies this message as a Propulsion Power Status extension message.

Command Status – The status of the command that this Propulsion Power Status message is acknowledging (see HLC Command Status Codes).

Supply ID – The ID of the propulsion power supply for which status is being requested.

Type – The type of the specified propulsion power supply.

Value	Description
1	Diode Front End (DFE) propulsion power supply

State – The state of the propulsion power supply. (see Table 4-7 on page 257).

Message CRC – A check value on the entire message (excluding the CRC) to verify integrity.

See Also

Get Propulsion Power Status on page 254

Examples

See Get Propulsion Power Status on page 254 for examples.

This page intentionally left blank.

Troubleshooting 5

Overview

This chapter describes the common difficulties that may be encountered when using the MagneMotion® Host Controller TCP/IP Communication Protocol. Included in this chapter are:

- Initial troubleshooting.
- Motor fault troubleshooting.

For assistance, see Rockwell Automation Support.

TCP/IP Communications Troubleshooting

This section describes the common difficulties that may be encountered with the Host Controller TCP/IP Communication Protocol, and general solutions.

Table 5-1: Initial Host Controller Communications Troubleshooting

Symptom	Problem	Solution
The host controller does not connect to the high-level controller even though it did previously.	The IP address of the HLC has been changed.	Check the IP setting through the node controller web interface or the console interface to verify the address.
	The network is not connected or there are network problems.	Verify all network connections.
Vehicles do not move to the expected positions.	The vehicle command message is configured incorrectly.	Verify that the distance to the positions is measured from the start of the path.
		Verify that the correct path is specified.

Table 5-1: Initial Host Controller Communications Troubleshooting (Continued)

Symptom	Problem	Solution
Vehicles do not cross a Gateway Node.	Path in the target system is suspended.	Issue a Resume command to the path.
		Wait for Interlock or E-stop on the path to be cleared.
	Duplicate Vehicle ID exists in the target system.	Verify Min vehicle ID = 1 and Max vehicle ID = 65535 in the Global Settings for all Control Groups.
	The Gateway Node is disabled.	Enable the Gateway Node (see Set Node Parameters on page 177).
Vehicles do not cross a Terminus Node.	Handshaking is not properly implemented.	Verify entry/exit handshaking is structured correctly.
Commands that are sent to the HLC cause the connection to the HLC to be closed.	The command is not formatted properly.	Verify that the CRC is calculated correctly (see Error Checking on page 400) and the message and framing are correct.
Vehicles do not move.	Propulsion power is disabled.	Turn on the propulsion power.
	A physical obstruction to vehicle motion.	Verify that vehicle motion is not obstructed.
	The vehicle is binding on the guideway.	Verify that the vehicle can move freely.
	A merge or diverge node is not responding or functioning properly.	Verify that the merge or diverge node is configured correctly and the switch hardware is operational.

Motor Fault Troubleshooting

Table 5-2 describes the common motor faults and general solutions. All motor faults are identified through the Motor Status response. Motor faults are provided in the tables that follow:

- Table 5-3 provides detailed descriptions and solutions for all MagneMover® LITE
- Table 5-4 provides detailed descriptions and solutions for all QuickStick® 100
- Table 5-5 provides detailed descriptions and solutions for all QuickStick 150
- Table 5-6 provides detailed descriptions and solutions for all QSHT and QSHT 5700

Table 5-2: Initial Motor Fault Troubleshooting

Fault	Problem	Solution
The upstream connection for the motor is reported as inoperative.	NC cannot connect to the upstream end of the motor.	Check the communication connection between the motor and the previous motor or NC.
The downstream connection for the motor is reported as inoperative.	NC cannot connect to the downstream end of the motor.	Check the communication connection between the motor and the next motor or NC.
		Check the power being supplied to the motor downstream.
	No motor downstream, but the configuration indicates that there is a motor.	Correct the Node Controller Configuration File.
Soft Start not complete.	Propulsion power insufficient to activate hot-plug protection circuits.	Check the power being supplied to the motor.
Motor reports the node controller has suspended motion.	NC has issued a Suspend command to the motor.	Issue a Reset command.
		Clear the E-stop or Interlock.
Motor does not respond.	NC cannot communicate with the motor.	Check the communication connection.
		Check the power being supplied to the motor.
Motor reports “Not in operational mode”.	All motors currently enter this state for 100 ms, and then automatically exit. This wait lets sampled A/D inputs and observers settle before using this data. This fault is informational only, there is no lockout of motor operation.	Wait 100 ms after reset or power on before sending any commands to the motor.
Unexpected Slave Module reset.	A slave board in the motor reset without being commanded to reset.	Check the power being supplied to the motor.
		Verify the motor and the vehicle are properly grounded.

Table 5-2: Initial Motor Fault Troubleshooting (Continued)

Fault	Problem	Solution
Motor reports the Slave Module is not responding.	A slave board in the motor is not responding to commands or queries from the motor controller.	Verify that the motor is programmed before commanding it.
		Verify the motor and the vehicle are properly grounded.
		Return the motor for service if this fault continues.
Over-current fault.	Too much power being pulled through the motor or block.	Check all motor power wiring. Reset the motor to clear the fault. If the problem persists, contact Technical Support.
		See the following motor faults: MML™: Over-current fault Gen 2 on page 351. QS 100: Over-current fault on page 361. QS 150: Over-current fault on page 367 QSHT: Software over-current fault (QSHT + Gen 2 Motor Controller) on page 383 and Hardware over-current fault on page 384.

Table 5-2: Initial Motor Fault Troubleshooting (Continued)

Fault	Problem	Solution
Under-voltage fault.	Not enough power being supplied to the motor.	Check the power being supplied to the motor.
		See the following motor faults: MML: Under-voltage fault on page 351 and Under-voltage fault on page 352. QS 100: Under-voltage fault on page 361. QS 150: Under-voltage fault on page 367 QSHT: Under-voltage fault (QSHT + Gen 2 Motor Controller) on page 385 and Under-voltage fault (QSHT + 5700 Motor Controller) on page 385.
Over-voltage fault.	Too much power (wrong voltage) being supplied to motor.	Check the power being supplied to the motor.
		See the following motor faults: MML: Over-voltage fault on page 351 and Over-voltage fault on page 353. QS 100: Over-voltage fault on page 361. QS 150: Over-voltage fault on page 368 QSHT: Over-voltage fault (QSHT + Gen 2 Motor Controller) on page 385 and Over-voltage fault (QSHT + 5700 Motor Controller) on page 385.

Table 5-2: Initial Motor Fault Troubleshooting (Continued)

Fault	Problem	Solution
Over-temperature fault.	Motor controller temperature has exceeded limits.	Verify that there is nothing preventing vehicle motion on the motor.
		Verify that the temperature of the operating environment is within specifications.
		Verify that the demand rms current on the motor does not exceed motor limitations.
		See the following motor faults: MML: Over-temperature fault on page 351 and Over-temperature fault on page 353. QS 100: Over-temperature fault on page 362. QS 150: Over-temperature fault on page 368 QSHT: Inverter over-temperature fault on page 386.
Estimated coil temperature too high.	Coil temperature estimate that is based on current draw has exceeded limits.	Verify that there is nothing preventing vehicle motion on the motor.
		Verify that the temperature of the operating environment is within specifications.
		See the following motor faults: MML: Estimated coil temperature too high on page 352.

Table 5-3 provides detailed fault definitions, set and clear conditions, and user actions for the MagneMover LITE motor faults reported in Table 4-12,MagneMover LITE Motor Fault Data, on page 291. The Motor Suspend Fault field shows when motion on the motor is suspended. Motion cannot resume until all faults causing a Motor Suspend are cleared. Vehicles are not allowed to enter the section of the path where the motor is located while it is suspended.

Table 5-3: MagneMover LITE Motor Fault Troubleshooting

OS Scheduler Faults	
Scheduler not initiated	
Definition	Scheduler didn't finish initialization.
Set condition	Scheduler was requested to schedule a nonexistent function.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes
Scheduler event queue full	
Definition	Software fault condition that is used for reporting when the scheduler event queue in the motor controller is full.
Set condition	The motor controller detects that the scheduler event queue is full.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes
Upstream and Downstream Comm Faults	
Connection inoperative	
Definition	The motor controller sends periodic ping requests and expects ping responses to determine the health of the communication link.
Set condition	The motor controller did not receive a ping response after sending four consecutive ping requests.
Clear condition	The motor controller received a ping response after sending a ping request.
User action	Check communication link cabling and/or power cabling to adjacent motors.
Motor Suspend Fault	Yes

Table 5-3: MagneMover LITE Motor Fault Troubleshooting (Continued)

Miscellaneous comm warning	
Definition	This fault has been deprecated in the firmware and is no longer applicable.
Set condition	
Clear condition	
User action	
Motor Suspend Fault	
Transmit buffer full	
Definition	The communication transmit buffer in the motor controller is full and cannot send the current message. This message is diagnostic only, no action takes place.
Set condition	The motor controller cannot send a message because the transmit buffer is full.
Clear condition	The motor controller is able to send a message.
User action	• Check to see if a Virtual Scope data stream has been started but not stopped. • Verify that the path length does not exceed the limit that is shown in the MagneMotion Node Controller Interface User Manual, publication MMI-UM001.
Motor Suspend Fault	No
UART settings corrected	
Definition	The UART settings are not correct. **NOTE**: Only used for serial type motors.
Set condition	External UART settings are not correct.
Clear condition	Cleared on reset.
User action	None.
Motor Suspend Fault	No
Motor Overall Faults	
Motor not in operational mode	
Definition	Initial state of motor when it comes out of reset.
Set condition	Reset complete.
Clear condition	Cleared after 100 ms.
User action	None. Fault clears automatically.
Motor Suspend Fault	Yes

Table 5-3: MagneMover LITE Motor Fault Troubleshooting (Continued)

Motor in configuration mode	
Definition	Master board is being configured.
Set condition	Node Controller sets the motor in configuration mode.
Clear condition	Node Controller clears motor from configuration mode.
User action	None
Motor Suspend Fault	Yes
Motor in diagnostic mode	
Definition	Master board is in a diagnostic mode.
Set condition	Node Controller sets the motor in diagnostic mode.
Clear condition	Node Controller clears the motor from diagnostic mode.
User action	None - Could be user driven
Motor Suspend Fault	Yes
Motion suspended by node controller	
Definition	A node controller issued a suspend command to the motor. This message can be in response to a suspend command or a digital I/O based interlock or emergency stop.
Set condition	Suspend command is sent from a node controller.
Clear condition	Resume command is sent from a node controller.
User action	Not Applicable
Motor Suspend Fault	Yes
Motor not responding	
Definition	A node controller is not able to communicate with a motor.
Set condition	No status report is received from a motor for 300 ms.
Clear condition	Status report is received from a non-responsive motor.
User action	• Check communication connections between the node controller and the first motor in the path that is not responding. • Check that power is being supplied to the first motor in the path that is not responding.
Motor Suspend Fault	Yes

