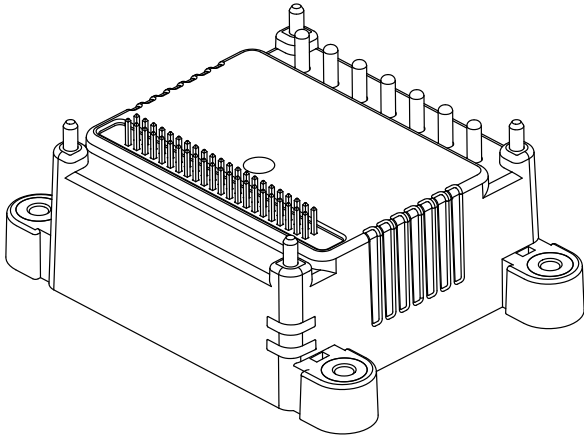


**PERFORMANCE
MOTION DEVICES**
MOTION CONTROL AT ITS CORE



ION®/CME N-Series Digital Drive

User Manual

Revision 1.1 / July 2025

Performance Motion Devices, Inc.

80 Central Street, Boxborough, MA 01719

www.pmdcorp.com



NOTICE

This document contains proprietary and confidential information of Performance Motion Devices, Inc., and is protected by federal copyright law. The contents of this document may not be disclosed to third parties, translated, copied, or duplicated in any form, in whole or in part, without the express written permission of PMD.

The information contained in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form, by any means, electronic or mechanical, for any purpose, without the express written permission of PMD.

Copyright 1998–2025 by Performance Motion Devices, Inc.

Juno, Atlas, Magellan, ION, Prodigy, Pro-Motion, C-Motion and VB-Motion are trademarks of Performance Motion Devices, Inc.

Warranty

Performance Motion Devices, Inc. warrants that its products shall substantially comply with the specifications applicable at the time of sale, provided that this warranty does not extend to any use of any Performance Motion Devices, Inc. product in an Unauthorized Application (as defined below). Except as specifically provided in this paragraph, each Performance Motion Devices, Inc. product is provided “as is” and without warranty of any type, including without limitation implied warranties of merchantability and fitness for any particular purpose.

Performance Motion Devices, Inc. reserves the right to modify its products, and to discontinue any product or service, without notice and advises customers to obtain the latest version of relevant information (including without limitation product specifications) before placing orders to verify the performance capabilities of the products being purchased. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement and limitation of liability.

Unauthorized Applications

Performance Motion Devices, Inc. products are not designed, approved or warranted for use in any application where failure of the Performance Motion Devices, Inc. product could result in death, personal injury or significant property or environmental damage (each, an “Unauthorized Application”). By way of example and not limitation, a life support system, an aircraft control system and a motor vehicle control system would all be considered “Unauthorized Applications” and use of a Performance Motion Devices, Inc. product in such a system would not be warranted or approved by Performance Motion Devices, Inc.

By using any Performance Motion Devices, Inc. product in connection with an Unauthorized Application, the customer agrees to defend, indemnify and hold harmless Performance Motion Devices, Inc., its officers, directors, employees and agents, from and against any and all claims, losses, liabilities, damages, costs and expenses, including without limitation reasonable attorneys’ fees, (collectively, “Damages”) arising out of or relating to such use, including without limitation any Damages arising out of the failure of the Performance Motion Devices, Inc. product to conform to specifications.

In order to minimize risks associated with the customer’s applications, adequate design and operating safeguards must be provided by the customer to minimize inherent procedural hazards.

Disclaimer

Performance Motion Devices, Inc. assumes no liability for applications assistance or customer product design. Performance Motion Devices, Inc. does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of Performance Motion Devices, Inc. covering or relating to any combination, machine, or process in which such products or services might be or are used. Performance Motion Devices, Inc.’s publication of information regarding any third party’s products or services does not constitute Performance Motion Devices, Inc.’s approval, warranty or endorsement thereof.

Patents

Performance Motion Devices, Inc. may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Patents and/or pending patent applications of Performance Motion Devices, Inc. are listed at <https://www.pmdcorp.com/company/patents>.

Related Documents

ION/CME N-Series Digital Drive Developer Kit User Manual

How to install, configure, and operate N-Series ION developer kits.

Magellan Motion Control IC User Guide

Complete description of the Magellan Motion Control IC features and functions with detailed theory of its operation.

C-Motion Magellan Programming Reference

Descriptions of all Magellan Motion Control IC commands, with coding syntax and examples, listed alphabetically for quick reference.

C-Motion Engine Development Tools Manual

Complete guide to developing software for PMD controller products. Includes examples on how to use the C-Motion Magellan, C-Motion PRP, and C-Motion PRP II C-language libraries, and details the recommended code development sequence for microcontroller-based, PC-based, and C-Motion Engine-based systems.

C-Motion PRP II Programming Reference

Describes C-Motion language function calls and associated PRP-formatted packets along with data types for communication between the host and ION/CME N-Series Digital Drives.

Table of Contents

1. Introduction	9
1.1 ION/CME N-Series Digital Drive Overview	9
1.2 Part Numbers and Configurations	10
1.3 Developer Kits	11
1.4 Feature List	12
1.5 Typical Applications	13
2. Specifications	21
2.1 Physical Dimensions	22
2.2 Configurations, Parameters, and Performance	24
2.3 Drive Ratings	26
2.4 Electrical Characteristics	27
2.5 Absolute Maximum Ratings	29
2.6 Environmental Ratings	29
2.7 Safety and Compliance	30
2.8 Timing Characteristics	30
2.9 Timing Diagrams	31
2.10 Pinout Quick Reference	32
2.11 Pin Descriptions	35
3. Operation	43
3.1 Function Summary	44
3.2 Magellan IC Functions	45
3.3 Encoder Functions	48
3.4 Motion Peripheral Functions	62
3.5 I/O Functions	67
3.6 Communications Functions	71
3.7 Power Stage	83
3.8 Drive Safety Functions	87
3.9 C-Motion Engine	90
3.10 General Functions	96
3.11 Device NVRAM	105
3.12 N-Series ION Powerup & Initialization	105
4. Installation and Production-Related	109
4.1 Mechanical Mounting	109
4.2 Production Unit Configuration	113
Appendix A. Scale Factors, Defaults and Limits	121
A.1 Power Level Related	121
A.2 N-Series ION NVRAM Default Settings	126
Appendix B. BiSS-C Encoder Settings	131
B.1 Renishaw Resolute Absolute Encoder, Linear	131
B.2 Renishaw Resolute Absolute Encoder, Angular	131
B.3 Parker Hannifin P-Series AC Servo Motor with BiSS-C Feedback	132
Appendix C. Application Notes & Example Schematics	133
C.1 General Design Notes	133
C.2 Power Supplies	134
C.3 RS232 Host Interface and Programming Port	136
C.4 RS485 Host Interface	138
C.5 CAN Host Interface	140
C.6 SPI Host Interface	142

C.7	Ethernet Host Interface	144
C.8	Expansion CAN Network and Motor Connection Setup	146
C.9	Expansion SPI Network	148
C.10	Connecting to an MC58113 IC	150
C.11	Digital IO Connection Schemes	152
C.12	Analog Input	154
Appendix D.Example Interconnect Board Layout.		157
D.1	Layer 1	157
D.2	Layer 2	158
D.3	Layer 3	159
D.4	Layer 4	160

List of Figures

1-1	N-Series ION Digital Drive	9
1-2	N-Series ION Development Kit	11
1-3	Position, Velocity, and Torque Control of Brushless DC and DC Brush Motors	13
1-4	Pulse & Direction Control of Step Motors	14
1-5	Pulse & Direction Control of Brushless DC and DC Brush Motors	15
1-6	Closed Loop Operation of Step Motors	15
1-7	Force Control	16
1-8	CAM Profile Control	16
1-9	Network Connected Drive	17
1-10	Network-Based Three-Axis Motion Controller Application	17
1-11	PCB-Based Four-Axis Motion Controller Application	18
1-12	Microprocessor Based Motion Controller Application	19
1-13	Single-Axis Standalone Controller Application	19
1-14	Two-Axis Controller with FPGA-Based Logic Expansion Application	20
2-1	ION/CME N-Series Digital Drive Physical Dimensions	22
2-2	Host SPI Timing	31
2-3	Expansion SPI Timing	31
3-1	ION/CME N-Series Digital Drive Internal Block Diagram	43
3-2	Quadrature AB Encoder Signal Waveform	49
3-3	QuadA1 & QuadB1 Internal Processing Circuitry	50
3-4	QuadA2, QuadB2, Index1, and Index2 Internal Processing Circuitry	51
3-5	Pulse & Direction Signal Format	51
3-6	Sin/Cos Encoder Signal Waveforms	52
3-7	Sin/Cos Signal Correction Calculation Flow	53
3-8	Example Correction Waveform	54
3-9	BiSS Encoder Format Interconnection Diagram and Serial Data Stream	58
3-10	BiSS Encoder Format Interconnection Diagram and Serial Data Stream	58
3-11	Enable, PosLim, NegLim, and Brake Signal Internal Processing Circuitry	63
3-12	FaultOut Internal Processing Circuitry	63
3-13	HallA, HallB, HallC and Home Signal Internal Processing Circuitry	64
3-14	Connection Schemes for Multiple Unit Synching	66
3-15	DigitalIO1-4 Internal Processing Circuitry	69
3-16	DigitalIO5-8 Internal Processing Circuitry	70
3-17	AnalogIn Internal Processing Circuitry	71
3-18	Daisy Chain Node ID Assignment Connection Scheme	73
3-19	Serial Communication Signals Internal Processing Circuitry	76
3-20	SPI Data Transaction Signals	80
3-21	SPI Master/Slave Data Buffering	82
3-22	PRP-based Connection Scheme Using Expansion SPI Port	82
3-23	Brushless DC Motor Bridge Configuration	84
3-24	DC Brush Motor Bridge Configuration	85
3-25	Two-phase Step Motor Bridge Configuration	85
3-26	Shunt Resistor Wiring Diagram	89
3-27	Block Diagram of C-Motion Engine	91
3-28	Reset Signal Internal Processing Circuitry	100
4-1	Mechanical Elements of N-Series ION Drive	109
4-2	N-Series ION Mounting Options	110
4-3	Recommended N-Series ION Thermal Transfer Material Dimensions	112
4-4	N-Series ION Programming Board	114
4-5	3-Pin Programming Interface Connection Diagram	116
4-6	Example PSF Script File Content	117
C-1	N-Series ION Power Supply Connections	135

C-2	N-Series ION RS232 Host Interface and Programming Port	137
C-3	N-Series ION RS485 Host Interface	139
C-4	N-Series ION CAN Host Interface	141
C-5	N-Series ION SPI Host Interface	143
C-6	Internal Schematic of RJ-45 with Integrated Magnetics	144
C-7	N-Series ION Ethernet host interface with integrated magnetics	145
C-8	N-Series ION Expansion CAN network and Motor Connections	147
C-9	N-Series ION Expansion SPI Connection	149
C-10	N-Series ION Expansion SPI with Slave MC58113	151
C-11	N-Series ION Digital IO Connection Schemes	153
C-12	N-Series ION Analog Signal Input Connection Schemes	155
D-1	PCB-1047-01 Layer 1	157
D-2	PCB-1047-01 Layer 2	158
D-3	PCB-1047-01 Layer 3	159
D-4	PCB-1047-01 Layer 4	160

1. Introduction

1

In This Chapter

- ▶ ION/CME N-Series Digital Drive Overview
- ▶ Part Numbers and Configurations
- ▶ Developer Kits
- ▶ Feature List
- ▶ Typical Applications

1.1 ION/CME N-Series Digital Drive Overview

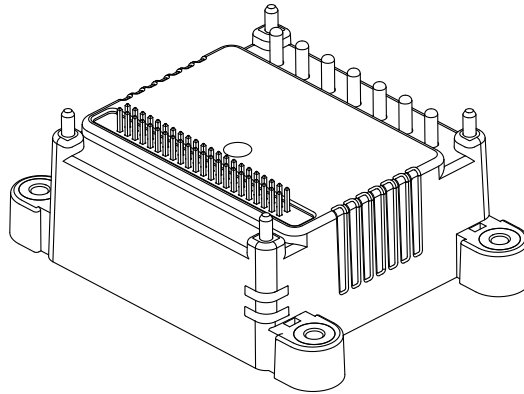


Figure 1-1:
N-Series ION
Digital Drive

This manual provides a complete user guide for the ION/CME N-Series Digital Drives. For documentation on other members of the ION Digital Drive family please consult the appropriate documentation.

ION/CME N-Series Digital Drives are single-axis motion controllers with integrated power electronics and network communications. Various models are available to drive DC Brush, Brushless DC, and step motors. Their very compact size, range of power output levels, and high level of connectivity make them an ideal solution for embedded or distributed motion control applications that require high performance in a small envelope.

ION N-Series Drives are based on PMD's Magellan Motion Control IC and perform profile generation, encoder position feedback, position servo compensation, step motor stall detection, brushless DC motor commutation, microstep generation, digital current/torque control, and more. All members of the ION family have integrated, high-power drive stages which fully protect from overcurrent, undervoltage, overvoltage, overtemperature, and short-circuit faults.

N-Series ION host communication options include Ethernet, CAN, RS232/RS485, and SPI (Serial Peripheral Interface). Each drive also supports an additional CAN and SPI expansion network for connecting to other N-Series ION or to other peripherals. All ION/CME N-Series Drives include a C-Motion Engine, allowing user application code to be downloaded and executed directly in the drive, along with NVRAM memory and trace memory for permanent and temporary storage of control parameters and performance trace results.

The N-Series ION Drives are PCB (printed circuit board) mounted and are packaged in a plastic and metal solderable module measuring 1.48" x 1.48" x .66" (37.6 mm x 37.6 mm x 16.8 mm). They come in three power levels; 75 watts, 350 watts, and 1,000 watts and utilize a 44 pin (2 x 22) 50-mil header for signal connections and a 7-pin high current connector for the DC bus and motor connections.

1.2 Part Numbers and Configurations

There are 36 different ION/CME N-Series Digital Drives in all, consisting of the combinations of four motor types (step motor, Brushless DC, DC Brush, Multi-Motor), three host interfaces (Serial, CAN/SPI, Ethernet), and three power levels (low, medium, high). Note that multi-motor units allow the motor type, either Brushless DC, DC Brush, or step motor, to be user programmed.

The following table shows the available N-Series ION part numbers:

P/N	Host Interface	Power Level	Voltage	Motor Type
Step Motor				
DD441S0056/02	Serial	Low (75W)	12-56V	Step Motor
DD441S0056/06	Serial	Medium (350W)	12-56V	Step Motor
DD441S0056/18	Serial	High (1,000W)	12-56V	Step Motor
DD441C0056/02	CAN/SPI	Low (75W)	12-56V	Step Motor
DD441C0056/06	CAN/SPI	Medium (350W)	12-56V	Step Motor
DD441C0056/18	CAN/SPI	High (1,000W)	12-56V	Step Motor
DD441D0056/02	Ethernet	Low (75W)	12-56V	Step Motor
DD441D0056/06	Ethernet	Medium (350W)	12-56V	Step Motor
DD441D0056/18	Ethernet	High (1,000W)	12-56V	Step Motor
Brushless DC				
DD431S0056/02	Serial	Low (75W)	12-56V	Brushless DC
DD431S0056/06	Serial	Medium (350W)	12-56V	Brushless DC
DD431S0056/18	Serial	High (1,000W)	12-56V	Brushless DC
DD431C0056/02	CAN/SPI	Low (75W)	12-56V	Brushless DC
DD431C0056/06	CAN/SPI	Medium (350W)	12-56V	Brushless DC
DD431C0056/18	CAN/SPI	High (1,000W)	12-56V	Brushless DC
DD431D0056/02	Ethernet	Low (75W)	12-56V	Brushless DC
DD431D0056/06	Ethernet	Medium (350W)	12-56V	Brushless DC
DD431D0056/18	Ethernet	High (1,000W)	12-56V	Brushless DC
DC Brush				
DD411S0056/02	Serial	Low (75W)	12-56V	DC Brush
DD411S0056/06	Serial	Medium (350W)	12-56V	DC Brush
DD411S0056/18	Serial	High (1,000W)	12-56V	DC Brush
DD411C0056/02	CAN/SPI	Low (75W)	12-56V	DC Brush
DD411C0056/06	CAN/SPI	Medium (350W)	12-56V	DC Brush
DD411C0056/18	CAN/SPI	High (1,000W)	12-56V	DC Brush
DD411D0056/02	Ethernet	Low (75W)	12-56V	DC Brush
DD411D0056/06	Ethernet	Medium (350W)	12-56V	DC Brush
DD411D0056/18	Ethernet	High (1,000W)	12-56V	DC Brush
Multi Motor				
DD481S0056/02	Serial	Low (75W)	12-56V	Multi Motor
DD481S0056/06	Serial	Medium (350W)	12-56V	Multi Motor
DD481S0056/18	Serial	High (1,000W)	12-56V	Multi Motor
DD481C0056/02	CAN/SPI	Low (75W)	12-56V	Multi Motor
DD481C0056/06	CAN/SPI	Medium (350W)	12-56V	Multi Motor
DD481C0056/18	CAN/SPI	High (1,000W)	12-56V	Multi Motor
DD481D0056/02	Ethernet	Low (75W)	12-56V	Multi Motor
DD481D0056/06	Ethernet	Medium (350W)	12-56V	Multi Motor
DD481D0056/18	Ethernet	High (1,000W)	12-56V	Multi Motor

1.3 Developer Kits

N-Series IONs are available in a developer kit configuration. Each developer kit comes with the specified N-Series ION soldered onto an interconnect board and mounted on a baseplate/heatsink for convenient use on a bench top surface.

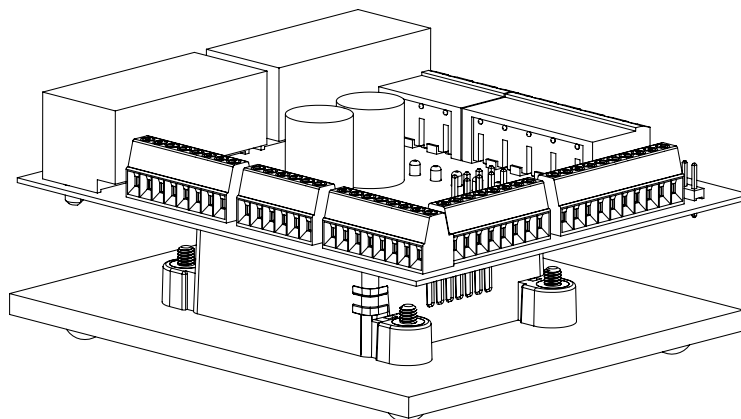


Figure 1-2:
N-Series ION
Development
Kit

The DK interconnect boards measure approximately 3.25" x 2.75" and are compact enough that if not used on a bench top they are suitable for mounting directly on the actual machine hardware during prototyping or even for the production application.

For complete information on how to install and operate developer kits for N-Series IONs refer to the *ION/CME N-Series Digital Drive Developer Kit User Manual*.

1.3.1 Developer Kit Part Numbers

1.3.1.1 Pre-Assembled Developer Kits

The following N-Series ION Developer Kits are fully assembled and installed with a multi-motor ION unit type. These various P/Ns represent three different host interface types (serial, CAN/SPI, Ethernet) and three different power levels (low, medium, high):

DK P/N	ION P/N Installed	Host Interface	Power Level
DK481S0056/02	DD481S0056/02	Serial	Low
DK481S0056/06	DD481S0056/06	Serial	Medium
DK481S0056/18	DD481S0056/18	Serial	High
DK481C0056/02	DD481C0056/02	CAN/SPI	Low
DK481C0056/06	DD481C0056/06	CAN/SPI	Medium
DK481C0056/18	DD481C0056/18	CAN/SPI	High
DK481D0056/02	DD481D0056/02	Ethernet	Low
DK481D0056/06	DD481D0056/06	Ethernet	Medium
DK481D0056/18	DD481D0056/18	Ethernet	High

1.3.1.2 Component Developer Kits

There may be occasions where it is preferable for the user to assemble an N-Series ION developer kit themselves. The main advantage of this is that it allows developer kit setups to be created with N-Series IONs other than the multi-motor type.

To facilitate this three non pre-assembled component developer kits are available as shown in the table below:

P/N	Host Interface	Comments
DK4X1S	Serial	Supports all serial host interface N-Series ION units.
DK4X1C	CAN/SPI	Supports all CAN/SPI host interface N-Series ION units.
DK4X1D	Ethernet	Supports all Ethernet host interface N-Series ION units.

Each of these component-only DKs contain a DK interconnect PCB, an aluminum base plate, and various mounting hardware such as screws and a thermal pad to allow the user to assemble a complete DK setup. Note that these DKs P/Ns do not include the N-Series ION unit itself. The N-Series ION must be ordered separately.

1.4 Feature List

The ION/CME N-Series Digital Drives provide a wealth of motion control capabilities and features. Here is a summary:

Overview

- Controls Brushless DC, Step Motor, and DC Brush motors
- Three power levels available; 75W, 350W, 1,000W
- 12-56V single DC supply operation
- Magellan Motion Control IC engine provides high performance positioning, current control, and drive safety features
- User downloadable code allows complete customizability of protocol processing and machine controller behavior
- Host-connected or fully autonomous operation
- Two status indicator LEDs
- Real time clock
- 8 general purpose digital I/Os
- 1 general purpose analog input

Communications

- Four available host connections; Ethernet, Serial, CAN FD, and SPI (Serial Peripheral Interface)
- Two expansion networks; CAN FD and SPI for ION to ION communication or communication to other devices and peripherals
- 100BASE-T Ethernet
- Support for UDP & TCP protocols with DHCP (Dynamic Host Configuration Protocol)
- RS232 & RS485 asynchronous serial

Motion Control IC

- Advanced PID filter with velocity and acceleration feedforward
- Programmable position & PID loop rate up to 50 μ Sec
- 50 μ Sec commutation and current loop rate
- Dual biquad filters
- Autotuning for easy setup and optimal operating performance
- 32KB Internal trace buffer with simultaneous capture of up to four variables
- Incremental quadrature encoder input up to 32.0 Mcounts/sec

- Sin/cos encoder support with up to 16,384 x interpolation factor
- SSI and BiSS-C format encoder support
- Second (auxiliary) axis of encoder input with quadrature A, B and index inputs
- Hall sensor inputs, two directional limit switches, index input, home input, *AxisIn* and *AxisOut* signals

C-Motion Engine

- High speed user code execution module
- Multitasking
- 256 Kbytes of NVRAM user program space
- 256 Kbytes user stack space
- Event manager
- 428 MIPS code execution speed

Amplifier & Safety

- Ultra efficient fully digital switching power amplifier
- Proprietary current/torque technique provides quiet and efficient motor operation
- Field-oriented control & space vector modulation
- I^2t current foldback management
- Selectable 20, 40, 80, or 120 kHz PWM frequency operation
- Continuous overvoltage, undervoltage, overcurrent, short circuit, and overtemperature monitoring & protection

1.5 Typical Applications

The following sections provide information on typical applications utilizing ION/CME N-Series Digital Drives.

1.5.1 Position, Velocity, and Torque Control of Brushless DC and DC Brush Motors

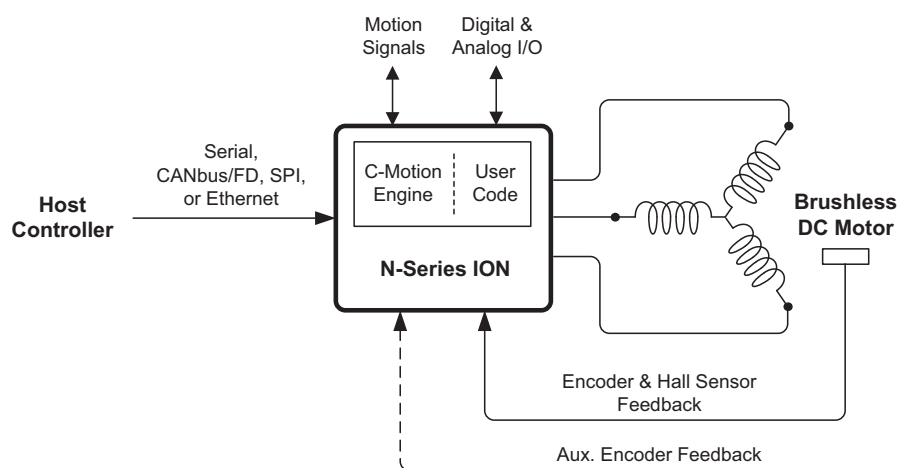


Figure 1-3:
Position,
Velocity, and
Torque Control
of Brushless
DC and DC
Brush Motors

In this application the ION/CME N-Series Digital Drive receives commands from a host via an RS232, RS485, CAN, SPI (Serial Peripheral Interface), or Ethernet network and provides high performance profile generation and position, velocity, and torque control of DC Brush or Brushless DC motors via a PID (Proportional, Integral, Derivative) or PI

servo control loop. Alternatively, command sequences can originate from user code downloaded into the ION unit's C-Motion Engine.

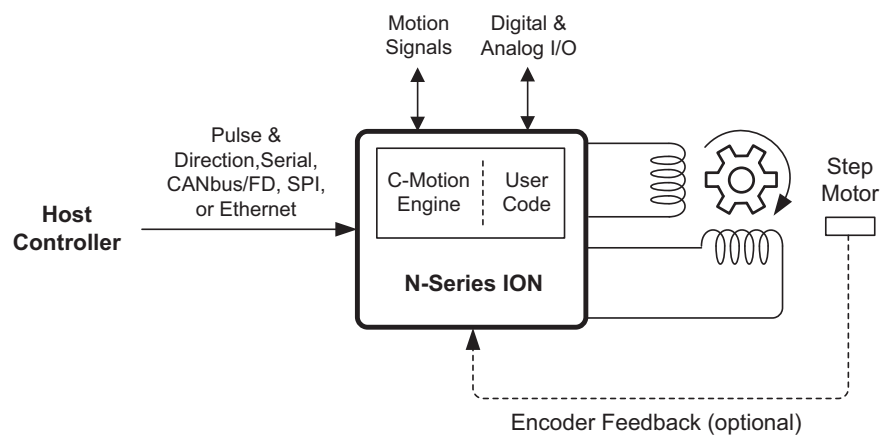
Quadrature encoder, sin/cos, or BiSS-C format encoders or Hall signals are used for position feedback. For Brushless DC motors Hall sensors normally provide commutation feedback. If encoder signals are available however Halls are not required as long as the motor can move freely during startup. Additional signals include home & index inputs, directional limit switches, general purpose digital and analog I/O, SynchIn & SynchOut, Enable, Brake, FaultOut signals and an auxiliary encoder input channel.

Note that in the diagram above a Brushless DC motor is shown but similar control can be provided for DC Brush motors.

Additional Hardware Needed: The N-Series ION is powered directly via a single + HV voltage and provides all electrical functionality for this application. The only additional hardware needed is an interconnect board to connect the N-Series ION's power, motor, and signal connections to external connectors. This functionality is typically provided via a user-designed interconnect board or the N-Series ION Developer Kit board.

1.5.2 Position & Velocity Control of Step Motors

**Figure 1-4:
Pulse &
Direction
Control of Step
Motors**



In this application the ION/CME N-Series Digital Drive receives commands from a host via pulse & direction signals, RS232, RS485, CAN, SPI (Serial Peripheral Interface), or Ethernet network and provides high performance profile generation and position, velocity, and torque control of step motors. Alternatively command sequences can originate from user code downloaded into the ION unit's C-Motion Engine. Encoder feedback for stall detection is supported but optional.

Additional signals include home & index inputs, directional limit switches, general purpose digital and analog I/O, SynchIn & SynchOut, Enable, Brake, FaultOut signals and an auxiliary encoder input channel.

Additional Hardware Needed: The N-Series ION is powered directly via a single + HV voltage and provides all electrical functionality for this application. The only additional hardware needed is an interconnect board to connect the N-Series ION's power, motor, and signal connections to external connectors. This functionality is typically provided via a user-designed interconnect board or the N-Series ION Developer Kit board.

1.5.3 Pulse & Direction Control of Brushless DC and DC Brush Motors

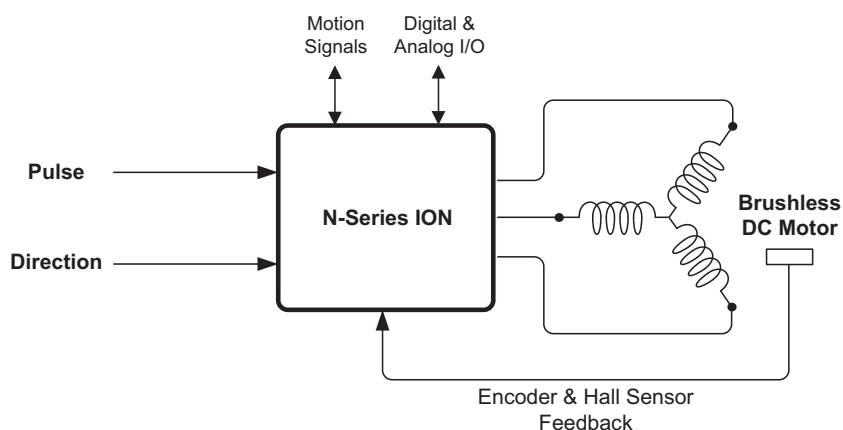


Figure 1-5:
Pulse &
Direction
Control of
Brushless DC
and DC Brush
Motors

In this application a PLC (programmable logic controller), motion control board or other external controller provides pulse and direction commands to the N-Series ION which controls a servo motor such as a DC Brush or Brushless DC motor. The incoming pulse & direction signals provide the position command which is used together with a PID (Proportional, Integral, Derivative) loop to provide position control. Quadrature, sin/cos, or BiSS-C format encoders or Hall sensor feedback is used for servo motor position feedback. For Brushless DC motors Hall sensors normally provide commutation feedback. If encoder signals are available however Halls are not required as long as the motor can move freely during startup.

Additional Hardware Needed: The N-Series ION is powered directly via a single + HV voltage and provides all electrical functionality for this application. The only additional hardware needed is an interconnect board to connect the N-Series ION's power, motor, and signal connections to external connectors. This functionality is typically provided via a user-designed interconnect board or the N-Series ION Developer Kit board.

1.5.4 Closed Loop Operation of Step Motors

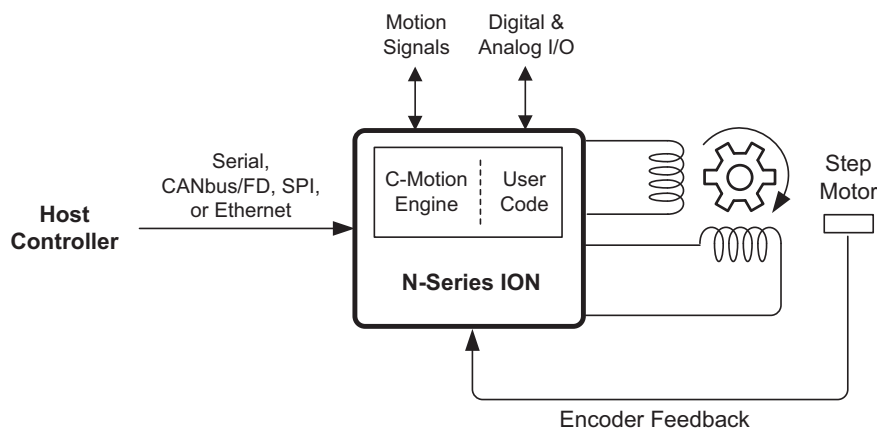


Figure 1-6:
Closed Loop
Operation of
Step Motors

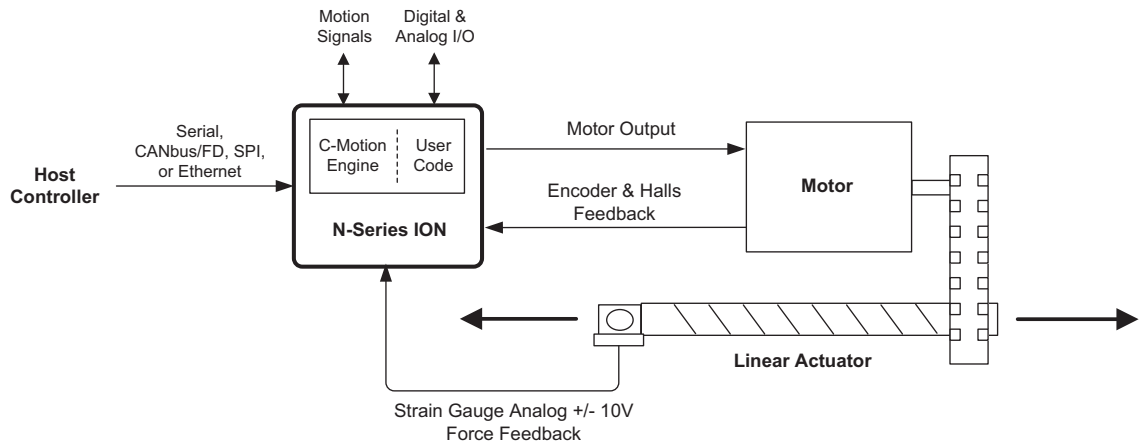
In this application the N-Series ION executes closed loop control (also called servo stepper operation) of a step motor. In this control mode the step motor is operated as a commutated two-phase servo motor. Relative to traditional microstepping step motor operation closed loop control provides smoother motion, less noise, higher acceleration, and eliminates the problem of lost steps.

Encoder input is required for closed loop step motor operation. Other than the approach toward step motor control, the features provided with this configuration are the same as for previously shown configurations.

Additional Hardware Needed: The N-Series ION is powered directly via a single + HV voltage and provides all electrical functionality for this application. The only additional hardware needed is an interconnect board to connect the N-Series ION's power, motor, and signal connections to external connectors. This functionality is typically provided via a user-designed interconnect board or the N-Series ION Developer Kit board.