Table 5-3: MagneMover LITE Motor Fault Troubleshooting (Continued)

Master Board Faults A

Unlocated vehicle fault

Definition	The motor detects that there are blocks with a magnet array present (detects field array) but no Vehicle ID.
Set condition	A block detects a magnet array with no Vehicle ID assigned.
Clear condition	All blocks that detect a magnet array have an associated Vehicle ID.
User action	Start up the path to locate the vehicle.
Motor Suspend Fault	No

Switch position fault (Gen 3)

Definition	Switch flipper is not in its expected position. **NOTE:** Used for MagneMover LITE Gen 3 switches only.
Set condition	No sensor active detected for 30 ms after expected arrival.
Clear condition	Once the switch automatically re-homes.
User action	None
Motor Suspend Fault	No

Propulsion power not ready

Definition	The Soft Start circuit that limits inrush current upon application of the DC bus has not turned on. If the Soft Start circuit does not turn on, the motor is in suspend mode.
Set condition	At initial power-up, until the DC Input bus is higher than +32V DC and the difference between the DC Input bus and the downstream load (through the Soft Start resistor) is less than 1V DC. Once cleared, it is not set again until the DC Input bus is lower than +27V DC. Repeatedly toggling the DC bus on/off causes the Soft Start PTC resistor to heat up and open.
Clear condition	Input DC bus is greater than +32V DC and the difference between the DC input bus and the downstream load is less than 1V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • Remove power and wait 60 seconds before reapplying power (limit cycling of the propulsion power to three times or less per minute to allow the PTC to cool-down).
Motor Suspend Fault	Yes

Table 5-3: MagneMover LITE Motor Fault Troubleshooting (Continued)

Under-voltage fault	
Definition	Propulsion power to a block is lower than +27V DC.
Set condition	Propulsion power falls lower than +27V DC.
Clear condition	Propulsion power rises higher than +28V DC (note hysteresis).
User action	Verify that propulsion power is being provided to the motor at the correct voltage.
Motor Suspend Fault	Yes
Over-voltage fault	
Definition	Propulsion power to a block is higher than +42V DC.
Set condition	Propulsion power rises higher than +42V DC.
Clear condition	Propulsion power falls lower than +38V DC (note hysteresis).
User action	Verify that propulsion power is being provided to the motor at the correct voltage.
Motor Suspend Fault	Yes
Over-temperature fault	
Definition	The temperature of the motor controller for the motor has exceeded safe limits for the hardware and the motor has shut down to prevent thermal damage.
Set condition	Ambient internal temperature exceeds 80 °C.
Clear condition	Ambient internal temperature is lower than 75 °C (note hysteresis).
User action	• Verify that there is nothing preventing vehicle motion on the motor. • Verify that the temperature of the operating environment is within specifications. • Verify that the demanded rms current on the motor does not exceed motor limitations.
Motor Suspend Fault	Yes
Over-current fault Gen 2	
Definition	The motor controller has detected an over-current condition. **NOTE:** Used for MagneMover LITE Gen 2 motors only.
Set condition	Filtered input current is greater than 10 A.
Clear condition	Filtered input current is less than 9 A (note hysteresis).
User action	None
Motor Suspend Fault	Yes

Table 5-3: MagneMover LITE Motor Fault Troubleshooting (Continued)

Configuration fault	
Definition	The number of blocks that are configured is less than 1 or greater than 60.
Set condition	Number of blocks is configured to be an invalid number.
Clear condition	Number of blocks is configured to be within valid limits.
User action	• Verify that the correct MagneMover LITE Motor Type Files are loaded onto the NC. • Restart services on the NC and reset the motors to load the new Node Controller Configuration Files.
Motor Suspend Fault	Yes
Master Board Faults B	
Reserved	
Driver Board 1…8 Faults	
Estimated coil temperature too high	
Definition	The motor controller has a thermal model of each coil in the motor. If the model exceeds its temperature threshold, the coil is shut off.
Set condition	The estimated coil temperature exceeds 100 °C.
Clear condition	The estimated coil temperature falls lower than 100 °C.
User action	• This fault may be due to a stalled vehicle or mechanical binding on the track. Check to see if these conditions exist. • This fault may arise from areas of the track with high duty cycles, such as start/stop areas with little dwell and high acceleration. These areas are application-dependent.
Motor Suspend Fault	No
Under-voltage fault	
Definition	Propulsion power to the driver board is lower than +27V DC.
Set condition	Propulsion power falls lower than +27V DC.
Clear condition	Propulsion power rises higher than +28V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one driver board, contact Technical Support.
Motor Suspend Fault	No

Table 5-3: MagneMover LITE Motor Fault Troubleshooting (Continued)

Over-voltage fault	
Definition	Propulsion power to the driver board is higher than +42V DC.
Set condition	Propulsion power rises higher than +42V DC.
Clear condition	Propulsion power falls lower than +38V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one driver board, contact Technical Support.
Motor Suspend Fault	Yes
Over-temperature fault	
Definition	Coil board temperature exceeds a threshold. The coil board temperature limits are higher than the motor controller temperature limits due to proximity of the coils.
Set condition	Coil board temperature exceeds 90 °C.
Clear condition	Coil board temperature is lower than 85 °C (note hysteresis).
User action	• Verify that there is nothing preventing vehicle motion on the motor. • Verify that the temperature of the operating environment is within specifications. • Verify that the demanded rms current on the motor does not exceed motor limitations.
Motor Suspend Fault	Yes
Not responding (Gen 4)	
Definition	The motor controller in the motor is unable to communicate with this driver board.
Set condition	Main board in the motor does not get a response from the driver board within 10 ms.
Clear condition	Main board in the motor receives a new update from the driver board.
User action	Make sure the driver board is programmed. Reset the motor. • If the fault clears on reset, make sure that the motor is not receiving a static shock from the vehicle or an external source. • If the fault does not clear on reset, contact Technical Support and replace the motor.
Motor Suspend Fault	Yes

Table 5-3: MagneMover LITE Motor Fault Troubleshooting (Continued)

Switch unable to complete move	
Definition	A standard or high payload switch flipper is commanded to move and after some retries, it does not arrive at its position (the stepper moved but the encoder is not updating).
Set condition	Flipper stepper is moving but encoder is not responding.
Clear condition	Encoder responds.
User action	Make sure propulsion power is applied so the flipper can move. If the fault does not clear on reset, contact Technical Support and replace the motor.
Motor Suspend Fault	Yes
Switch mechanism out of position	
Definition	The switch flipper on a standard or high payload switch is at its commanded position, but is outside its tolerance window.
Set condition	Flipper is moved out of its tolerance window.
Clear condition	Flipper is within tolerance window.
User action	Make sure flipper is not being moved out of position externally. Make sure propulsion power is applied so the flipper can move. If the fault does not clear on reset, contact Technical Support and replace the motor.
Motor Suspend Fault	Yes

Table 5-4 provides detailed fault definitions, set and clear conditions, and user actions for the QuickStick 100 motor faults reported in Table 4-13,QuickStick 100 Motor Fault Data, on page 293. The Motor Suspend Fault field shows when motion on the motor is suspended. Motion cannot resume until all faults causing a Motor Suspend are cleared. Vehicles are not allowed to enter the section of the path where the motor is located while it is suspended.

Table 5-4: QuickStick 100 Motor Fault Troubleshooting

OS Scheduler Faults	
Scheduler not initiated	
Definition	Scheduler didn't finish initialization.
Set condition	Scheduler was requested to schedule a nonexistent function.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes
Scheduler event queue full	
Definition	Software fault condition that is used for reporting when the scheduler event queue in the motor controller is full.
Set condition	The motor controller detects that the scheduler event queue is full.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes
Upstream and Downstream Comm Faults	
Connection inoperative	
Definition	The motor controller sends periodic ping requests and expects ping responses to determine the health of the communication link.
Set condition	The motor controller did not receive a ping response after sending four consecutive ping requests.
Clear condition	The motor controller received a ping response after sending a ping request.
User action	Check communication link cabling and/or power cabling to adjacent motors.
Motor Suspend Fault	Yes

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Miscellaneous comm warning	
Definition Set condition Clear condition User action Motor Suspend Fault	This fault has been deprecated in the firmware and is no longer applicable.
Transmit buffer full	
Definition	The communication transmit buffer in the motor controller is full and cannot send the current message. This message is diagnostic only, no action takes place.
Set condition	The motor controller cannot send a message because the transmit buffer is full.
Clear condition	The motor controller is able to send a message.
User action	• Check to see if a Virtual Scope data stream has been started but not stopped. • Verify that the path length does not exceed the limit that is shown in the MagneMotion Node Controller Interface User Manual, publication MMI-UM001.
Motor Suspend Fault	No
UART settings corrected	
Definition	The UART settings are not correct. **NOTE**: Only used for serial type motors.
Set condition	External UART settings are not correct.
Clear condition	Cleared on reset.
User action	None.
Motor Suspend Fault	No
Motor Overall Faults	
Motor not in operational mode	
Definition	Initial state of motor when it comes out of reset.
Set condition	Reset complete.
Clear condition	Cleared after 100 ms.
User action	None. Fault clears automatically.
Motor Suspend Fault	Yes

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Motor in configuration mode	
Definition	Master board is being configured.
Set condition	Node Controller sets the motor in configuration mode.
Clear condition	Node Controller clears motor from configuration mode.
User action	None
Motor Suspend Fault	Yes
Motor in diagnostic mode	
Definition	Master board is in a diagnostic mode.
Set condition	Node Controller sets the motor in diagnostic mode.
Clear condition	Node Controller clears the motor from diagnostic mode.
User action	None - Could be user driven.
Motor Suspend Fault	Yes
Motion suspended by node controller	
Definition	A node controller issued a suspend command to the motor. This message can be in response to a suspend command or a digital I/O based interlock or emergency stop.
Set condition	Suspend command is sent from a node controller.
Clear condition	Resume command is sent from a node controller.
User action	Not Applicable
Motor Suspend Fault	Yes
Motion suspended – FastStop active	
Definition	Motors were sent a FastStop command, which decelerates the vehicle at the maximum thrust until velocity reaches zero. At that point, the motors hold the vehicle in place and suspend. A FastStop suspend applies to the full path. **NOTE:** Valid for NC Software Image version 7.2.21 and higher.
Set condition	Node controller receives a FastStop command.
Clear condition	Node controller receives a Resume command.
User action	Not Applicable
Motor Suspend Fault	Yes

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Motor not responding	
Definition	A node controller is not able to communicate with a motor.
Set condition	No status report is received from a motor for 300 ms.
Clear condition	Status report is received from a non-responsive motor.
User action	• Check communication connections between the node controller and the first motor in the path that is not responding. • Check that power is being supplied to the first motor in the path that is not responding.
Motor Suspend Fault	Yes
Block 1…10 Faults A	
Inverter disabled	
Definition	Inverter is disabled due to one of the other block faults.
Set condition	See the block fault section that caused the inverter to be disabled for Set condition, Clear condition, User action, and Motor Suspend Fault information.
Clear condition	
User action	
Motor Suspend Fault	Yes
Motor under-voltage warning (Block 1 only)	
Definition	Motor detects the bus voltage to be close to the under-voltage threshold.
Set condition	Bus voltage at the motor drops below +42.5 VDC.
Clear condition	Bus voltage at the motor rises above +43.0 VDC.
User action	Check the propulsion power. Make sure that the propulsion power supply and the cables are sized properly to avoid excessive voltage drop.
Motor Suspend Fault	No
Vehicle not located (Blocks 2…10)	
Definition	Coil board has detected a vehicle but vehicle ID is unknown.
Set condition	Anomaly that occurred.
Clear condition	Issue is Startup on that path to relocate the vehicle.
User action	Perform a Startup on that path to locate the vehicle to delete any vehicle ID set to none.
Motor Suspend Fault	No

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Motor over-voltage warning (Block 1 only)	
Definition	Motor detects the bus voltage to be close to the over-voltage threshold.
Set condition	Bus voltage at the motor rises above +57.0 VDC.
Clear condition	Bus voltage at the motor drops below +56.5 VDC.
User action	Check the propulsion power. Make sure that the propulsion power supply and the cables are sized properly to avoid excessive voltage drop.
Motor Suspend Fault	No
Motor over-voltage warning (Block 1 only)	
Definition	Motor detects the bus voltage to be close to the over-voltage threshold.
Set condition	Bus voltage at the motor rises above +57.0 VDC.
Clear condition	Bus voltage at the motor drops below +56.5 VDC.
User action	Check the propulsion power. Make sure that the propulsion power supply and the cables are sized properly to avoid excessive voltage drop.
Motor Suspend Fault	No
Unexpected slave module reset	
Definition	Slave module was reset.
Set condition	Slave module went through a reset state.
Clear condition	N/A
User action	N/A
Motor Suspend Fault	No
In bootload mode	
Definition	Motor in programming state.
Set condition	N/A
Clear condition	N/A
User action	Finalize programming via node controller.
Motor Suspend Fault	No

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Slave module not responding	
Definition	Slave board is not responding to inquiries and commands from the motor controller.
Set condition	The motor controller detects no slave communication for 10 ms.
Clear condition	Motor controller-slave communication is restored.
User action	• Verify that the motor has been programmed and that motors and vehicles are properly grounded. • If this fault reoccurs, return the motor.
Motor Suspend Fault	Yes
Soft Start not complete (Block 1 only)	
Definition	This fault appears only on block 1, which indicates the Soft Start circuit that limits inrush current upon application of the DC bus has not turned on. If the Soft Start circuit does not turn on, the motor is in suspend mode.
Set condition	The DC Input bus is higher than +43V DC and the difference between the DC Input bus and the downstream load (through the Soft Start resistor) is less than 2V DC. Typically, this fault only occurs at initial power-up. Once cleared, it is not set again until the DC Input bus falls lower than +41V DC. Repeatedly toggling the DC bus on/off causes the Soft Start PTC resistor to heat up and open.
Clear condition	DC Input bus is greater than +43V DC and the difference between the DC input bus and the downstream load is less than 2V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • Remove power and wait 60 seconds before reapplying power (limit cycling of the propulsion power to three times or less per minute to allow the PTC to cool-down).
Motor Suspend Fault	Yes
Slave processor reset initiated (Block 1 only)	
Definition	A slave processor is being reset due to not responding.
Set condition	Slave processor not responding for over 2 s.
Clear condition	Slave processors responds.
User action	Check for static on the track – slaves processor could be reset.
Motor Suspend Fault	Yes

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Block 1…10 Faults B	
Over-current fault	
Definition	The motor has detected an over-current condition on the inverter for a motor block. The motor controller attempts to reset the fault three times before raising an over-current fault.
Set condition	The motor has detected an over-current fault condition on the inverter for a motor block.
Clear condition	Clear fault command.
User action	• If this fault occurs on all motor blocks, check the motor power wiring for intermittent connections and reset the motor to clear the fault. • If this fault only occurs on one block or the problem persists after a reset, contact Technical Support.
Motor Suspend Fault	Yes
Under-voltage fault	
Definition	Propulsion power to a block is lower than +41V DC.
Set condition	Propulsion power falls lower than +41V DC.
Clear condition	Propulsion power rises higher than +43V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one block, contact Technical Support.
Motor Suspend Fault	Yes
Over-voltage fault	
Definition	Propulsion power to a block is higher than +59V DC.
Set condition	Propulsion power rises higher than +59V DC.
Clear condition	Propulsion power falls lower than +57V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one block, contact Technical Support.
Motor Suspend Fault	Yes

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Motor stall detected	
Definition	The block has a thermal model of its inverter. If the model exceeds its thermal threshold, this fault is set. Also, fault modes that damage the current sensing hardware cause the motor to go into this mode.
Set condition	Thermal model rises higher than the thermal limit.
Clear condition	Thermal model falls lower than the lower limit (note hysteresis).
User action	• Check for a stalled vehicle or mechanical binding in the track. • Reset the motor. If the fault persists, return the motor.
Motor Suspend Fault	Yes
Slave module not configured	
Definition	The slave board has not been configured with all of its motor-specific configuration items. This fault is indicative of a slave board reset since all configuration is last downloaded when the slave board is reset.
Set condition	Set at reset and remains set until all slave board configuration items are received from the motor controller.
Clear condition	Required slave board configuration items have been received from the motor controller.
User action	Check for proper grounding of the track and vehicle.
Motor Suspend Fault	Yes
Over-temperature fault	
Definition	Motor controller temperature has exceeded safe limits for hardware and the motor has shut down to prevent thermal damage.
Set condition	Ambient internal temperature exceeds 95 °C.
Clear condition	Ambient internal temperature is lower than 90 °C (note hysteresis).
User action	• Verify that there is nothing preventing vehicle motion on the motor. • Verify that the temperature of the operating environment is within specifications. • Verify that the demanded rms current on the motor does not exceed the motor limitations.
Motor Suspend Fault	Yes

Table 5-4: QuickStick 100 Motor Fault Troubleshooting (Continued)

Hall Effect board sensor fault	
Definition	Occurs if there is a mismatch of the signal from the corresponding HES pairs.
Set condition	Either because a Hall Effect sensor fails or because the sensor signal is not strong enough to be recognized. Signal strength is typically low when there is mechanical misalignment between the magnet array and the stator's sense board.
Clear condition	None
User action	Reset the motor. If the problem persists, replace the motor.
Motor Suspend Fault	No
Slave communication fault	
Definition	Controller board unable to communicate with driver board.
Set condition	None.
Clear condition	None.
User action	Reset if automatically cleared.
Motor Suspend Fault	Yes

Table 5-5 provides detailed fault definitions, set and clear conditions, and user actions for the QuickStick 150 motor faults reported in Table 4-14,QuickStick 150 Motor Fault Data, on page 295. The Motor Suspend Fault field shows when motion on the motor is suspended. Motion cannot resume until all faults causing a Motor Suspend are cleared. Vehicles are not allowed to enter the section of the path where the motor is located while it is suspended.

Table 5-5: QuickStick 150 Motor Fault Troubleshooting

OS Scheduler Faults	
Scheduler not initiated	
Definition	Scheduler didn't finish initialization.
Set condition	Scheduler was requested to schedule a nonexistent function.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Scheduler event queue full	
Definition	Software fault condition that is used for reporting when the scheduler event queue in the motor controller is full.
Set condition	The motor controller detects that the scheduler event queue is full.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes
Upstream and Downstream Comm Faults	
Connection inoperative	
Definition	The motor controller sends periodic ping requests and expects ping responses to determine the health of the communication link.
Set condition	The motor controller did not receive a ping response after sending four consecutive ping requests.
Clear condition	The motor controller received a ping response after sending a ping request.
User action	Check communication link cabling and/or power cabling to adjacent motors.
Motor Suspend Fault	Yes
Transmit buffer full	
Definition	The communication transmit buffer in the motor controller is full and cannot send the current message. This message is diagnostic only, no action takes place.
Set condition	The motor controller cannot send a message because the transmit buffer is full.
Clear condition	The motor controller is able to send a message.
User action	• Check to see if a Virtual Scope data stream has been started but not stopped. • Verify that the path length does not exceed the limit that is shown in the MagneMotion Node Controller Interface User Manual, publication MMI-UM001.
Motor Suspend Fault	No