1.5.5 Force Control With Brushless DC or DC Brush Motors

**Figure 1-7:
Force Control**



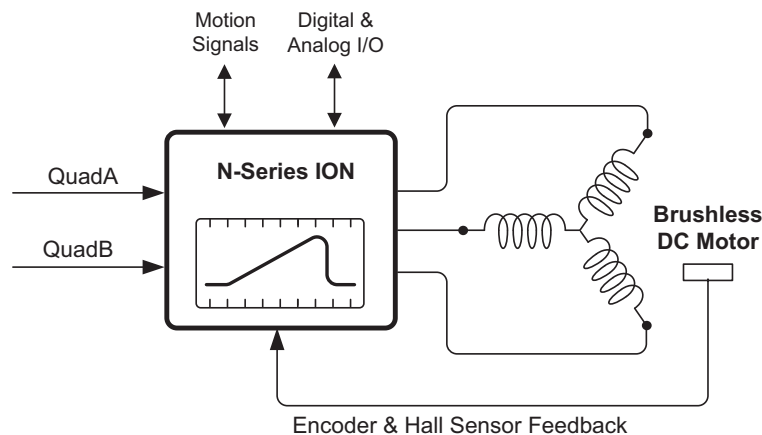
In this application the N-Series ION receives host commands from an external network or internally from downloaded user code representing a desired force. An external force sensor (strain gauge) provides an analog feedback signal which is input directly to the N-Series ION and used to create a corrective motion command which is then output by the ION to achieve precise force control.

Note that in the diagram above a linear actuator with a strain gauge is shown however there are many 'outer loop' control applications that can similarly be controlled using the N-Series ION. These include pressure control, temperature control, magnetic bearing control, liquid level control, and more. In all cases a feedback signal measures the desired quantity to be controlled and the ION commands the motor or coil which has a mechanical arrangement such that changes in its spin rate, commanded position, or field strength increase or decrease the measured quantity.

Additional Hardware Needed: The N-Series ION is powered directly via a single + HV voltage and provides all electrical functionality for this application. The only additional hardware needed is an interconnect board to connect the N-Series ION's power, motor, and signal connections to external connectors. This functionality is typically provided via a user-designed interconnect board or the N-Series ION Developer Kit board.

1.5.6 CAM Profile Control of Brushless DC, DC Brush, or Step Motors

**Figure 1-8:
CAM Profile Control**



In this application the N-Series ION inputs a quadrature position data stream from an external encoder via its Auxiliary encoder input and uses it as the master rate command for execution of an internally stored cam or electronic gear profile. Prior to profile operation the cam shape is stored by the user in the ION unit. Execution of camming functions and of other complex shapes is enabled via PMD's User Defined Profile Mode.

The above diagram shows a Brushless DC motor but similar control can be provided for DC Brush and step motors.

Additional Hardware Needed: The N-Series ION is powered directly via a single + HV voltage and provides all electrical functionality for this application. The only additional hardware needed is an interconnect board to connect the N-Series ION's power, motor, and signal connections to external connectors. This functionality is typically provided via a user-designed interconnect board or the N-Series ION Developer Kit board.

1.5.7 Network Connected Drive

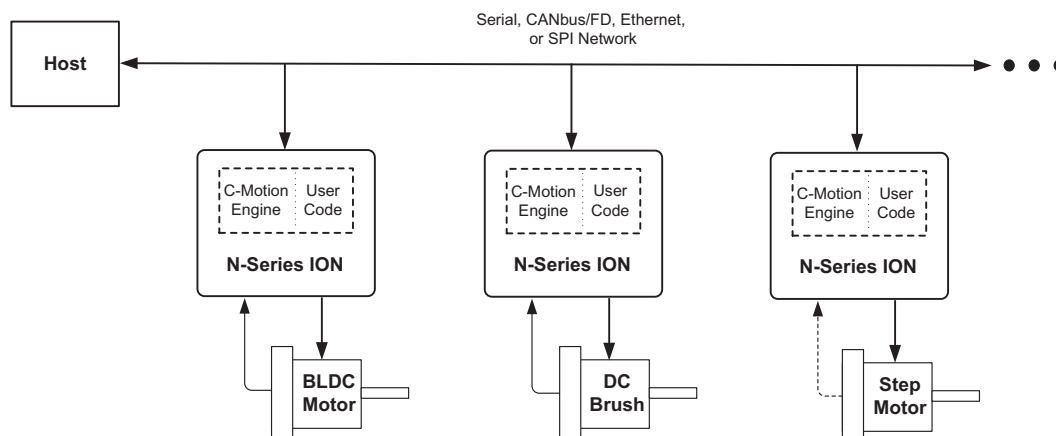


Figure 1-9:
Network
Connected
Drive

In this application one or more N-Series IONs connect directly to an RS422, RS485, CAN, Ethernet or SPI (Serial Peripheral Interface) network and provide control of Brushless DC, DC Brush, or step motors. Commands are sent by the host to each N-Series ION to accomplish the overall machine function. In addition to N-Series IONs other PMD network-attached devices may be added to the network or non-PMD devices as well.

In some such distributed applications, the user application software is executed only in the host. In other applications, one or more N-Series IONs may also execute user-written code. Typically this would be done to distribute the machine's overall computation load, to improve performance by locating the code in the drive itself, or to implement protocol translation if the host network must speak a particular protocol. Whatever the function of the code running in the N-Series ION, it is created by the user using PMD's C-Motion libraries and downloaded to the N-Series ION, where it is kept in NVRAM.

1.5.8 Machine Controller, Network Based

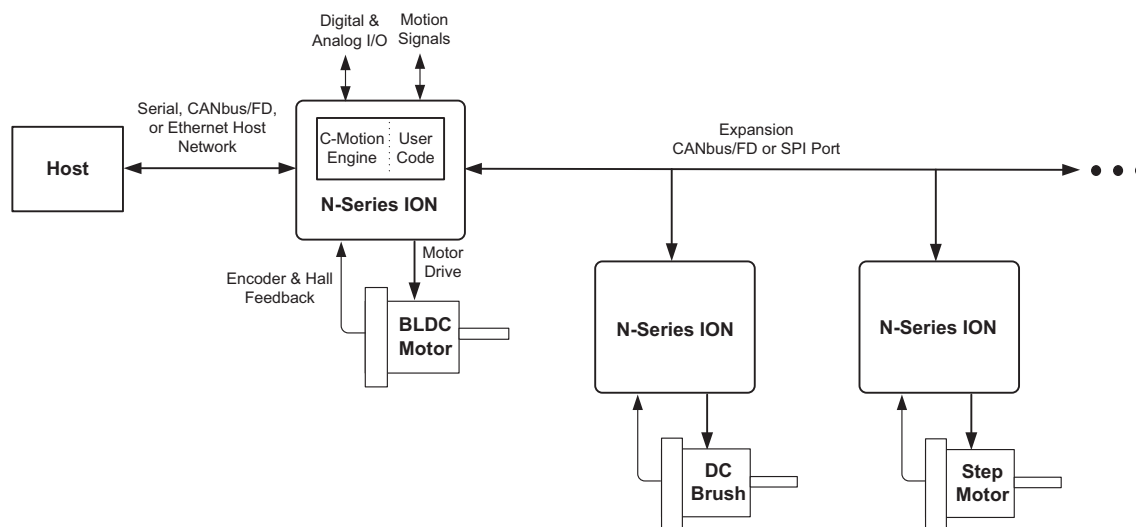


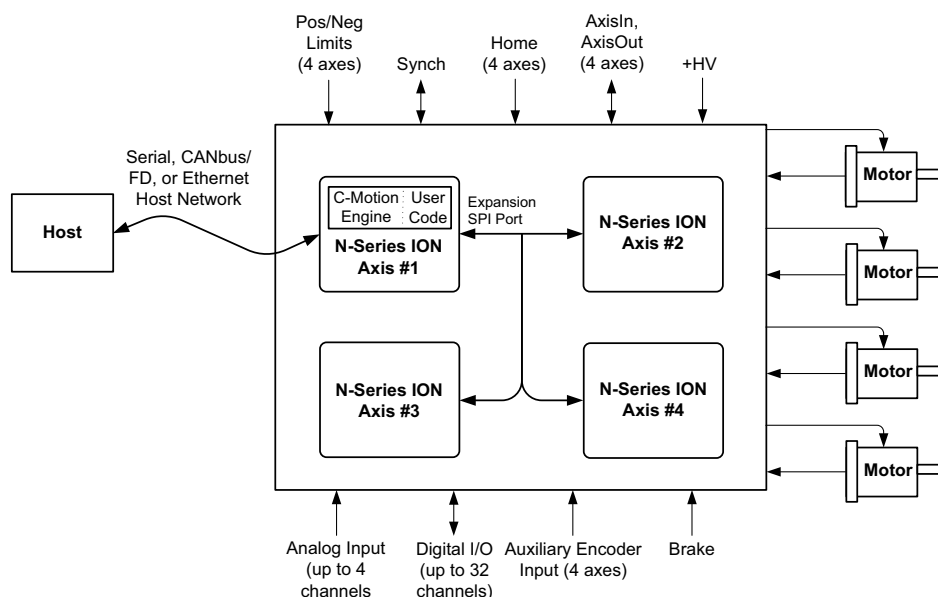
Figure 1-10:
Network-Based
Three-Axis
Motion
Controller
Application

In this application an N-Series ION interfaces directly to an RS422, RS485, CAN, or Ethernet host network and acts as the host network command processor and master machine behavior coordinator. This N-Series ION, which directly controls one motor axis, connects to two or more additional N-Series IONs via its expansion CAN or SPI network.

Use of the expansion network in this way allows creation of a virtual multi-axis machine controller. Commands to this virtual multi-axis machine are processed by the master N-Series ION, and motion sequences are partitioned as needed to the additional N-Series IONs via the expansion network. The code to accomplish this is created by the user using PMD's C-Motion libraries and is downloaded to the master N-Series ION's CME (C-Motion Engine).

1.5.9 Machine Controller, PCB-Based

**Figure 1-11:
PCB-Based
Four-Axis
Motion
Controller
Application**



In this application an N-Series ION interfaces directly to a RS422, RS485, CAN, or Ethernet network and acts as the host network command processor and master machine behavior coordinator. This N-Series ION, which directly controls one motor axis, connects to three additional N-Series IONs via its expansion SPI or CAN port. This application is similar to [Section 1.5.8, "Machine Controller, Network Based,"](#) except that rather than being cable-connected the IONs are all mounted on the same PCB (Printed Circuit Board).

Use of the N-Series ION's SPI expansion network in this way allows creation of a virtual multi-axis machine controller card. Commands to this virtual multi-axis machine are processed by the master N-Series ION, and motion sequences are partitioned as needed to the additional N-Series IONs via the expansion network. The code to accomplish this is created by the user using PMD's C-Motion libraries and is downloaded to the master N-Series ION's CME (C-Motion Engine).

1.5.10 Microprocessor Based Motion Controller, PCB-Based

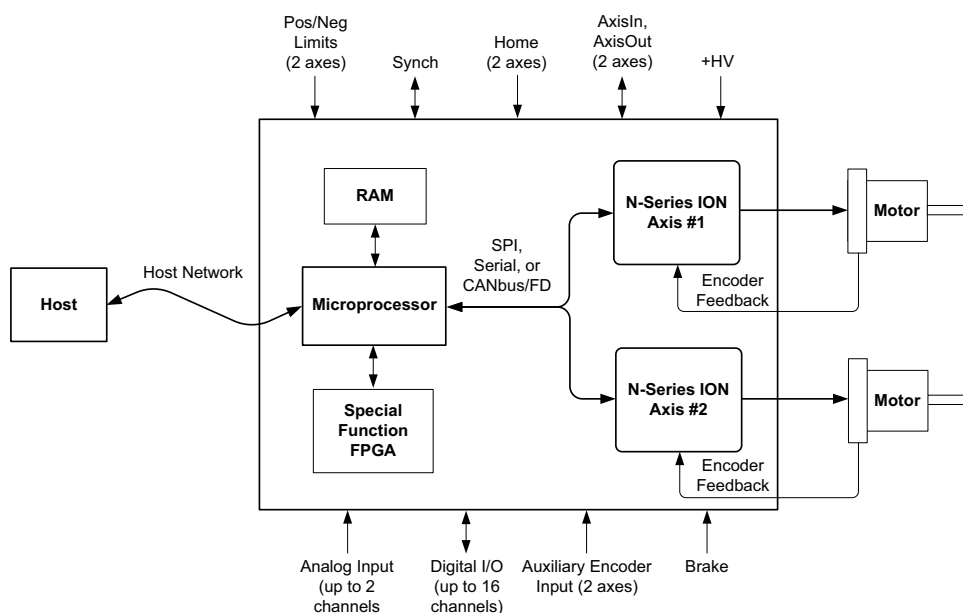


Figure 1-12:
Microprocessor
Based Motion
Controller
Application

In this application a microprocessor connects to one or more N-Series IONs via an SPI, serial, or CAN connection on a single PCB. In this application the user code resides in the microprocessor which commands the overall motion sequences. This application is similar to the [Section 1.5.9, "Machine Controller, PCB-Based,"](#) except that a microprocessor connects to the host network rather than an N-Series ION.

Use of a separate microprocessor for motion command sequencing may be useful especially where special computational requirements for the application exist or where special peripherals on the card need to be interfaced to that can not be accommodated by an N-Series ION.

1.5.11 Standalone Controller with User Interface

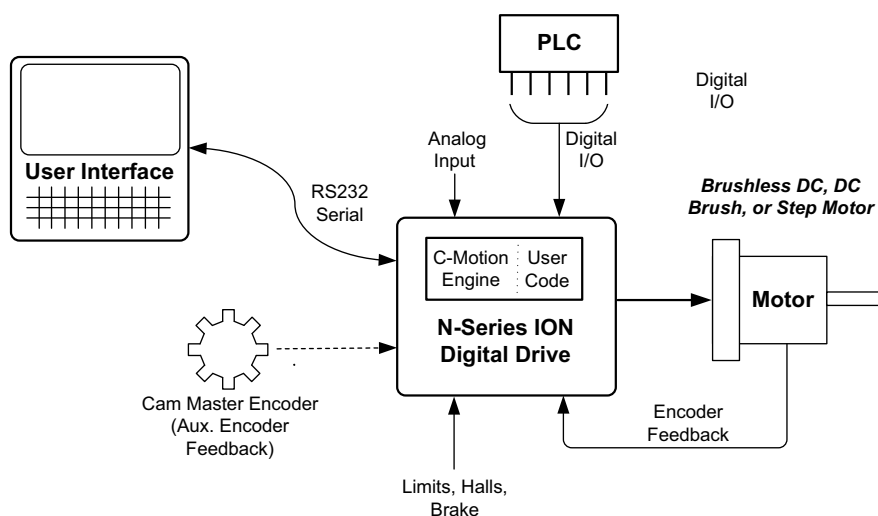


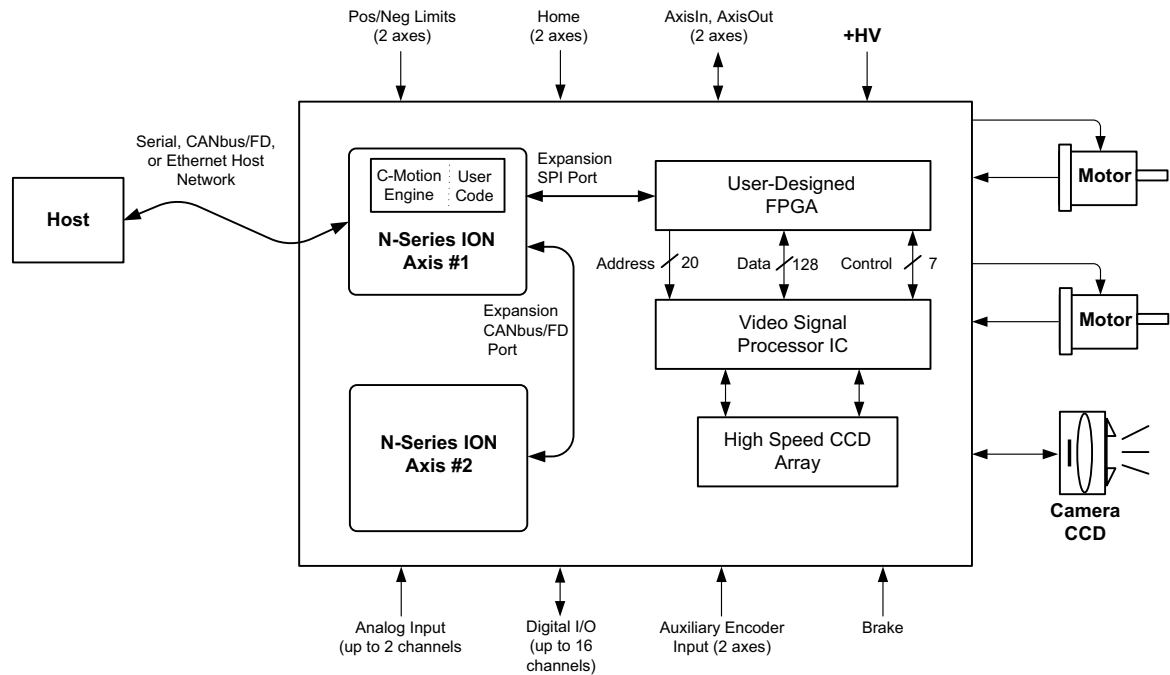
Figure 1-13:
Single-Axis
Standalone
Controller
Application

In this application an N-Series ION functions as a complete standalone controller. The Serial N-Series ION connects via its RS232 port to a user interface module which allows an operator to configure and control a cam profile function. The cam master encoder data stream is input via the N-Series ION's auxiliary encoder input. Finally, digital I/Os from a PLC select various cam functions to execute.

User created and downloaded code executing in the N-Series ION's C-Motion Engine provides the overall controller functionality including controlling the user interface and providing the PLC bit-command processing.

1.5.12 FPGA Access Using SPI Port

Figure 1-14:
Two-Axis
Controller with
FPGA-Based
Logic
Expansion
Application



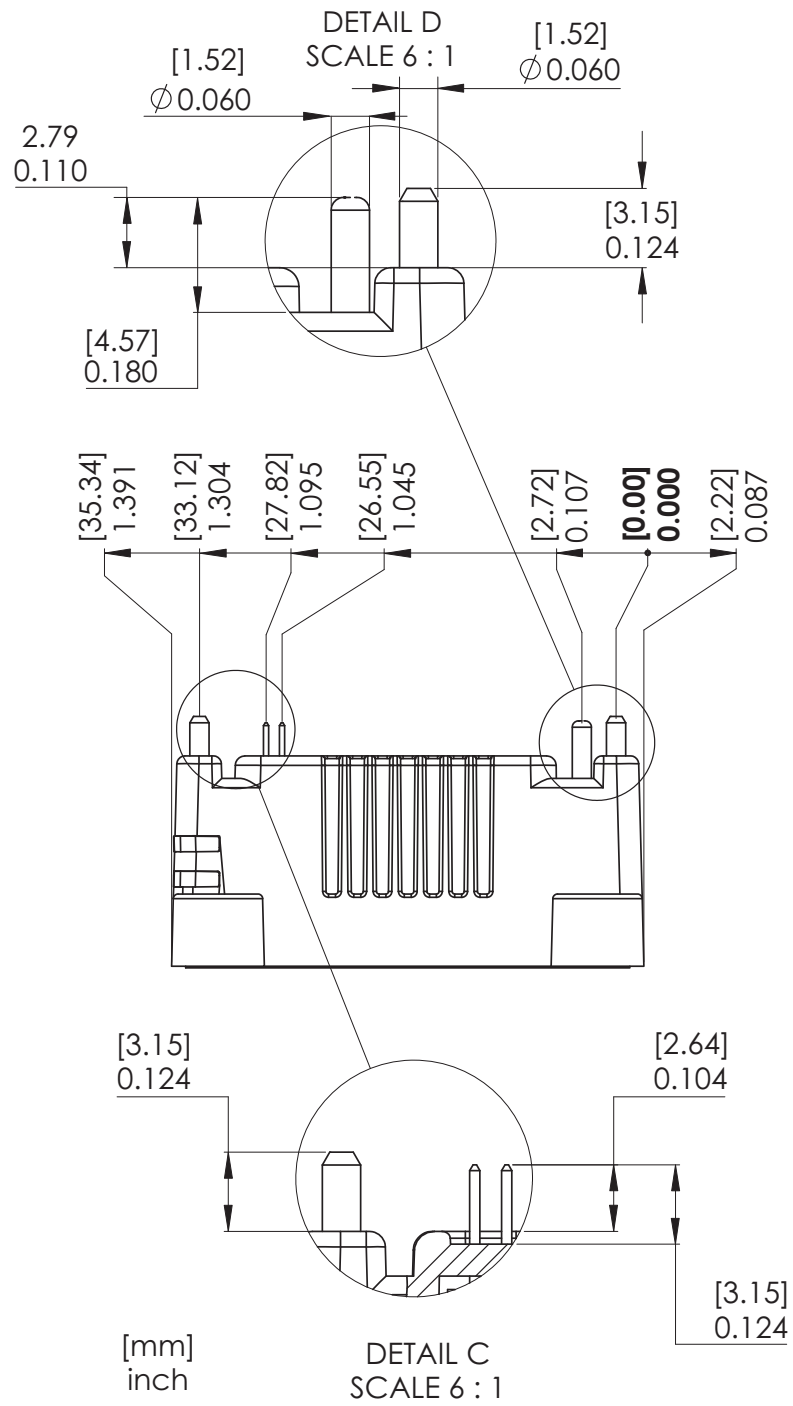
In this application an N-Series ION's SPI expansion network is used to address a user-designed FPGA (field programmable gate array) which contains specialized high speed logic utilized for application-dependent signal processing. This application demonstrates that the N-Series ION's expansion SPI port can be used as a general purpose memory or logic expansion port, greatly expanding the possibilities of on-card processing circuitry while still allowing control by the N-Series ION executing user created and downloaded C-Motion code.

2. Specifications

2

In This Chapter

- ▶ Physical Dimensions
- ▶ Configurations, Parameters, and Performance
- ▶ Drive Ratings
- ▶ Electrical Characteristics
- ▶ Absolute Maximum Ratings
- ▶ Environmental Ratings
- ▶ Safety and Compliance
- ▶ Timing Characteristics
- ▶ Timing Diagrams
- ▶ Pinout Quick Reference
- ▶ Pin Descriptions



2.2 Configurations, Parameters, and Performance

Supported motor types	DC Brush, Brushless DC, Step motor
Profile modes	S-curve point-to-point - Position, velocity, acceleration/ deceleration, jerk Trapezoidal point-to-point - Position, velocity, acceleration, deceleration Velocity-contouring - Velocity, acceleration, deceleration Electronic gearing - gear ratio
Position loop	Scalable PID with velocity and acceleration feedforward, integra- tion limit, offset bias, dual biquad filter, and settable derivative sam- pling time. Also supports dual encoder feedback.
Current loop	FOC (Field Oriented Control) with space-vector PWM, leg current sensing, scalable PI with integration limit and torque/current limit
Motion Control IC NVRAM	4 KB
Motion Control IC trace RAM	32 KB
Maximum number of simultaneous trace variables	4
Commutation rate	19.53 kHz
Current loop rate	19.53 kHz
PWM frequency	20 kHz, 40 kHz, 80 kHz, 120 kHz (user selectable)
Servo cycle time range	51.2 μ Seconds (19.53 kHz) to 1.114 seconds
Position range	-2,147,483,648 to +2,147,483,647 counts or steps
Velocity range	-32,768 to +32,767 counts or steps per cycle with a resolution of 1/65,536 counts or steps per cycle
Acceleration and deceleration ranges	0 to +32,767 counts or steps per cycle ² with a resolution of 1/ 65,536 counts or steps per cycle ²
Jerk range	0 to $\frac{1}{2}$ counts or steps per cycle ³ with a resolution of 1/ 4,294,967,296 counts or steps per cycle ³
Electronic gear ratio range	-32,768 to +32,767 with a resolution of 1/65,536 (negative and positive direction)
Position error tracking	Motion error window allows axis to be stopped upon exceeding pro- grammable window Tracking window allows flag to be set if position error is within a programmable position window Axis settled function allows flag to be set if position error is within programmable position window for a programmable amount of time after trajectory motion is complete
Multi-axis synchronization	< 1 μ sec difference between master and slave servo cycle
Current measurement resolution	12 bits
Brushless DC commutation sources	Hall sensors, encoder
Microstepping resolution	Up to 256 microsteps per full step

Axis control signals	Enable, AxisIn, AxisOut, Synch, HostInterrupt, PosLim, NegLim, Home, FaultOut
Motor amplifier	Four quadrant MOSFET-based switching amplifier with individual leg current sensing
Current control options	Individual phase, FOC (Field Oriented Control), third leg floating
Drive safety functions	Over current detect, over temperature detect, over voltage detect, under voltage detect, I ² t current foldback, Brake signal, 5V overcurrent detect
Brake modes	Passive braking, full disable
Shunt resistor control	Yes (for DC Brush and Brushless DC motors)
Host communications options	Serial, CAN, Ethernet, SPI (Serial Peripheral Interface)
Serial ports	Serial1, Serial2 (serial host interface unit only), Serial3 (all units)
Serial protocols supported	RS232 (Serial1, Serial2), RS422 (Serial1), RS485 half duplex (Serial1), RS485 full duplex (Serial1), UART (Serial3)
Serial port baud range	1,200 to 921, 600 (Serial1, Serial2, Serial3)
CAN ports	HostCAN (CAN/SPI host interface unit only), ExpCAN (all units)
CAN protocols supported	CAN FD (HostCAN, ExpCAN)
CAN FD nominal baud rate	50,000 to 1,000,000
CAN FD data baud rate	50,000 to 5,000,000
SPI ports	HostSPI (CAN/SPI host interface unit only), ExpSPI (all units)
SPI clock range	0.10 to 20.0 MHz
Expansion SPI clock frequency	625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz, 20 MHz
Ethernet port	100 Base-T Ethernet (Ethernet host interface unit only)
Ethernet protocols supported	TCP, UDP, DHCP
Encoder formats supported	Quadrature AB, pulse & direction, sin/cos, BiSS-C, SSI
Position capture signal sources	Index, home
Quadrature max rate	32 Mcounts/sec
Pulse input max rate	10 Mpulses/sec
Sin/Cos max frequency	10 MHz
Sin/Cos interpolation factor range	4 to 65,536
Sin/Cos calibration settings	Sin analog offset, cos analog offset, Sin analog scale, Cos analog scale, Kphase, 32-entry derived angle correction table
C-Motion engine CPU speed	428 MIPS
User program space NVRAM	256 KB
User stack RAM	256 KB
User scratch RAM	64 KB
Max # of user tasks	16

High priority task response time	5.0 μ Sec or less
Nominal task switch time	1.0 mSec
Real time clock accuracy	+/-50 ppm
Powered-off battery voltage	1.5V-3.3V
Powered-off battery draw	2 μ A @ 1.5V, 5 μ A @ 3V, 8 μ A @ 3.3V
General purpose device NVRAM size	128 KB
General purpose device RAM size	64 KB
General purpose digital I/Os	8 bi-directional digital I/Os (4 push-pull, 4 open collector)
General purpose analog input	1 channel differential +/-10V analog input
Analog input measurement resolution	16 bits
Weight of N-Series ION unit	32.5 grams

2.3 Drive Ratings

2.3.1 Low Power Units (P/Ns DD4x1x0056/02)

Specifications*	DC Brush Motor	Brushless DC Motor	Step Motor
Nominal supply voltage	12-56 VDC	12-56 VDC	12-56 VDC
Continuous phase current	1.5 ADC	1.5 Arms	1.5 Arms
Peak current (per phase)	3.8 A	3.8 A	3.8 A
Maximum continuous power	84 W	103 W	119 W

* Transformer isolated power supply, $T < 40^{\circ}C$

A coldplate or a heatsink in an environment with sufficient airflow can be used to achieve the above drive ratings.

For temperature operation beyond the standard 0-40 $^{\circ}$ C range above-listed ratings may change. Contact your PMD representative for additional information on extended temperature operation including higher temperature drive ratings.

2.3.2 Medium Power Units (P/Ns DD4x1x0056/06)

Specifications*	DC Brush Motor	Brushless DC Motor	Step Motor
Nominal supply voltage	12-56 VDC	12-56 VDC	12-56 VDC
Continuous phase current	7.1 ADC	5.5 Arms	5.0 Arms
Peak current (per phase)	12.0 A	12.0 A	12.0 A
Maximum continuous power	396 W	379 W	396 W

* Transformer isolated power supply, $T < 40^{\circ}C$

A coldplate or a heatsink in an environment with sufficient airflow can be used to achieve the above drive ratings.

For temperature operation beyond the standard 0-40 $^{\circ}$ C range above-listed ratings may change. Contact your PMD representative for additional information on extended temperature operation including higher temperature drive ratings.

2.3.3 High Power Units (P/Ns DD4x1x0056/18)

Specifications*	DC Brush Motor	Brushless DC Motor	Step Motor
Nominal supply voltage	12-56 VDC	12-56 VDC	12-56 VDC
Continuous phase current	19.0 ADC	14.8 Arms	13.4 Arms
Peak current (per phase)	36.0 A	36.0 A	36.0 A
Maximum continuous power	1,064 W	1,018 W	1,064 W

* Transformer isolated power supply, $T < 40^{\circ} \text{C}$.

A coldplate or a heatsink in an environment with sufficient airflow can be used to achieve the above drive ratings.

For temperature operation beyond the standard 0-40° C range above-listed ratings may change. Contact your PMD representative for additional information on extended temperature operation including higher temperature drive ratings.

2.4 Electrical Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Description
Supply						
V_{HV}	Supply voltage	12		56	V	Supplies motor
V_{HV_Aux}	Auxiliary supply voltage	12		56	V	HV-Aux supplies internal logic Typical current shown at 24V V_{HV_Aux}
I_{HV_Aux}	Auxiliary supply current		0.07	0.5	A	
Host SPI Signals						
	Signals					HostSPIClock, HostSPIRcv, HostSPISelect
	Input voltage range	0		5	V	TTL compatible
V_{IH_IO1}	Input high	2			V	
V_{IL_IO1}	Input low			0.8	V	
	Signals					HostSPIXmt
V_{OH_SPI}	Output voltage high	2		5	V	TTL compatible
V_{OL_SPI}	Output voltage low	0		0.8	V	
	Clock Frequency			20	MHz	See Section 2.8, "Timing Characteristics," for timing information
RS232/RS485 Communications						
	Signals					Srl2Xmt/RS485Xmt+, Srl2Xmt/RS485Xmt-, Srl1Rcv/RS485Rcv+, Srl1Rcv/RS485Rcv-
	Termination					Programmable
	Isolation					None
Differential Encoder Inputs						
	Signals					Main encoder (A+, A-, B+, B-, Index+, Index-) Auxiliary encoder (A+, A-, B+, B-)
	Input voltage range	0		5	V	RS422 compatible
	Maximum frequency			10	MHz	
Sin Cos Encoder Inputs						
	Signals					Sin Cos encoder (Sin+, Sin-, Cos+ Cos-, Index+, Index-)

Symbol	Parameter	Min	Typ	Max	Units	Description
	Input voltage range	0		5	V	
	Maximum frequency			10	MHz	
Dedicated Digital Inputs						
	Signals					Hall A, Hall B, Hall C, Home
	Input voltage range	0		24	V	
V _{IH_IO1}	Input high	2			V	TTL Compatible
V _{IL_IO1}	Input low			0.8	V	
	Maximum frequency			2	MHz	
	Signals					PosLim, NegLim, Enable, Brake
	Input voltage range	0		24	V	
V _{IH_IO1}	Input high	2			V	TTL Compatible
V _{IL_IO1}	Input low			0.8	V	
	Maximum frequency			22	kHz	
Dedicated Digital Outputs						
	Signals					FaultOut
	Output Voltage	0		5	V	Open collector with internal 5V pull up
	Voltage Range	0		24	V	
	Sinking current			50	mA	
General purpose digital inputs/outputs						
	Signals					DigitalIO1-4
	Input voltage range	0		5	V	TTL compatible
V _{IH_IO1}	Input high	2			V	
V _{IL_IO1}	Input low			0.8	V	
V _{OH_1}	Output voltage high			5	V	
	Sourcing current			8	mA	
	Signals					DigitalIO5-8
	Input voltage range	0		24	V	TTL compatible, open collector
V _{IH_IO2}	Input high	2			V	
V _{IL_IO2}	Input low			0.8	V	
	Sourcing current			4	mA	
	Output voltage			0	V	Open collector with internal 5V pull up
	Output Voltage Range	0		24	V	
	Sinking current			50	mA	
Analog Input						
	Signals					AnalogIn + , AnalogIn-
	Differential input voltage range	-10		10	V	
	Common mode voltage range	-10		25	V	
	Resolution			16	Bits	

Symbol	Parameter	Min	Typ	Max	Units	Description
	Maximum recommended input frequency			25	kHz	
	Throughput			1.0	MSPS	
	Integral non-linearity error	-4	1	4	LSB	
	Differential impedance		143		kOhms	
	Common mode rejection ratio	60	71		dB	

2.5 Absolute Maximum Ratings

Parameter	Rating
HV voltage range, low power units	0 V to +60 V
HV voltage range, medium power units	0 V to +60 V
HV voltage range, high power units	0 V to +60 V
Serial signals: Srl1Rcv/RS485Rcv +, Srl1Rcv/RS485Rcv- voltage range	-15 V to +15 V
CAN signals: HostCAN +, HostCAN-, ExpCAN +, ExpCAN- voltage range	-14 V to +14 V
SPI signals: HostSPIClock, HostSPIRcv, HostSPISelect voltage range	-0.5 V to 6.5 V
Digital IO1-4	-0.5 V to 6.5 V
Digital IO5-8	-0.5 V to 30 V
Misc. signals: Enable, Hall A, Hall B, Hall C, Home, PosLimit, NegLimit, AxisIn voltage range	-0.3V to 30V
FaultOut voltage range	-0.3V to 30V
5V output current	50 mA

All voltage values are with respect to GND unless otherwise noted.