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Link down	
Definition	Link between the motor and the node controller is down.
Set condition	Link is not detected for 100 ms.
Clear condition	Link is operational.
User action	Verify the connection between the NC and the motor.
Motor Suspend Fault	Yes (configurable)
Motor Overall Faults	
Motor not in operational mode	
Definition	Initial state of motor when it comes out of reset.
Set condition	Reset complete.
Clear condition	Cleared after 100 ms.
User action	None. Fault clears automatically.
Motor Suspend Fault	Yes
Motor in configuration mode	
Definition	Master board is being configured.
Set condition	Node Controller sets the motor in configuration mode.
Clear condition	Node Controller clears motor from configuration mode.
User action	None
Motor Suspend Fault	Yes
Motor in diagnostic mode	
Definition	Master board is in a diagnostic mode.
Set condition	Node Controller sets the motor in diagnostic mode.
Clear condition	Node Controller clears the motor from diagnostic mode.
User action	None - Could be user driven.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Motion suspended by node controller	
Definition	A node controller issued a suspend command to the motor. This message can be in response to a suspend command or a digital I/O based interlock or emergency stop.
Set condition	Suspend command is sent from a node controller.
Clear condition	Resume command is sent from a node controller.
User action	Not Applicable
Motor Suspend Fault	Yes
Motion suspended – FastStop active	
Definition	Motors were sent a FastStop command, which decelerates the vehicle at the maximum thrust until velocity reaches zero. At that point, the motors hold the vehicle in place and suspend. A FastStop suspend applies to the full path. **NOTE:** Valid for NC Software Image version 7.2.21 and higher.
Set condition	Node controller receives a FastStop command.
Clear condition	Node controller receives a Resume command.
User action	Not Applicable
Motor Suspend Fault	Yes
Motor not responding	
Definition	A node controller is not able to communicate with a motor.
Set condition	No status report is received from a motor for 300 ms.
Clear condition	Status report is received from a non-responsive motor.
User action	• Check communication connections between the node controller and the first motor in the path that is not responding. • Check that power is being supplied to the first motor in the path that is not responding.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Block 1…10 Faults A	
Gate driver not responding	
Definition	Gate driver does not respond to a command from the motor.
Set condition	Loss of communication to the gate driver.
Clear condition	Re-establishment of communication to the gate driver.
User action	Reset motor.
Motor Suspend Fault	Yes
Block 1…10 Faults B	
Over-current fault	
Definition	The motor has detected an over-current condition on the inverter for a motor block. The motor controller attempts to reset the fault three times before raising an over-current fault.
Set condition	The motor has detected an abnormal current draw within the motor block that cannot be cleared automatically.
Clear condition	Reset of the motor commanded by the node controller or power cycle of the motor.
User action	• If this fault occurs on all motor blocks, check the motor power wiring for intermittent connections and reset the motor to clear the fault. • If this fault only occurs on one block or the problem persists after a reset, contact Technical Support.
Motor Suspend Fault	Yes
Under-voltage fault	
Definition	Propulsion power to a block is lower than +41V DC.
Set condition	Propulsion power falls lower than +41V DC.
Clear condition	Propulsion power rises higher than +43V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one block, contact Technical Support.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Over-voltage fault	
Definition	For propulsion power input voltage of 48V the level is higher than +59V DC and for propulsion power input voltage of 72V the level is higher than +83V DC.
Set condition	For propulsion power input voltage of 48V the level is lower than +59V DC and for propulsion power input voltage of 72V the level is lower than +82V DC.
Clear condition	For propulsion power input voltage of 48V the level is lower than +57V DC and for propulsion power input voltage of 72V the level is lower than +82V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one block, contact Technical Support.
Motor Suspend Fault	Yes
Motor stall detected	
Definition	The block has a thermal model of its inverter. If the model exceeds its thermal threshold, this fault is set. Also, fault modes that damage the current sensing hardware cause the motor to go into this mode.
Set condition	Thermal model rises higher than the thermal limit.
Clear condition	Thermal model falls lower than the lower limit (note hysteresis).
User action	• Check for a stalled vehicle or mechanical binding in the track. • Reset the motor. If the fault persists, return the motor.
Motor Suspend Fault	Yes
Over-temperature fault	
Definition	Motor controller temperature has exceeded safe limits for hardware and the motor has shut down to prevent thermal damage.
Set condition	Ambient internal temperature exceeds 95 °C.
Clear condition	Ambient internal temperature is lower than 90 °C (note hysteresis).
User action	• Verify that there is nothing preventing vehicle motion on the motor. • Verify that the temperature of the operating environment is within specifications. • Verify that the demanded rms current on the motor does not exceed the motor limitations.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Hall Effect board sensor fault	
Definition	Occurs if there is a mismatch of the signal from the corresponding HES pairs.
Set condition	Either because a Hall Effect sensor fails or because the sensor signal is not strong enough to be recognized. Signal strength is typically low when there is mechanical misalignment between the magnet array and the stator’s sense board.
Clear condition	None
User action	Reset the motor. If the problem persists, replace the motor.
Motor Suspend Fault	No
Hall Effect sensor not responding	
Definition	Occurs if there is no reading from the Hall Effect sensors.
Set condition	Hall Effect sensor is not being read.
Clear condition	None
User action	Reset the motor. If the problem persists, replace the motor.
Motor Suspend Fault	No
Master board and secondary core faults	
Aux core note responding	
Definition	Communication to the auxiliary core has communication issues and has not received a response in 10 ms.
Set condition	The motor controller has detected a communication fault and has timed-out.
Clear condition	Aux Core Communication is re-established to the motor controller.
User action	• Power cycle the motor. • If the problem persists, return the motor.
Motor Suspend Fault	Yes
Unexpected aux core reset detected	
Definition	Secondary processor issued a reset without host command
Set condition	Secondary processor reset.
Clear condition	Upon reading the secondary processor status this will get cleared.
User action	• Power cycle the motor. • If the problem persists, return the motor.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Manufacturing data CRC error	
Definition	Manufacturing data consistency failed.
Set condition	Manufacturing data experienced a change.
Clear condition	Reset and reconfigure motor.
User action	Reset and reconfigure motor.
Motor Suspend Fault	Yes
Calibration data CRC error	
Definition	Calibration data consistency failed.
Set condition	Calibration data experienced a changed.
Clear condition	None.
User action	None.
Motor Suspend Fault	Yes
Driver core 1 and driver core 2 faults	
Bus under-voltage fault	
Definition	Propulsion power to a block is lower than +41V DC.
Set condition	Propulsion power input voltage falls lower than +41V DC.
Clear condition	Propulsion power input voltage rises higher than +43V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one block, contact Technical Support.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Bus over-voltage fault	
Definition	For propulsion power input voltage of 48V the level is higher than +59V DC and for propulsion power input voltage of 72V the level is higher than +83V DC.
Set condition	For propulsion power input voltage of 48V the level is higher than +59V DC and for propulsion power input voltage of 72 V the level is higher than +83V DC.
Clear condition	Bus voltage at the motor drops below +59 VDC when using an input propulsion power voltage of +48 VDC and motor drops below +82 VDC when using an input propulsion power voltage of +72 VDC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • If this fault only occurs on one block, contact Technical Support.
Motor Suspend Fault	Yes
Fuse open	
Definition	The inverter board fuse is detected open.
Set condition	Fuse is blown open.
Clear condition	None.
User action	Replace motor.
Motor Suspend Fault	Yes

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Soft start not complete	
Definition	The Soft Start circuit that limits inrush current upon application of the DC bus has not turned on. If the Soft Start circuit does not turn on, the motor is in suspend mode.
Set condition	The DC Input bus is higher than +43V DC and the difference between the DC Input bus and the downstream load (through the Soft Start resistor) is less than 2V DC. Typically, this fault only occurs at initial power-up. • Once cleared, it is not set again until the DC Input bus fall lower than +41V DC. • Repeatedly toggling the DC bus on/off causes the Soft Start PTC resistor to heat up and open
Clear condition	DC Input bus is greater than +43V DC and the difference between the DC input bus and the downstream load is less than 2V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • Remove power and wait 60 seconds before reapplying power (limit cycling of the propulsion power to three times or less per minute to all the PTC to cool-down).
Motor Suspend Fault	Yes
Bus under-voltage warning	
Definition	Motor detects the bus voltage to be close to the under-voltage threshold.
Set condition	Bus voltage at the motor drops below +42.5V DC.
Clear condition	Bus voltage at the motor rises above +43.0V DC.
User action	Check the propulsion power. Make sure that the propulsion power supply and the cables are sized properly to avoid excessive voltage drop.
Motor Suspend Fault	No

Table 5-5: QuickStick 150 Motor Fault Troubleshooting (Continued)

Bus over-voltage warning	
Definition	Motor detects the bus voltage to be close to the over-voltage threshold.
Set condition	Bus voltage at the motor rises above +57.0V DC when using an input propulsion power voltage of +48V DC and motor rises above +82.0V DC when using an input propulsion power voltage of +72V DC.
Clear condition	Bus voltage at the motor drops below +56.5V DC when using an input propulsion power voltage of +48 VDC and motor drops below +81.5V DC when using an input propulsion power voltage of +72V DC (note hysteresis).
User action	Check the propulsion power. Make sure that the propulsion power supply and cables are sized properly to avoid excessive voltage drop.
Motor Suspend Fault	No
Ethernet comm errors	
Duplicate IP address detected	
Definition	A duplicate IP address has been detected in the system.
Set condition	A device with the same IP address has been detected.
Clear condition	System no longer detects a duplicate IP address
User action	Correct the duplicate IP address environment to another IP address.
Motor Suspend Fault	Yes
IP address not configured	
Definition	Motor missing an IP address or IP address has not been set.
Set condition	System detects an IP address is missing or not configured correctly.
Clear condition	IP address has been configured and system can communicate with IP address.
User action	Configure the IP address in question.
Motor Suspend Fault	Yes

Table 5-6 provides detailed fault definitions, set and clear conditions, and user actions for the QuickStick HT motor faults reported in Table 4-15,QuickStick HT Gen 2 Motor Fault Data, on page 297. The Motor Suspend Fault field shows when motion on the motor is suspended. Motion cannot resume until all faults causing a Motor Suspend are cleared. Vehicles are not allowed to enter the section of the path where the motor is located while it is suspended.

For definitions of motors, motor controller, or inverters see MagneMotion Glossary of Terms, publication MMI-RM003.

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting

OS Scheduler Faults	
Scheduler not initiated	
Definition	Scheduler didn't finish initialization.
Set condition	Scheduler was requested to schedule a nonexistent function.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes
Scheduler event queue full	
Definition	Software fault condition that is used for reporting when the scheduler event queue in the motor controller is full.
Set condition	The motor controller detects that the scheduler event queue is full.
Clear condition	Motor reset.
User action	• Reset motor. • Send NC/HLC log files and motor runtime version to Technical Support for evaluation.
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Upstream and Downstream Comm Faults

Connection inoperative

Definition	The motor controller sends periodic ping requests and expects ping responses to determine the health of the communication link.
Set condition	The motor controller did not receive a ping response after sending four consecutive ping requests.
Clear condition	The motor controller received a ping response after sending a ping request.
User action	Check communication link cabling and/or power cabling to adjacent motors.
Motor Suspend Fault	Yes

Transmit buffer full

Definition	The communication transmit buffer in the motor controller is full and cannot send the current message. This message is diagnostic only, no action takes place.
Set condition	The motor controller cannot send a message because the transmit buffer is full.
Clear condition	The motor controller is able to send a message.
User action	• Check to see if a Virtual Scope data stream has been started but not stopped. • Verify that the path length does not exceed the limit that is shown in the MagneMotion Node Controller Interface User Manual, publication MMI-UM001.
Motor Suspend Fault	No

UART settings corrected

Definition	The UART settings are not correct. **NOTE**: Only used for serial type motors.
Set condition	External UART settings are not correct.
Clear condition	Cleared on reset.
User action	None.
Motor Suspend Fault	No

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Link Down	
Definition	Link between the motor and the upstream node controller is down.
Set condition	Link is not detected for 100 ms.
Clear condition	Link is operational.
User action	Verify the connection between NC and motor.
Motor Suspend Fault	Yes (Configurable)
Motor Overall Faults	
Motor not in operational mode	
Definition	Initial state of motor when it comes out of reset.
Set condition	Reset complete.
Clear condition	Cleared after 100 ms.
User action	None. Fault clears automatically.
Motor Suspend Fault	Yes
Motor in configuration mode	
Definition	Master board is being configured.
Set condition	Node Controller sets the motor in configuration mode.
Clear condition	Node Controller clears motor from configuration mode.
User action	None
Motor Suspend Fault	Yes
Motor in diagnostic mode	
Definition	Master board is in a diagnostic mode.
Set condition	Node Controller sets the motor in diagnostic mode.
Clear condition	Node Controller clears the motor from diagnostic mode.
User action	None - Could be user driven.
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Motion suspended by node controller	
Definition	A node controller issued a suspend command to the motor. This message can be in response to a suspend command or a digital I/O based interlock or emergency stop.
Set condition	Suspend command is sent from a node controller.
Clear condition	Resume command is sent from a node controller.
User action	Not Applicable
Motor Suspend Fault	Yes
Motion suspended – FastStop active	
Definition	Motors were sent a FastStop command, which decelerates the vehicle at the maximum thrust until velocity reaches zero. At that point, the motors hold the vehicle in place and suspend. A FastStop suspend applies to the full path. **NOTE:** Valid for NC Software Image version 7.2.12 and higher.
Set condition	Node controller receives a FastStop command.
Clear condition	Node controller receives a Resume command.
User action	Not Applicable
Motor Suspend Fault	Yes
Motor not responding	
Definition	A node controller is not able to communicate with a motor.
Set condition	No status report is received from a motor for 300 ms.
Clear condition	Status report is received from a non-responsive motor.
User action	• Check communication connections between the node controller and the first motor in the path that is not responding. • Check that power is being supplied to the first motor in the path that is not responding.
Motor Suspend Fault	Yes
Master Board Faults A	
Reserved	

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Master Board Faults B

Logic under-voltage fault (QSHT + Gen 2 Motor Controller)

Definition	Logic power to a motor is lower than +19V DC.
Set condition	Logic power falls lower than +19V DC.
Clear condition	Logic power rises higher than +20V DC (note hysteresis).
User action	Verify that logic power is provided to the motor at the correct voltage.
Motor Suspend Fault	No

Logic under-voltage fault (QSHT + 5700 Motor Controller)

Definition	Logic power to a motor is lower than +22V DC.
Set condition	Logic power falls lower than +22V DC.
Clear condition	Logic power rises higher than +22V DC (note hysteresis).
User action	Verify that logic power is provided to the motor at the correct voltage.
Motor Suspend Fault	No

Logic over-voltage fault (QSHT + Gen 2 Motor Controller)

Definition	Logic power to a motor is higher than +42V DC.
Set condition	Logic power rises higher than +42V DC.
Clear condition	Logic power falls lower than +38V DC (note hysteresis).
User action	Verify that logic power is provided to the motor at the correct voltage.
Motor Suspend Fault	No

Logic over-voltage fault (QSHT + 5700 Motor Controller)

Definition	Logic power to a motor is higher than +26V DC.
Set condition	Logic power rises higher than +26V DC.
Clear condition	Logic power falls lower than +25V DC (note hysteresis).
User action	Verify that logic power is provided to the motor at the correct voltage.
Motor Suspend Fault	No

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Logic over-temperature fault	
Definition	The temperature of the motor controller has exceeded safe limits for the hardware and the motor has shut down to prevent thermal damage.
Set condition	Ambient internal temperature exceeds 80 °C.
Clear condition	Ambient internal temperature is lower than 75 °C (note hysteresis).
User action	• Verify that the fan on the motor controller enclosure is not blocked. • Verify that the temperature of the operating environment is within specifications.
Motor Suspend Fault	Yes
Safety Core not responding (QSHT + 5700 Motor Controller)	
Definition	Communication to the safety core has timed-out after 100 ms. This is a major recoverable fault.
Set condition	The motor controller has detected a communication fault and has timed-out.
Clear condition	Safety Core communication is re-established to the motor controller.
User action	• Power cycle the inverter. • If the problem persists, return the inverter.
Motor Suspend Fault	Yes
Aux Core not responding (QSHT + 5700 Motor Controller)	
Definition	Communication to the auxiliary core has communication issues and has not received a response in 10 ms.
Set condition	The motor controller has detected a communication fault and has timed-out.
Clear condition	Aux Core Communication is re-established to the motor controller.
User action	• Power cycle the inverter. • If the problem persists, return the inverter.
Motor Suspend Fault	No

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Safety Core invalid software (QSHT + 5700 Motor Controller)	
Definition	Safety core does not have valid software.
Set condition	Detected an invalid safety core software.
Clear condition	None
User action	• Reset motor. • If the problem persists, return the inverter.
Motor Suspend Fault	Yes
HES 1, 2 Faults A	
Stator Mismatch	
Definition	Inverter detects a mismatch between detected and configured stators.
Set condition	The inverter detects an invalid stator connected to this unit.
Clear condition	None
User action	• Check HES stator connections to inverter. • Check the motor configuration in the configuration file. • Reset inverter.
Motor Suspend Fault	Yes
In diagnostic mode	
Definition	The HES board is in diagnostic mode.
Set condition	The motor controller detects the HES board is in diagnostic mode.
Clear condition	The motor controller detects the HES board is in operational mode.
User action	Reset the unit.
Motor Suspend Fault	Yes
In boot load mode	
Definition	The HES board is kept in bootloader mode.
Set condition	The motor controller detects the HES board is in bootloader mode.
Clear condition	The motor controller detects the HES board is in operational mode.
User action	• Program the HES board. • Check inverter and stator cables.
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

HES 1, 2 Faults B

Stator over-temperature fault

Definition	Stator temperature as measured by the HES board has exceeded 90 °C.
Set condition	Stator temperature exceeds 90 °C.
Clear condition	Stator temperature falls lower than 85 °C (note hysteresis).
User action	• Verify that there is nothing preventing vehicle motion on the motor. • Check environmental conditions of stator. • Verify that the demanded rms current on the motor does not exceed the motor limitations.
Motor Suspend Fault	Yes

HES board not responding

Definition	HES board is not responding to inquiries and commands from the motor controller.
Set condition	The motor controller detects no HES communication for 10 ms.
Clear condition	Communication between the motor controller and the HES board is restored.
User action	• Check the sense cable between the inverter and the stator. • Check the stator module. • If the problem persists, return the inverter or stator module.
Motor Suspend Fault	Yes

Inverter 1, 2 Faults A

Inverter disabled by command

Definition	Operational state of this inverter.
Set condition	The inverter is disabled by a command from the host.
Clear condition	Normal operational state of this inverter as commanded by the host.
User action	None
Motor Suspend Fault	No

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Soft Start switch Off (QSHT + Gen 2 Motor Controller)	
Definition	Propulsion power not ready – The Soft Start circuit that limits inrush current upon application of the DC bus has not turned on. If the Soft Start circuit does not turn on, the motor is in suspend mode.
Set condition	The DC Input bus is higher than +270V DC and the difference between the DC Input bus and the downstream load (through the Soft Start resistor) is less than 5V DC. Typically, this fault only occurs at initial power-up. If this fault is cleared, it is not set again until the DC Input bus is lower than +250V DC. Repeatedly toggling the DC bus on/off causes the Soft Start PTC resistor to heat up and open.
Clear condition	DC Input bus is higher than +270V DC and the difference between the DC input bus and the downstream load is less than 5V DC (note hysteresis).
User action	• Verify that propulsion power is being provided to the motor at the correct voltage. • Remove power and wait 60 seconds before reapplying power (limit cycling of the propulsion power to three times or less per minute to allow the PTC to cool-down).
Motor Suspend Fault	Yes
Inverter disabled by power supply not ready (QSHT + 5700 Motor Controller)	
Definition	The inverter board has not received a state update that the power supply is in the running state.
Set condition	CIP Axis State of the Power Supply is not in the **Running** state.
Clear condition	CIP Axis State of the Power Supply is in the **Running** state.
User action	Wait for the Power Supply to close its internal relay before driving current out the inverter.
Motor Suspend Fault	Yes
Fuse Open (QSHT + 5700 Motor Controller)	
Definition	The inverter board fuse is detected open.
Set condition	Fuse is blown open.
Clear condition	Replace the motor controller.
User action	Replace the motor controller.
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Hardware over-current warning	
Definition	The inverter receives an over-current shutdown.
Set condition	The motor has detected an over-current condition on the inverter for a motor block.
Clear condition	The inverter clears the fault and retries 10 ms after the fault has occurred. • If 5 seconds pass and another fault HAS happened, the inverter latches off (issue a reset to clear the latched condition). Once the inverter latches off, a Hardware over-current fault is sent. • If 5 seconds pass and another fault has NOT happened, the inverter is allowed to retry if a subsequent over-current fault occurs.
User action	• Check that propulsion voltage is within range. • Check stator cable between motor controller and stators. • Reset the motor.
Motor Suspend Fault	Yes
Software over-current fault (QSHT + Gen 2 Motor Controller)	
Definition	The motor software has detected an over-current condition.
Set condition	Set if any of the three phase currents measurements are greater than ±25 A.
Clear condition	Clear if all three-phase current measurements fall within ±25 A. Normal operation current limit is set for 15 A.
User action	• Check that propulsion voltage is within range. • Check the drive cable between the motor controller and the stator.
Motor Suspend Fault	Yes
In bootload mode	
Definition	The inverter is kept in bootloader mode.
Set condition	The motor controller detects the inverter board is in bootloader mode.
Clear condition	The motor controller detects the inverter board is in operational mode.
User action	• Program the inverter board. • Might occur when the Node Controller resets the motor.
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Fan fault (QSHT + Gen 2 Motor Controller)	
Definition	Status of the fan on the front of the motor controller.
Set condition	Set if fan status is OFF.
Clear condition	Clear if fan status is ON.
User action	Check that the fan is not blocked or clogged.
Motor Suspend Fault	Yes
Inverter fault	
Definition	Indicates that any of the inverter faults that would shut down the inverter are active.
Set condition	**QSHT + Gen 2 Motor Controller:** Either hardware over-current, propulsion over-voltage, device thermal overload, inverter (heatsink) over-temperature, inverter disabled, or software over-current is detected. **QSHT + 5700 Motor Controller:** Either hardware over-current, propulsion over-voltage, device thermal overload, inverter disabled by Safety Core, inverter disabled, gate driver under-voltage lockout, or software over-current is detected.
Clear condition	Cleared if all inverter faults are clear.
User action	See the individual fault descriptions for troubleshooting a particular fault.
Motor Suspend Fault	Yes
Inverter 1, 2 Faults B	
Hardware over-current fault	
Definition	The motor has detected an over-current condition.
Set condition	Any phase current or the propulsion power input current exceeds ±25 A.
Clear condition	Reset of motor controller.
User action	• Reset the motor. • Check propulsion power input cable. • Check the drive cable between the motor controller and the stator. • If the problem persists, replace the motor controller.
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Under-voltage fault (QSHT + Gen 2 Motor Controller)	
Definition	Propulsion power is lower than +250V DC.
Set condition	Propulsion power falls lower than +250V DC.
Clear condition	Propulsion power rises higher than +270V DC (note hysteresis).
User action	Verify that propulsion power is being provided to the motor at the correct voltage.
Motor Suspend Fault	Yes
Under-voltage fault (QSHT + 5700 Motor Controller)	
Definition	Propulsion power is lower than +265V DC.
Set condition	Propulsion power falls lower than +265V DC.
Clear condition	Propulsion power rises higher than +275V DC (note hysteresis).
User action	Verify that propulsion power is being provided to the motor at the correct voltage.
Motor Suspend Fault	Yes
Over-voltage fault (QSHT + Gen 2 Motor Controller)	
Definition	Propulsion power is higher than +430V DC.
Set condition	Propulsion power rises higher than +430V DC.
Clear condition	Propulsion power falls lower than +420V DC (note hysteresis).
User action	Verify that propulsion power is being provided to the motor at the correct voltage.
Motor Suspend Fault	Yes
Over-voltage fault (QSHT + 5700 Motor Controller)	
Definition	Propulsion power is higher than +830V DC.
Set condition	Propulsion power rises higher than +830V DC.
Clear condition	Propulsion power falls lower than +820V DC (note hysteresis).
User action	Verify that propulsion power is being provided to the motor at the correct voltage.
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Motor stall detected	
Definition	Device thermal overload – The inverter board has a thermal model. If the model exceeds its thermal threshold, this fault is set. The model allows the maximum commanded current to be present in the phases for 6 seconds (lower currents are allowed for longer times). Once tripped, the model takes approximately 6 seconds to fall lower than the lower threshold to turn the inverter back on.
Set condition	Thermal model exceeds the thermal threshold.
Clear condition	Thermal model lowers to within the acceptable range.
User action	• Check for a stalled vehicle or mechanical binding of the vehicle in the track. • Lower Acceleration. • Reset the motor. • If the fault persists, return the inverter.
Motor Suspend Fault	Yes
Guard stop request status (QSHT + 5700 Motor Controller)	
Definition	The inverter’s gate drivers are disabled due to a STO function from the safety core.
Set condition	Hardwired STO is enabled.
Clear condition	Hardwired STO is disabled.
User action	• Check the Safety Core Faults. • If the problem persists after a power cycle, replace the motor controller.
Motor Suspend Fault	Yes
Inverter over-temperature fault	
Definition	The inverter monitors the inverter heatsink/IGBT temperature (different than the Device Thermal Overload).
Set condition	The inverter heatsink/IGBT temperature is higher than 60 °C/100 °C.
Clear condition	The inverter heatsink/IGBT temperature falls lower than 55 °C/90 °C (note hysteresis).
User action	• Check environmental conditions of motor controller. • Check that the fan on the motor controller is not blocked or clogged. • Verify that the demanded rms current on the motor does not exceed the motor limitation
Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Inverter not responding	
Definition	The inverter board has not responded to the motor controller.
Set condition	Inverter board does not respond to motor controller queries within 20 ms.
Clear condition	Inverter board responds to motor controller queries.
User action	Reset the motor. If the problem persists after a reset, replace the motor controller.
Motor Suspend Fault	Yes
Gate driver under-voltage lockout (QSHT + 5700 Motor Controller)	
Definition	The inverter board detects an under-voltage level on the gate drivers.
Set condition	Gate driver voltage drops below 13 V.
Clear condition	Gate driver voltage rises above 13 V.
User action	Power cycle the motor. If the problem persists after a power cycle, replace the motor controller.
Motor Suspend Fault	Yes
Ethernet Comm Faults	
Reserved	
Block 1, 2 Safety Faults	
Safety Core Fault (QSHT + 5700 Motor Controller)	
Definition	The safety core has detected a non-recoverable internal fault.
Set condition	Internal non-recoverable fault.
Clear condition	Power Cycle Module.