2.6 Environmental Ratings

Specification	Value
Operating ambient temperature	0 to 40 C
Maximum base plate temperature	70 C
Storage temperature	-20 to 85 C
Reflow soldering temperature	300 C (1.5mm for 10 seconds)
Humidity	0 to 95%, non-condensing
Altitude	Up to 2,000 meters without derating
Contamination	Pollution Degree 2

2.7 Safety and Compliance

Specification	Standard
CE	LVD: EN60204-1 EMC-D: EN61000-6-1, EN61000-6-3, EN55011 IEC 61000-4-2 +/-4kV contact, +/-8kV air
Electrical safety	Designed to UL61800-5-1, UL508C, UL840 and EN60204-1
Hazardous materials	RoHS compliant
Flammability	UL94-HB
Enclosure	IP20

2.8 Timing Characteristics

2.8.1 Host SPI

Timing Interval	No.	Min	Typ	Max
SPI clock cycle time ⁽¹⁾	T1	50 nSec		1600 nSec
SPI clock high	T2	T1/2-2 nSec		T1/2 + 2 nSec
Slave select setup time	T3	2 nSec		
Slave select hold time	T4	1 nSec		
Data output access time	T5	13 nSec	21 nSec	38 nSec
Data output valid time	T6		21 nSec	27 nSec
Data output hold time	T7	14 nSec		
Data output disable time	T8	4 nSec	9 nSec	16 nSec
Data input setup time	T9	1 nSec		
Data input hold time	T10	2 nSec		

2.8.2 Expansion SPI

Timing Interval	No.	Min	Typ	Max
SPI clock cycle time ⁽²⁾	T1	50 nSec		1600 nSec
SPI clock high	T2	T1/2-2 nSec	T1	T1/2 + 2 nSec
Data input setup time	T3	1 nSec		
Data input hold time	T4	4 nSec		
Data output valid time	T5		1 nSec	3 nSec
Data output hold time	T6	0 nSec		
CS to clock high	T7	T1		

Notes:

(1) Up to 20 MHz

(2) Settable from 625 kHz to 20 MHz, see [Section 3.6.4, "SPI Communications."](#)
for details

2.9 Timing Diagrams

2.9.1 Host SPI

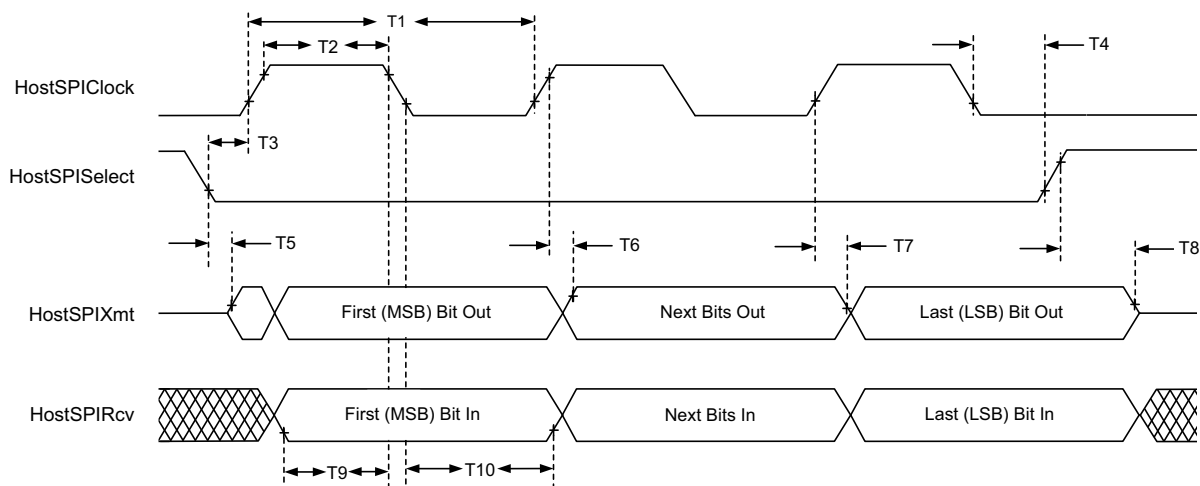


Figure 2-2:
Host SPI
Timing

2.9.2 Expansion SPI

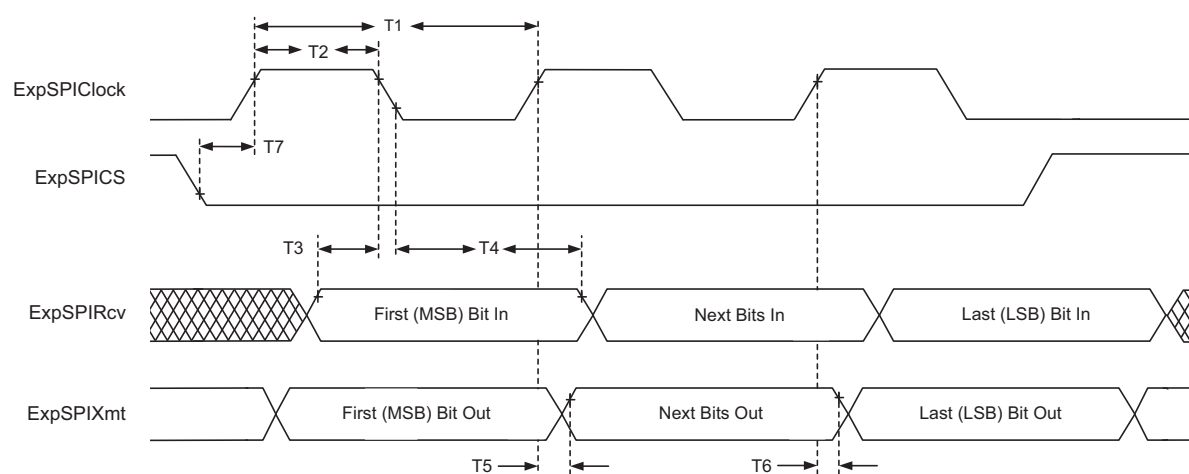


Figure 2-3:
Expansion SPI
Timing

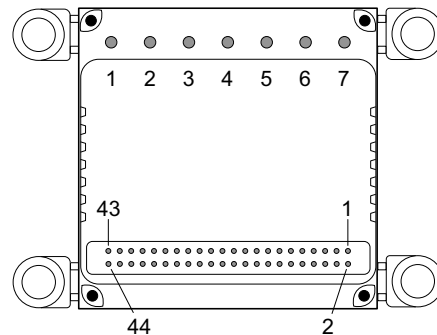
2.10 Pinout Quick Reference

The following sections provide detailed pinouts for the three N-Series ION host interface options; Serial, CAN/SPI, Ethernet.

2.10.1 N-Series ION Pinouts, Serial Host Interface

Power Connector

Pin	Name
1	HV Aux
2	HV
3	GND
4	Motor A
5	Motor B
6	Motor C
7	Motor D/Shunt



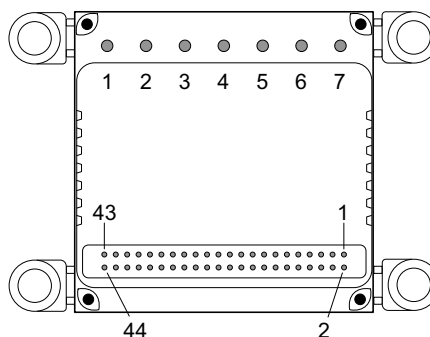
Signal Connector

Pin	Name	Pin	Name
1	Srl2Xmt/RS485Xmt-	2	Srl1Xmt/RS485Xmt +
3	Srl2Rcv/RS485Rcv-	4	Srl1Rcv/RS485Rcv +
5	GND	6	GND
7	ExpCAN +	8	ExpCAN-
9	RS485Sel/DigitalIO8	10	GND
11	QuadA2 + /Pulse2 +	12	QuadA2- /Pulse2-
13	QuadB2 + /Direction2 +	14	QuadB2- /Direction2-
15	Index1 + /BiSSClock +	16	Index1- /BiSSClock-
17	Index2 + /BiSSData +	18	Index2- /BiSSData-
19	HallA	20	HallB
21	HallC	22	PosLim
23	NegLim	24	Home
25	Enable	26	FaultOut
27	Reset	28	Brake
29	5V	30	GND
31	Srl3Xmt/DigitalIO1/ExpSPIXmt	32	Srl3Rcv/DigitalIO2/ExpSPIRcv
33	AxisIn/DigitalIO3/ExpSPIClock	34	SynchIn/DigitalIO4/ExpSPICS1
35	AxisOut/DigitalIO5/ExpSPICS2	36	HostInterrupt/DigitalIO6/ExpSPICS3
37	SynchOut/DigitalIO7/ExpSPIStatus/Exp- SPICS4	38	GND
39	QuadB1 + /Direction1 + /Sin +	40	QuadB1- /Direction1- /Sin-
41	QuadA1 + /Pulse1 + /Cos +	42	QuadA1- /Pulse1- /Cos-
43	AnalogIn +	44	AnalogIn-

2.10.2 N-Series ION Pinouts, CAN/SPI Host Interface

Power Connector

Pin	Name
1	HV Aux
2	HV
3	GND
4	Motor A
5	Motor B
6	Motor C
7	Motor D/Shunt



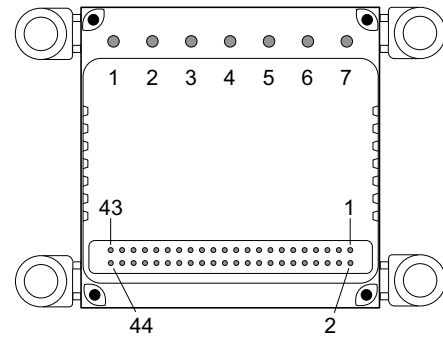
Signal Connector

Pin	Name	Pin	Name
1	HostSPiXmt	2	HostSPIClock
3	HostSPiRcv	4	HostSPiSelect
5	HostCAN +	6	HostCAN-
7	ExpCAN +	8	ExpCAN-
9	HostSPiStatus/DigitalIO8	10	GND
11	QuadA2 + /Pulse2 +	12	QuadA2- /Pulse2-
13	QuadB2 + /Direction2 +	14	QuadB2- /Direction2-
15	Index1 + /BiSSClock +	16	Index1- /BiSSClock-
17	Index2 + /BiSSData +	18	Index2- /BiSSData-
19	HallA	20	HallB
21	HallC	22	PosLim
23	NegLim	24	Home
25	Enable	26	FaultOut
27	Reset	28	Brake
29	5V	30	GND
31	Srl3Xmt/DigitalIO1/ExpSPiXmt	32	Srl3Rcv/DigitalIO2/ExpSPiRcv
33	AxisIn/DigitalIO3/ExpSPIClock	34	SynchIn/DigitalIO4/ExpSPiCS1
35	AxisOut/DigitalIO5/ExpSPiCS2	36	HostInterrupt/DigitalIO6/ExpSPiCS3
37	SynchOut/DigitalIO7/ExpSPiStatus/Exp-SPiCS4	38	GND
39	QuadB1 + /Direction1 + /Sin +	40	QuadB1- /Direction1- /Sin-
41	QuadA1 + /Pulse1 + /Cos +	42	QuadA1- /Pulse1- /Cos-
43	AnalogIn +	44	AnalogIn-

2.10.3 N-Series ION Pinouts, Ethernet Host Interface

Power Connector

Pin	Name
1	HV Aux
2	HV
3	GND
4	Motor A
5	Motor B
6	Motor C
7	Motor D/Shunt



Signal Connector

Pin	Name	Pin	Name
1	EthernetTx +	2	EthernetTx-
3	EthernetRx +	4	EthernetRx-
5	TapTx	6	TapRx
7	ExpCAN +	8	ExpCAN-
9	DigitalIO8	10	GND
11	QuadA2 + /Pulse2 +	12	QuadA2- /Pulse2-
13	QuadB2 + /Direction2 +	14	QuadB2- /Direction2-
15	Index1 + /BiSSClock +	16	Index1- /BiSSClock-
17	Index2 + /BiSSData +	18	Index2- /BiSSData-
19	HallA	20	HallB
21	HallC	22	PosLim
23	NegLim	24	Home
25	Enable	26	FaultOut
27	Reset	28	Brake
29	5V	30	GND
31	Srl3Xmt/DigitalIO1/ExpSPIXmt	32	Srl3Rcv/DigitalIO2/ExpSPIRcv
33	AxisIn/DigitalIO3/ExpSPIClock	34	SynchIn/DigitalIO4/ExpSPICS1
35	AxisOut/DigitalIO5/ExpSPICS2	36	HostInterrupt/DigitalIO6/ExpSPICS3/Eth-SpeedLED
37	SynchOut/DigitalIO7/ExpSPIStatus/ExpSPICS4/Eth-LinkLED	38	GND
39	QuadB1 + /Direction1 + /Sin +	40	QuadB1- /Direction1- /Sin-
41	QuadA1 + /Pulse1 + /Cos +	42	QuadA1- /Pulse1- /Cos-
43	AnalogIn +	44	AnalogIn-

2.11 Pin Descriptions

The following sections detail the pin descriptions for the N-Series ION's Signal Connector. This 44 pin connector provides host interface connections at pins 1 through 9, and the remaining pin functions (which do not change from one N-Series ION host interface type to another) at pins 10 through 44.

Accordingly, the next three sections detail pins 1-9 separately for each of the three different host interface types. This is followed by a section that details pins 10-44, which are common for all N-Series ION units.

2.11.1 Signal Connector Pins 1-9, Serial Host Interface

Pin Name	#	Direction	Description
Srl2Xmt/RS485Xmt-	1	output/output	When RS232 mode is selected this pin carries the transmit of the RS232 channel 2 host connection. When RS485 mode is selected this pin carries the negative differential transmit of the RS485 host network.
Srl1Xmt/RS485Xmt +	2	output/output	When RS232 mode is selected this pin carries the transmit of the RS232 channel 1 host connection. When RS485 mode is selected this pin carries the positive differential transmit of the RS485 host network.
Srl2Rcv/RS485Rcv-	3	input/input	When RS232 mode is selected this pin carries the receive of the RS232 channel 2 host connection. When RS485 mode is selected this pin carries the negative differential receive of the RS485 host network.
Srl1Rcv/RS485Rcv +	4	input/input	When RS232 mode is selected this pin carries the receive of the RS232 channel 1 host connection. When RS485 mode is selected this pin carries the positive differential receive of the RS485 host network.
GND	5, 6	N/A	These pins provide a digital ground.
ExpCAN +	7	bi-dir	These pins carry the positive and negative differential expansion network CAN FD signals.
ExpCAN-	8		
RS485Sel/DigitalIO8	9	input/bi-dir	<p>The function of this pin is user programmable. When programmed as RS485Sel, which is the default setting, this pin is used to select RS232 or RS485 mode operation of the host serial interface. If tied high or left unconnected RS232 will be selected. If tied to ground RS485 will be selected.</p> <p>When programmed to the DigitalIO8 function this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When functioning as an output this signal is an open collector output with a voltage range of 0-24 V with 4mA sourcing and 50mA sinking capability.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-24V with a 30V maximum.</p>

2.11.2 Signal Connector Pins 1-9, CAN/SPI Host Interface

Pin Name	#	Direction	Description
HostSPIXmt	1	output	This pin transmits synchronous serial data for the host SPI bus.
HostSPIClock	2	input	This pin inputs the clock signal used with the host SPI bus transfer.
HostSPIRcv	3	input	This pin inputs synchronous serial data for the host SPI bus.
HostSPISelect	4	input	This pin inputs an enable signal for the host SPI bus. This signal is active low, meaning it should be low when an SPI host communication is occurring, and high at all other times.
HostCAN +	5	bi-dir	These pins carry the positive and negative differential host CAN FD signals.
HostCAN-	6		
ExpCAN +	7	bi-dir	These pins carry the positive and negative differential expansion CAN FD signals.
ExpCAN-	8		
HostSPIStatus/DigitalIO8	9	output/bi-dir	<p>The function of this pin is user programmable. When programmed as HostSPIStatus, which is the default setting, this pin is used with Host SPI communications to indicate when the ION has a command result that is ready to be read by the host. A high signal indicates that the command result is ready.</p> <p>When programmed to the DigitalIO8 function, this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When functioning as an output this signal is an open collector output with a voltage range of 0-24 V with 4mA sourcing and 50mA sinking capability.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-24V with a 30V maximum.</p>

2.11.3 Signal Connector Pins 1-9, Ethernet Host Interface

Pin Name	#	Direction	Description
EthernetTx +	1	output	These pins carry the positive and negative differential Ethernet transmit.
EthernetTx-	2		
EthernetRx +	3	input	These pins carry the positive and negative differential Ethernet receive.
EthernetRx-	4		
TapTx	5	N/A	This pin carries the transmit center tap signal connected to the Ethernet isolation transformer.
TapRx	6	N/A	This pin carries the receive center tap signal connected to the Ethernet isolation transformer.
ExpCAN +	7	bi-dir	These pins carry the positive and negative differential expansion network CAN FD signals.
ExpCAN-	8		
DigitalIO8	9	bi-dir	<p>This pin provides a general purpose bi-directional digital I/O bit.</p> <p>When functioning as an output this signal is an open collector output with a voltage range of 0-24 V with 4mA sourcing and 50mA sinking capability.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-24V with a 30V maximum.</p>

Pin Name	#	Direction	Description
EthSpeedLED	36	output	Standard Ethernet Speed LED indicator output. Signal is low, which will drive LED on, if link is 100 MBS. Output is high, turning off LED, if link is 10 MBS.
EthLinkLED	37	output	Standard Ethernet Link LED indicator output. Signal toggles high and low blinking the LED if there is port activity. Signal is steady low, turning on LED, if port is connected but there is no activity. Signal is steady high, turning off LED, if port is not connected.

2.11.4 Signal Connector Pins 10-44

Pin Name	#	Direction	Description
GND	10	N/A	This pin provides a digital ground.
QuadA2 + /Pulse2 +	11	input/input	<p>If the auxiliary axis (axis #2) encoder type is selected as quadrature these pins carry the positive and negative differential quadrature A input. The interpretation of these signals can be user programmed.</p> <p>If the auxiliary axis (axis #2) encoder type is selected as pulse & direction these pins carry the positive and negative differential pulse input. A pulse (position step command) occurs when these signals transition from a high state to a low state, although this default interpretation can be user programmed.</p>
QuadA2-/Pulse2-	12		
QuadB2 + /Direction2 +	13	input/input	<p>If the auxiliary axis (axis #2) encoder type is selected as quadrature these pins carry the positive and negative differential quadrature B input. The interpretation of these signals can be user programmed.</p> <p>If the auxiliary axis (axis #2) encoder type is selected as pulse & direction these pins carry the positive and negative differential direction input. When these signals specify a high state the position increases upon a pulse command, although this default interpretation can be user programmed.</p>
QuadB2-/Direction2-	14		
Index1 + /BissClock +	15	input/output	<p>If the primary axis (axis #1) encoder type is selected as quadrature or sin/cos these pins carry the positive and negative differential Index input for the primary axis (axis #1).</p> <p>If the primary axis (axis #1) encoder type is selected as BiSS-C, these pins carry the positive and negative differential BiSS clock output.</p>
Index1-/BiSSClock-	16		
Index2 + /BiSSData +	17	input/input	<p>If the auxiliary axis (axis #2) encoder type is selected as quadrature and the primary axis encoder type is quadrature or sin/cos, these pins carry the positive and negative differential Index input for the auxiliary axis (axis #2).</p> <p>If the primary axis (axis #1) encoder type is selected as BiSS-C, these pins carry the positive and negative differential BiSS data input.</p>
Index2-/BiSSData-	18		
HallA	19	input	<p>These pins input the Hall-encoded inputs for brushless DC motors. The A, B, and C Hall signals encode six valid states as follows: A on, A and B on, B on, B and C on, C on, C and A on. By default a sensor is defined as being on when its signal is high, however this signal interpretation can be user programmed.</p> <p>These are a TTL compatible inputs and accept 0-24V with a 30V maximum.</p>
HallB	20		
HallC	21		

Pin Name	#	Direction	Description
PosLim	22	input	These pins input a positive-side (forward) and negative-side (reverse) travel limit switch signals. By default this signal has active low interpretation, however this interpretation can be user programmed. These are TTL compatible inputs and accept 0-24V with a 30V maximum. If not connected these signals are pulled to an inactive state.
NegLim	23		
Home	24	input	<p>This pin inputs a home signal, which is a general-purpose input to the position capture mechanism. By default, a valid home signal is recognized when Home goes low, however this interpretation can be user programmed.</p> <p>This is a TTL compatible input and accepts 0-24V with a 30V maximum. If not connected this signal is pulled to an inactive state.</p>
Enable	25	input	<p>This pin is an enable input. To allow normal operation a low signal is asserted. When a high signal is asserted motor control operation is disabled, although host communications and various other operations are still available.</p> <p>This is a TTL compatible input and accepts 0-24V with a 30V maximum. If not connected this signal is pulled to an inactive state.</p>
FaultOut	26	output	<p>This pin provides a general purpose fault indicator that can be programmed to indicate a number of conditions including a motion error, amplifier error, or various other conditions. A high indicates that a fault condition is present.</p> <p>This is an open collector output with a voltage range of 0-24 V, with 50mA sinking capability. In addition this signal is pulled up to 5V.</p>
Reset	27	input	<p>This pin is the master reset input. It may be temporarily brought low to reset the unit to its startup conditions and then restored to high for normal operation.</p> <p>During powerup it is not required to provide a reset. The ION generates an internal reset upon powerup.</p> <p>This is a TTL compatible input and accepts 0-5V with a 6.5V maximum. If not connected this signal is pulled to an inactive state.</p>
Brake	28	input	<p>This pin provides a high speed amplifier braking function that may be useful for some applications. When low, normal motor amplifier operation is overridden to execute a brake function, or if so user programmed, to be disabled entirely. Operation is normal when this signal is high.</p> <p>This is a TTL compatible input and accepts 0-24V with a 30V maximum. If not connected this signal is pulled to an inactive state.</p>
5V	29	N/A	This pin provides a 5V output that may be useful for powering encoders or other low power sensors or peripherals. The maximum output current is 50 mA.
GND	30	N/A	This pin provides a digital ground.

Pin Name	#	Direction	Description
Srl3Xmt/DigitalIO1/ExpSPIXmt	31	output/bi-dir/ output	<p>The function of this pin is user programmable. When programmed as Srl3Xmt, which is the default setting, this pin provides the transmit signal for the point-to-point serial UART programming port.</p> <p>When programmed as DigitalIO1 this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When programmed as ExpSPIXmt this pin carries the SPI expansion network data transmit signal.</p> <p>When functioning as an output this signal is tri-state TTL compatible providing 0-5V and 8mA drive rating.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-5V with a 6.5V maximum.</p>
Srl3Rcv/DigitalIO2/ExpSPIRcv	32	input/bi-dir/ input	<p>The function of this pin is user programmable. When programmed as Srl3Rcv, which is the default setting, this pin provides the receive signal for the point-to-point serial UART programming port.</p> <p>When programmed as DigitalIO2 this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When programmed as ExpSPIRcv this pin carries the SPI expansion network data receive signal.</p> <p>When functioning as an output this signal is tri-state TTL compatible providing 0-5V and 8mA drive rating.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-5V with a 6.5V maximum.</p>
AxisIn/DigitalIO3/ExpSPIClock/Battery	33	input/bi-dir/ output/input	<p>The function of this pin is user programmable. When programmed as AxisIn, which is the default setting, this pin inputs the Magellan general purpose AxisIn signal.</p> <p>When programmed as DigitalIO3 this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When programmed as ExpSPIClock this pin carries the SPI expansion network clock output signal.</p> <p>When programmed as battery this pin is used to provide low current power backup for the dedicated real time clock circuitry. The allowed range of battery backup voltage is 1.5V to 3.3V.</p> <p>When functioning as an output this signal is tri-state TTL compatible providing 0-5V and 8mA drive rating.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-5V with a 6.5V maximum.</p>
SynchIn/DigitalIO4/ExpSPICS1	34	input/bi-dir/ output	<p>The function of this pin is user programmable. When programmed as SynchIn, which is the default setting, this pin inputs a pulse signal that can be used to synchronize the loop rate to an external N-Series ION, PMD Magellan IC, or other source. A synch pulse is defined as a transition from high to low.</p> <p>When programmed as DigitalIO4 this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When programmed as ExpSPICS1 this pin carries the SPI expansion network device 1 select signal.</p> <p>When functioning as an output this signal is tri-state TTL compatible providing 0-5V and 8mA drive rating.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-5V with a 6.5V maximum.</p>

Pin Name	#	Direction	Description
AxisOut/DigitalIO5/ExpSPICS2	35	output/bi-dir/ output	<p>The function of this pin is user programmable. When programmed as AxisOut, which is the default setting, this pin outputs the Magellan AxisOut signal.</p> <p>When programmed as DigitalIO5 this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When programmed as ExpSPICS2 this pin carries the SPI expansion network device 2 select signal.</p> <p>When functioning as an output this signal is an open collector output with a voltage range of 0-24V with 4mA sourcing and 50mA sinking capability.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-24V with a 30V maximum.</p>
HostInterrupt/DigitalIO6/ExpSPICS3/ EthSpeedLED	36	output/bi-dir/ output/output	<p>The function of this pin is user programmable. When programmed as HostInterrupt, which is the default setting, this pin outputs the Magellan HostInterrupt signal.</p> <p>When programmed as DigitalIO6 this pin provides a general purpose bi-directional digital I/O bit.</p> <p>When programmed as ExpSPICS3 this pin carries the SPI expansion network device 3 select signal.</p> <p>When programmed as EthSpeedLED, which is only available for Ethernet host type ION units, this pin is intended to drive an LED and has a low output (LED on) if the link is 100 MBS and a high output (LED is off) if the link is 10 MBS.</p> <p>When functioning as an output this signal is an open collector output with a voltage range of 0-24V with 4mA sourcing and 50mA sinking capability.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-24V with a 30V maximum.</p>
SynchOut/DigitalIO7/ExpSPIStatus/ ExpSPICS4/EthLinkLED	37	output/bi-dir/ input/output/ output	<p>The function of this pin is user programmable. When programmed as SynchOut, which is the default setting for Serial and CAN/SPI host type units, this pin outputs a pulse signal that can be used to synchronize the loop rates of other N-Series IONs, Magellan ICs, or external circuitry. A synch pulse is defined as a transition from high to low.</p> <p>When programmed as DigitalIO7/ExpSPIStatus this pin provides a general purpose bi-directional digital I/O bit unless PRP device communications are conducted on the expansion SPI port, in which case this pin will carry the ExpSPI-Status function. A low signal indicates that the SPI command result is ready.</p> <p>When programmed as ExpSPICS4 this pin carries the SPI expansion network device 4 select signal.</p> <p>When programmed as EthLinkLED, which is only available for Ethernet host type ION units, and which for that unit type is the default value, this pin is intended to drive an LED. This signal toggles high and low blinking the LED if there is port activity, has a steady low output (LED on) if port is connected but no activity, and has a steady high output (LED off) if the port is not connected.</p> <p>When functioning as an output this signal is an open collector output with a voltage range of 0-24V with 4mA sourcing and 50mA sinking capability.</p> <p>When functioning as an input this signal is a TTL compatible input and accepts 0-24V with a 30V maximum.</p>
GND	38	N/A	This pin provides a digital ground.

Pin Name	#	Direction	Description
QuadB1 + /Direction1 + /Sin +	39	input/input/	<p>If the primary axis (axis #1) encoder type is selected as quadrature, these pins carry the positive and negative differential quadrature B channel input for the primary axis (axis #1).</p> <p>If the primary axis (axis #1) encoder type is selected as pulse & direction these pins carry the positive and negative differential direction input. When these signals specify a high state the position increases upon a pulse command, although this default interpretation can be user programmed.</p> <p>If the primary axis (axis #1) encoder type is selected as sin/cos, these pins carry the positive and negative differential analog Sin signals.</p> <p>With sin/cos operation these are analog inputs with a range of 0-5V and a center point value of 2.5V.</p>
QuadB1 - /Direction1 - /Sin -	40	input	
QuadA1 + /Pulse1 + /Cos +	41	input/input/	<p>If the primary axis (axis #1) encoder type is selected as quadrature, these pins carry the positive and negative differential quadrature A channel input for the primary axis (axis #1).</p> <p>If the primary axis (axis #1) encoder type is selected as pulse & direction these pins carry the positive and negative differential pulse input. A pulse (position step command) occurs when these signals transition from a high state to a low state, although this default interpretation can be user programmed.</p> <p>If the primary axis (axis #1) encoder type is selected as sin/cos, these pins carry the positive and negative differential analog cos signals.</p> <p>With sin/cos operation these are analog inputs with a range of 0-5V and a center point value of 2.5V.</p>
QuadA1 - /Pulse1 - /Cos -	42	input/input	
AnalogIn +	43	input	<p>These pins carry the positive and negative differential signals of the general purpose analog input.</p> <p>These analog inputs has a range of -10V to +10V.</p>
AnalogIn -	44		

2.11.5 Power Connector

Pin Name	#	Description
HV Aux	1	Auxiliary HV (powers logic)
HV	2	HV supply (powers motor)
GND	3	Power ground
Motor A	4	Motor A coil connection (used with Brushless DC, DC Brush, and step motors)
Motor B	5	Motor B coil connection (used with Brushless DC, DC Brush, and step motors)
Motor C	6	Motor C coil connection (used with Brushless DC and step motors)
Motor D/Shunt	7	For step motors provides the Motor D coil connection. With DC Brush or Brushless DC motors may be used to provide a shunt output.

This page intentionally left blank.

3. Operation

3

In This Chapter

- ▶ Function Summary
- ▶ Magellan IC Functions
- ▶ Encoder Functions
- ▶ Motion Peripheral Functions
- ▶ I/O Functions
- ▶ Communications Functions
- ▶ Power Stage
- ▶ Drive Safety Functions
- ▶ C-Motion Engine
- ▶ General Functions
- ▶ Device NVRAM
- ▶ N-Series ION Powerup & Initialization

The following diagram provides a functional block diagram of N-Series IONs:

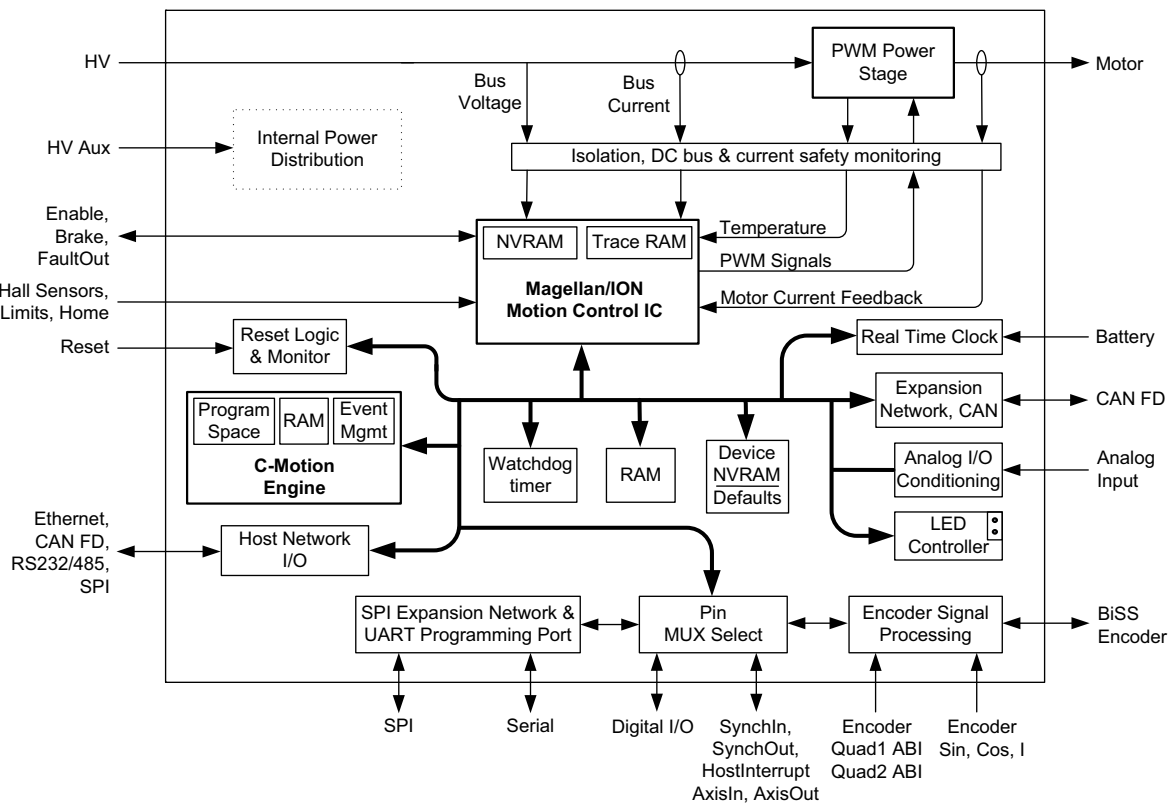


Figure 3-1:
ION/CME N-Series Digital Drive Internal Block Diagram

3.1 Function Summary

N-Series ION functions can be broken down into seven overall categories:

Magellan IC functions - These are functions which reside in the Magellan Motion Control IC. Included are profile generation, DC Brush and Brushless DC servoloop closure, breakpoint processing, current control, PWM signal generation, and much more. These functions are accessed through the Magellan Motion Control IC's command set, which allows for sophisticated control of the motion axes and associated hardware.

Encoder functions - These are the functions related to encoder input processing. The encoder formats supported by N-Series IONs are quadrature, pulse & direction, sin/cos, SSI, and BiSS-C.

Motion peripheral functions - These are the functions directly related to Magellan IC axis signals such as Enable, FaultOut, HostInterrupt, Home, limit switch inputs, Hall inputs, AxisIn & AxisOut, and SynchIn & SynchOut.

I/O functions - These are digital and analog input and output functions. There are 8 general purpose bi-directional digital channels, and there is one general purpose +/- 10V analog input channel.

Communications functions - The N-Series IONs provide sophisticated communication facilities consisting of one of four host interfaces depending on the specific ION unit type being used; Serial, CAN FD, SPI, or Ethernet. In addition, CAN FD and SPI ports, referred to as expansion networks, are available on every N-Series ION.