User action	• Verify safety wiring and connections:
	  • Wire terminations at the Safe Torque Off (STO) connector.
	  • Cable/header not seated correctly.
	  • +24V power.
	• Check state of safety inputs.
	• Reset fault and run proof test.
	• If fault persists, return the inverter.

Motor Suspend Fault	Yes

Table 5-6: QSHT and QSHT 5700 Motor Fault Troubleshooting (Continued)

Safe Torque Off Fault (QSHT + 5700 Motor Controller)

Definition	The Safe Torque Off (STO) function detected a fault.
Set condition	Internal fault detected when the STO function is requested.
Clear condition	Turn both inputs to the OFF-state for more than 1 second, then return the inputs to ON.

User action	• Verify safety wiring and connections:
	  • Wire terminations at the Safe Torque Off (STO) connector.
	  • Cable/header not seated correctly.
	  • +24V power.
	• Check state of safety inputs.
	• Reset fault and run proof test.
	• If fault persists, return the inverter.

Motor Suspend Fault	Yes

Guard Stop Input Fault (QSHT + 5700 Motor Controller)

Definition	Monitors the Safe Torque Off function inputs.
Set condition	Safe Torque Off mismatch is detected when safety inputs are in different state for more than 1.0 second.
Clear condition	Turn both inputs to the OFF-state for more than 1 second, then return the inputs to ON.

User action	• Verify safety wiring and connections:
	  • Wire terminations at the Safe Torque Off (STO) connector.
	  • Cable/header not seated correctly.
	  • +24V power.
	• Check state of safety inputs.
	• Reset fault and run proof test.
	• If fault persists, return the inverter.

Motor Suspend Fault	Yes

Digital Inputs (QSHT + 5700 Motor Controller)

IN 1…IN 4

Definition	Digital Input Status of the input pin. Action is user-configurable.
Set condition	Input is held HIGH (24V).
Clear condition	Input is held low (0V) or not connected.
User action	None
Motor Suspend Fault	No

Node Fault Troubleshooting

This section describes the common node faults, and general solutions.

Table 5-7: Initial Node Troubleshooting

Fault	Problem	Solution
Node reports a Device Status of Faulted.	Detected abnormal operation of the device that is associated with the node.	Troubleshoot the node hardware.

Node Controller Fault Troubleshooting

This section describes the common node controller and high-level controller faults, and general solutions.

Table 5-8: Initial Node Controller Troubleshooting

Fault	Problem	Solution
HLC or NC is “stuck” in the **Initialization** state.	The Node Controller Configuration File is invalid.	See the node controller log file for identification of the specific fault.
	Unable to collect motor information.	
HLC reports a **Degraded** status.	Network communication issues.	See the node controller log file for identification of the specific fault.
NC is reported as **Disconnected**.	Network communication issues.	Verify all network connections.
	Power issues.	Verify all power connections.
HLC reports PLC communication status as **Not Configured**.	The host controller is not correctly defined in the Node Controller Configuration File.	Verify that the host controller IP address is correctly defined.
HLC reports EtherNet/IP communication status as **Link Down**.	Network communication issues.	Verify all network connections.
		Verify all network configuration.
		Verify that multiple host controllers are not using the same IP address.

Path Fault Troubleshooting

This section describes the common path faults, and general solutions.

Table 5-9: Initial Path Troubleshooting

Fault	Problem	Solution
Upstream link/connection failed.	NC cannot connect to the upstream end of the motor at the node.	Check the communication connection between the motor and the NC.
	Ensure node controller is not in an initialization state, if so, see Node Controller in Initialization State Troubleshooting Table 5-7.	Check the power being supplied to the motor.
Downstream link/connection failed.	NC cannot connect to the downstream end of the motor at the node.	Check the communication connection between the motor and the NC.
	Ensure node controller is not in an initialization state, if so, see Node Controller in Initialization State Troubleshooting Table 5-7.	Check the power being supplied to the motor.

Vehicle Fault Troubleshooting

This section describes the common vehicle faults, and general solutions.

Table 5-10: Initial Vehicle Troubleshooting

Fault	Problem	Solution
Vehicle Signal for the vehicle is reported as low “0”.	Vehicle is not being detected due to being removed from system.	Issue a Delete Vehicle command and Restart the path.
	Vehicle is not being detected due to being moved from its expected position.	

Table 5-10: Initial Vehicle Troubleshooting (Continued)

Fault	Problem	Solution
Vehicle Hindered Status is being reported as high “1”.	Mechanical blocking of the vehicle.	Clear any obstruction from the motor.
	A vehicle is not in motion while under sync control.	Status clears once the vehicle moves.
	A vehicle command uses a velocity of zero.	Reissue the vehicle motion command with a positive velocity.
	A vehicle command uses a PID set equal to zero.	Reissue the vehicle motion command using a different PID set.
	The Control Off Position Tolerance or the Integrator Distance Threshold positions are set outside of the arrival tolerance. This causes the vehicle to appear to have arrived even though it is not in the expected location.	Redefine the Control Off Position Tolerance or the Integrator Distance Threshold positions in the Motors section of the Node Controller Configuration File.
	Motor does not have propulsion power.	Check the power being supplied to the motor.
Vehicle Obstructed Status is being reported as high “1”.	There is a vehicle in the way.	This occurs during normal operation when vehicles are in a queue or when a vehicle is in a switch, which keeps another vehicle from entering the switch.
	There is a hardware fault.	Vehicles will not enter motors with a suspending fault. Resolve anomalies on faulted motor(s).
	Attempting to move on, or to, a path that is suspended.	The vehicle is attempting to move onto a path that is suspended.
	Attempting to position part of the vehicle past the end of the path.	Send a new vehicle command to reposition the vehicle.
	The vehicle is stopped at a red traffic light.	Change the traffic light to green.

This page intentionally left blank.

Communications Protocol 6

Communications Format

The high-level controller supports communication with the host controller via TCP/IP over Ethernet as defined in the OSI model. Each layer of network communication within the standard OSI (Open Systems Interconnect, ISO/IEC 7498-1) model is described here. Ethernet/TCP runs directly over the TCP and UDP layer 4 transport layers in the OSI model. The implementation of TCP/IP only uses the TCP transport layer.

Calculating the CRC for Messages using the TCP/IP Communication Protocol

This section explains step by step how to calculate the Cyclic Redundancy Check (CRC) that is part of every message sent between the HLC and the Host when using the TCP/IP communication protocol. The Example is intended to help customers who are not familiar with the CRC to understand the calculation and determine how to best implement it in their Host controller code.

Introduction

A cyclic redundancy check (CRC) is an error-detecting technique that is commonly used to detect accidental changes to data in networks. A check value is attached to the original data prior to transmitting it. When the data is received after being transmitted through a network the check value can be used to determine if the data has been corrupted.

Use the CRC-CCITT standard with an initial value of 0xFFFF and the mask polynomial 0x1021 = $2^{16} + 2^{12} + 2^{5} + 1 = 10001000000100001_{2}$. Note that the hexadecimal representation (0x1021) omits the leading 1 of the binary representation according to common practice.

Calculating the CRC - Theory

The CRC value is calculated in a polynomial long division, which takes the message as the dividend and the mask polynomial as the divisor. The quotient is discarded and the remainder becomes the resulting check value.

The CRC-CCITT is a 16-bit CRC, which means the check value will be 16 bits long and the mask polynomial is of the order 16 and therefore has 17 terms as shown in the previous section.

The main advantage of using a CRC for error-detecting is that they are simple to implement in binary hardware. However, this will not discuss the implementation of the CRC calculation since a large number of efficient commercial or freeware implementations are available. Instead the following section will go through a full example of calculating the check sum for an example message to give the reader a step by step instruction that can be used to select or develop a CRC implementation.

Calculating the CRC - Example

The following example is transforming the CRC from hexadecimal to a binary. For additional guidance understanding hexadecimal formats, see Manual Conventions.

Message

In this example we will calculate the CRC for the following example message:

$$AB_{16}\ BA_{16}\ 03_{16}\ 00_{16}\ CRCByte1_{16}\ CRCByte2_{16}$$

Following the TCP/IP communications format:

- The first two Bytes (0xAB and 0xBA) represent the two start bytes.
- The third byte represents the message length (number of bytes after the length byte including the CRC bytes) and is correctly indicating that there are 3 more bytes in the message.
- The fourth byte represents the message type. Message type 00 is not used for an actual message within TCP/IP communications protocol and it is used as an example here to avoid any confusion with a real message. Normally a real message would include additional bytes that contain message-specific information after the message type byte. In order to keep this step by step example reasonably short, a 00 message type with no message-specific content will be used.
- The last two bytes will contain the 16-bit CRC value that will be calculated in the example below.

Initial Value and Appendix

In order to calculate the CRC value, the original message has to be rewritten and modified according to standard CRC theory.

First, the bytes of the message in front of the two CRC bytes are transformed to their binary representation.

$$\begin{gathered} \text{0xAB 0xBA 0x03 0x00} \\ = \\ 1010\ 1011\ 1011\ 1010\ 0000\ 0011\ 0000\ 0000_2 \end{gathered}$$

Because an initial value of 0xFFFF is used, the first 16 bits of the message need to be inverted. The message thus becomes:

0101 0100 0100 0101 0000 0011 0000 0000$_2$

Finally, an appendix consisting of zeros is added to the message to allow the whole message to be processed using the sliding XOR method described in the following sections. The length of the appendix is determined by the length of the mask polynomial minus 1. In this example a 17-bit mask is used and thus 16 zeros are appended to the message:

0101 0100 0100 0101 0000 0011 0000 0000 0000 0000 0000 0000$_2$

Mask Polynomial

As mentioned above, the 17-bit mask polynomial 0x1021 is used and its binary representation is:

1 0001 0000 0010 0001$_2$

Computation of the polynomial long division

The extended and partially inverted message is now used as the dividend with the polynomial mask as the divisor. In every step, the dividend is divided using the XOR (exclusive OR; 1 XOR 1 = 0, 1 XOR 0 = 1, 0 XOR 1 = 1, 0 XOR 0 = 0) operator starting with the first 1 from the left.

The first step aligns the **mask** with the first 1 from the left of the message and carries out the XOR operation for the 17 digits where the mask overlaps the message. The *tail end* of the message remains unchanged.

0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 *0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0*

1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

0 0 1 0 0 0 0 0 1 0 0 1 1 0 1 0 1 *0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0*

This operation is repeated until a remainder where the leftmost 1 is located less than 17 bits from the right is found.

```
00 1 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
   0 0 0 0 1 0 1 0 0 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
           1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
           0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
               1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
               0 0 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                   1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                   0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                     1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                     0 0 1 0 1 1 1 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                         1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                         0 0 1 1 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                             1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                             0 1 0 1 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                               1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                               0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                   1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                   0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
                                       1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                       0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0
                                               1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                               0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 1 0 0 0 0 0 0
                                                       1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                                       0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
```

The remainder is:

$$0001\ 1100\ 1100\ 0100_2$$
$$=$$
$$0x1C\ 0xC4$$

And it follows that:

CRCByte1 = 0x1C

CRCByte2 = 0xC4

With that, the original message is completed to:

0xAB 0xBA 0x03 0x00 0x1C 0xC4

Error Detection

When the message including the CRC value is received, the same CRC calculation can be carried out on the whole message that has been received (including the CRC bytes).

0xAB 0xBA 0x03 0x00 0x1C 0xC4

=

$1010\ 1011\ 1011\ 1010\ 0000\ 0011\ 0000\ 0000\ 0001\ 1100\ 1100\ 0100_2$

As before, the first 16 bits need to be inverted, but no appendix is required:

$\underline{0101\ 0100\ 0100\ 0101}\ 0000\ 0011\ 0000\ 0000\ 0001\ 1100\ 1100\ 0100_2$