Power stage functions - The on-board amplifier accepts motor output commands from the Magellan Motion Control IC and provides high performance amplification and current control for step, DC Brush, or Brushless DC motors.

Drive safety functions - ION provides numerous safety and DC bus management functions including detection and protection from over current, over temperature, over/under voltage, shunt control, and more.

C-Motion Engine functions - The C-Motion Engine is a self-contained, high performance code execution unit that allows C-Motion code to be downloaded and executed on the N-Series ION. It can communicate with various resources on the board including the Magellan Motion Control IC, the drive's serial, CAN, Ethernet and SPI ports, and other resources such as the NVRAM.

General drive functions - These are functions associated with general drive facilities including non-volatile RAM, reset, a real time clock, LED control, pin muxing, and other functions.

3.1.1 N-Series ION Access Basics

Access to the N-Series ION from the serial, CAN, Ethernet, or SPI ports is provided by a protocol called the *PMD Resource access Protocol* (PRP). This easy-to-use yet powerful system utilizes actions, resources, and addresses to access the N-Series ION's functions. Various ION functions are organized into resources, and resources process actions sent to them. Actions can send information, request information, or command specific events to occur. Addresses allow access to a specific resource on the ION, or connected to the ION, via the serial, CAN, Ethernet, or SPI connections.

A basic communication to the N-Series ION consists of a 16-bit PRP header, and an optional message body. The message body contains data associated with the specified PRP action, but some actions do not require a message body. After a PRP communication is sent, a return communication is sent by the N-Series ION which consists of a PRP header and an optional return message body. The return message body may contain information associated with the requested PRP action, or it may contain error information if there was a problem processing the requested action.

There are five different resource types supported by the N-Series ION. The **Device** resource indicates functionality that is addressed to the entire ION unit, the **MotionProcessor** resource indicates a Magellan Motion Control IC, the **CMotionEngine** resource indicates the C-Motion Engine, the **Memory** resource indicates the RAM and the non-volatile RAM (Random Access Memory), and the **Peripheral** resource indicates a communications connection, the digital and analog I/O functions, or other on-board registers.

There are ten different PRP actions including **Command**, which is used to send commands to resources such as the Magellan Motion Control IC, **Send** and **Receive**, which are used to communicate using the serial, CAN, Ethernet, and SPI ports, **Read** and **Write**, which are used to access memory-type devices such as the ION unit's Device non-volatile RAM, and **Set** and **Get**, which are used to load or read parameters.

For complete information on the format and function of PMD Resource Access Protocol refer to the *C-Motion PRP II Programming Reference*.

Although it may be useful to be familiar with the N-Series ION's command processing architecture, most users will not need to know all of these details. Most users will prototype motion sequences using PMD's Pro-Motion Windows-based motion development application, and will use PMD's high-level C-Motion library to develop user application code. C-Motion provides function calls which hide the low level details of communication so that the user may directly access the functions of interest to them.



3.2 Magellan IC Functions

The Magellan Motion Control IC in [Figure 2-1](#) forms the core of the N-Series IONs. Here is an overview of the functions provided by the Magellan Motion Control IC:

- Profile generation
- DC brush and brushless DC servo loop closure
- Breakpoint processing
- AxisIn and AxisOut processing
- Trace
- Motion error detection, tracking windows, and axis-settled indicator
- Limit switch processing
- Current control
- PWM (pulse width modulation) signal generation
- DC bus and amplifier related safety monitoring
- NVRAM application-specific initialization storage

Magellan instructions are encoded in packets, which are sent to and from the Magellan Motion Control IC. The Magellan processes these packets, performs requested functions, and returns requested data. Generally speaking each command packet has its own C-Motion command associated with it.

Within the N-Series ION the Magellan IC connects to the ION's internal communications bus, which allows the Magellan to be controlled via the C-Motion Engine, or via an external host controller connected to the N-Series ION by serial, CAN, Ethernet, or SPI.

3.2.1 Example Magellan Instructions

The Magellan instruction set is very flexible and powerful. The following example, which sets up and executes a simple trapezoidal profile, illustrates just a small part of the overall command set.

```

SetProfileMode Axis1, trapezoidal    // set profile mode to trapezoidal for axis 1
SetPosition Axis1, 12345             // load a destination position for axis 1
SetVelocity Axis1, 223344            // load a velocity for axis 1
SetAcceleration Axis1, 1000          // load an acceleration for axis 1
SetDeceleration Axis1, 2000          // load a deceleration for axis 1
Update Axis1                         // Double buffered registers are copied into
                                     // the active registers, thereby initiating the move

```


This form of displaying Magellan IC commands is known as the mnemonic format, and is the primary format that will be used throughout this manual when referencing specific commands. Here is the same sequence as it would look in actual C-Motion calls:

```
PMDSetProfileMode(&hAxis1, PMDProfileTrapezoidal);
PMDSetPosition(&hAxis1, 12345);
PMDSetVelocity(&hAxis1, 223344);
PMDSetAcceleration(&hAxis1, 1000);
PMDSetDeceleration(&hAxis1, 2000);
PMDUpdate(&hAxis1);
```

Note that the overall format of C-Motion calls is very similar to their corresponding mnemonic command representation.

3.2.1.1 C-Motion Conventions & Structures

In the above C-Motion example `hAxis1` is a handle to a structure known as a `PMDAxisHandle`. There are other commands that address different ION resources. For example C-Motion functions that address the **Device** resource are prefaced with `PMDDevice`, and C-Motion functions that address the **Peripheral** resource are prefaced with `PMDPeriph`. Each of these command types take a handle to a variable with that structure. For example commands prefaced with `PMDDevice` take a handle to a structure known as a `PMDDeviceHandle`, and commands prefaced with `PMDPeriph` take a handle to a structure known as a `PMDPeripheralHandle`.

There are many different C-Motion structures, and in general these structures are used to make accessing N-Series IONs simpler and more flexible. In particular, by developing code with C-Motion, it is very easy to change the physical location of a PMD axis or other resource type such as a **Peripheral** without any changes to the developed C-Motion code sequences.

Two manuals describe how the Magellan Motion Control IC operates and how it is programmed: the *Magellan Motion Control IC User Guide*, and the *Magellan Motion Control IC Programming Reference*. These documents also describe Visual Basic Support, and C-Motion, which are the software libraries that are used to send commands to the Magellan chip and exercise its many functions.

The manual that describes the PRP C-Motion system is the *C-Motion PRP II Programming Reference*.



To simplify the presentation of command names, most portions of this manual will provide C-Motion commands without the “PMD” preface. The actual commands however should include the PMD preface. For example the C-Motion command given in the manual as **PeripheralRead** is actually called **PMDPeripheralRead**.

3.2.2 Magellan Initialization

In its factory default condition the Magellan IC is not yet loaded with application specific control settings, for example specifying gain factors, safety limit values, breakpoint settings, and other settings. Commonly these settings are sent via Magellan command sometime after initialization by user code loaded into the ION’s C-Motion Engine, or by external host communication ports.

Alternatively however, it is possible to store initialization information into the Magellan IC so that it is already initialized with the application control settings after power on.

This is accomplished by an NVRAM memory segment called the Magellan IC NVRAM. Magellan commands that would normally be sent from the C-Motion Engine or from host ports are instead stored into the NVRAM. Upon initialization the Magellan IC determines if command sequences are stored in its NVRAM, and if so executes these before beginning normal operations.

3.2.2.1 Initialization Execution Control

To make the Magellan IC initialization sequence as flexible as possible N-Series IONs provide a facility to control execution of the stored initialization commands. This function is provided by the Magellan IC’s **ExecutionControl** command.

Using this command, initialization can be suspended for a specific period of time or until various internal or external conditions are satisfied. The **ExecutionControl** command may be inserted into a sequence of Magellan initialization commands one or more times (or not at all). Each time it is encountered the conditions specified are applied and initialization proceeds, or stops, accordingly. Together these features are useful for coordinating Magellan IC startup with external processes on the user's PCB, or to execute simple motion sequences prior to normal operation.

For more detailed information on the function and format of the **ExecutionControl** command refer to the *Magellan Motion Control IC User Guide* and the *Magellan Motion Control IC Programming Reference*.

The ExecutionControl command, while powerful and flexible, can be programmed in such a way that Magellan initialization halts indefinitely. This most often occurs when initialization is made contingent upon an external ION signal state, and the signal never transitions to the conditioned state. To avoid such a condition it is up to the user to verify that ExecutionControl commands do not inadvertently permanently halt the Magellan Initialization sequence.



To learn more about how to check the status of Magellan initialization and confirm that it is ready for normal operations refer to [Section 3.12.2, "Magellan IC Initialization."](#)

3.2.2.2 Magellan IC NVRAM

The Magellan IC contains its own NVRAM area that can store initialization sequences as discussed above as well as more general purpose information such as information to identify the contents of the stored Magellan initialization sequence, information related to the machine being controlled, or for any purpose.

All Magellan IC NVRAM content uses a storage format known as PSF (PMD Structured data Format). This storage format allows various categories of storage information to be accessed including initialization command sequences. An additional useful feature of PSF is that the machine-format data stored in NVRAM can be converted to ASCII script files which can be edited by any standard text editor, and then converted back into machine NVRAM format.

Note that while it is possible for the user to write their own software sequences to store and retrieve NVRAM content in PSF format, most users will instead use the Pro-Motion program to conveniently retrieve, edit, and store Magellan IC NVRAM content.

For more information on the Magellan IC NVRAM refer to the *Magellan Motion Control IC User Guide*. For general information on PSF refer to the *Magellan Motion Control IC Programming Reference*. For information on loading the Magellan IC NVRAM in the production process refer to [Section 4.2.3.1, "Loading the Magellan IC NVRAM."](#)

3.2.3 C-Motion Commands

To initiate C-Motion access to the Magellan Motion Control IC the command **MotionProcessorOpen** is used. To initiate access to a single axis the command **AxisOpen** is used. Detailed information on these and other commands can be found in the *C-Motion PRP II Programming Reference*.

As indicated earlier in this section there are many different commands which set or retrieve information from the motion control IC or initiate actions. Examples include **SetProfileMode**, **GetAcceleration**, and **Update**. For detailed information on these C-Motion commands refer to the *Magellan Motion Control IC Programming Reference*.

3.2.4 Connection & Associated Signals

The table below shows the signals directly associated with Magellan IC functions. This list is intended only as a summary view for convenience to the user. All of these signals are detailed in other sections of the manual which include additional information on their function and electrical interfacing.

Pin Name	Pin #(s)	Description
QuadA1 +, QuadA1 -*	41, 42	Differential A quadrature inputs for the primary channel (axis #1)
QuadB1 +, QuadB1 -*	39, 40	Differential B quadrature inputs for the primary channel (axis #1)
QuadA2 +, QuadA2 -	11, 12	Differential A quadrature inputs for the auxiliary channel (axis #2)

Pin Name	Pin #(s)	Description
QuadB2 +, QuadB2-	13, 14	Differential B quadrature inputs for the auxiliary channel (axis #2)
Index1 +, Index1-*	15, 16	Differential Index inputs for the primary channel (axis #1)
Index2 +, Index2-*	17, 18	Differential Index inputs for the auxiliary channel (axis #2) when the primary axis encoder type is set to quadrature AB or sin/cos.
BiSSData +, BiSSData-*	17, 18	BiSS encoder differential BiSS data inputs
BiSSClock +, BiSSClock-*	15, 16	BiSS encoder differential BiSS clock outputs
Sin +, Sin-*	39, 40	Sin/Cos encoder differential sin signal inputs
Cos +, Cos-*	41, 42	Sin/Cos encoder differential cos signal inputs
Enable	25	Magellan IC Enable input signal
FaultOut	26	Magellan IC FaultOut output signal
HostInterrupt*	36	Magellan IC HostInterrupt output signal
Home	24	Home signal input
PosLim, NegLim	22, 23	Positive & negative limit signal inputs
HallA, HallB, HallC	19, 20, 21	Hall signal inputs
AxisIn*	33	Magellan IC AxisIn signal input
AxisOut*	35	Magellan IC AxisOut signal output
Brake	28	Motor brake signal input
SynchIn, SynchOut*	34, 37	Magellan IC synch signals

* These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed.

3.3 Encoder Functions

N-Series IONs support two separate position encoder channels, one functioning as a primary axis, and one as an auxiliary axis. Position feedback input can be used for a number of purposes including servo position compensation, electronic gear or CAM command input, data capture synchronization, motion error detection, and other functions. In addition to position encoder input, an *Index* signal for both the primary and auxiliary axis and a *Home* signal for the primary axis is supported as a position capture trigger.

N-Series IONs support four different encoder formats; quadrature AB, pulse & direction, sin/cos, and BiSS-C. These encoder formats are always supported on the primary axis and for some formats also supported on the auxiliary axis (axis #2). The following table provides a summary. Unless noted otherwise the primary and auxiliary axis can be programmed independently. For example the primary axis can be set to quadrature AB and the auxiliary axis to pulse & direction.

Encoder Format & Axis #	Available?	Available Trigger Captures	Comments
Quadrature AB			
Primary	Yes	Index1 Home	
Auxiliary	Yes	Index2	Index2 signals carry the auxiliary axis index capture signal when the primary axis is set to quadrature AB or sin/cos. If the primary axis is set to BiSS-C no capture source is available for the auxiliary encoder.
Pulse & Direction			
Primary	Yes	N/A	Position capture not available with this encoder format.
Auxiliary	Yes	N/A	
Sin/Cos			
Primary	Yes	Index1 Home	
Auxiliary	No		

Encoder Format & Axis #	Available?	Available Trigger Captures	Comments
BiSS-C			
Primary	Yes	Home	One BiSS-C encoder is supported which may be used either as the primary or the auxiliary encoder input.
Auxiliary	Yes		

3.3.1 Quadrature Encoders

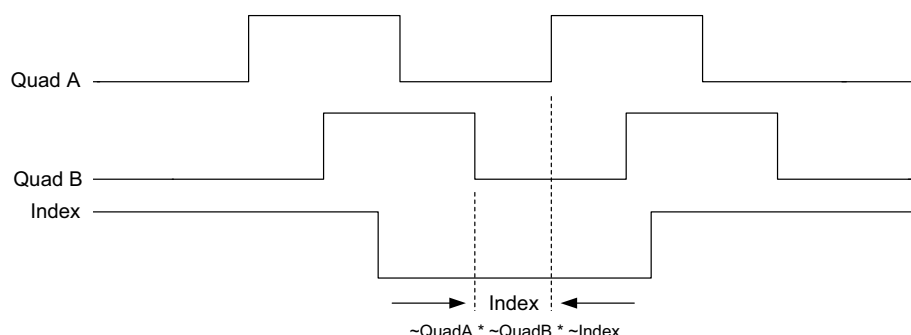


Figure 3-2:
Quadrature AB
Encoder Signal
Waveform

Figure 3-2 shows the format of quadrature AB signals. The A and B signals are 90 electrical degrees apart. At transition points of the *QuadA* and *QuadB* signals the tracked position either increases or decreases depending on which signal changes state and whether it changes from high to low or vice versa.

To reduce the possibility of a spurious position change both the A and B quadrature signals are passed through analog as well as digital filters within the N-Series ION. Quadrature positions can be tracked at velocities up to 32 Mcounts/sec in both the primary and auxiliary axis. N-Series IONs accept quadrature signals in both differential and signal ended format. For reliable quadrature encoder tracking use of differential signals is always recommended.

3.3.1.1 Position Capture

N-Series IONs allow input of a trigger signal to capture the instantaneous position value and store this value in a register for later retrieval by the user. This may be useful for a number of reasons including to check whether quadrature encoder counts have been lost, to maintain commutation accuracy if the quadrature encoder rather than Hall sensors is used for commutation, during axis homing, or for other application-specific purposes.

For the primary axis quadrature AB input two separate triggers are supported, the *Index* signal and the *Home* signal. For the auxiliary encoder input only the *Index* signal is supported. Position capture occurs via dedicated hardware inside the ION unit and is therefore nearly instantaneous.

3.3.1.2 Position Capture Signal Qualification

The *Index* signal, when used as the position capture trigger, results in a trigger occurring when it, the *QuadA* signal, and the *QuadB* signal are all low. This is shown in Figure 3-2. The *Index1* signal triggers the Axis1 position capture, and the *Index2* signal triggers the Axis2 capture.

Index signal capture qualification may be useful for a number of reasons including improving repeatability of the captured location and insuring that the capture location is the same whether the axis approaches from the positive or negative direction. To accommodate a range of encoder signal waveforms the sense of all of these signals (*Index*, *QuadA*, *QuadB*) can be user programmed.

The *Home* signal, when used as a position capture trigger, is not qualified with respect to the state of the A and B signals, but similarly to the *Index* signal its sense interpretation can be used programmed. For more information on quadrature encoder input, position tracking, position capture, and other related functions refer to the *Magellan Motion Control IC User Guide*.

3.3.1.3 C-Motion Commands

There are numerous Magellan C-Motion commands that relate to the encoder feedback and index position, including commands that retrieve, capture, compare, set, or otherwise utilize the current encoder position. Refer to the *Magellan Motion Control IC User Guide* for more information.

3.3.1.4 Connections & Associated Signals

The table below shows the signals associated with quadrature encoder processing. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
QuadA1 + *	41	Differential A + quadrature input for the primary channel (axis #1).
QuadA1 - *	42	Differential A - quadrature input for the primary channel (axis #1).
QuadB1 + *	39	Differential B + quadrature input for the primary channel (axis #1).
QuadB1 - *	38	Differential B - quadrature input for the primary channel (axis #1).
QuadA2 + *	11	Differential A + quadrature input for the auxiliary channel (axis #2).
QuadA2 - *	12	Differential A - quadrature input for the auxiliary channel (axis #2).
QuadB2 + *	13	Differential B + quadrature input for the auxiliary channel (axis #2).
QuadB2 - *	14	Differential B - quadrature input for the auxiliary channel (axis #2).
Index1 + *	15	Differential Index + input for the primary channel (axis #1).
Index1 - *	16	Differential Index - input for the primary channel (axis #1).
Index2 + *	17	Differential Index + input for the auxiliary channel (axis #2) when the primary axis encoder type is set to quadrature AB or sin/cos.
Index2 - *	18	Differential Index - input for the auxiliary channel (axis #2) when the primary axis encoder type is set to quadrature AB or sin/cos.
5V	29	+ 5V power output which may be used to power encoder circuitry.
GND	30	Ground. Note that any of the N-Series ION's ground pins can be used including 10, 30, and 38.

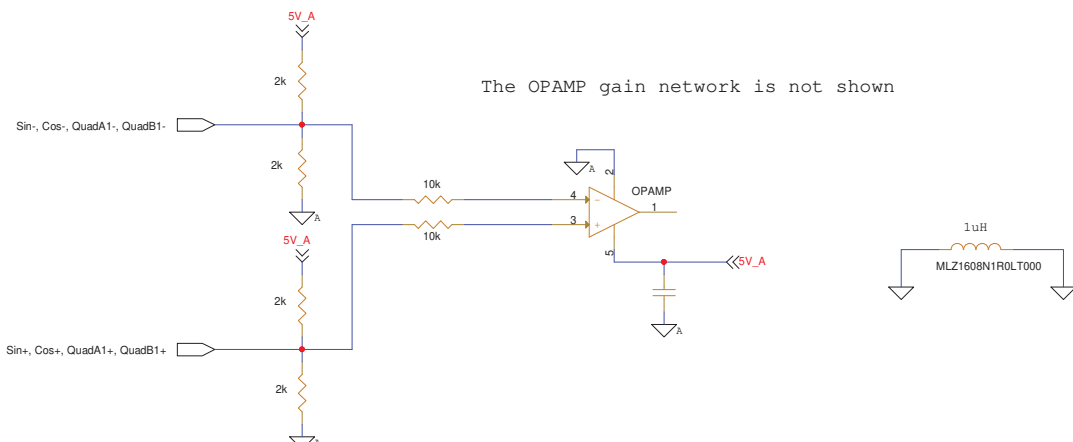
* These signals are located on pins with multiple programmable functions. Unless otherwise indicated in the description the pin function applies only when the encoder type has been programmed to quadrature.

The quadrature and index signals can be connected in one of two ways. Single-ended means that only one wire per signal is used, while differential means two wires encode each signal (labeled + and -). Differential connections are recommended for the highest level of reliability. If single-ended connections are used, only the + wire is connected and the - wire should be left floating.

3.3.1.5 Quadrature Encoder Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process quadrature and index signal input.

The following diagram shows this information for the QuadA1 and QuadB1 signals:



**Figure 3-3:
QuadA1 &
QuadB1
Internal
Processing
Circuitry**

The QuadA2, QuadB2, Index1, and Index2 signals use a slightly processing scheme as shown in the diagram below.

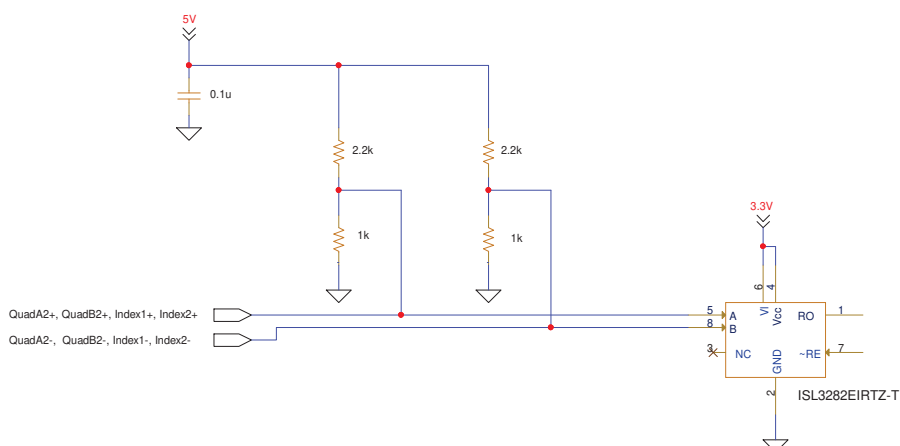


Figure 3-4:
QuadA2,
QuadB2,
Index1, and
Index2 Internal
Processing
Circuitry

3.3.2 Pulse & Direction Input

Pulse & Direction is a two-signal incremental position encoding scheme. It is most often used to command a desired position rather than to record the position of a sensor such as from a shaft encoder. As such, pulse & direction encoder type is most often used with the auxiliary axis when it is used to command the master command position during electronic gear or cam profiling.

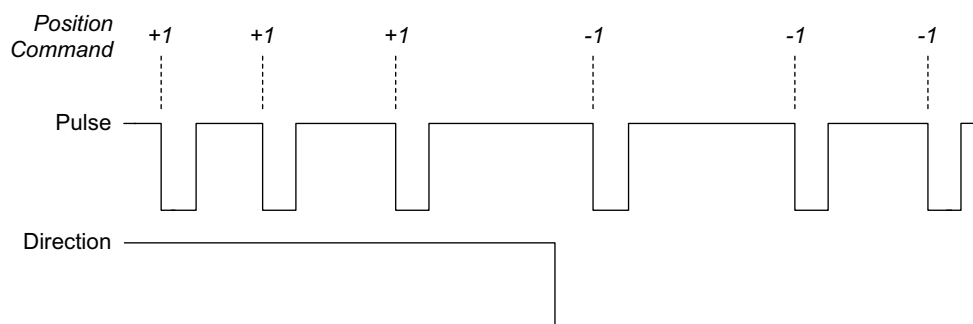


Figure 3-5:
Pulse &
Direction Signal
Format

[Figure 3-5](#) shows an overview of the pulse & direction scheme. A pulse (position step command) occurs when the *Pulse* signal transitions from high to low (although this default sense interpretation can be user programmed). When this occurs the commanded position increases if the *Direction* signal is high, and decreases if low (although again this default interpretation can be used user programmed). Pulse & direction input supports a maximum rate of 10 Mpulses/sec.

Pulse & direction format does not support a position capture function. For more information on pulse & direction position input, position tracking, and related functions, refer to the *Magellan Motion Control IC User Guide*.

3.3.2.1 C-Motion Commands

There are numerous Magellan C-Motion commands that relate to the encoder feedback and index position, including commands that retrieve, capture, compare, set, or otherwise utilize the current encoder position. Refer to the *Magellan Motion Control IC User Guide* for more information.

3.3.2.2 Connections & Associated Signals

The table below shows the signals associated with pulse & direction encoder format. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Pulse1 + *	41	Differential Pulse + input for the primary channel (axis #1).
Pulse1 - *	42	Differential Pulse- input for the primary channel (axis #1).
Direction1 + *	39	Differential Direction + input for the primary channel (axis #1).
Direction1 - *	38	Differential Direction- input for the primary channel (axis #1).
Pulse2 + *	11	Differential Pulse + input for the auxiliary channel (axis #2).
Pulse2 - *	12	Differential Pulse- input for the auxiliary channel (axis #2).
Direction2 + *	13	Differential Direction + input for the auxiliary channel (axis #2).
Direction2 - *	14	Differential Direction- input for the auxiliary channel (axis #2).
5V	29	+ 5V power output which may be used to power encoder circuitry.
GND	30	Ground. Note that any of the N-Series ION's ground pins can be used including 10, 30, and 38.

* These signals are located on pins with multiple programmable functions. The pin function applies only when the encoder type has been programmed to pulse & direction.

The *Pulse* and *Direction* signals can be connected in one of two ways. Single-ended means that only one wire per signal is used, while differential means two wires encode each signal (labeled + and -). Differential transmission is recommended for the highest level of reliability. If single-ended connections are used only the + wire is connected, and the - wire should be left floating. If differential connections are used, both the + and - signals are used.

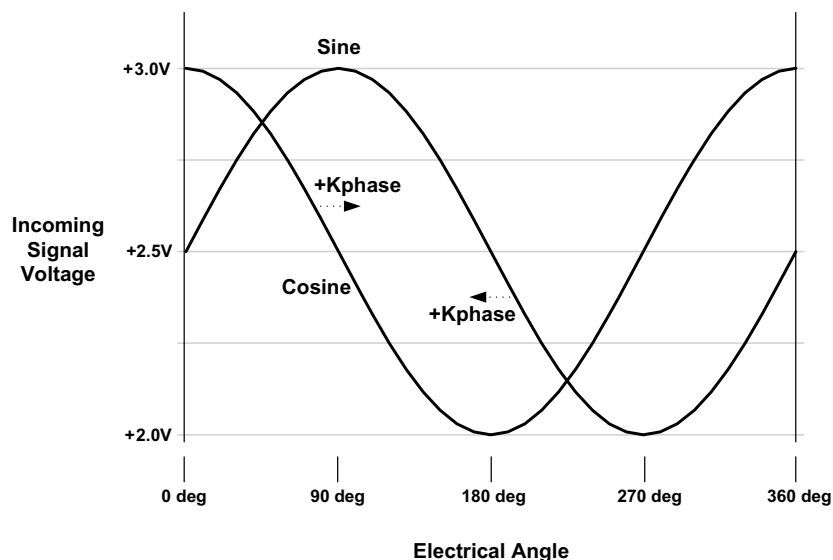
3.3.2.3 Pulse & Direction Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process these signals. Pulse1, Direction1, Pulse2, and Direction2 signals share the same processing circuitry as QuadA1, QuadB1, QuadA2, and QuadB2 signals respectively. Therefore refer to [Section 3.3.1.5, "Quadrature Encoder Signal Processing,"](#) for information on this processing circuitry.

3.3.3 Sin/Cos Encoders

Some encoders output a two signal format similar to quadrature AB but containing a sinusoidal analog output rather than a digital output. This type of encoder format is known as a sin/cos or a sine-cosine format encoder.

Figure 3-6:
Sin/Cos
Encoder Signal
Waveforms



Sin/cos encoder signals are output with one signal offset by 90 degrees from the other as shown in [Figure 3-6](#). The primary benefit to sinusoidal waveforms is that they can be used to interpolate much higher position resolutions compared to digital quadrature encoding. In an ideal setup the interpolation factor can be as high as 16,384, although practical limitations due to noise and other factors often lower this.

N-Series IONs provide a single channel of direct analog sin/cos encoder input for the primary axis (axis #1). The center value of the input analog signals is expected to be + 2.5V, with an allowed maximum magnitude $\pm 2.5V$, or 0.0V to 5.0V. Note that a more common magnitude is 1.0Vpp or an input range of 2.0V to 3.0V. Sin/cos input is not available on the auxiliary axis. If an auxiliary axis encoder input is desired then either quadrature AB or pulse & direction format can be selected.

3.3.3.1 Signal Processing & Waveform Correction

A wide variety of encoders are available that support sin/cos output, and there is an equally large range in the quality and characteristics of the signals output by these encoders. Ideal sin/cos encoder waveforms are noise free, follow the exact mathematical form of a sinusoid, have an exact centerpoint voltage of 0.0V, have the exact same peak to peak height, and are located 90 degrees apart.

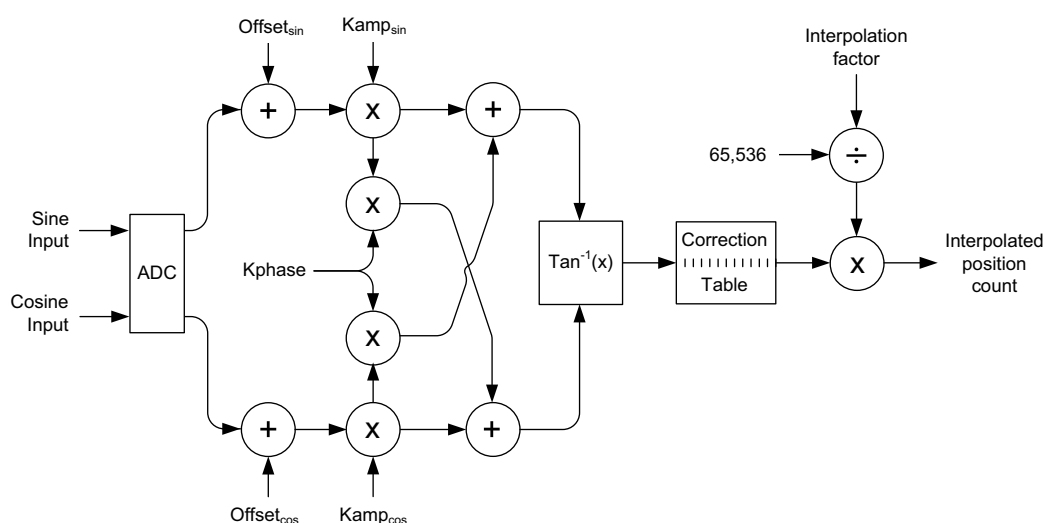


Figure 3-7:
Sin/Cos Signal
Correction
Calculation
Flow

For cases where the encoder output waveforms are not ideal N-Series IONs provide an algorithmic correction scheme along with user-settable adjustment coefficients to correct the incoming signals. [Figure 3-7](#) shows the calculation flow of this correction scheme.

Parameter settings for these coefficients can be determined in one of two ways; the first is that the ION unit can determine these values automatically. For this method to provide accurate coefficient values the motor must traverse a significant number of sin/cos electrical cycles, at least 10, but often more. For information on how to operate the N-Series ION for automatic correction coefficient determination see [Section 3.3.3.4, "Automatic Correction Coefficient Determination."](#)

Alternatively, the user can capture the signals from the encoder separately from the process of operating the motor and attached encoder, analyze the resulting sin/cos encoder output signals, and program the coefficients from an analysis of this data. In many cases use of an additional independent measuring system such as an interferometer or high precision encoder is incorporated into this process. A complete discussion of how best to undertake such an analysis is beyond the scope of this product manual. For assistance contact your PMD representative.

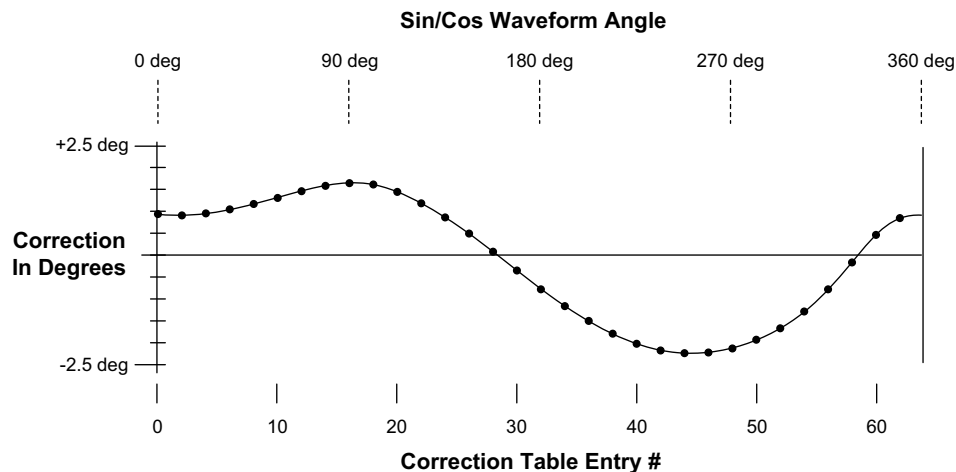
The following table details the settable waveform correction coefficients that can be determined automatically or set manually. Refer to [Figure 3-7](#) to better understand terms and how these adjustment coefficients are used.

Coefficient Name	Default Value	Range	Description & Usage
Offset _{sin} Offset _{cos}	0	-32,768 to +32,767	Sin and cos signal offsets. This correction allows the center-point of the incoming sin & cos signals to be centered at a numerical value of 0. The numerical representation of both the raw A/D input values and the Offset _{sin} and Offset _{cos} values are signed numbers from -32,768 to +32,767 corresponding to voltages of -2.5V and +2.5V respectively. For example to program an offset of -0.123 volts, the value $-0.123V * 32,768 \text{ counts} / 2.5V = -1,612$ is used.
Kamp _{sin} Kamp _{cos}	16,384	0 to 32,767	Sin and cos amplitude scale factors. This correction factor allows the amplitude of the sin signal to be matched to the cos signal. The default value of 16,384 represents a unity (1.0) scale factor. For example to set a scale factor of .950 the value $.950 * 16,384 / 1.0 = 15,565$ is used.
Kphase	0	-16,384 to 16,383	Quadrature phase adjustment factor. This correction factor allows the phase of the sin with respect to the cos waveform to be adjusted to 90 degrees. Refer to Figure 3-6 for a visual representation of this. Kphase is a signed quantity and has a unity (1.0) scaling value of 16,384. To create a shift in both the sin and the cos waveforms of N degrees a Kphase value of $-\tan(N)$ is programmed. For example to shift the sin signal backwards by 3.5 degrees and the cos signal forward by 3.5 degrees a Kphase value of $-\tan(-3.5 \text{ deg}) * 16,384 = -0.06116 * 16,384 = -1,002$ is used.

The above correction coefficients apply at all times once programmed. As such, the waveform correction entries can only effectively correct waveform inaccuracies that repeat across each 360 degree sin/cos cycle. To insure the best correction coefficient settings, data for multiple sin/cos electrical cycles should therefore be collected and averaged.

3.3.3.2 Derived Angle Correction Table

Figure 3-8:
Example
Correction
Waveform



As shown in [Figure 3-7](#) N-Series IONs provide the ability to make arbitrary adjustments to the derived angle output by the atan() calculation. If used, 64 entries are programmed into what is called the derived angle correction table. These table entries specify the output angle adjustment for a given input angle. Linear interpolation is used to smooth transitions from one programmed entry to the next.

A single 360 degree electrical cycle is encoded, with each stored adjustment value applying to an 5.625 degree wide (360 degrees / 64) portion of the 360 degree derived angle space. Each stored value represents an angle offset value. For example if the recorded adjustment value for the lookup input angle of 10.0 degrees is -1.535 degrees, then the output value will be $10.0 - 1.535 = 8.465$ degrees. Note however that due to the linear interpolation function, depending on the value of the neighboring offset values stored in the table and the input angle, the actual applied offset may not be exactly 1.535 degrees.

The scaling of each offset is 8,192 counts per 180.0 degrees, or approximately 45.5 counts/degree. The programmed value is a signed quantity with a range of -32,768 to +32,767.

The default configuration for this function is that the derived angle correction table is not used. To enable use of this function, table entries should first be specified, followed by the function being explicitly enabled.

Example:

The table below shows the first few entries of a hypothetical correcting table.

Table Index #	Centerpoint of Input Angle in Degrees	Offset in Degrees	Table Entry Value (Offset in Counts)
0	0.0	.95	43
1	5.625	.93	42
2	11.25	1.02	46
3	16.875	1.18	54
4	22.5	1.24	56
•			
•			
•			
63	354.375	.96	44

3.3.3.3 Interpolation Factor

The goal of the calculation flow shown in [Figure 3-7](#) is to determine the electrical angle, from 0 to 360 degrees, that is represented by the analog sin & cos signals. Once this angle has been determined a parameter known as the interpolation factor specifies how this angle value contributes to the motion IC's 32-bit register that holds the encoder position. Larger values mean a given change in angle translates to a larger change in portion count, and vice versa for smaller values of the interpolation factor.

The allowed range of the interpolation factor is 4 to 16,384. A value of 4 is effectively equivalent to quadrature AB, resulting in four resolved position values per complete sin & cos waveform cycle. The maximum value of 16,384 results in 16,384 resolved position locations per full sin/cos cycle. The selected interpolation factor must be a power of two. For example 4, 8, 16, 32, etc... The default value for the interpolation factor is 4,096.

The interpolation factor setting directly affects the axis' position resolution. For example if a linear sin/cos encoder outputs 100 sinusoids per millimeter, an interpolation factor of 4,096 results in an effective position resolution of $100 * 4,096 = 409,600$ counts / mm. And for the same system if the axis moves at 10 mm/sec the speed is $10 \text{ mm/sec} * 409,600 \text{ counts / mm} = 40,960,000 \text{ counts / sec}$.

Sin/cos encoder resolution in counts, settable via the interpolation factor, does not necessarily equate to accuracy. Many variables can reduce the effective accuracy of sin/cos encoders including noise, waveform variations that can not be corrected for, and variations over time of the encoder signal output.



3.3.3.4 Automatic Correction Coefficient Determination

As indicated in [Section 3.3.3.1, "Signal Processing & Waveform Correction,"](#) sin/cos waveform correction coefficients can be determined automatically by the N-Series ION. Activation of this process occurs via user command, and once commanded, during movement of the sin/cos encoder special observer circuitry records and characterizes the incoming sin and cos input waveforms. From this process the Offset_{\sin} , Offset_{\cos} , Kamp_{\sin} , Kamp_{\cos} , and Kphase coefficients are automatically determined.

To function correctly the observer must be able to characterize a sufficient number of incoming sin/cos electrical cycles. This number may be as few as ten cycles but it may be more. To satisfy this requirement after activating the automatic calibration process many applications will command a special initialization move before regular axis operations begin. Note that during such an initialization move (or moves), the motion of the axis may not be as stable or as accurate as after characterization is complete.

When characterization is complete the 'calibrated' bit in the Magellan Motion Control IC's Drive Status register will transition from 0 to 1, and the automatically determined coefficients become the active parameter settings. These parameters can be retrieved by the user if so desired.

The user has two choices for how to utilize the N-Series ION's automatic sin/cos correction determination circuitry. The first is to always execute automatic determination so that sin/cos encoder correction values are newly derived at each and every powerup or reset. This approach has the advantage that any changes over time in the sin/cos output signals will be tracked automatically.

The second is to run the automatic determination process once, and then read and store the resultant values to the Magellan Motion Control IC's NVRAM where they will be applied at each startup. This approach has the benefit that the correction values will always be consistent. For more information on the N-Series ION Magellan Motion Control IC's NVRAM refer to [Section 4.2.3.1, "Loading the Magellan IC NVRAM."](#)



It is up to the user to determine whether the N-Series ION's process for automatically determining correction coefficients is sufficiently accurate for a given application. In general, automatically determined values will not be as accurate as those determined by a carefully constructed manual process for sin/cos signal characterization and coefficient setting.



For sufficiently distorted uncorrected sin/cos encoder signals it is possible that an initialization move, used during the N-Series ION's automatic correction determination process, will result in unstable motion. It is up to the user to determine whether axis motion can be safely commanded in a given application before sin/cos encoder correction parameters are applied.

3.3.3.5 Position Capture

As for quadrature AB encoder format, both the *Index* and *Home* signals may be used as position capture triggers for the primary axis when it is set to sin/cos encoder type. Unlike quadrature AB encoders however there is no signal qualification of the *Index* signal with respect to the sin and cos signal inputs. Both *Index* and *Home*, when transitioning from high to low, will generate a capture trigger regardless of the state of the sin or cos signals. Note that this default *Index* and *Home* signal sense interpretation can be user programmed.

Another change relative to quadrature AB encoder signal processing is the latency of the position capture after the selected trigger signal (either index or home) goes active. For sin/cos encoder position capture the worst case latency is 51.2 μ Sec, which is the same as the N-Series ION's minimum sample time setting.

If the rate of the incoming sin/cos signal frequency is greater than 4.88 kHz however the worst case latency will be reduced. For a sin/cos encoder outputting encoder signals at a particular frequency ($\sin_{\text{frequency}}$) above 4.88 kHz, the worst case position capture latency can be approximated by the expression: $\text{latency} = 51.2 \mu\text{Sec} * 4.88 \text{ kHz} / \sin_{\text{frequency}}$. For example if a motor is moving such that its sin/cos encoder output signal frequency is 75 kHz, the worst case position capture latency will be approximately $51.2 \mu\text{Sec} * 4.88 / 75.0 \text{ kHz} = 3.3 \mu\text{Sec}$.

3.3.3.6 C-Motion Commands

All of the parameters and actions associated with these functions are managed by the Magellan Motion Control IC.

To set and read back the correction coefficients used to process sin/cos signals the **SetAnalogCalibration** and **GetAnalogCalibration** commands are used, respectively. Similarly, to set and read back as well as enable or disable use of the derived angle correction table the **SetAnalogCalibration** and **GetAnalogCalibration** commands are used. To initiate automatic correction coefficient determination the **CalibrateAnalog** command is used.

There are many additional commands used in association with encoder input including specifying the encoder type, reading the encoder position, and reading the captured position value. For detailed information on these C-Motion commands and others refer to the *Magellan Motion Control IC Programming Reference*.

3.3.3.7 Connections & Associated Signals

The table below shows the signals associated with sin/cos encoder format. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Sin + *	39	Differential positive sin signal input.
Sin-*, **	40	Differential negative sin signal input.
Cos + *	41	Differential positive cos signal input.
Cos-*, **	42	Differential negative cos signal input.
Index1 + *	15	Differential positive index signal input.
Index1-*	16	Differential negative index signal input.
Home	24	Home signal input for position capture of the primary (BiSS-C) encoder.
5V	29	+ 5V power output which may be used to power encoder circuitry.
GND	30	Ground. Note that any of the N-Series ION's ground pins can be used including 10, 30, and 38.

* These signals are located on pins with multiple programmable functions. The pin function applies only when the encoder type has been programmed to sin/cos.

** If a single-ended encoder connection is used the + pin carries the signal and this pin should be connected to 2.5V.

The index signals can be connected in one of two ways. Single-ended means that only one wire per signal is used, while differential means two wires encode each signal (labeled + and -). Differential connections are recommended for the highest level of reliability. If single-ended connections are used, only the + wire is connected and the - wire should be left floating.

3.3.3.8 Using Sin/Cos Signals As General Purpose Analog Inputs

Rather than using the sin/cos signal inputs for encoder input it is also possible to use these signals for general purpose analog input, thereby providing up to two additional general purpose analog inputs to the N-Series ION. Note however that unlike the AnalogIn +/- signals which have an input range of +/- 10V, the sin/cos signal inputs have an analog input range of 0.0 to 5.0V.

To operate the Sin and Cos signals as general purpose analog inputs the CMESetDefault command is used. Refer to [Section A.2, "N-Series ION NVRAM Default Settings,"](#) for details. To read the sin and cos signal voltages a 32 bit read at PIO register 0x344 is used. The 16-bit sin value is stored in the high 16 bits and the cos value in the low 16 bits.

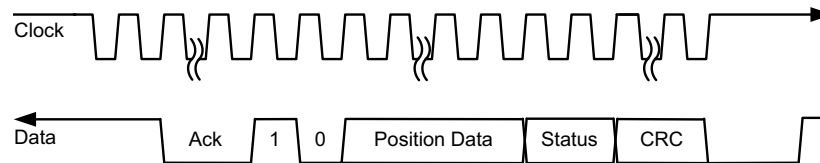
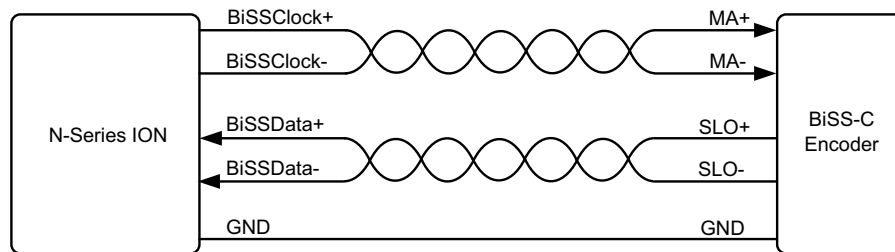
Each returned 16-bit value is an unsigned integer representing a voltage from 0V to 5V. For example if 0.0V is presented at the sin or cos pin the returned value is 0, and if 5.0V is presented the returned value is 65,535.

3.3.3.9 Sin/Cos Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process these signals. Sin and Cos signals share the same processing circuitry as QuadB1 and QuadA1 respectively. Therefore refer to [Section 3.3.1.5, "Quadrature Encoder Signal Processing,"](#) for information on this processing circuitry.

3.3.4 BiSS-C & SSI Encoders

Figure 3-9:
BiSS Encoder
Format
Interconnection
Diagram and
Serial Data
Stream



BiSS-C (Bidirectional Serial/Synchronous interface) is a synchronous serial interface for acquiring position data from an encoder. It is a master/slave interface, with the N-Series ION functioning as the master and controlling the timing of position acquisition, and the encoder functioning as the slave. The signal interfacing scheme and serial data transfer format for typical BiSS-C encoders are shown in [Figure 3-9](#).

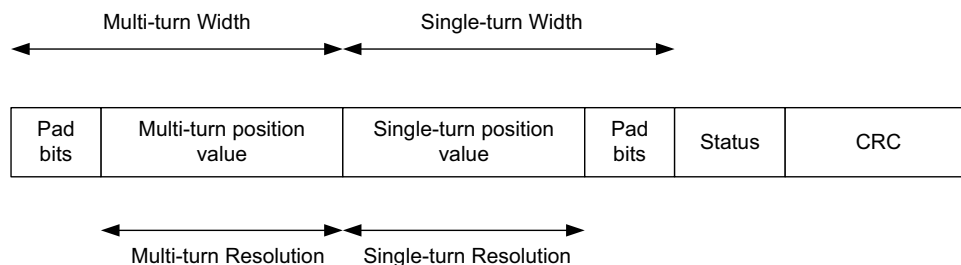
The electrical interface consists of two unidirectional differential pairs of lines, known as MA +/– which transmits position acquisition requests and timing information (clock) to the encoder, and SLO +/– which transfers position data from the encoder to the master synchronized to MA. The corresponding N-Series ION signals are named BiSSClock +, BiSSClock–, BiSSData +, and BiSSData–.

The N-Series ION circuitry which supports BiSS-C encoders also supports SSI (Synchronous Serial Interface) encoders. Much of the information detailed here for BiSS-C encoders also applies to SSI, however for information specific to interfacing to SSI encoders refer to [Section 3.3.4.4, “SSI Encoder Interfacing.”](#)

N-Series IONs support a single BiSS-C or SSI encoder on the primary axis. There is no index capture supported, but the home input can be used as a position trigger capture for the BiSS-C or SSI encoder. If used, the auxiliary axis can be set for quadrature AB or pulse & direction input.

3.3.4.1 Interface Data Format

Figure 3-10:
BiSS Encoder
Format
Interconnection
Diagram and
Serial Data
Stream



There is considerable variability in the total set of interface commands and response words supported by BiSS-C encoders. This is because the BiSS-C standard itself encompasses a number of different functions, and because some vendors add functions that are specific to their encoder offerings.

Nevertheless nearly all BiSS-C encoders support a basic query and response to provide the current position of the encoder. [Figure 3-9](#) shows this word in the overall BiSS-C data stream and [Figure 3-10](#) shows the specific format of this position word. There are two separate position information blocks within the overall word, one carrying the

multi-turn position and the other carrying the single turn position. Encoders which do not provide a multi-turn function will not use the multi-turn data block.

Both the multi-turn and single turn blocks support separate sizes (in number of bits) of the entire block and the bits that carry just the position. The total block size is referred to as the width, and the position word size is referred to as the resolution. The width may be the same as the resolution or it may be larger. If larger, leading or trailing "pad" zeroes are inserted into the data stream according to the difference in size. For the multi-turn data block the pad bits lead the position data, and for the single turn block the pad bits trail the position data.

Four separate parameters are user-specified to allow the N-Series ION to correctly decode the provided encoder data; multi-turn width, multi-turn resolution, single turn width, and single turn resolution. In addition a bit shift parameter is specified to reference the position data within the overall transmission from the encoder. Refer to [Section 3.3.4.2, "Interface Control Registers,"](#) for more information on these registers.

In calculating the 32-bit position within the Magellan position registers N-Series IONs assume that the single turn position word is a power of two. The multi-turn and single turn position data values provided by the encoder therefore effectively form a signed binary word. For example if the resolution of the multi-turn word is 6 bits and the resolution of the single turn word is 12 bits, the maximum # of different positions that can be specified by the encoder is 266,144 ($2^6 * 2^{12}$), and the range of values is -131,072 to +131,071.

If the total resolution of the combined multi-turn and single turn position word is less than 32 bits the encoder value will be sign extended to the Magellan IC's actual position word size of 32 bits. In addition, if the combined multi-turn and single turn position word exceeds 32 bits, bits above 32 will be truncated.



3.3.4.2 Interface Control Registers

There are a number of user settable values that configure the interface for a particular BiSS-C or SSI encoder. These values, which are all accessed via PIO registers, are listed in the table below:

Name	Bit field	Allowed Values	Description
BiSS Config (PIO register 0x100)			
Position Data Format*	2	0 - Binary 1 - Gray Code	(SSI only) This one-bit field specifies whether the attached encoder returns position data in binary format or gray code format.
BiSS-C or SSI encoder format	3	0 - SSI 1 - BiSS-C	This one-bit field specifies whether the attached encoder is of SSI format type or BiSS-C format type.
CRC Word Format**	4:8	0 - No CRC 6 - 6-bit, (0x43)	This five bit field specifies how the attached encoder returns CRC information. Note that the value in parenthesis (0x43) details the CRC polynomial that is expected. A value of 0x43 (hexadecimal 43) uses a polynomial of $X^6 + X^1 + X^0$.
BiSS Encoder Resolution (PIO register 0x102)			
Position word size	0:4	7-32	This five-bit field specifies the size of the position word returned by a BiSS-C or SSI encoder. A value of 7 means a word size of 7 bits, a value of 8 specifies a word of 8 bits, etc. The value should be equal to the sum of the BiSS multi-turn resolution and the BiSS single-turn resolution.
BiSS Frequency (PIO register 0x104)			
Clock Frequency	0:7	3-19 (10 MHz to 1 MHz)	This 8-bit field sets the clock frequency - that is, the frequency output on the N-Series ION's BiSSClock signals. The allowed values are 3, 7, 9, 15, 17, and 19 corresponding to a clock frequency of 10 MHz, 5 MHz, 4 MHz, 2.5 MHz, 2 MHz, and 1 MHz.
BiSS Enable (PIO register 0x106)			

Name	Bit field	Allowed Values	Description
Enable/Disable	0	0 - Disabled 1 - Enabled	This 1-bit field determines whether BiSS-C or SSI encoder operation is active (enabled) or inactive (disabled). At powerup this bit is set to 0 (disabled) and to change any of the control settings detailed in this table this value must be at this disabled state. When this field is changed to enabled the N-Series ION will use the programmed settings and initialize BiSS-C or SSI encoder operation and thereafter continually track the encoder position. Encoder operation can be halted by setting this field to 0 (disable), however see warning below regarding safety.
BiSS Single-Turn Word (PIO register 0x108)			
Resolution	0:7	0-32	This 8 bit field specifies the resolution of the single turn position word in number of bits.
Width	8:15	0-32	This 8 bit field specifies the width of the single turn position word. The width must be greater than or equal to the resolution.
BiSS Multi-Turn Word (PIO register 0x10A)			
Resolution	0:7	0-32	This 8 bit field specifies the resolution of the multi-turn position word in number of bits.
Width	8:15	0-32	This 8 bit field specifies the width of the multi-turn position word in number of bits. The width must be greater than or equal to the resolution.
BiSS Position Word Shift (PIO register 0x10C)			
Shift	0:3	0-15	This 4 bit field specifies the number of bits to right-shift the data transmission to correctly read the position word. A value of 0 means there is no shift, a value of 3 means the shift is 3 bits, etc. BiSS-C encoders typically require a shift of at least 2 to account for error and warning bits.

* This setting is only used with SSI format encoders. It is not used with BiSS-C format encoders.

** This setting is only used with BiSS-C format encoders. It is not used with SSI format encoders.

For information on accessing Peripheral I/O space (PIO) registers refer to [Section 3.10.1, "Peripheral I/O Space."](#)



All of the above settings configure the N-Series ION's BiSS-C interface circuitry so that the data format and content of messages transmitted by the encoder are interpreted correctly. These settings do not affect the behavior of the encoder which are considered to output a fixed, unchangeable format. To the extent that some BiSS-C encoders may support format programmability, this function must be undertaken by the user separately from the settings of the above parameters.

3.3.4.3 Configuring and Initializing BiSS-C Encoder Operation

For a factory default N-Series ION the Enabled/Disabled field of the BiSS Enable register (PIO address 0x106) is disabled, meaning no clock signals or other communications occur between the ION and the encoder. To initialize the N-Series ION for operation with a BiSS-C encoder the various control settings indicated in the table above should be programmed, and then the Enabled/Disabled bit should be changed to Enabled. For more information on PIO registers refer to [Section 3.10.1, "Peripheral I/O Space."](#) Some or all these PIO settings can be stored permanently in NVRAM to avoid having to program them at each power up.

The success or failure of the initialization of the BiSS-C encoder will be indicated by the return value of the C-Motion call used to enable BiSS-C operation. Once the encoder is initialized and operating correctly an encoder type of BiSS is set via a C-Motion command to the N-Series ION's on-board Magellan Motion Control IC. Similar to PIO register settings, if desired this setting can be stored permanently in NVRAM however this is accomplished via the Magellan's NVRAM memory rather than the Device NVRAM where PIO register settings are stored.

For more information on storing values into the N-Series ION Device NVRAM refer to [Section 3.10.7, “Setting Default Values.”](#) For more information on storing values into the Magellan NVRAM refer to [Section 3.2.2, “Magellan Initialization.”](#)

[Appendix B, “BiSS-C Encoder Settings,”](#) provides several examples of specific BiSS-C encoders and their corresponding control register settings.

Disabling an operating BiSS-C encoder should only be undertaken when it is not being used to actively control the axis.



3.3.4.4 SSI Encoder Interfacing

In addition to BiSS-C format encoders N-Series IONs also support SSI format encoders. The SSI encoder interface is essentially a simplified version of the BiSS-C encoder interface, but there are a few differences to be aware of.

By convention the interface signal names for SSI encoders are Data + , Data- , Clock + , and Clock- rather than SLO + , SLO- , MO + , and MO- . However these signals connect to the same N-Series ION signals which are named BiSSData + , BiSSData- , BiSSClock + , and BiSSClock- .

Whether the attached encoder is of SSI type or BiSS-C type must be explicitly specified during setup, and is accomplished via a PIO register as detailed earlier. SSI encoders do not support CRC data transmission checking, and unlike BiSS-C encoders SSI encoders can support gray code as well as binary code position words so this must be specified if the SSI encoder type is selected.

For the available BiSS-C and SSI encoder settings refer to [Section 3.3.4.2, “Interface Control Registers.”](#)

3.3.4.5 Position Capture

The *Home* signal may be used as position capture trigger for the primary axis when it is set to BiSS-C encoder type. There is no index signal capture available when the encoder type is BiSS-C. There is no signal qualification of the *Home* signal with respect to the BiSS-C encoder signals. Note that the *Home* signal sense interpretation can be user programmed. For BiSS-C or SSI encoder position capture with the *Home* signal the worst case latency is 51.2 μ Sec.

3.3.4.6 C-Motion Commands

The primary C-Motions commands used to access the Peripheral I/O registers are **PeriphOpenPIO**, **PeriphRead**, and **PeriphWrite**. For complete information refer to the *C-Motion PRP II Programming Reference*.

There are numerous Magellan C-Motion commands that relate to encoder feedback, including commands that set up, retrieve, capture, compare, set, or otherwise utilize the current encoder position. Refer to the *Magellan Motion Control IC Programming Reference*.

3.3.4.7 Connections & Associated Signals

The table below shows the signals associated with BiSS-C and SSI encoder format. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
BiSSClock + *	15	Differential positive BiSS-C or SSI encoder clock signal output
BiSSClock-*	16	Differential negative BiSS-C or SSI encoder clock signal output
BiSSData + *	17	Differential positive BiSS-C or SSI encoder data signal input
BiSSData-*	18	Differential negative BiSS-C or SSI encoder data signal input
Home	24	Home signal input
5V	29	+ 5V power output which may be used to power encoder circuitry
GND	30	Ground. Note that any of the N-Series ION's ground pins can be used including 10, 30, and 38

* These signals are located on pins with multiple programmable functions. The pin function applies only when the encoder type has been programmed to BiSS-C.

3.4 Motion Peripheral Functions

Motion peripheral functions are functions related to axis status and control signals that connect to, and are processed by, the Magellan Motion Control IC. These signals include *Enable*, *FaultOut*, *HostInterrupt*, *Home*, *PosLim*, *NegLim*, *HallA-C*, *AxisIn*, and *AxisOut*.

How these signals are interpreted or controlled by the Magellan IC is explained in detail in the *Magellan Motion Control IC User Guide*.

3.4.1 Enable, FaultOut, and HostInterrupt Signals

This group of Magellan IC-connected signals provide critical control and status functions.

The *Enable* input signal provides a global drive control signal that must be active for the drive to function. This signal is typically set active (low) by connecting to ground, or can be integrated with control logic that may dynamically set this signal inactive (high) or active if particular external conditions occur. If this signal is inactive the motor drive output will be disabled. If the motor is moving at the time of disable this will have the effect of freewheeling the motor. For more precise control of a moving motor in a safety-related situation the *Brake* signal may be used. The *Brake* signal input is described in [Section 3.8.6, "Motor Brake."](#)

The *FaultOut* input signal is used to indicate a fault condition of the Magellan Control IC. This signal is always active high, and its sense cannot be changed. The default setting of this signal is to indicate overtemperature, overcurrent, and over/under voltage faults, however these conditions can be user programmed.

The *HostInterrupt* output signal is used to notify external circuitry that an event has occurred within the Magellan IC. This signal is active low and its sense can not be changed. To control which internal events result in activation of this signal the user programs a mask register. The default setting of this mask is all off, meaning that the *HostInterrupt* signal is inactive regardless of the internal status of the motion IC.



Disabling the N-Series ION unit via the *Enable* signal results in the motor drive being disabled and the motor freewheeling. It is up to the user to determine whether disabling of the drive is safe under the application conditions.

3.4.1.1 C-Motion Commands

All of the parameters associated with these functions are managed by the Magellan Motion Control IC. For detailed information on these C-Motion commands refer to the *Magellan Motion Control IC Programming Reference*.

3.4.1.2 Connections & Associated Signals

The table below shows the signals associated with these signals. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Enable	25	Magellan IC Enable input signal
FaultOut	26	Magellan IC FaultOut output signal
HostInterrupt*	36	Magellan IC HostInterrupt output signal
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38

* This signal is located on a pin with multiple programmable functions. The pin function applies only if so programmed. For internal signal processing information for this pin refer to the corresponding DigitalIO signal (Digital IO6) described in [Section 3.5.1.7, "Digital I/O Signal Processing."](#)

For these signals to function properly one or more of the digital grounds must be connected.

3.4.1.3 Enable Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process I/O signals.

The following diagram shows this for the *Enable*, *PosLim*, *NegLim*, and *Brake* signals.

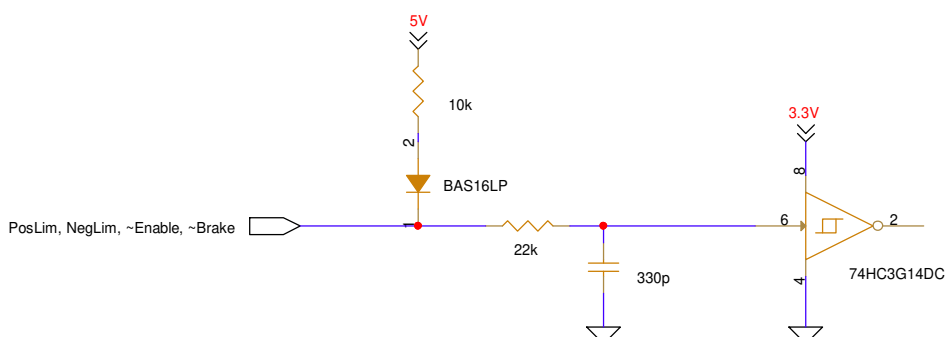


Figure 3-11:
Enable, PosLim,
NegLim, and
Brake Signal
Internal
Processing
Circuitry

3.4.1.4 FaultOut Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process I/O signals.

The following diagram shows this for the *FaultOut* signal.

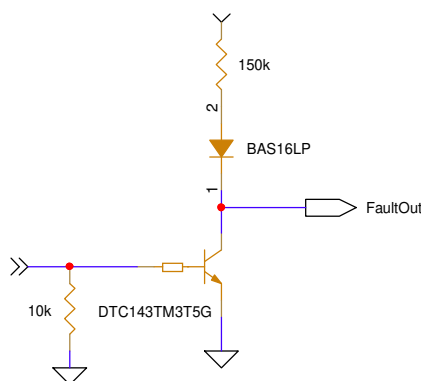


Figure 3-12:
FaultOut
Internal
Processing
Circuitry

3.4.2 Home, Limits, and Hall Signals

This group of signals are used to operate, or safeguard, the motor axis during motion.

The *Home* input signal provides a general purpose axis initialization signal that can be set to trigger capture of the encoder position. In a typical homing sequence the axis is commanded to move in the direction of the home signal and stop, or capture, the encoder position when the home signal is encountered. This signal has a default sense of active low (axis is in home switch when low) however this interpretation can be user programmed.

The *PosLim* and *NegLim* input signals can be set to halt the motor motion when encountered. In a typical limit switch setup these inputs are located at the extreme motion travel limits, one for the forward motion direction (*PosLim*) and one for the backwards motion direction (*NegLim*). These signals have a default sense of active low (axis is in limit switch when low) however this interpretation can be user programmed.

The *HallA-C* signals accept Hall-encoded inputs for Brushless DC motors. They provide commutation information used to initialize phasing, or to actively drive the motor. These commutation signals can also be used as the encoder source in cases where the motor does not have a separate position encoder. The A, B, and C Hall signals encode six valid states as follows: A on, A and B on, B on, B and C on, C on, C and A on. By default a sensor is defined as being on when its signal is high, however this interpretation can be user programmed.

See the *Magellan Motion Control IC User Guide* for more information on how any of the above signals are configured and used during motion operation.

3.4.2.1 C-Motion Commands

All of the parameters and actions associated with these functions are managed by the Magellan Motion Control IC. For information on these C-Motion commands refer to the *Magellan Motion Control IC Programming Reference*.

3.4.2.2 Connections & Associated Signals

The table below shows the signals associated with these functions. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Home	24	Home signal input. By default a low signal indicates a valid home condition, but this signal sense can be user programmed.
PosLim	22	Positive limit signal input. By default a low signal indicates a valid limit switch condition, but this signal sense can be user programmed.
NegLim	23	Negative limit signal input. By default a low signal indicates a valid limit switch condition, but this signal sense can be user programmed.
HallA	19	Hall signal input A for Brushless DC motors.
HallB	20	Hall signal input B for Brushless DC motors.
HallC	21	Hall signal input C for Brushless DC motors.
5V	29	+ 5V power output may be used to power sensor circuitry or the motor's Hall sensor/encoder circuitry.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used with these signals, including pins 10, 30, or 38.

For these signals to function properly one or more of the digital grounds must be connected.

3.4.2.3 PosLim, NegLim Signal Processing

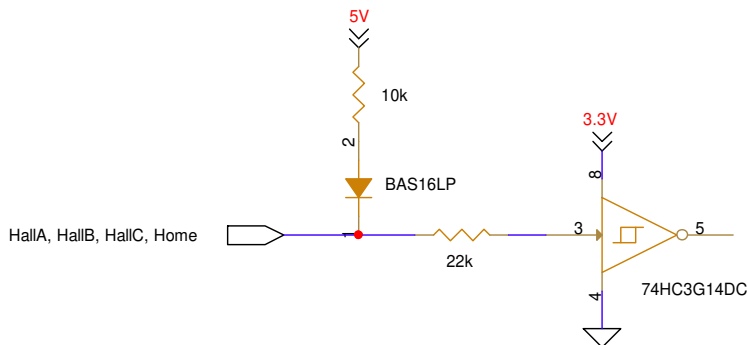
Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process these signals. *PosLim* and *NegLim* signals share the same type of processing circuitry as the *Enable* signal. Therefore refer to [Section 3.4.1.3, "Enable Signal Processing,"](#) for information on this processing circuitry.

3.4.2.4 HallA, HallB, HallC, and Home Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process I/O signals.

The following diagram shows this for the *HallA*, *HallB*, *HallC*, and *Home* signals.

Figure 3-13:
HallA, HallB,
HallC and
Home Signal
Internal
Processing
Circuitry



3.4.3 AxisIn, AxisOut Signals

These signals are input to, or output by, the Magellan Motion Control IC and facilitate coordination of motion sequences with external hardware.

The *AxisIn* signal is a general purpose digital input which can be utilized to allow external events to directly control internal events in the Magellan IC. A common use is to use this signal to affect a motion profile change on the fly, but many other externally triggered changes are possible. The default condition of this signal is that its function is unprogrammed. This signal does not have a default sense interpretation, the eventual interpretation depending on how its function is programmed.

The *AxisOut* signal is a general purpose digital output which can be utilized to signal various internal conditions of the motion control IC to external circuitry. The default condition of this signal is that its function is unprogrammed. This signal does not have a default sense interpretation its eventual interpretation depending on how its function is programmed.

See the *Magellan Motion Control IC User Guide* for more information on how these signal functions are utilized and programmed.

3.4.3.1 C-Motion Commands

All of the parameters and actions associated with these functions are managed by the Magellan Motion Control IC. For information on these C-Motion commands refer to the *Magellan Motion Control IC Programming Reference*.

3.4.3.2 Connections & Associated Signals

The table below shows the signals associated with these functions. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
AxisIn [*]	33	Magellan IC AxisIn signal input.
AxisOut [*]	35	Magellan IC AxisOut signal output.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

**These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed. For internal signal processing information for these pins refer to the corresponding DigitalIO signals (DigitalIO3 and DigitalIO5 for AxisIn and AxisOut respectively) described in [Section 3.5.1.7, "Digital I/O Signal Processing,"](#)*

For these signals to function properly, one or more of the digital grounds must be connected.