The CRC calculation then follows the process introduced above:

```
0   1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
    1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
    0 0 1 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
        1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
        0 0 0 0 1 0 1 0 0 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                    1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                    0 0 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                        1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                        0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                          1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                          0 0 1 0 1 1 1 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                              1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                              0 0 1 1 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                                  1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                  0 1 0 1 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                                    1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                    0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 1 0 0
                                        1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                        0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 0
                                            1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                            0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0
                                                    1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                                    0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0
                                                            1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1
                                                            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

When the remainder is zero, it is very likely that the transmitted message has not been corrupted accidentally. Note that it is comparatively easy to intentionally manipulate the message with errors that aren't detected by CRC, but it is assumed that this case is not relevant to detecting errors in EtherNet/IP messages.

Ethernet TCP/IP

The standard OSI model for layers 1…4 are outlined here. The high-level controller application runs on top of the TCP transport layer. Standard 10/100/1000 Base-TX half-duplex or full-duplex twisted-pair Ethernet is used for all network communications.

Physical Layer

The electrical interface that is supported for communication between the high-level controller software running on the node controller hardware and the host controller is twisted-pair Ethernet. This interface is based on the IEEE 802.3 and IEEE 802.3u Ethernet and Fast Ethernet communication standards.

The interface uses IEEE 802.3 and IEEE 802.3u standard signal levels for 10/100 Base-TX. A standard Category 5 or better cable is used to connect the HLC directly to the host controller when a hub, switch, or router is not used for the connection.

Data Link Layer

The data link layer is standard 802.3/802.3u Ethernet packet framing. This framing uses 6 byte destination and source MAC addresses, 2 byte protocol type, 1500 byte MTU (maximum transmission unit) of payload data, and 2 byte FCS (frame check sequence). 10 Mbps or 100 Mbps signaling rates in full-duplex and half-duplex modes are supported. It is expected that any standard Ethernet device using a twisted-pair physical layer will have no problems inter-operating with the HLC computer at the physical and data link layers.

Network Layer

The network layer communications for the high-level controller is Internet Protocol version 4 (IPv4). For diagnostic purposes during initial setup and optional network layer health status checks, the HLC supports accepting Internet Control Message Protocol (ICMP) echo requests and responding to echo requests with ICMP echo responses. The HLC supports the additional ICMP messages that are required to support IPv4 fragmentation and redirection of network layer packets to the next hop IP router if customer premise networking equipment requires the HLC to honor such IP fragmentation or packet routing redirection. It is expected that any standard modern IPv4 implementation will have no problems inter-operating with the HLC computer.

The high-level controller does not support dynamic routing protocols and does not act as a router in the customer premise network. The HLC drops any packets that it receives via some other peer on the network using the HLC computer as its next hop IPv4 router.

The high-level controller supports configuration with a static IPv4 address, network mask, and gateway address of the next hop IP router.

The high-level controller supports IPv4 Address Resolution Protocol (ARP) to support the discovery of peer Ethernet data link layer MAC addresses.

Transport Layer

The high-level controller uses a Transmission Control Protocol (TCP) stream over the IPv4 network layer to support multiple end-to-end connections.

The high-level controller is the server and the host controller is the client for TCP endpoint communication roles. The HLC listens for incoming TCP connections from the host controller (general-purpose computer) on the following ports:

- 799 for the Host Control connection.
- 800 for the Host Status connections.
- 8000 for the MagneMotion Virtual Scope connection.

The high-level controller supports one Host Control TCP connection from the host controller at a time. If a new TCP connection is made to TCP port 799 when the HLC already has an established connection, the established connection is dropped and the new connection is used. Even though TCP/IP supports a more or less arbitrary number of peers at the transport layer, the HLC supports only one Host Control TCP connection at a time.

The high-level controller supports up to four Host Status TCP connections from the host controller at a time. If a new TCP connection is made to port 800 when the HLC already has four established connections, the oldest connection is dropped and the new connection is used.

NOTE: The Host Status connections can be used to monitor the status of the transport system using the various status request commands. Any other commands are dropped.

The high-level controller supports one Virtual Scope TCP connection from the host controller at a time. If a new TCP connection is made to port 8000 when the HLC already has an established connection, the established connection is dropped and the new connection is used.

When the HLC receives an invalid message on an established TCP connection as determined by the framing that is described in Command Message Framing, the HLC drops the connection.

The high-level controller throttles incoming TCP connections to port 799 and 800 if a problem in the customer premise host controller software causes an excessive number of connections to be established by the host controller and dropped by the HLC due to the requirements described. If the connection is dropped, the HLC waits 1 second before allowing a new connection.

Command Message Framing

To decode a communication message, it is necessary to frame the information at transmission so the receiver knows how to find and decode the various message components.

General Message Formats

Table 6-1 illustrates the general message format.

Table 6-1: General Message Format

Item	When Used	Size in Bytes	RANGE
Start Bytes	Always	2	0xAB followed by 0xBA
Message Length (in bytes)	Always	1	The length of the message following the length byte including the CRC bytes. Range is 3…61 bytes.
Message Type	Always	1	0…255
Data	Optional	0…58	The length of the data varies depending upon the command.
CRC	Always	2	0…65535

The first bytes in a message are 0xAB followed by 0xBA, which signals the beginning of a message. The next byte is the length byte, and represents the number of bytes in the message after the length byte, including the CRC bytes. The maximum message size is limited to 64 bytes. The receiver can lock on to the start of a message by verifying the CRC check using the CRC sent as the last 2 bytes of the message (designated by the length byte). If the bytes 0xABBA appear sequentially in the message, two checks are performed to verify that it is not the start of a message: 1) the length byte is within bounds, and 2) if the length byte is within bounds, the CRC must match the CRC sent in the message.

If the occurrence of 0xABBA is deemed not to be the start of a message, a search must be performed for the next occurrence to check if that is the start of the message.

The Message Type and Data items of the general message format are defined in the Protocol Reference that begins on page 133.

NOTE: When parsing any message from the HLC, the CRC is always the last 2 bytes of any valid message. The position of the CRC is determined using the length byte. This structure allows additional parameters/fields to be inserted after the current data definitions, but before the CRC, which maintains backward compatibility.

Error Checking

A Cyclic Redundancy Check (CRC) is performed on all messages. The last 2 bytes of a message always contain the CRC of the entire message before the CRC – as transmitted. A CRC check of all bytes including the CRC returns 0. The CRC used is the CRC-CCITT with an initial value of 0xFFFF and a mask value of 0x1021 derived from the following polynomial: $2^{16}+2^{12}+2^{5}+1$. Note that the mask value (0x1021) omits the leading bit (1) according to common practice.

NOTE: Reference Numerical Recipes in C: The Art of Scientific Computing by Press et al.

An example of a command message would contain the following bytes.

0xAB 0xBA 0x03 0x00 0x1C 0xC4

In this example, the two CRC bytes are calculated from the first four bytes (every byte before the CRC). Using 0xFFFF as the start value, the CRC generated is 0x1C followed by 0xC4.

Data Encoding

Floating point data is transmitted in little-endian byte-order (least significant byte (LSB) first). All other data, 16 bits and larger, is transmitted in big-endian byte-order (most significant byte (MSB) first), also known as Network Byte Order.

Commands

The Protocol Reference that begins on page 133 details all messages between the host controller and the high-level controller.

Most commands from the host controller are acknowledged with either a Command Accepted or Command Rejected response (see Command Status on page 261). Some commands, such as motion commands, take time to complete. In these cases, after the initial acceptance response, a Command Complete message is transmitted to the host controller when the command has been completed. Status request commands simply provide the requested status and are not acknowledged otherwise.

This page intentionally left blank.

Index

Numerics

A

B

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Changes

Overview

This manual is changed as required to keep it accurate and up-to-date to provide the most complete documentation possible for the MagneMotion® Host Controller TCP/IP Communication Protocol. This section provides a brief description of each significant change.

NOTE: Distribution of this manual and all addenda and attachments is not controlled. To identify the current revision, see the Literature Library on the Rockwell Automation website.

Rev. A

Initial release to support the following NC Software Image Versions:

- MM LITE™ – 1.1.19
- QS 100 – 0.9.147

Rev. B

Revised to support the following NC Software Image Versions:

- MM LITE – 4.1.25
- QS 100 – 7.1.20

Added the following:

- New host controller to HLC commands:
  - Set Node Parameters
  - FastStop Motion
  - *Extended Vehicle Status Request, V2 (0xBF 03 02)*
  - Get Node Controller Digital I/O Status
  - Set Node Controller Digital I/O Outputs
  - Set Node Controller Digital I/O Notification Mask
- New HLC to host controller status responses:
  - *Extended Vehicle Status, V2 (0xDF 03 02)*
  - Node Controller Digital I/O Status

- Support for QSHT motors.
- Added new HLC Command Status Codes.

Updated the following:

- The title of the manual from “TCP Communication Protocol” to “TCP/IP Communication Protocol”.
- Trademark and copyright information.
- The Overview Note in this section.
- In the About This Manual chapter, the Manual Conventions section to describe number and measurement conventions better. The Notes, Safety Notices, and Symbols section to include Symbol Identification. And, the list of Additional Resources.
- In Introduction, the Transport System Components Overview to include the NC-12 node controller. The Transport System Software Overview to update the descriptions of Motor and Magnet Array Type files. And, expanded the Getting Started with the TCP/IP Communication Protocol procedure.
- In Chapter 2, Transport System Control, all figures that are used in the examples and all examples to reflect a typical transport system layout better.
- In Chapter 3, Application Notes, the E-stops cautions and description and the figure for the Gateway Node.
- In Chapter 4, Protocol Reference, all PID set references to 16 PID sets, and the Node Status response.
- The Glossary and Index.

Removed the following:

- All references to the standard node controller (replaced by the NC-12 node controller). Support for the standard node controller, including software, spare parts, technical support, and service continues to be available.
- All references to Zero Headway as this feature is no longer supported.
- All references to Automatic Path Recovery as this feature is deprecated.
- All references to unsupported node types (Turntable and Host Switch).

Rev. C

Revised to support the following NC Software Image Versions:

- MM LITE – 4.1.37
- QS 100 – 7.2.15
- QSHT – 7.2.12

Added the following:

- In Chapter 4, Protocol Reference, Suspect Status bit description for the Vehicle Status response. The over/under-voltage note to the Motor Status response. And, software version support identification for all commands and responses.
- In Chapter 5, Troubleshooting, fault level troubleshooting for MagneMover LITE Motor Fault Troubleshooting, QuickStick 100 Motor Fault Troubleshooting, and QSHT and QSHT 5700 Motor Fault Troubleshooting.

Updated the following:

- In Chapter 2, Transport System Control, the Running the Transport System procedure to include verifying the node controller status.
- In Chapter 3, Application Notes, the description of Safe Stopping Distance, the Moving Path Node to include requirement for vehicle to move past configured clearance distances. The Platooning to define limits on Gateway Node usage. The description of queuing at Merge and Diverge Nodes. The description of Terminus Node operation to include clearance requirement.
- In Chapter 4, Protocol Reference, the description of Suspend Motion and FastStop Motion commands. Corrected the range of the Station ID to 255 max. The description of the 0x0E command status code. The description of the motor faults for QuickStick 100 Motor Fault Data,MagneMover LITE Motor Fault Data, and QuickStick HT Gen 2 Motor Fault Data.

Rev. D

Revised to support the following NC Software Image Versions:

- MM LITE – 4.1.37
- QS 100 – 7.2.21
- QSHT – 7.2.12

Added the following:

- In Chapter 1, Introduction, information about the Virtual Scope utility. Description of the node controller console interface.
- In Chapter 2, Transport System Control, safety warnings that are related to vehicle motion. Examples for FastStop and Node Controller Digital I/O Control operation. Examples for vehicle motion Following Another Vehicle Through a Switch and for Move to Station Using an Offset.

- In Chapter 3, Application Notes, a process for Inserting and Removing Vehicles.
- In Chapter 4, Protocol Reference, offset to the Move Vehicle To Station command. Motor status to QuickStick 100 Motor Fault Data. Motor status and hardware over-current warning to QuickStick HT Gen 2 Motor Fault Data and QSHT and QSHT 5700 Motor Fault Troubleshooting. Additional HLC Command Status Codes (0x1A, 0x1B, 0x26, 0x27, 0x28, 0x43, 0x45).
- In Chapter 5, Troubleshooting, added motor status and hardware over-current warning to QSHT and QSHT 5700 Motor Fault Troubleshooting.

Updated the following:

- Changed the revision from alpha (Rev. D) to numeric (Ver. 04).
- Changed the logo to "A Rockwell Automation Company" version.
- Trademark and copyright information.
- In the About This Manual chapter, updated Symbol Identification to show the change in the Pinch/Crush hazard symbol and updated the description of the hazard.
- In Chapter 1, Introduction, the Transport System Software Overview and the Simplified View of Transport System Software Organization.
- In Chapter 2, Transport System Control, the Move to Position and Move to Station examples. The descriptions of keep-out area operation for Suspend and FastStop.
- In Chapter 3, Application Notes, the Moving Path Node, Terminus Node, and Platooning examples. The Node Type Descriptions and Usage reference section to use the same order as other manuals.
- In Chapter 4, Protocol Reference, the Vehicle Status response. The "Jammed" vehicle status flag is changed to "Hindered".
- In Chapter 5, Troubleshooting, the temperature range for the QS 100 Over-temperature fault. The current range for the QSHT Hardware over-current fault.
- In the Communications Protocol, the system limit for Vehicles and the QSHT Velocity limit.

Rev. E

Added the QuickStick 150 throughout the manual.

Added the following:

- New host controller to HLC commands:
  - Vehicle Follow Order
  - Warm Reset
  - Create Station
  - Delete Station
  - Get Station Status
  - Create Traffic Light
  - Set Traffic Light
  - Get Traffic Light Status

Delete Traffic Light
Lock Vehicle
Vehicle Command
Get Extended Vehicle Status
Get Node Controller Digital I/O Status
Set Node Controller Digital I/O Outputs
MP Get Path End Status
MP Link Command
MP Unlink Command
SM Subscription Command
SM Poll Command
Motor Inverter Command
Set Node Controller Configuration
Restart Node Controllers
Get Extended Node Controller Status
Get Extended High-Level Controller Status
Get Propulsion Power Status
Set Propulsion Power State

- New HLC to host controller status responses:
  Extended Vehicle Status,
  MP Path End Status Report
  MP Alignment Request
  SM Subscription Response
  SM Subscription Data Response
  SM Poll Data Response
  Extended Node Controller Status
  Extended High-Level Controller Status
  Propulsion Power Supply Status
- In Chapter 1, Introduction, added descriptions of new utilities (NC File Retrieval Tool and Ethernet Motor Commissioning Tool) and new file types.
- In Chapter 2, Transport System Control, added Warm Reset, Station Control, Traffic Light Control, Motor Controller Control, Propulsion Power Supply Control to the Transport System Reset, Startup, and Operation section. Added Station Status, Traffic Light Status, Moving Path Node Status, and Propulsion Power Supply Status to the Monitoring Transport System Status section. Added Platooning to the Moving Vehicles section.
- In Chapter 3, Application Notes, added Thrust Limitations information. Added Moving Path Node, Platooning, and System Monitoring examples. Added Overtravel Node and Moving Path Node descriptions.

Updated the following:

- Changed the revision to alpha (Rev. E) only.
- Updated all trademark and copyright information and moved to the back cover.
- Updated the titles and part numbers for all referenced manuals.

- Updated the appearance of the safety notices to match Rockwell Automation standards.
- In the About This Manual chapter, updated the descriptions of the safety notices and the list of Additional Resources.
- In Chapter 1, Introduction, updated the Transport System Components Overview, the Transport System Software Overview, and Getting Started with the TCP/IP Communication Protocol.
- In Chapter 2, Transport System Control, updated Vehicle Status to use the Get Extended Vehicle Status.
- Chapter 3, Application Notes, updated Moving Path Node to support current node.
- In Chapter 4, Protocol Reference, removed Follow ID option from Move Vehicle To Position. Updated the status responses in Command Status.
- In Chapter 5, Troubleshooting, updated the Motor Fault Troubleshooting, and the Contact Rockwell Automation Technical Support.
- Updated the Glossary and Index.

Removed the following:

- In Chapter 2, Transport System Control, removed Following Another Vehicle Through a Switch.
- In Chapter 3, Application Notes, removed the Shuttle node, which is deprecated.
- In Chapter 4, Protocol Reference, removed the *Extended Vehicle Status Request, V2 (0xBF 03 02)* and *Extended Vehicle Status, V2 (0xDF 03 02)*, which have been replaced by V3 to provide support for platooning.
- List of Table, List of Figures, Glossary, Manual Conventions, Transport System Limits, Audience, Manual Structure, and, Contact Rockwell Automation Support.

Consolidated the following publications into this document:

- MMI-AT031, Calculating the CRC for Messages using TCP/IP
- MMI-AT035, QuickStick 100 Startup using Moving Paths
- MMI-UM018, TCP/IP System Monitoring
- MMI-UM029, Suspect Vehicle Bit
- MMI-UM033, Moving Path Node
- MMI-UM036, Vehicle Platooning
- MMI-UM040, Managed Configurations

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, Knowledgebase, and product notification updates.	rok.auto/support
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Technical Documentation Center	Quickly access and download technical specifications, installation instructions, and user manuals.	rok.auto/techdocs
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	rok.auto/pcdc

Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)

At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental compliance information on its website at rok.auto/pec.

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) 2663 0600, Fax: (32) 2 663 0640
ASIA PACIFIC: Rockwell Automation SEA Pte Ltd, 2 Corporation Road, #04-05, Main Lobby, Corporation Place, Singapore 618494, Tel: (65) 6510 6608, FAX: (65) 6510 6699
UNITED KINGDOM: Rockwell Automation Ltd., Pitfield, Kiln Farm, Milton Keynes, MK11 3DR, United Kingdom, Tel: (44)(1908) 838-800, Fax: (44)(1908) 261-917

Publication MMI-UM003E-EN-P - October 2023
Supersedes Publication MMI-UM003D-EN-P - June 2018