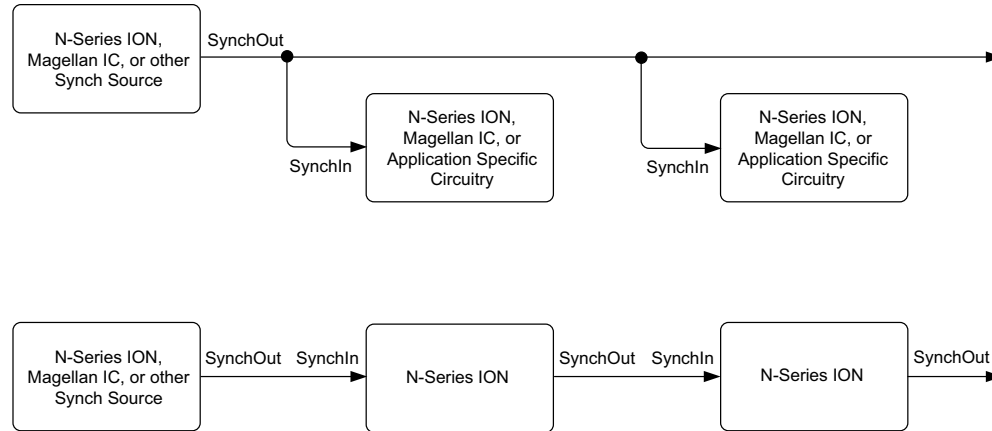
3.4.4 SynchIn, SynchOut Signals

There are two overall uses for the *SynchIn* and *SynchOut* signals. The first is to synchronize servo loops between multiple N-Series IONs and the second is to initiate Node IDs during startup. In the section below the servo loop synchronization function is described. For information on daisy chain Node ID assignment refer to [Section 3.6.1.3, "Daisy-Chain Node ID Assignment."](#)

The *SynchIn* and *SynchOut* signals are input to, or output by, the Magellan Motion Control IC and allow precise synchronization of the servo loop of one N-Series ION with the servo loop of another. This is most often used for multi-axis motion coordination. One unit is specified as the master, which outputs a synch signal every 51.2 μ Sec, and all other units are specified as slaves. Slave units, rather than generating their servo loop timing internally, use the external synching signal to clock their servo loops.

Synch pulses are defined as a transition of the *SynchOut* signal from high to low.

Figure 3-14:
Connection
Schemes for
Multiple Unit
Synching



As shown in [Figure 3-14](#), using N-Series IONs there are two available connection schemes from synching the units together. The one-signal method, which is the scheme shown at top, has the advantage of using only one signal per device, either the *SynchOut* signal for the master unit or the *SynchIn* signal for the slave units. In addition this method is compatible with other PMD devices such as Magellan ICs and Prodigy boards, allowing synching of devices other than N-Series IONs.

*The two-signal method, which is the bottom scheme shown in the diagram, daisy chains the *SynchIn* and *SynchOut* signals thus requiring two signals per N-Series ION. The main advantage of this method is that it is compatible with CAN or RS485 daisy chain Node ID assignment, which also uses the *SynchIn* and *SynchOut* signals. For more information on this feature refer to [Section 3.6.1.3, "Daisy-Chain Node ID Assignment."](#)*

There is additional important information related to operating devices that are synched, particularly related to start up and transition from internally generated servo timing to external signal-based timing. Refer to the *Magellan Motion Control IC User Guide* for complete details.



For synched slave units precautions should be taken to insure a reliable, noise-free stream of synch pulses, along with a seamless transition from internally generated servo timing to externally determined timing. Dropouts or glitches of the *SynchIn* signal may result in unsafe motion.

3.4.4.1 C-Motion Commands

All of the parameters and actions associated with these functions are managed by the Magellan Motion Control IC. For information on these C-Motion commands refer to the *Magellan Motion Control IC Programming Reference*.

3.4.4.2 Connections & Associated Signals

The table below provides signal information for the *SynchIn* and *SynchOut* signals when used for servo loop synching. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
SynchIn*	34	Magellan IC synch signal when unit is operated as slave. A synch pulse is received when this signal transitions from high to low.
SynchOut*	37	Magellan IC synch output signal when unit is operated as master or when daisy chain connection scheme is used. A synch pulse is commanded upon transition of this signal from high to low.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

The table below provides signal information for the *SynchIn* and *SynchOut* signals when used for daisy chain Node ID assignment.

Pin Name	Pin #	Description
SynchIn*	34	When used for daisy-chain Node ID assignment this input gates the RS485 and CAN network SetNodeID command such that an active level (low) on this signal as well as a NodeID/address match is required for network message receipt.
SynchOut*	37	When used for daisy-chain Node ID assignment this output determines whether the next device in the daisy chain will be enabled to receive SetNodeID commands.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

* These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed. For internal signal processing information for these pins refer to the corresponding *DigitalIO* signals (*DigitalIO4* and *DigitalIO7* for *SynchIn* and *SynchOut* respectively) described in [Section 3.5.1.7, "Digital I/O Signal Processing."](#)

3.5 I/O Functions

3.5.1 General-Purpose Digital I/O

In addition to signals that are directly associated with the Magellan Motion Control IC such as *AxisIn*, *AxisOut*, *Home*, *QuadA*, *QuadB*, and *Index*, N-Series IONs support eight general-purpose bi-directional digital I/O signals that are an available resource within the drive device.

All of the bi-directional I/O bits, which are labeled *DigitalIO1-8*, are shared with other N-Series ION pin functions. So for whichever bits are intended to be used as digital I/Os, those pin function settings must be programmed accordingly. For information on setting the pin multiplexing function refer to [Section 3.10.2, "Pin Multiplexing Control."](#)

Digital I/O resources are accessed through the Peripheral I/O space, or PIO for short. See [Section 5.1.2, "Peripheral I/O Space,"](#) for more information on the N-Series ION's PIO address map.

3.5.1.1 Setting the I/O Direction

Four of the bi-directional signals *DigitalIO1-4* must have their directions explicitly programmed while signals *DigitalIO5-8* do not require this.

Specifying the direction of one of the *DigitalIO1-4* signals is accomplished using a mask/value method. The desired direction (input or output) for each bit is written using an I/O direction value register (PIO address 0x222), and the specific bits that are to be written to are specified via a separate I/O direction mask (PIO address 0x220).

Each bit position in the mask that has a value of one will result in the corresponding I/O direction bit being updated by the write command. Each bit position in the mask with a value of zero will result in the corresponding bit being ignored.

A zero bit value written to the I/O direction register selects the corresponding signal as an input, and a one selects it as an output. The default value of all digital IO direction settings is input.

To read the current active direction settings the PIO 0x224 register is used.

3.5.1.2 Reading Digital Inputs

For *DigitalIO1-4* bits programmed as inputs and for all *DigitalIO5-8* bits, the status of each signal can be read via the digital I/O input value register (PIO address 0x200). When the signal input is low (0V), the bit value in this register will be zero, and when the input is high (5V), the bit value will be one.

3.5.1.3 Writing Digital Outputs

Similar to setting the direction of each bit, setting the output state of each *DigitalIO* 1-4 bit programmed as an output and for all *DigitalIO* 5-8 bits, is accomplished using a mask/value method. The digital output command is set via a write value register (PIO address 0x212) that holds the desired output signal levels along with a separate write value mask register (PIO address 0x210) that holds the write mask.

Each bit position in the write mask with a value of one will result in the corresponding bit value in the write value register becoming the commanded digital output value for that signal. Each bit position in the mask with a value of zero will result in the corresponding bit of the write value being ignored. The register (PIO address 0x214) can be read to determine the current active output signal settings.

Each zero bit value written to the write value register results in a low (0V) output and each one bit value results in a high (5V) output. The default value of all digital IOs programmed as outputs is zero (low).

3.5.1.4 Bit Mapping

The following table correlates the N-Series ION's digital I/O signals with the bit positions of the various registers described above used to write, read, and configure the digital I/Os.

Signal Name	I/O Direction Value (0x222)	I/O Direction Mask (0x220)	I/O Input Value (0x200)	I/O Write Value (0x212)	I/O Write Mask (0x210)
DigitalIO1	0	0	0	0	0
DigitalIO2	1	1	1	1	1
DigitalIO3	2	2	2	2	2
DigitalIO4	3	3	3	3	3
DigitalIO5	N/A	N/A	4	4	4
DigitalIO6	N/A	N/A	5	5	5
DigitalIO7	N/A	N/A	6	6	6
DigitalIO8	N/A	N/A	7	7	7

Examples

In an application, four of the N-Series ION's general purpose digital IOs will be used. DigitalIO2 and DigitalIO5 will be utilized as outputs and DigitalIO6 and DigitalIO8 will be utilized as inputs.

To program the direction, first the direction mask is written with a value of 2 specifying '1' mask values at bits 1. Next the direction value is written with a value of 2 specifying an output function at bit 1.

To command low signal output values on both the DigitalIO2 and DigitalIO5 outputs a 0x12 (decimal 18) value is written to the write value mask register and a 0 is written to the write value register.

To command DigitalIO5 high without affecting the state of DigitalIO2 a 0x10 (decimal 16) value is written to the write value mask register and a 0x10 (decimal 16) is written to the write value register.

To command DigitalIO2 high without affecting the state of DigitalIO5 a 2 value is written to the write value mask register and a 2 value is written to the write value register.

3.5.1.5 C-Motion Commands

To set or retrieve the value of the N-Series ION's general purpose digital I/O the user must first open a Peripheral of type PIO. This is accomplished using the C-Motion command **PeriphOpenPIO**.

Using the returned *Peripheral* handle, the user can write the 16-bit word value and mask using the C-Motion command **PeriphWrite**. Note that a single two-word write must be used with the write mask in the high 16-bit word and the write value in the low 16-bit word.

To read back a previously written write value or mask, or to read back the current actual output signals commands, or to read back the current input signal states, the command **PeriphRead** is used.

For complete information on these commands refer to the *C-Motion PRP II Programming Reference*.

PIO space writes to the general purpose digital I/O write value and mask register should always be made via a single call to **PeriphWrite**.



3.5.1.6 Connections & Associated Signals

The table below shows the signals associated with these functions. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
DigitalIO1*	31	Digital I/O 1
DigitalIO2*	32	Digital I/O 2
DigitalIO3*	33	Digital I/O 3
DigitalIO4*	34	Digital I/O 4
DigitalIO5*	35	Digital I/O 5
DigitalIO6*	36	Digital I/O 6
DigitalIO7*	37	Digital I/O 7
DigitalIO8*	9	Digital I/O 8
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38

*These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed.

For digital I/O signals to function properly one or more of the digital grounds must be connected.

3.5.1.7 Digital I/O Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process Digital I/O signals.

The following diagram shows this for signals DigitalIO1-4, which use a push-pull methodology:

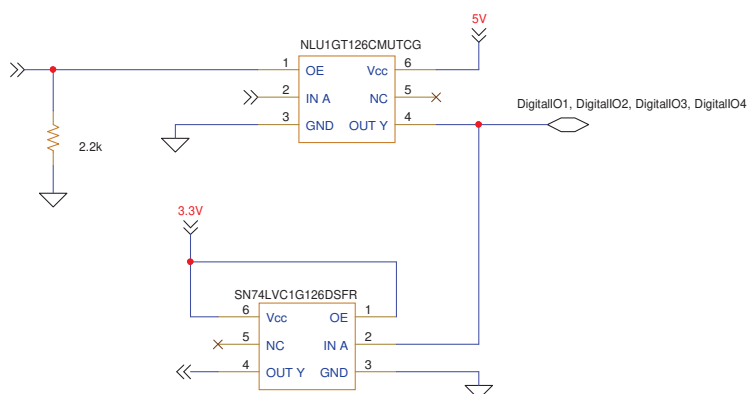
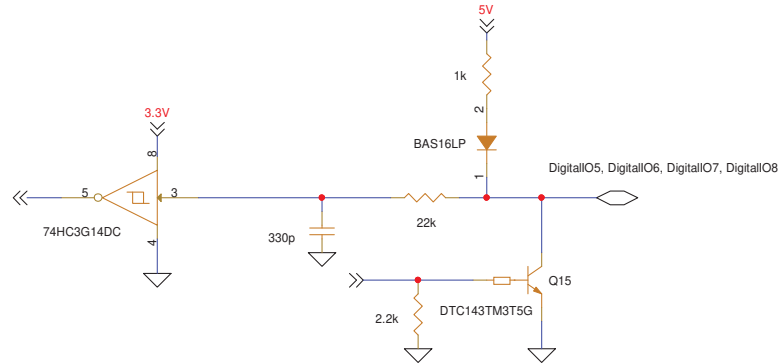


Figure 3-15:
DigitalIO1-4
Internal
Processing
Circuitry

The Digital5-8 signals use an open collector methodology and are shown in the diagram below.

Figure 3-16:
DigitalIO5-8
Internal
Processing
Circuitry



3.5.2 General Purpose Analog Input

The N-Series ION supports a general purpose +/- 10V differential analog input. The A/D sampling resolution is 16 bits, and the range of the returned value is -100,000 to +100,000 where -100,000 = -10V, and +100,000 = +10V.

The N-Series ION can interface with both single-ended and differential analog inputs. When used in single ended configuration AnalogIn- should be connected to ground of the transmitter. When used in differential configuration, AnalogIn+ and AnalogIn- are connected to the differential output of the transmitter. In addition, it is recommended to connect the ground of the transmitter to an N-Series ION ground.

To determine the numerical value that will be read by the N-Series ION given a specific voltage at the input pins, the following formula is used:

$$\text{ReadValue} = \text{AnalogVoltage} * 100,000 / 10.0V$$

Conversely, given a read value, the voltage at the connection is calculated as:

$$\text{AnalogVoltage} = \text{ReadValue} * 10.0V / 100,000$$

3.5.2.1 C-Motion Commands

To read the analog input channels the user must first open a Peripheral of type PIO. This is accomplished using the C-Motion command **PeriphOpenPIO**. Using the returned *Peripheral* handle, the user can read one or more 16-bit word values representing the current analog voltage using the **PeriphRead** command.

For complete information on these commands refer to the *C-Motion PRP II Programming Reference*.

3.5.2.2 Connections & Associated Signals

The table below shows the signals associated with the general purpose analog input of the N-Series ION. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name*	Pin #	Description
AnalogIn +	43	Positive differential input for +/-10V general purpose analog input.
AnalogIn-	44	Negative differential input for +/-10V general purpose analog input.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

For additional information on the analog input signals refer to [Section 2.4, "Electrical Characteristics."](#)

3.5.2.3 Analog Input Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process the general purpose analog input signals.

The following diagram shows this:

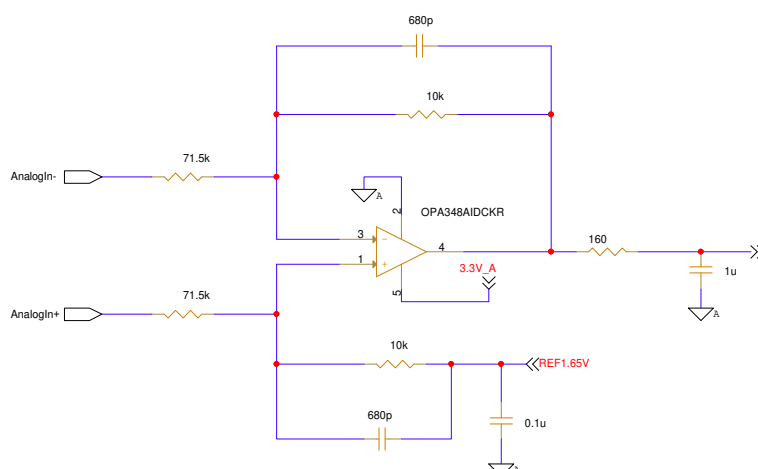


Figure 3-17:
AnalogIn
Internal
Processing
Circuitry

3.6 Communications Functions

N-Series IONs provide four host communication interfaces; Serial, CAN, Ethernet, and SPI (Serial Peripheral Interface). The host interface type of the N-Series ION determines which of these four host interface connections is present in the unit. All units regardless of host interface type also provide an expansion CAN, an expansion SPI, and a serial programming port interface. This is shown in the following table along with the specific port identifier labels for each interface.

N-Series ION Unit				
Type	Host Interface(s)	Expansion CAN	Expansion SPI	Programming Port
Serial:	RS232 - Serial1, Serial2 RS485 - Serial1	ExpCAN	ExpSPI	Serial3
CAN/SPI:	HostCAN, HostSPI	ExpCAN	ExpSPI	Serial3
Ethernet:	Ethernet	ExpCAN	ExpSPI	Serial3

For a complete listing of N-Series ION P/Ns and associated host interface refer to [Section 1.2, “Part Numbers and Configurations.”](#)

In the subsequent sections the operational characteristics of the Serial, CAN, Ethernet, and SPI network ports will be detailed.

3.6.1 Serial Communications

3.6.1.1 Serial1 & Serial2 Interfaces

N-Series IONs that are of serial host interface type provide asynchronous serial communications in both RS232 and RS485 format.

In RS232 mode two serial ports are supported, referred to as Serial1 and Serial2. Serial1 serves as the host serial interface for the ION unit, while Serial2 serves as an extra general purpose RS232 port. While some applications will not need to use two serial ports, the second port may be useful during C-Motion application code debugging, or to communicate with various serial devices connected to the ION unit.

In RS485 mode, a single serial port is supported, referred to as Serial1. RS485 may be operated in either half duplex or full duplex mode. The table below shows which RS485 signals are used in half duplex and full duplex mode.

Pin Name*	RS485 Full Duplex Function	RS485 Half Duplex Function
RS485Xmt +	Positive (non-inverting) transmit output	Positive transmit/receive

Pin Name*	RS485 Full Duplex Function	RS485 Half Duplex Function
RS485Xmt-	Negative (inverting) transmit output	Negative transmit/receive
RS485Rcv +	Positive (non-inverting) receive input	not used
RS485Rcv-	Negative (inverting) receive input	not used

By default the RS485Sel pin, pin #9 of the N-Series ION's Signal Connector, selects whether RS232 or RS485 communications mode is used. If left floating, the ION operates the serial port in RS232 mode. If tied to ground the serial port is operated in RS485 mode. Note that a change in the status of this pin will not properly take effect until after a power on or reset of the ION unit. Whether or not this pin selects the RS232/RS485 mode can be user programmed however. See [Section 3.6.1.2, "RS485 Settings,"](#) for details.

Both Serial Port 1 and Serial Port 2 can be operated at various communication settings as shown in the following table. All settable serial port parameters can be programmed separately for Serial 1 and Serial 2.

Settings & default values for Serial1 and Serial2:

Parameter	Range	Serial1 Default	Serial2 Default
Baud rate	1,200 to 921,600	57,600	115,200
Parity	none, even, odd	none	none
# Data bits	5, 6, 7, 8	8	8
# Stop bits	1, 2	1	1

After a reset or at power-up, the N-Series ION retrieves default information for the Serial ports from the N-Series ION's non-volatile RAM. To change these default values see [Section 3.10.7, "Setting Default Values."](#)

3.6.1.2 RS485 Settings

The table below provides a summary of the available serial network settings if Serial1 is operated in RS485 mode:

Parameter	Default Setting	Setting Options	Comment
Override RS485Sel pin	no	no yes	If this field is set to "no" the value of RS485Sel, pin 9 of the N-Series ION's Signal Connector, will be read to select RS232 or RS485 (high = RS232, low = RS485). If this field is set to "yes" the pin value will be ignored and the RS232/RS485 setting will be read from the setting of the next field.
RS232 or RS485	RS232	RS232 RS485	See above. This setting is only used if 'Override RS485Sel pin' is programmed to "yes".
Node ID	0	0-31	Each ION unit operating in RS485 mode should be programmed with a node address.
Duplex (full/half)	full duplex	half duplex full duplex	If using a four wire connection scheme full duplex should be selected. Two wire schemes should select half duplex.
Termination active	no	yes no	N-Series IONs of serial host type have the ability to provide internal termination, which can be programmed by the user. The first and last device on the bus should have termination active. All other units should have termination inactive.

After a reset or at power-up the ION retrieves default information for the Serial ports from the non-volatile RAM including the RS485 settings listed above. See [Section 3.10.7, "Setting Default Values,"](#) for more information on changing these default values.

3.6.1.3 Daisy-Chain Node ID Assignment

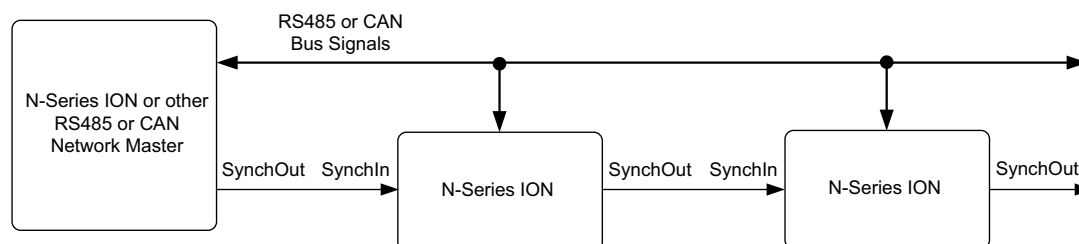


Figure 3-18:
Daisy Chain
Node ID
Assignment
Connection
Scheme

As detailed above each network-attached RS485 device requires a unique Node ID. There are many circumstances where programming these unique Node IDs can be accomplished in a straightforward way as part of the user's application PCB production process. There are other circumstances however, particularly in cases where multiple N-Series IONs are installed on a single PCB, where it is convenient for Node ID assignment to occur without a special production assembly procedure.

The N-Series ION supports this capability via a daisy-chain Node ID assignment scheme utilizing its *SynchIn* and *SynchOut* signals. This scheme can be used for Node ID initialization with both RS485 and CAN networks.

The scheme works via the use of a special command called **SetNodeID**. Normally when the master sends a message to a particular Node ID any attached device with a matching Node ID will receive and process the message and send a response. However when the **SetNodeID** command is received an additional condition is required namely that the *SynchIn* signal be active before it will process the message and send a response.

Here is how this additional network message receive criteria is used to initialize each device on the network with a unique Node ID when the master and attached network devices are wired as shown in the diagram above. Note that the N-Series ION's factory default Node ID settings for RS485 and CAN is 0 (CAN address of 0x580).

- The master N-Series ION sets the pin which carries the *SynchOut* signal low (active) by programming this pin function to *DigitalIO7* (Pin #37 of the Signal Connector carries both the *SynchOut* function and the *DigitalIO7* function), and then setting the *DigitalIO7* output value low. For information on how to program pin functions for pins which carry multiple functions refer to [Section 3.10.2, "Pin Multiplexing Control."](#) For information on setting digital I/O signal values refer to [Section 3.5.1, "General-Purpose Digital I/O."](#)
- The master then sends, to Node ID 0, a **SetNodeID** command specifying a non-zero Node ID value for the first device in the chain. Although all attached slave devices will receive the **SetNodeID** command (because they are all at their default Node ID setting of 0), only one, the first unit in the chain will process the message because only that unit has a *SynchIn* signal which is in an active state. All other units will see an inactive level on their *SynchIn* inputs and therefore ignore the **SetNodeID** message. After the first unit receives this command it will reprogram its Node ID as directed with the value contained in the **SetNodeID** command message and then change its *SynchOut* signal from inactive to active.
- The master then sends, to Node ID 0, a **SetNodeID** command specifying a non-zero Node ID value for the next sequential receiving device. Because the first attached slave no longer has a Node ID of 0 (it has been reprogrammed to a non-zero value as detailed in the previous step) and because all downstream attached slave devices have *SynchIn* signal inputs which are inactive, only one, the second unit in the chain processes the message. As before when this unit processes this command it will program its Node ID as directed and change its *SynchOut* signal output level from inactive to active.
- This sequence continues with the master assigning one Node ID after another until all attached units are programmed with their desired network Node IDs. At that point normal RS485 and CAN network communications may occur. If desired, these programmed Node ID settings can be made permanent by storing in the N-Series ION's NVRAM. This eliminates the need to execute the above-described sequence at each power up. For more information refer to [Section 3.10.7, "Setting Default Values."](#)

For a complete list of N-Series ION initialization methods refer to [Section 4.2, "Production Unit Configuration."](#) For additional information on the *SynchIn* and *SynchOut* signals including electrical interfacing refer to [Section 3.4.4, "SynchIn, SynchOut Signals."](#)

Whether or not the daisy-chain NodeID assignment method should be used for initialization is application specific. It has the advantage of allowing network-attached factory default N-Series IONs to be uniquely addressed and initialized by electronic means only. This may save hardware expense and production setup time, and may simplify field replacement of boards populated with N-Series IONs.

On the other hand this approach requires the use of two signals per attached network device; the *SynchIn* and the *SynchOut* signals. If the other functions provided on those pins are needed in a particular application (for example the *DigitalIO4* and *DigitalIO7* functions) either an external device on the user PCB such as an FPGA must be implemented to alter the signal routing after NodeID initialization, or the daisy-chain assignment method should not be used.

3.6.1.4 Serial3 Interface

All N-Series ION units regardless of host interface type support a serial connection known as the programming port on Serial3.

Serial3 provides UART functionality and has TTL level signals only. It does not directly support RS232 interconnection. Consequently Serial3 is most often used in conjunction with PMD's USB to 3-pin cable, which contains the needed circuitry to communicate via the Serial3 programming port from an external device such as a PC.

The settings and default values for Serial3 are provided in the table below:

Parameter	Range	Serial3 Default
Baud rate	1,200 to 921,600	57,600
Parity	none, even, odd	none
# Data bits	5, 6, 7, 8	8
# Stop bits	1, 2	1

After a reset or at power-up, the N-Series ION retrieves default information for the Serial ports from the N-Series ION's non-volatile RAM. To change these default values see [Section 3.10.7, "Setting Default Values."](#)

The Serial3 port is not generally intended to be used for purposes other than setup and programming of the N-Series ION unit, but there is no exclusion from using it for other purposes. To utilize Serial3 as a general purpose serial port similar to how one might use Serial1 and Serial2 the normal commands indicated in [Section 3.6.2.2, "C-Motion Commands,"](#) are used.

3.6.1.5 Serial3 Access Recovery

Since Serial3 is commonly used as the main programming interface to the ION unit, care should be taken not to alter the Serial3 settings such that communication is lost. The most likely ways for communications to be lost are if one or both Serial3 port pins, pins #43 and #44, are programmed to another function, if the Serial3 communication parameters are changed to unknown or invalid parameters, or if downloaded user code is set to automatically start after reset and takes over Serial3 via the **PeriphOpenCOM** command. Note that for these changes to result in a loss of communication they must be saved in NVRAM. This is because if these parameters are changed in RAM but not stored in NVRAM the unit can simply be reset and the default settings will be restored.

If communication via Serial3 is lost, and if there are no other operating communication paths to the ION, then a special Serial3 access recovery sequence is available that can be used to restore its function.

Here is the sequence that is used to accomplish this: Within 500 mSec of an ION unit reset a host sends one or more 'a' (ASCII 97) characters to the Serial3 port at the default Serial3 settings shown in the table above. If the 'a' character has been received it will be echoed back by the ION. Once the echoed 'a' is received, with a pause of no more than 10 mSec between each sent character, the access recovery sequence "recover" (ASCII 114, 101, 99, 111, 118, 101, 114) is then sent by the host, at the default Serial3 settings.

Upon receipt of the full access recovery sequence the ION unit will set the pin MUX settings to the Serial3 pins and the communicate parameters to the Serial3 default parameters, thereby allowing normal communications via Serial3. Note that these restored settings are stored in RAM only. If the ION's NVRAM settings are not reprogrammed, upon the next reset or power up cycle the pin MUX programming and communication settings will revert to whatever values are stored in the NVRAM. For more information on storing N-Series ION parameters into NVRAM see [Section 3.10.7, "Setting Default Values."](#)

3.6.1.6 C-Motion Commands

Access to either the Serial, CAN, Ethernet, or SPI port is accomplished by sending a C-Motion command with the detailed connection parameters that will be used. To create a serial port peripheral connection the parameters are specified in the C-Motion command **PeriphOpenCOM**. Messages to and from the Serial port are transmitted via the **PeriphSend** and **PeriphReceive** commands. Whenever a new serial port peripheral is opened its previous function is canceled.

To communicate with other Magellan or PRP protocol devices via the peripheral connection established using the above commands a device connection must be established. The command **MPDeviceOpen** is used to connect with Magellan protocol devices such as Magellan ICs, and the command **RPDeviceOpen** is used to connect with PRP protocol devices such as ION/CME or Prodigy/CME devices.

For detailed information on these commands refer to the *C-Motion PRP II Programming Reference*.

3.6.1.7 Connections & Associated Signals

The table below shows the signals associated with the host serial interface. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Srl2Xmt/RS485Xmt-	1	RS232 serial 2 transmit, or RS485 negative differential transmit signal.
Srl1Xmt/ RS485Xmt +	2	RS232 serial 1 transmit, or RS485 positive differential transmit signal.
Srl2Rcv/RS485Rcv-	3	RS232 serial 2 transmit, or RS485 negative differential transmit signal.
Srl1Rcv/ RS485Rcv +	4	RS232 serial 1 transmit, or RS485 positive differential transmit signal.
RS485Sel*	9	This signal selects RS232 or RS485 operation of the host serial interface. If left unconnected or tied high RS232 is selected, If tied low RS485 is selected.
Srl3Xmt*	31	Programming port serial transmit. Note that Srl3Xmt and Srl3Rcv are TTL level signals only and do not support direct RS232 connection.
Srl3Rcv*	32	Programming port serial receive.
GND	3	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

* These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed.

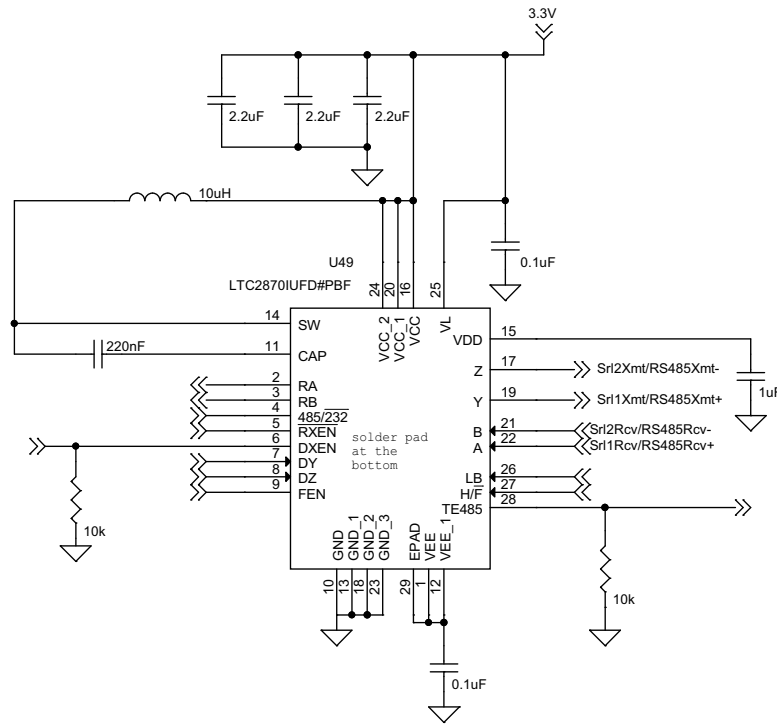
For these signals to function properly one or more of the digital grounds must be connected.

3.6.1.8 Serial Communications Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process I/O signals.

The following diagram shows this for the *Srl2Xmt/RS485Xmt-*, *Srl1Xmt/RS485Xmt +*, *Srl2Rcv/RS485Rcv*, and *Srl1Rcv/RS485Rcv +* signals.

Figure 3-19:
Serial
Communic-
ation Signals
Internal
Processing
Circuitry



3.6.2 CAN Communications

The N-Series IONs provide either one or two general purpose CAN ports, both compatible with the CAN FD standard. CAN/SPI type host interface N-Series IONs support a host interface CAN FD port. All N-Series units support an expansion CAN FD port which is useful for networking IONs together.

The CAN interfaces may be operated at various communication rates from 50,000 to 5,000,000 bps (bits per second). In addition, each CAN device is assigned two CAN identifiers (also called addresses); one for transmission of PRP messages, and one for reception of PRP messages. CAN FD supports two separate baud rates; known as the nominal baud rate and the data baud rate. This information is summarized in the following table:

Parameter	Range	Default
Nominal baud rate	50,000 to 1,000,000 bps	1,000,000 (host CAN), 1,000,000 (expansion CAN)
Data baud rate*	50,000 to 5,000,000 bps	1,000,000 (host CAN), 1,000,000 (expansion CAN)
Host send address	0 - 0x800	0x580
Host receive address	0 - 0x800	0x600

* After a reset or at power-up the ION retrieves default information for the CAN port nominal baud rate/data baud rate from the non-volatile RAM. See [Section 3.10.7, "Setting Default Values,"](#) for more information on changing these default values.

3.6.2.1 Daisy-Chain Node ID Assignment

As detailed above each network attached CAN device requires a unique address. There are many circumstances where programming these unique addresses can be accomplished in a straightforward way as part of the user's application PCB production process. There are other circumstances however, particularly when multiple N-Series IONs are installed on a single PCB, where it is convenient for address assignment to occur without need of a special production assembly procedure.

N-Series ION devices support this capability via a daisy-chain address assignment scheme utilizing the *SynchIn* and *SynchOut* signals. This scheme can be used for Node ID or address initialization with both RS485 and CAN networks, and is described in detail in [Section 3.6.1.3, "Daisy-Chain Node ID Assignment."](#)

3.6.2.2 C-Motion Commands

Access to either the Serial, CAN, Ethernet, or SPI port is accomplished by sending a C-Motion command with the detailed connection parameters that will be used. To create a CAN peripheral connection, the parameters are specified in the C-Motion command **PeriphOpenCAN**. Messages to and from the CAN port are transmitted via the **PeriphSend** and **PeriphReceive** commands.

To communicate with other Magellan or PRP protocol devices via the peripheral connection established using the above commands a device connection must be established. The command **MPDeviceOpen** is used to connect with Magellan protocol devices such as Magellan ICs, and the command **RPDeviceOpen** is used to connect with PRP protocol devices such as ION/CME or Prodigy/CME devices.

For detailed information on these commands refer to the *C-Motion PRP II Programming Reference*.

3.6.2.3 Connections & Associated Signals

The table below shows the signals associated with CAN communications. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
HostCAN +	5	These signals are used with CAN/SPI Host Interface unit types only and provide the positive differential host network CAN signal.
HostCAN-	6	These signals are used with CAN/SPI Host Interface unit types only and provide the negative differential host network CAN signal.
ExpCAN +	7	Positive differential expansion network CAN signal.
ExpCAN-	8	Negative differential expansion network CAN signal.

3.6.3 Ethernet Communications

N-Series IONs that are of Ethernet host interface type provide 100 Base-T full duplex Ethernet communications. Two different Ethernet protocols are supported; TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). TCP is used for primary Ethernet communications to the ION unit, while UDP is typically used for non-critical applications such as data logging, or for the Pro-Motion console window. See [Section 3.9.4, "Debug Console,"](#) for more information on the C-Motion Engine console window.

When used to receive PRP messages the physical node on the Ethernet network controller is assigned a 32-bit IP (Internet Protocol) address, along with a 32-bit netmask and a 32-bit gateway value. The Netmask is used to indicate which IP addresses are local, and the gateway value is used to route non-local addresses. To correctly receive communications from the host controller, a 16-bit identifier known as a port must also be specified. Note that when used as the connection between the host controller and the N-Series ION, TCP rather than UDP communications are used. To determine what the unused IP addresses are for your Ethernet network, and what values for netmask and gateway to use, you should contact your network administrator.

By convention, the 32 bit values for IP Address, Netmask, and Gateway are shown in Dotted Quad Notation. In this notation each of the four numbers are separated by dots, and denote a decimal value for each byte of the four byte word.



The table below shows the range and default settings for the Ethernet controller of the N-Series ION:

Parameter	Range	Default
IPaddress	0.0.0.0 – 255.255.255.255	192.168.2.2
Netmask	0.0.0.0 – 255.255.255.255	255.255.255.
Gateway	0.0.0.0 – 255.255.255.255	0.0.0.0
PRPListenTCPPort	0 - 65,535	40100

Each physical hardware device on an Ethernet network is assigned one IP address, however, a given IP address can have multiple ports. This is useful because it allows user application code running on the C-Motion Engine to open up peripheral connections using port numbers other than the PRP communications port (which has a default value

of 40100), thereby allowing PRP messages and application-specific data in any format to co-exist on the same Ethernet IP node.

After a reset or at power-up, the N-Series ION retrieves default information for its Ethernet port from the N-Series ION's non-volatile RAM. To change these default values see [Section 3.10.7, "Setting Default Values."](#)

3.6.3.1 TCP Connection Keep-Alive

All N-Series ION TCP connections, including the PRP communications port, use a 'keep-alive' mechanism to detect whether a connection is still valid.

The keep-alive mechanism is a standard part of the TCP protocol specification, and is useful for preventing the N-Series ION from leaving connections open if the host has not properly closed a connection. This may occur, for example, if the host has been physically disconnected, or otherwise stops functioning.

The default keep-alive parameters for the N-Series IONs are:

Parameter	Range	Default
idle time	0-65,535 seconds	60 seconds
interval time	not settable	30 seconds
retry count	not settable	2

The *idle time* is the amount of time after the last message on the port that must occur for a 'keep-alive' message to be sent. When sent, the keep-alive message requests the host connection to acknowledge that it is still functioning properly. If it provides this acknowledgment, the *idle time* counter is reset to 0. If it does not, a second 'keep-alive' message will be sent after the *interval time*, and this will be repeated a total of *retry count* number of times. If the host ultimately does not correctly respond, the ION unit connection will automatically be closed.

Note that all of these functions are handled automatically by the N-Series ION's TCP processing system, and should in turn also automatically be handled by the host's TCP system. No user action is required to initiate or monitor these automatic TCP 'keep-alive' messages.

See [Section 3.10.7, "Setting Default Values,"](#) for information on how to change machine controller default values including the keep-alive parameters described above.

3.6.3.2 Magnetism

The N-Series ION's internal circuitry and signaling provide the electrical functions needed for a direct Ethernet connection, however external magnetism are required for providing galvanic isolation. The simplest way to accomplish this is to utilize an RJ-45 connector with integrated magnetism. This option is convenient and compact, and is the approach used with PMD's Ethernet N-Series ION developer kit.

3.6.3.3 C-Motion Commands

Access to either the Serial, CAN, Ethernet, or SPI port is accomplished by sending a C-Motion command with the detailed connection parameters that will be used. For example to create an Ethernet TCP connection, the IP Address and port number is provided. If the connection is successfully established, a *peripheralID* is generated and thereafter used as the reference for any future communications through that connection.

To create an Ethernet/TCP or Ethernet/UDP peripheral conversation the IP address and port are specified in the C-Motion commands **PeriphOpenTCP** or **PeriphOpenUDP**, respectively. To transfer messages via this peripheral connection the **PeriphSend** and **PeriphReceive** commands are used

To communicate with other Magellan or PRP protocol devices via the peripheral connection established using the above commands a device connection must be established. The command **MPDeviceOpen** is used to connect with Magellan protocol devices such as Magellan ICs, and the command **RPDeviceOpen** is used to connect with PRP protocol devices such as ION/CME or Prodigy/CME devices.

For detailed information on these commands refer to the *C-Motion PRP II Programming Reference*.

3.6.3.4 Connections & Associated Signals

The table below shows the signals associated with the Ethernet port functions of the N-Series ION. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
EthernetTx +	1	These pins carry the positive differential Ethernet transmit.
EthernetTx-	2	These pins carry the negative differential Ethernet transmit.
EthernetRx +	3	These pins carry the positive differential Ethernet receive.
EthernetRx-	4	These pins carry the negative differential Ethernet receive.
TapTx	5	This pin carries the transmit center tap signal that connects to the Ethernet isolation transformer.
TapRx	6	This pin carries the receive center tap signal that connects to the Ethernet isolation transformer.
EthSpeedLED*	36	Standard Ethernet Speed LED indicator open collector output. Signal is low, which will drive LED on, if link is 100 MBS. Output is high, turning off LED, if link is 10 MBS.
EthLinkLED*	37	Standard Ethernet Link LED indicator open collector output. Signal toggles high and low blinking the LED if there is port activity. Signal is low, turning on LED if there is a connection but no activity. Signal is high, turning off LED, if there is no connection.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

*These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed.

For differential Ethernet signals and the *TapTx* and *TapRx* signals ground connections are not needed. For the two LED control signals to function properly one or more of the digital grounds must be connected.

Using the Ethernet N-Series ION without external magnetics will result in faulty and unreliable Ethernet communications.



3.6.4 SPI Communications

3.6.4.1 Host SPI Port

N-Series IONs that are of CAN/SPI host interface type provide an SPI (Serial Peripheral Interface) port for host communications. This SPI port operates as an SPI slave, and is most often used as a PRP-formatted connection from another N-Series ION or from an on-board microprocessor. Alternatively the host SPI port can be used in conjunction with user code stored in the N-Series ION's C-Motion Engine to execute an application specific protocol.

The host SPI interface consists of four signals; *HostSPISelect*, *HostSPIXmt*, *HostSPIRcv*, and *HostSPIClock*. When operating as a PRP network an additional signal is used called *HostSPIStatus*.

When the host SPI port is used to carry PRP communications the default PRP settings are automatically applied and no SPI port settings are normally specified. When the host SPI port is used with an application specific protocol the port settings must match the settings of the SPI master for communication to occur correctly. Refer to [Section 3.6.4.3, "SPI Port Settings,"](#) for information on available SPI port operating settings.

For more information on how PRP communications are hosted on an SPI connection refer to the *C-Motion PRP II Programming Reference*.

3.6.4.2 Expansion SPI Port

All N-Series IONs, regardless of host interface type, provide an expansion SPI port. This port functions as an SPI master and supports up to four chip selects allowing communication with up to four separate SPI-connected

devices. Typical uses of the expansion SPI port are connecting to other N-Series IONs and connecting to SPI peripherals such as ADCs (Analog to Digital Converters), DACs (Digital to Analog Converters), flash memory, RAM, accelerometers, and other devices.

The signals comprising the expansion SPI port are *ExpSPIXmt*, *ExpSPIRcv*, *ExpSPIClock*, *ExpSPIStatus*, *ExpSPICS1*, *ExpSPICS2*, *ExpSPICS3*, and *ExpSPICS4*.

PRP and non-PRP expansion SPI connections can be individually programmed, as can their associated SPI port settings. For example expansion SPI CS1 (chip select #1) could be set up for a PRP connection with an N-Series ION with a clock rate of 20 MHz, while CS2 could be set up for an application-specific connection (for example to a RAM chip) at 10 MHz, etc. See the section below for information on settable SPI port parameters.



Although there are four expansion SPI chip select signals in total, the maximum number of PRP connections via the expansion SPI port is three. This is because *ExpSPICS4* is shared with *ExpSPIStatus*, and *ExpSPIStatus* is a required signal for PRP communications. For more information on PRP-based expansion SPI connections see [Section 3.6.4.7, "PRP-Based Expansion SPI Networks."](#)

3.6.4.3 SPI Port Settings

Several user programmable SPI port settings are available to operate the host and expansion SPI ports. These SPI port settings may be useful to accommodate different SPI-based devices or application conditions. The following table provides information on these parameters:

Parameter	Available Settings	PRP Default	Comments
Clock rate	.625 MHz 1.25 MHz 2.5 MHz 5.0 MHz 10 MHz 20 MHz	N/A	The master clock rate should be set based on the max rate supported by the slowest attached slave device and application specific factors such as signal run distances. In general slower clock rates will operate with greater reliability than higher clock rates.
Data size in bits	4 - 16	16	The data size of the SPI master and slave must be equal. Data is transferred most significant bit first.
Clock polarity	idle low idle high	idle low	This parameter defines the signal level of the clock signal when in its idle state.
Clock phase	first edge second edge	second edge	This parameter determines whether SPI data capture will start on the first clock transition or the second clock transition.

3.6.4.4 SPI Data Transfer Transactions

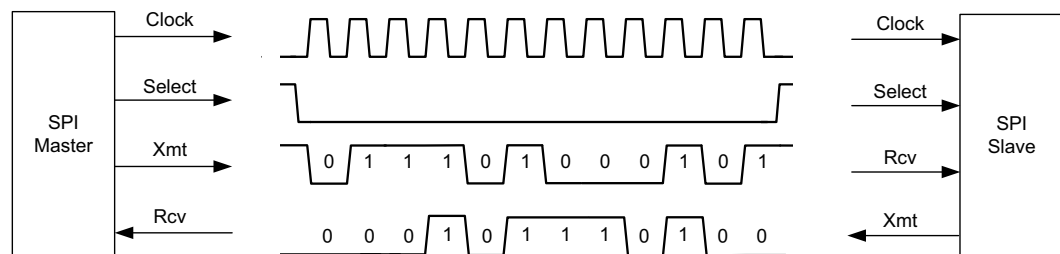


Figure 3-20:
SPI Data
Transaction
Signals

[Figure 3-20](#) shows an overview of a typical SPI data transaction - in this example a 12 bit transaction. To send data to a slave device the master activates the chip select for that device and transfers data synchronously with the clock signal, most significant bit first. Simultaneous to the master transmitting data the slave transmits data back on the receive signal. In this way data is simultaneously sent both to, and received from, the slave.

SPI data transfer transactions that utilize the PRP protocol are handled by the N-Series ION unit automatically. For other types of SPI-based communications, whether via the host SPI or the expansion SPI port, user code must be written and downloaded into the ION unit's C-Motion Engine to manage the SPI transactions.

Similar to serial, CAN, and Ethernet port transactions the peripheral resource is used to manage SPI transactions. To access the host SPI via C-Motion commands a single peripheral connection is created, while for the expansion SPI port one peripheral for each utilized chip select is opened. Therefore the maximum number of expansion SPI peripherals that can be created is four.

Once a C-Motion peripheral is created the peripheral send and peripheral receive commands are used to send and receive data. As mentioned above SPI transactions can transmit and receive data transfer simultaneously. From C-Motion this is handled by separate calls to the send and receive commands. For C-motion code controlling an SPI master ION a peripheral send call is made followed by a peripheral receive call. The data to be sent to the slave is specified by the send command, and the data sent by the slave is accessed via the receive command.

For C-motion code controlling an SPI slave ION the sequence is the same. First a peripheral send command is made specifying the data that will be sent to the master, and then a peripheral receive call is made to access the data sent by the master.

The above description applies when the master and slave want to specify and access both the transmit and receive data transfers. Depending on the application however that may not always be the case. If the master does not care what data is being sent back by the slave it is not obliged to execute a peripheral receive. Similarly if the slave does not care what data has been sent by the master, after executing the peripheral send it is not obliged to execute a peripheral receive. And if the slave does not care what data it sends to the master it is not obliged to execute a peripheral send. In that case the data sent by the slave to the master will consist of all zeroes.

Note that in cases where data received on the SPI bus signals is not acted on by C-Motion commands data in the receive buffer will be overwritten by subsequent SPI transactions. For transmit buffers that fill up further send commands will be suspended until the previous SPI transaction is completed. The easiest way to arrange this is to call the peripheral send function such that the task suspends until the send completes. For more information refer to the *C-Motion PRP II Programming Reference*.

3.6.4.5 SPI Data Word Format

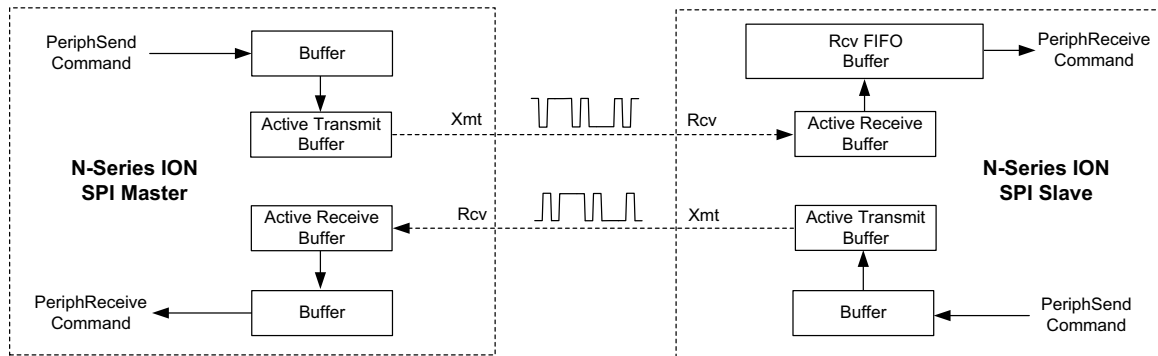
The number of data words being sent or received must be specified by the C-Motion code. For multi-word transfers the bits are transmitted in a continuous stream with no de-assertion of the chip select signal or SPI clock. For example if the SPI data word size is 12 bits, and a send command of three-words is specified a continuous serial stream of 36 bits will appear on the SPI port. After this 36 bit stream is fully sent the SPI control signals will be de-asserted.

Similarly for a receive command, when more than one word of data is requested the total number of uninterrupted (with no intervening de-assertion of the control signals) bits expected to be received is the SPI data word size times the number of words. The maximum length of a sent or received SPI serial data stream is 2,048 bits (256 bytes).

If the programmed SPI data word size is 8 bits or less than data is stored in an array of bytes, and if the SPI data word size is 9 through 16 bits in size the data is stored in an array of 16-bit words. Valid bits are kept in the low order bits of the byte or word. For example if the SPI data word size is 5 bits the peripheral send command expects an array of 8 bit bytes where each byte carries the data value stored in the low 5 bits, and if the SPI data word size is 11 bits the peripheral send command expects an array of 16-bit words with the data value in the low 11 bits.

Figure 3-21:
SPI Master/
Slave Data
Buffering

3.6.4.6 SPI Data Buffering



To simplify synchronization and management of SPI transactions separate send and receive message buffers are implemented. How these buffers and associated peripheral commands function when the N-Series ION operates as a master versus a slave is somewhat different.

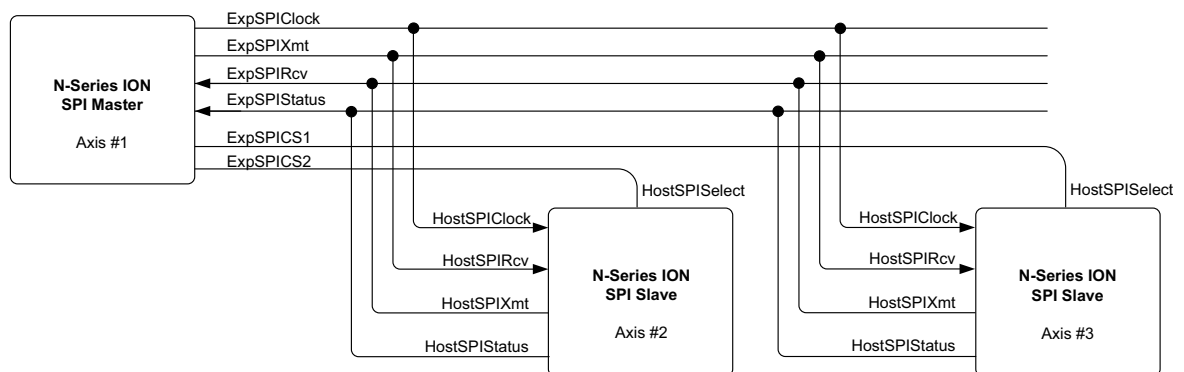
When functioning as a master the **PeriphSend** command is used to transfer a message to a buffer up to 256 bytes in length. A subsequent call to **PeriphSend** will execute immediately if the original message has completed transmission on the SPI bus. If not, the call will suspend until transmission is complete, after which the new message begins transmission and the call to **PeriphSend** returns function. Calls to the **PeriphReceive** function will suspend until the message being sent by the master completes transmission.

When functioning as a slave the **PeriphReceive** command is again used, however control is returned not when the master completes transmission, but when the requested amount of data has been received. In addition the slave receive buffer functions as a FIFO. It can store multiple previous SPI bus transmissions sent by the master up to its size of 256 bytes. If desired, the command **FlushBuffer** can be used to empty this slave receive FIFO buffer.

To send messages from the slave the **PeriphSend** command is used, however unlike when functioning as a master, control after execution of this command immediately suspends. Control is returned when the specified amount of data has been sent by the slave. Control is not affected by whether the master has completed transmission of its entire message. For example if the slave has specified a send of 24 bits and the master begins sending what will be a 32 bit message, control from the **PeriphSend** command at the slave will return after 24 bits have been transmitted.

3.6.4.7 PRP-Based Expansion SPI Networks

Figure 3-22:
PRP-based
Connection
Scheme Using
Expansion SPI
Port



[Figure 3-22](#) shows a multi-ION PRP connection with the ION unit's expansion SPI port operating as an SPI master to command two SPI host interface ION units operating as slaves.

Separate select signals connect the master ION and each slave ION. The SPI clock, data receive, data transmit, and SPI status signals are tied together, forming a bus. Signals output by the slave units (host SPI transmit and SPI status) are asserted only by the slave SPI device selected via the select signal. Non-selected IONs leave these signals in a floating condition thereby avoiding signal contention.

3.6.4.8 C-Motion Commands

To initialize an SPI peripheral the command **PeriphSPIOpen** is used. This command contains arguments for specifying the SPI port settings listed in [Section 3.6.4.3, "SPI Port Settings."](#) To transmit a peripheral message the command **PeriphSend** is used.

To receive a peripheral message the command **PeriphReceive** is used. Similar to other peripheral port types, with SPI receive operations a timeout value can be specified, allowing program control to return if message reception does not complete within a specified amount of time. A specified timeout of 0 means that if the full expected message is not immediately available the function call returns and an error is returned.

To communicate with other Magellan or PRP protocol devices via the peripheral connection established using the above commands a device connection must be established. The command **MPDeviceOpen** is used to connect with Magellan protocol devices such as Magellan ICs, and the command **RPDeviceOpen** is used to connect with PRP protocol devices such as ION/CME or Prodigy/CME devices.

For additional information on these commands refer to the *C-Motion PRP II Programming Reference*.

3.6.4.9 Connections & Associated Signals

The table below shows the signals associated with the host and expansion SPI ports. All of these signals are located on the NSeries ION's Signal Connector.

Pin Name	Pin #	Description
HostSPIXmt	1	Host SPI port data transmit output
HostSPIClock	2	Host SPI port clock input
HostSPIRcv	3	Host SPI port data receive input
HostSPISelect	4	Host SPI port select input
HostSPIStatus*	9	Host SPI port status output
ExpSPIXmt*	31	Expansion SPI port data transmit output
ExpSPIRcv*	32	Expansion SPI port data receive input
ExpSPIClock*	33	Expansion SPI port clock output
ExpSPICS1*	34	Expansion SPI port select 1 output
ExpSPICS2*	35	Expansion SPI port select 2 output
ExpSPICS3*	36	Expansion SPI port select 3 output
ExpSPICS4**	37	Expansion SPI port select 4 output
ExpSPIStatus**	37	Expansion SPI port data receive input

* These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed.

** ExpSPICS4 and ExpSPIStatus share the same N-Series ION pin. When PRP communications are used on one or more SPI expansion ports the ExpSPICS4 pin function can not be selected because the ExpSPIStatus function is required for PRP communications.

SPI (Serial Peripheral Interface) is intended for communication between devices on the same PCB (Printed Circuit Board) only. SPI should not be used to connect devices on different PCBs. It is up to the user to determine whether timing and signal integrity specifications have been met such that SPI communications will be reliable.



3.7 Power Stage

The N-Series IONs contain a high-efficiency fully digital MOSFET power stage with PWM (Pulse Width Modulation) control and leg current feedback. A slightly different configuration is used for each motor type.

- DC brush motors are driven with an H-Bridge consisting of 4 MOSFET switches.
- Brushless DC motors are driven with a 3-phase bridge consisting of 6 MOSFET switches.

- Step motors are driven with two H-Bridges, one for each phase, for a total of 8 MOSFET switches.

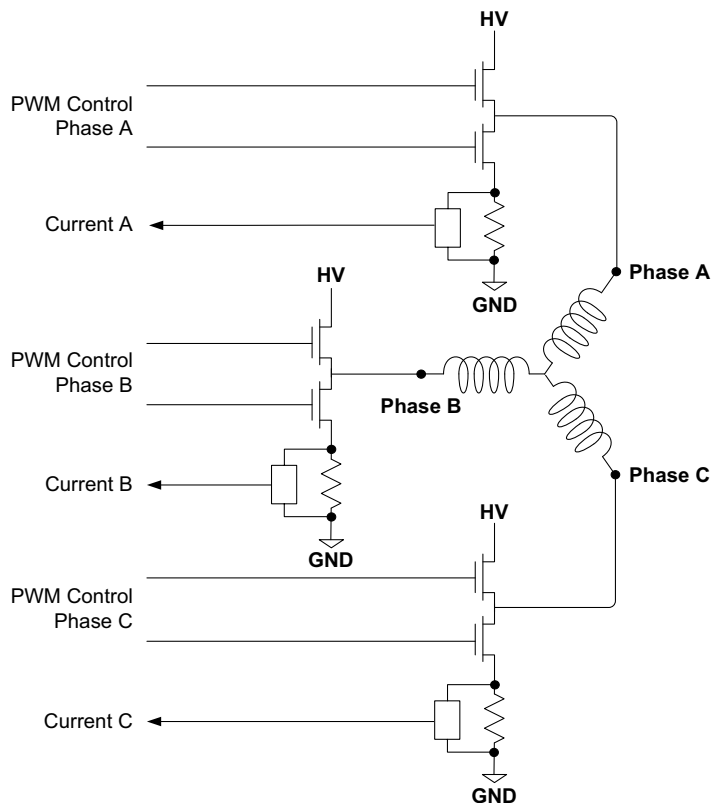
The use of 3-phase and H-Bridge topologies provides full 4-quadrant operation.

N-Series IONs use an advanced PWM switching scheme that minimizes the ripple current on the motor windings while maximizing the current loop performance. The PWM frequency is selectable between 20 kHz and 120 kHz to cover a broad range of motor inductance. The fundamental frequency of the ripple current is at twice the PWM frequency and well out of the audible range in all cases.

3.7.1 Brushless DC Motor Configuration

[Figure 3-23](#) shows the bridge and current sense arrangement when the N-Series ION is used with a Brushless DC motor.

Figure 3-23:
Brushless DC
Motor Bridge
Configuration



3.7.2 DC Brush Motor Configuration

[Figure 3-24](#) shows the bridge and current sense arrangement when the N-Series ION is used with a DC Brush motor.

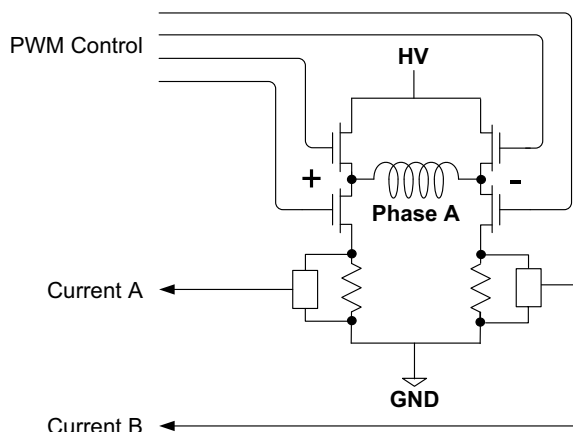


Figure 3-24:
DC Brush
Motor Bridge
Configuration

3.7.3 Step Motor Drive Configuration

[Figure 3-25](#) shows the bridge and current sense arrangement when the N-Series ION is used with a step motor.

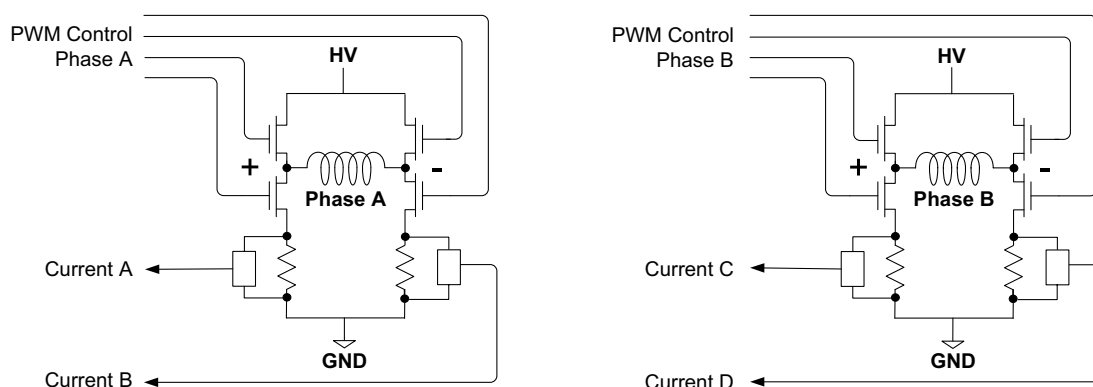


Figure 3-25:
Two-phase
Step Motor
Bridge
Configuration

3.7.4 PWM (Pulse Width Modulation) Frequency

N-Series IONs provide four selectable PWM (Pulse Width Modulation) rates; 20 kHz (which is the default), 40 kHz, 80 kHz, and 120 kHz. Higher inductance motors should use 20 kHz. Lower inductance motors may use higher PWM frequency values to maximize current control accuracy and minimize heat generation.

3.7.5 Current Control

In the N-Series ION's normal operating mode current control is used to determine the amplifier bridge's voltage command output. Current control is a technique used for DC Brush, Brushless DC, and step motors for controlling the current (and therefore the torque) through each winding of the motor. By controlling the current response times improve, motor efficiency is higher, and motion smoothness increases. For single-phase motors such as DC Brush one current loop is used, while for Brushless DC motors and step motors, which are multi-phase devices, two current loops are used.

There are three overall methods of current control provided by N-Series IONs, however not all methods are used with all motor types. The table below summarizes which current control modes are available with each motor type, along with the default configuration for that motor type.

Current Control Method	Brushless DC	DC Brush	Step Motor
Individual Phase Control		Y (default)	Y
Field Oriented Control (FOC)	Y (default)		Y (default)
Third Leg Floating	Y		

The large majority of applications will use Field Oriented Control (FOC) to drive Brushless DC or step motors. FOC usually provides the highest top speeds and more energy efficient operation of the motor compared to individual phase control. Third leg floating is an option that should generally only be considered for Hall-commutated motors. In that configuration, third leg floating can sometimes provide a higher top speed than FOC. Finally, individual phase control is always used with DC Brush motors, and may, under certain specialized conditions, provide improved performance for step motors over FOC.

It is possible to operate the amplifier bridge without current control, an operating mode referred to as voltage mode. While not recommended for general operation of the motor, this mode can be useful for special operating conditions or for calibration or testing.

The *Magellan Motion Control IC User Guide* provides much more information on the available current control modes including control flow diagrams and how to set up and read various current control related parameters.

3.7.6 C-Motion Commands

There are several settable parameters associated with the drive stage function including selection of the current loop method, current loop gain settings, and the PWM frequency. All of these parameters are managed by the Magellan Motion Control IC and are therefore accessed via Magellan IC C-Motion commands. For detailed information on available commands refer to the *Magellan Motion Control IC Programming Reference*.

3.7.7 Connections & Associated Signals

The table below shows the signals associated with the N-Series ION drive stage. All of these signals are located on the N-Series ION's Power Connector.

Pin Name	Pin #	Description
Motor A	4	Motor A motor coil connection
Motor B	5	Motor B motor coil connection
Motor C	6	Motor C motor coil connection
Motor D	7	Motor D motor coil connection
GND	3	Power Connector Ground, used for case shielding connections

Different motor types use different Motor A-D connections. The following table provides detailed information on how to connect different motor types to the N-Series ION drive.

Motor Type	Pin Name	Motor Coil Connections
Brushless DC	Motor A	A winding connection
	Motor B	B winding connection
	Motor C	C winding connection
	GND	(optional) motor shield connection
DC Brush	Motor A	+ winding connection
	Motor B	- winding connection
	GND	(optional) motor shield connection
Step motor	Motor A	phase A + winding connection
	Motor B	phase A- winding connection
	Motor C	phase B + winding connection
	Motor D	phase B- winding connection
	GND	(optional) motor shield connection

3.8 Drive Safety Functions

N-Series IONs provide a number of features that detect safety-related conditions or provide safety-related functions. The subsequent sections describe these features.

3.8.1 Overcurrent Fault

N-Series IONs monitor both the positive and negative DC bus current to detect overcurrent conditions including: line-to-line, line-to-power supply, and line-to-case-ground short circuits. Both hard short circuits and excessive current conditions are detected. N-Series IONs can even detect some “ground fault” conditions caused by a partial winding short circuit between winding and case within a motor.

An overcurrent fault will cause the current loop and power stage modules to be disabled, thereby halting further motor output. To recover from this condition the user should determine the nature of the fault. It is generally desirable to power down the ION unit to check connections or otherwise correct the ION-attached hardware so that the problem does not occur again.

If the overcurrent condition has been resolved, when restart is attempted ION will resume normal operations. If the overcurrent condition has not been resolved, the overcurrent condition will immediately occur again.

For more information on overcurrent detection and recovery, as well as information on the other drive safety functions discussed in this and subsequent sections of this chapter refer to the *Magellan Motion Control IC User Guide*.

Over current faults are serious conditions and warrant the utmost caution before re-enabling the ION drive. It is the responsibility of the user to determine the cause and corrective action of an overcurrent condition.



3.8.2 I^2t Current Foldback

Current foldback, also known as I^2t foldback, is an additional tool to protect the drive output stage or the motor from excessive current.

I^2t current foldback works by integrating, over time, the difference of the square of the actual motor current and the square of a user-programmed continuous current limit. When the integrated value reaches a user-programmed energy limit, the ION unit goes into current foldback. When in this condition, and correctly programmed, the ION unit will attempt to clamp the maximum current to the continuous current limit value. Note that the N-Series ION's ability to do so depends on a properly functioning current loop.

For more information on the functions of the I^2t current foldback refer to the *Magellan Motion Control IC User Guide*.

3.8.3 Overtemperature Fault

N-Series IONs provide the capability to continually monitor and detect excessive internal temperature conditions. Such a condition may occur if excessive current is requested, if heat sinking of the ION unit is inadequate, or if some other problem results in elevated drive temperatures.

To detect this condition a programmable temperature threshold is continuously compared to an internal temperature sensor. If the value read from the internal sensor exceeds the programmed threshold, an overtemperature fault occurs. In addition, a settable overtemperature hysteresis allows the user to ensure that the ION temperature drops by a specified number of degrees before allowing drive restart.

The maximum allowed setting for the temperature threshold is 75.0° C, which is also the default value. The maximum allowed value of the hysteresis parameter is 50° C, and the default value is 5° C.



An overtemperature fault will cause the current loop and power stage modules to be disabled, thereby halting further motor output. To recover from this condition the user should determine the nature of the fault. It is generally desirable to power down ION to correct the condition.

If the overtemperature condition has been resolved, when restart is attempted the ION drive will resume normal operations. If the overtemperature condition has not been resolved, the condition will immediately occur again.

Overtemperature faults indicate that the internal safe limit of the drive temperature range has been exceeded. This potentially serious condition can result from incorrect motor connections, excessive power demands placed on the N-Series ION, or inadequate heat sinking. It is the responsibility of the user to operate IONs within safe limits.

3.8.4 DC Bus Overvoltage Fault

N-Series IONs continually monitor and detect excessive voltage on the incoming DC bus voltage supply. Such a condition may occur if there is a fault in the system power supply, if a large back EMF (electromotive force) is generated during motor deceleration, or if some other problem results in an elevated bus voltage.

To detect this condition a programmable bus voltage threshold is continuously compared to the bus voltage sensor. If the value read from the internal sensor exceeds the programmed threshold, an overvoltage fault occurs.

The maximum allowed setting for the overvoltage threshold is 60.0 volts, which is also the default value. The minimum allowed threshold is 10.0 volts.

An overvoltage fault will cause the current loop and power stage modules to be disabled, thereby halting motor output. To recover from this condition the user should determine the nature of the fault. In most cases it is desirable to power down the ION unit to correct the condition.

If the overvoltage condition has been resolved, when restart is attempted the ION unit will resume normal operations. If the overvoltage condition has not been resolved, the condition will immediately occur again.



Overvoltage faults indicate that a serious safety condition has occurred. It is the responsibility of the user to operate N-Series IONs within safe limits.

3.8.5 DC Bus Undervoltage Fault

N-Series IONs also provide the capability to sense an undervoltage condition. The undervoltage value is compared to the value read from the drive DC bus, and if the value read is less than the programmed threshold, an undervoltage fault occurs. The minimum allowed value for this threshold is 10.0 volts, which is also the default value. The maximum allowed value is 56.0 volts.

All other aspects of this feature are the same as for overvoltage sense. Just as for overvoltage conditions, it is the user's responsibility to determine the seriousness of, and appropriate response to, an undervoltage condition.

3.8.6 Motor Brake

The N-Series ION *Brake* signal input provides a high speed amplifier function that may be useful for safety protection when using Brushless DC, DC Brush motors, or step motors.

When this input is active amplifier output is driven to one of two user programmable states; a braking state or a fully disabled state. This signal is always active low and its sense can not be changed. When a brake command is asserted with the braking function programmed the ION's drive controls the switches in such a way that current will circulate within the coils and generate resistive back-EMF torque resulting in a deceleration of the motor.

When a brake command is asserted with the fully disabled mode programmed no back-EMF braking is applied and the motor will “free wheel” and decelerate more slowly to a stop.

Decelerating a motor in a braking mode (back EMF engaged), particularly one starting from a high speed and having substantial inertia may result in excessive heat generation in the motor. Whether or not this occurs is application dependent. It is up to the user to determine whether the braking function should be used in their application.

During braking the motor's velocity is not actively controlled by the drive. The actual motor motion response after a brake signal is applied is application dependent. Particularly in applications where external forces exist on the motor the motor may not decelerate as described above or it may even accelerate. It is the responsibility of the user to determine whether, and how, the *Brake* signal should be used in their application.



3.8.7 Shunt Resistor Control

N-Series IONs provide the ability to drive a shunt resistor via their Motor D output. Shunt control is a technique for managing the DC Bus voltage, particularly in situations where the motor axis is decelerating and acting as a generator, thereby driving the DC bus voltage higher.

Shunt control is an option for use with DC brush and Brushless DC motors. It is not used with step motors. Whether or not it should be used depends on a number of factors including the nature of the DC power supply, the motor inertia, and the worst case expected motor deceleration rate.

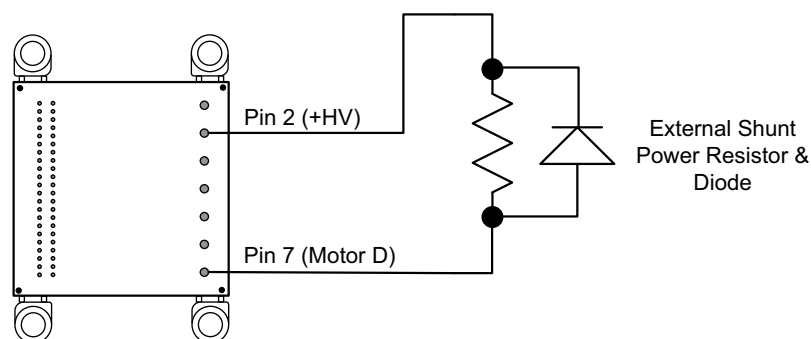


Figure 3-26:
Shunt Resistor
Wiring Diagram

As shown in [Figure 3-26](#) the N-Series ION provides a shunt PWM output at the Motor D output pin on the unit's Power Connector (Pin #4).

The shunt functions by continually comparing the DC bus voltage to a user programmable threshold. If the DC bus voltage exceeds the comparison threshold the Motor D pin signal outputs a PWM waveform at a user programmable duty cycle, variously connecting the motor D/shunt pin to GND in its active state, and HV in its inactive state. The PWM frequency used with shunt is equal to the motor drive PWM frequency. Once active, shunt PWM output will stop when the DC bus drops to 2.5% below the threshold comparison value.

The shunt resistor connected should have a resistance such that the current flow through the shunt switch, diode and resistor do not exceed the ratings of those components at the expected DC bus voltage. The diode, which is connected in parallel to the resistor, should have a voltage and current rating at least equal to the anticipated current flow.

3.8.8 C-Motion Commands

There are several settable parameters associated with the drive safety functions including I^2t continuous current limit and energy limit, the overtemperature threshold and hysteresis settings, and over and undervoltage thresholds. In addition the brake function is user programmed for a braking or a full amplifier disable stop, and the shunt function is programmed with a threshold value and duty cycle setting.

All of these parameters are managed by the Magellan Motion Control IC and are therefore accessed via standard C-Motion commands. For detailed information on these commands refer to the *Magellan Motion Control IC Programming Reference*.

3.8.9 Connections & Associated Signals

There are no connections or signals associated with the drive safety functions for overcurrent, overtemperature, or over/under voltage. All of the signaling, sensors, and detection circuitry is located internal to the drive.

The table below shows the signals associated with the shunt resistor function. All of these signals are located on the N-Series ION's Power Connector.

Pin Name*	Pin #	Description
Motor D/Shunt	7	Shunt resistor drive connection
HV	2	HV supply

The table below shows the signals associated with the motor brake function. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Brake	28	Motor brake signal input. When asserted low activates the motor brake.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

For this signal to function properly one of more of the digital grounds must be connected.

3.8.10 Brake Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process this signal. *Brake* shares the same type of processing circuitry as the *Enable* signal. Therefore refer to [Section 3.4.1.3, "Enable Signal Processing,"](#) for information on this processing circuitry.

3.9 C-Motion Engine

The C-Motion Engine (CME) allows user-written code to be downloaded and executed on the N-Series ION. The C-Motion Engine is a powerful and flexible engine that can be used to:

- Operate N-Series IONs in a standalone mode
- Offload time-critical code from the host to the ION unit
- Communicate via serial, CANbus, Ethernet or SPI with a cell controller or other host controller
- Communicate via the expansion SPI and expansion CAN networks to other N-Series IONs or control system peripherals
- Lower system cost by combining functions that would otherwise reside in a separate microprocessor.

Creating, compiling, downloading, and verifying a user C-Motion application on a N-Series ION is accomplished with the C-Motion Engine development system. The outcome of such a development sequence is a downloadable .bin code image that contains the user application code to be executed by the C-Motion Engine on the N-Series ION.

For detailed information refer to the *C-Motion Engine Development Tools Manual*. For information on creating a user code development setup with N-Series ION developer kits, refer to the *ION/CME N-Series Digital Drive Developer Kit User Manual*.

3.9.1 CME Function Overview

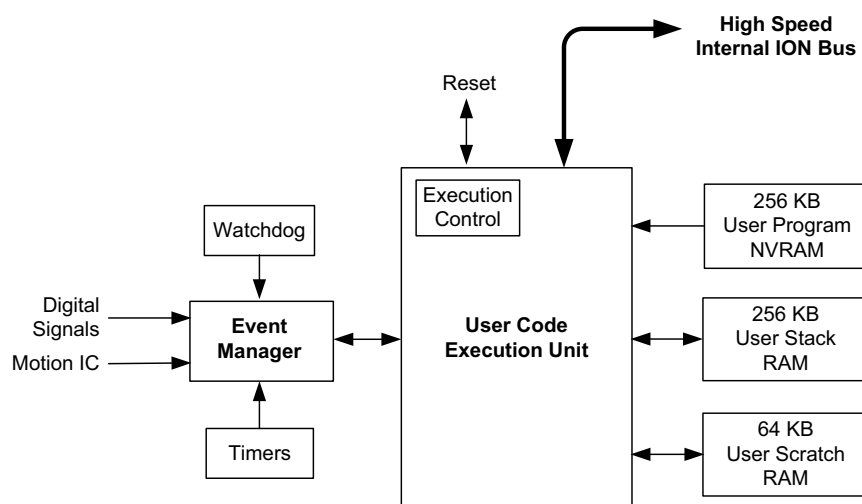


Figure 3-27:
Block Diagram
of C-Motion
Engine

The C-Motion Engine (CME) is a self-standing unit within the N-Series ION that provides user code execution and includes various specialized real-time functions such as event management, timers, and execution control. The CME is connected to other internal N-Series ION resources via a high speed bus, allowing it to access the ION's resources including the Magellan IC, the host communication ports, the expansion network ports, the LEDs, the device default NVRAM, the real time clock, and other internal N-Series ION resources.

Downloaded user code executes on the CME at a rate as high as 428 MIPS, although within the code execution unit there is also an internal real-time operating system (RTOS) that utilizes some of these code execution resources to provide functions such as multi-tasking, memory management, and other functions.

In the following sections we will provide summary information on functions such as CME startup & execution control, multi-tasking, event management, and others. For complete information on CME function refer to the *C-Motion PRP II Programming Reference* and the *C-Motion Engine Development Tools Manual*.

3.9.2 CME Powerup & Execution Control

Upon reset or powerup the C-Motion Engine initializes itself and checks to see whether execution of user application code, if downloaded, should automatically begin. If the factory default setting has not been changed user application code holds in a wait state and does not auto-execute. If this default setting is changed user code begins executing automatically and continues until the ION unit is powered down, reset, or the code aborts.

If holding in a wait state, to start code execution from Pro-Motion the C-Motion Engine dialog box is used. Whether by manual control or from auto-start, once execution has started the user code typically executes a sequence of instructions that serve the purpose of initializing the ION unit and attached peripherals for the axis or machine being controlled. Such an initialization process might include communicating with these external devices, checking that the ION unit's Magellan has completed its initialization, and other safety and initialization related checks prior to normal machine operation.

For additional information on powerup and initialization of the N-Series ION unit refer to [Section 3.12, "N-Series ION Powerup & Initialization."](#)

3.9.3 Run Time Environment & Resources

Application code developed for the C-Motion Engine is C-based, and thus there are limits to code size, RAM usage, and stack usage that should be observed during runtime operation of downloaded C-Motion our code. The following table provides some of these values along with various other CME parameters:

Resource	Specification
User program space (stored in flash)	256 KB
User scratch RAM space	64 KB
Total user stack space	256 KB
Maximum # of user tasks	16
# of available mailboxes	10
# of available digital signal events	2
# of available timers	2
MIPS (millions of instructions per second)	428

For user code that does not correctly observe these limits, or for user code files that have become corrupted, there are a number of fault conditions that can occur. These very serious run-time faults include instruction errors - indicating that an unknown instruction was encountered during execution of the user code, and address faults - indicating that either a program space or RAM space access limit was violated.

When an exception occurs the C-Motion Engine will immediately reset the N-Series ION. This C-Motion Engine-initiated reset is identical to the reset that occurs after sending a reset C-Motion command, except that the cause of the reset is also recorded as 'C-Motion Engine user code fault. See [Section 3.10.3, "Reset,"](#) also more information on the Reset command. See [Section 3.10.4, "Reset Monitor,"](#) for information on retrieving the reset cause.

3.9.3.1 Tasks

The C-Motion Engine supports a main task as well as the ability to spawn new tasks. Including the main task a total of 16 tasks are allowed, and these tasks share a total stack memory of 256 KB.

The **USER_CODE_TASK** C-Motion macro is used to create the main task and has a single argument which is a string that may be used to specify the task name. If auto-start is not enabled a parameter may be passed to the main task via a PRP communications link using the **TaskStart** command. If auto-start is enabled the passed parameter comes from the NVRAM defaults memory area. For more information on the NVRAM defaults area see [Section 3.10.7, "Setting Default Values."](#)

The command **TaskCreate** is used to create and begin execution of additional tasks. The arguments for this command include a function pointer to the task code, an ASCII string used to specify a task name, the stack size to be used for the task, a passed parameter, and the priority level of the task. A typical allocated stack size is 8 KB, but this is dependent on the function of the task itself. The command **TaskGetTaskInfo** may be used during development to determine, among other things, how much of its stack a particular task is using.

There are three available priority levels for tasks, low, normal, and high. The main task always has a priority of normal. High priority tasks should be created and used with caution because if such a task does not give up its round-robin scheduled time it may effectively disable lower priority tasks from receiving processor time. See [Section 3.9.3.3, "Task Switching,"](#) for guidelines on processing sharing.

Both the main task and any other created tasks are identified by a task number. The task number of the main task is 0 and subsequently created tasks have a task number of 1, 2, 3, etc... up to 15. The task number may be useful to check the status of a task using the **TaskGetTaskInfo** command.

The **TaskAbort** command is used by a task to stop further execution of itself. Note that neither the memory resources associated with an aborted task, nor the task number for that task, are freed up after sending a

TaskAbort command. Therefore aborting a task does not impact the ability to create more tasks. Once 16 tasks have been created, or once all of the available stack space has been allocated, no additional tasks can be created.

It is up to the user to determine that memory space is not overwritten during application execution, and that task priority settings in combination with task code execution does not monopolize processor access to other tasks in the system.



3.9.3.2 Inter-task Communication & Coordination

Some CME user applications may need to share data between tasks. To support this the CME supports a mailbox mechanism. Mailboxes are created using the **MailBoxCreate** command which has arguments to define the # of data items in the mailbox and the size of each data item. Up to 10 mailboxes can be created. Memory allocation for created mailboxes is taken from the total stack space pool.

The mailbox is a FIFO, so that the first data placed into the mailbox is also the first extracted. The command **MailBoxSend** places data into the mailbox while the command **MailBoxReceive** extracts sent data. It is also possible to look at the content of a mailbox without withdrawing the data via the **MailBoxPeek** command. A semaphore can be implemented using a mailbox with a depth of 1 and item size of 0.

In addition to mailboxes the CME supports a Mutex mechanism. Mutexes are useful for arbitrating access to resources that are shared by multiple tasks. The associated commands are **MutexCreate**, **MutexLock**, and **MutexUnlock**.

An example of mutex use is a mutex to protect communication to a device. Without access arbitration, multiple tasks could send simultaneous commands to the device, potentially corrupting communications. To address this a mutex dedicated to the device access is created. For each task that wants to communicate with the device the mutex is locked before a packet is sent and then unlocked after the response has been received.

3.9.3.3 Task Switching

Tasks of equal priorities share the available processing time using a time sliced round-robin scheduling scheme. Tasks of higher priority are always given processor time in preference to tasks of a lower priority. Therefore the primary purpose of assigning a user task a high priority is to ensure it responds to events in the minimum amount of time.

For example, in a particular application it is necessary to perform some action as rapidly as possible after a digital input changes state. To affect this sequence an **EventWait** command is used (more on this command in a moment). When the digital input signal changes state the higher priority task would immediately wake up and return from the **EventWait** call. Any currently running lower priority task would be pre-empted.

In this same example, if there was a task of equal priority currently running, then the **EventWait** call would not return until the time slice of the running task was over, or until that task itself entered a wait state thereby giving back its processing time to the task manager.

Although the longest amount of time a single task can execute is one time slice interval, it is seldom the case that a particular task will take up this entire allotted time slice. This is because most tasks, after executing a set of code instructions will then wait for some external event, or a period of time, before continuing execution.

Commands such as **EventWait** provide this function, indicating to the CME operating system that further task execution (and therefore utilization of its time slice) may be suspended until a particular condition occurs. Other commands with similar characteristics are **TaskWait** and **TaskWaitUntil**. These two commands allow the task to be suspended for a programmed amount of time, or to wait until a certain value of the system clock timer is reached. Note that calling **TaskWait** with a wait time of 0 suspends execution for that time slice and forces a task switch.

3.9.3.4 Events

One of the most useful mechanisms provided by the C-Motion Engine for coordination and synchronization of machine behavior is events. Events allow external real-world conditions to automatically trigger code execution. There are three different types of events; timer events; Magellan motion control IC events, and digital signal events. Here is more information on each.

Timer Events

Timer events use special hardware within the CME to count down a precise period of time. Timer events are instantiated using the command **EventOpenTimer**. The parameters associated with this command are the timer event number, the timer type, and the timer period. Up to four timer events can be operated simultaneously numbered 1 through 4. There are two timer types, one-time or continuous and as the names imply a one-time timer only 'fires' once, and a continuous timer fires indefinitely until disabled. The timer period value represents the time, in microseconds, to count down. When the count reaches a value of zero the timer event occurs (fires).

As is the case for all event types the command **EventWait** is used to suspend code execution until the programmed event occurs. When the event occurs, control returns from this function call and normal code execution resumes. If the timer counts down to zero before the **EventWait** function call is made execution control will immediately return from the **EventWait** call.

To disable a timer event, or in fact any already-defined event, the command **EventClose** is used.

Motion IC Events

Motion IC events are a type of event triggered by the *HostInterrupt* signal from the Magellan Motion Control IC inside the N-Series ION. An **EventOpenMotion** command is used to set up a motion event. Motion events are useful for coordinating machine activity so that, for example, a new motor move is started immediately after a previous move has completed, or to be automatically informed of problems with the motion function.

The conditions that control the *HostInterrupt* signal are determined by status bits within the Magellan IC's Event Status Register. For more information on this register and how to program the *Host Interrupt* function refer to the *Magellan Motion Control IC User Guide*.

Note that Pin 36 on the N-Series ION's Signal Connector, which carries the *HostInterrupt* signal, may also be programmed for other output functions. However the mux setting of this pin does not affect the function of the motion event. Therefore it is not a requirement that the *HostInterrupt* signal be selected for pin signal output to use the motion IC events mechanism.

An **EventWait** command suspends task code execution until the event condition becomes true, which in this case means the host interrupt goes active. Note that when the host interrupt does occur there may be Magellan instructions needed to clear and reset the interrupt. Refer to the *Magellan Motion Control IC User Guide* for more information.

Digital Signal Events

In addition to timer and motion IC events a general purpose mechanism for triggering events via an external digital signal pin is provided. To instantiate these events the command **EventOpenDI** is used. The parameters associated with this command are the digital signal event number, the digital signal, and the signal trigger condition. Up to two digital signal events can be operated simultaneously numbered 1 and 2. There are 23 selectable signals, shown in the table below, and three possible trigger types; positive edge, negative edge, and both edges (triggers on any signal change).

The trigger conditions apply to the raw signal state at the N-Series ION's digital pin, and are not affected by how the pin is programmed to be interpreted. For example the signal sense interpretation of the negative and positive limit switches is user programmable, but whether or not these signal states are programmed will not affect the functioning of the digital signal event if set to one of these signals.

The following table shows the available signals that may be selected to create an event trigger, the enumerated code used for the signal via the **EventOpenDI** command, and the associated pin # of the N-Series ION's Signal Connector:

Signal Name(s)	Enumerated Signal ID Value	N-Series ION Signal Connector Pin #s
DigitalIO1 - DigitalIO8	0 - 7	31, 32, 33, 34, 35, 36, 37, 9
HallA-C	8, 9, 10	19, 20, 21
PosLimit, NegLimit	11, 12	22, 23
Home	13	24
Enable	14	25
Brake	15	28
QuadA1	16	41, 42 (+, - differential inputs),
QuadB1	17	39, 40 (+, - differential inputs)
Index1	18	15, 16 (+, - differential inputs)

Signal Name(s)	Enumerated Signal ID Value	N-Series ION Signal Connector Pin #s
QuadA2	20	11, 12 (+, - differential inputs),
QuadB2	21	13, 14 (+, - differential inputs)
Index2	22	17, 18 (+, - differential inputs)
FaultOut	23	

As is the case for all event types the command **EventWait** is used to suspend code execution until the programmed event occurs. When the event occurs, control returns from this function call and normal code execution resumes.

If the digital signal event condition becomes true after the event is created (via **EventOpenDI**) but before the **EventWait** function call is made, task execution will not be suspended and control will immediately return from the **EventWait** call. If the event condition becomes true multiple times between **EventWait** calls, only the most recent such event is 'remembered' and any subsequent **EventWait** calls will wait until the condition event occurs again.

Digital signal events remain active regardless of whether, or how many times, the signal event condition becomes true. To disable a digital signal event the command **EventClose** is used. Note that to re-assign a particular digital signal event it is first necessary to close it.

3.9.4 Debug Console

During development, procedure calls similar to **printf()** and **puts()** can be made from the application running on the C-Motion Engine to send messages to a dedicated monitor or the PC development environment for display in a special console window. These console messages may be useful for checking code progress, displaying internal variables, or for other code development-related purposes.

Console messages are sent to one of the N-Series ION's communication ports, for example a serial port, CAN port, or Ethernet port depending on the ION unit host type being used. For host interfaces such as CAN and Ethernet, which accommodate more than one simultaneous connection, the console connection may not necessarily be a different physical port connection but rather just a separate node or port ID.

An alternative to specifying an ION communication port for the console channel is to specify Console_PRP. When set to Console_PRP console messages are transported via PRP packets using the existing PRP connection.

When set to Console_PRP the console channel can buffer up to 20 messages of 80 characters each. If console messages are not read from the PRP packet stream by Pro-Motion or a user-created program, the buffer will eventually fill up and further sendouts to the console will be lost.

For information on setting up a console link via Pro-Motion as well as communication-related information during development refer to the *ION/CME N-Series Digital Drive Developer Kit User Manual*.

3.9.5 Sending Messages to/from User Application Code

A common function of user application code running on the C-Motion Engine is to parse command messages sent to it by a host controller. For example a user might write code for the C-Motion Engine that responds to an "Extend RobotArm" command sent by the host controller, and then send a series of commands to the Magellan Motion Control IC to execute this motion sequence. At the end of the motion sequence the user application code might send an "Arm Extended" message confirming the movement sequence has completed.

One method of achieving this is to use the N-Series ION's peripheral mechanism to open, and operate, a low-level communications link via the SPI, serial, CANbus, or Ethernet link. This method has the advantage of giving relatively direct control over the communication traffic. The disadvantage is that the user has to implement specific send and receive communications in the host controller, and the user's C-Motion Engine code needs to have similar code implemented that can process these messages.

Another method that may be more convenient, particularly during early debugging of the user application code, is to use a capability of the PRP system to connect directly to the user application code on the C-Motion Engine. Messages sent and received by the C-Motion Engine from a host controller are stored in a special buffer, and can be easily read or written to by the user application code.

In addition, PMD's Pro-Motion application supports a simple way of entering, sending, and/or receiving such messages. This makes it easy to manually enter commands from Pro-Motion and exercise the user application code which is programmed to parse these messages. The main advantage of using PRP system for user packets is that

the same interface can be used for communicating to the CME device and the user code. The Serial and SPI interfaces benefit from this because they do not allow multiple communication channels over the same interface.

To utilize this approach, a new peripheral is opened using the C-Motion command **PeriphOpenCME**. Thereafter **PeriphSend** or **PeriphReceive** commands are used to send and receive these messages within the C-Motion Engine.

3.9.6 C-Motion Commands

Various C-Motion commands are highlighted in the previous sections. For additional information on these commands refer to the *C-Motion PRP II Programming Reference*.

3.9.7 Connection & Associated Signals

There are no signals directly associated with the C-Motion Engine. For information on the function of communication ports which may be used to carry C-Motion Engine console traffic and associated connections and signals refer to [Section 3.6, "Communications Functions."](#)

3.10 General Functions

3.10.1 Peripheral I/O Space

Some of the N-Series ION's bit and word-oriented resources are accessed through what is known as the Peripheral I/O address space, also called PIO for short. The following table shows the N-Series ION's Peripheral I/O address map:

Address	Function	Read/Write	Comments
0x100	BiSS-C & SSI encoder configuration	Read & Write	This register holds various settings for the BiSS-C and SSI encoder interface.
0x102	BiSS-C & SSI encoder resolution	Read & Write	This register holds the BiSS-C and SSI encoder word resolution.
0x104	BiSS-C & SSI encoder frequency	Read & Write	This register holds the BiSS-C and SSI encoder clock frequency.
0x106	BiSS-C & SSI encoder enable	Read & Write	This register allows enabling and disabling of the BiSS-C and SSI interface logic.
0x108	BiSS-C single-turn resolution and width	Read & Write	This register holds the BiSS-C single-turn encoder resolution and width.
0x10A	BiSS-C multi-turn resolution and width	Read & Write	This register holds the BiSS-C multi-turn encoder resolution and width.
0x10C	BiSS-C position shift amount	Read & Write	This register holds the BiSS-C number of bits to shift the position value right to exclude extra error or warning bits.
0x200	General purpose digital input values	Read	The 8 available digital I/O bit input values are stored in this register.
0x210	General purpose digital output write mask	Read & Write	The digital I/O output write mask is stored in this register.
0x212	General purpose digital output write value	Read & Write	The 8 digital I/O bit write values are stored in this register.
0x220	Bi-directional I/O direction mask	Read & Write	The bi-directional I/O direction mask is stored in this register.

Address	Function	Read/Write	Comments
0x222	Bi-directional I/O direction value	Read & Write	The 8 bi-directional digital I/O direction values are stored in this register.
0x224	Bi-directional I/O direction setting read	Read	The effective bi-directional digital I/O bit direction values are read using this register.
0x228	Muxed pin function selection	Read & Write	This register is used to select a specific function for pins that support multiple functions and that are user programmable.
0x250	Event signal source	Read & Write	This register selects the signal sources for programmable event #1.
0x252	Event signal source	Read & Write	This register selects the signal sources for programmable event #2.
0x340	Analog input channel value.	Read	This register holds the analog 16-bit input word.
0x344	Sin/Cos Analog Input Value	Read	This 32-bit register holds the sin and cos analog 16-bit input values. The Sin value is stored in the high 16 bits, the Cos in the low 16 bits.
0x412	Green LED control setting		This register programs the LED control setting for the N-Series ION's green LED.
0x414	Red LED control setting		This register programs the LED control setting for the N-Series ION's red LED.

3.10.1.1 C-Motion Commands

To initiate access to the PIO registers the command **PMDDeviceOpenPeriphPIO** is used with an address argument value of 0 (zero), eventIRQ argument value of 0, and datasize argument value of PMDDataSize16.

Once access has been created the commands **PMDPeriphRead** and **PMDPeriphWrite** are used to read and write to the PIO registers. For complete information refer to the *C-Motion PRP II Programming Reference*.

3.10.2 Pin Multiplexing Control

3.10.2.1 Direct Pin Function Selections

A number of the signal pin connections provided by N-Series IONs have multiple user selectable functions. Providing multiple functions maximizes the features available to the user while keeping the pin count of the ION unit manageable.

The direct user settable MUXed pins and their available selection settings are shown in the table below:

Pin #	0 Setting	1 Setting (Default)	2 Setting	3 Setting	Comment
9	DigitalIO8	HostSPIStatus			CAN/SPI host unit type only
9	DigitalIO8	RS485Sel			Serial host unit type only
9	DigitalIO8				Ethernet host unit type only
31	DigitalIO1	Srl3Xmt***	ExpSPIXmt		
32	DigitalIO2	Srl3Rcv***	ExpSPIXRcv		
33	DigitalIO3	AxisIn	ExpSPIClock	Battery	
34	DigitalIO4	SynchIn	ExpSPICS1		

Pin #	0 Setting	1 Setting (Default)	2 Setting	3 Setting	Comment
35	DigitalIO5	AxisOut	ExpSPICS2		
36	DigitalIO6	HostInterrupt*	ExpSPICS3	EthLinkLED*	
37	DigitalIO7/Exp-SPIStatus**	SynchOut	ExpSPICS4	EthSpeedLED	

* The default pin function for this pin when the unit host type is serial or CAN/SPI is HostInterrupt. When the host type is Ethernet the default function is EthLinkLED.

** If PRP communications through the expansion SPI port are commanded this pin will carry the ExpSPIStatus input function. Otherwise it will have the normal programmable DigitalIO7 function.

*** Special caution should be exercised when changing the pin function of pin 31 or 32 since these pins carry the serial signals for the 3-pin programming port. An alternate communication port should be used if the default mux setting of one or both of these pins is changed.

To program these pin functions the PIO register address **0x228** is used. The follow table shows the relationship between this register and the pin selection.

Each multiple use pin is selected via a two-bit field in the PIO register, with a value of 0 selecting function 1, a value of 2 selecting function 2 etc. The table below shows the locations of the two-bit fields for each programmable signal:

Pin #	Bits
9	14, 15
31	0, 1
32	2, 3
33	4, 5
34	6, 7
35	8, 9
36	10, 11
37	12, 13

While these settings can be changed dynamically, caution should be exercised particularly for output signals because depending on the new selected function relative to the previous, the pin output may change state.



It is recommended that the user plan the pin function selections for their application and then store these settings in the N-Series ION's NVRAM so that the correct pin MUX settings are active from power up. This is especially true when the default pin function is changed from an output to an input and external hardware will be driving the signal state. Failure to do so may result in signal contention after the unit powers up. The pins which are outputs in their default configuration are HostSpiStatus, AxisOut, HostInterrupt, and SynchOut.

3.10.2.2 Encoder-Based Pin Function Selections

The following pin #s also carry multiple functions, but for these pins the function is not explicitly programmed by the user, rather they are determined by the user's selection of the encoder type.

Pin #	Quadrature	Pulse & Direction	Sin/Cos	BiSS-C
11	QuadA2 +	Pulse2 +		
12	QuadA2-	Pulse2-		
13	QuadB2 +	Direction2 +		
14	QuadB2-	Direction2-		
15	Index1 +		Index1 +	BiSSClock +
16	Index1-		Index1-	BiSSClock-

Pin #	Quadrature	Pulse & Direction	Sin/Cos	BiSS-C
17	Index2 +		Index2 +	BiSSData +
18	Index2-		Index2-	BiSSData-
39	QuadB1 +	Direction1 +	Sin +	
40	QuadB1-	Direction1-	Sin-	
41	QuadA1 +	Pulse1 +	Cos +	
42	QuadA1-	Pulse1-	Cos-	

To program these pin functions the Magellan IC C-Motion command for selecting the encoder type is used. Similarly to the pin MUX settings described in the previous section, it is recommended that the encoder setting be stored in the Magellan's NVRAM so that it is active from power up.

3.10.2.3 C-Motion Commands

The primary C-Motions commands used to access the Peripheral I/O registers are **PeriphOpenPIO**, **PeriphRead**, and **PeriphWrite**. For complete information refer to the *C-Motion PRP II Programming Reference*.

To set and retrieve the setting of the encoder type the Magellan Motion Control IC commands **SetEncoderSource** and **GetEncoderSource** are used. For more information on these C-Motion commands refer to the *Magellan Motion Control IC Programming Reference*.

3.10.3 Reset

Although a reset occurs automatically during power-up, it is sometimes desirable to reset the N-Series ION unit explicitly. The user can accomplish this in two different ways; via external reset signal or via C-Motion user command.

To reset the ION using the reset pin this signal should be momentarily driven low and then returned to a high state. This signal is always active low and its sense can not be changed.

Resetting the ION unit will result in immediate discontinuation of all motor control operations and C-Motion Engine user code execution. It is up to the user to determine whether a reset can be safely executed based on the application conditions.



After a reset occurs the Magellan Motion Control IC and the C-Motion Engine modules will be reset, and many of the N-Series ION's output signals will be driven to known states. for detailed information refer to [Section 3.12.5, "Default Pin States."](#)

In addition, upon a drive reset all default parameters are reloaded. See [Section 3.10.7, "Setting Default Values,"](#) for more information on default values.

3.10.3.1 C-Motion Commands

The C-Motion command **DeviceReset** is used to reset the ION unit. This command, upon execution by the user responds in the normal way but thereafter executes the reset. For more information on this command refer to the *C-Motion PRP/II Programming Reference*.

3.10.3.2 Connections & Associated Signals

The table below shows the signals associated with the N-Series ION hardware reset. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Reset	27	When brought low this signal resets the ION unit, and when high restores normal operation.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

For the reset signal to function properly one or more of the digital grounds must be connected.

3.10.3.3 Reset Signal Processing

Some users may find it useful to know the exact internal processing circuitry used by the N-Series ION to process I/O signals.

The following diagram shows this for the *Reset* signal.

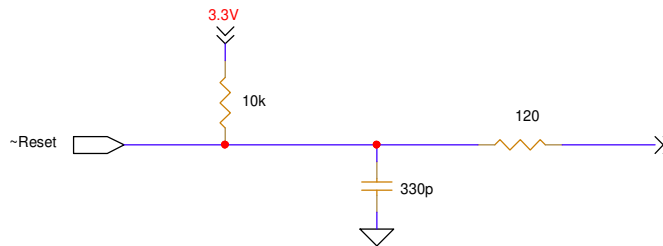


Figure 3-28:
Reset Signal
Internal
Processing
Circuitry

3.10.4 Reset Monitor

In addition to power-up, external reset via signal, or reset by C-Motion command, there are several other ways that the N-Series ION can reset.

To determine the cause of a reset, the N-Series ION maintains a reset monitor register that can be read by the user via C-Motion command. The following table details the encoding of the value returned by the reset monitor.

Reset Cause	Bit Location	Description
Software reset	9	A 1 value in this bit indicates a reset commanded via C-Motion command, or a reset due to a user code exception.
User code exception	10	A 1 value in this bit indicates an instruction or address access fault in the user code running on the C-Motion Engine.
Watchdog reset	11	A 1 value in this bit means that an internal watchdog has timed out indicating the task scheduler has stopped functioning. The most common reason for this is that the user has created a high priority task which preempts task switching to other tasks, including operating system tasks, thereby effectively disabling the task scheduler.
Pin reset	12	A 1 value in this bit indicates a reset commanded via the N-Series ION's reset pin (pin #27 of the Signal Connector).
Undervoltage	13	A 1 value in this bit indicates a reset caused by detection of an internal undervoltage condition. A reset due to this condition is generally not an error and in fact will always occur after external power (HV) to the N-Series ION is removed. However, undervoltage resets that occur when HV power is within nominal limits may indicate a serious internal fault of the unit.

The most common use of this feature is as a safety check, contained in user application code residing on the C-Motion Engine. Since user application code most often automatically executes after ION powerup, a check of this register by the user application code will allow anomalous occurrences of a reset to be flagged and investigated.

3.10.4.1 C-Motion Commands

To read the N-Series ION's reset monitor register the command **DeviceGetResetCause** is used. For more information on this command refer to the *C-Motion PRP II Programming Reference*.

3.10.4.2 Connections & Associated Signals

There are no signals associated with this function.

3.10.5 LEDs

N-Series IONs have two LEDs for indicating the status of the unit, one green and one red. The green LED is called the power LED and is solid on when the unit is powered and the *Enable* signal is active (low). If the unit is powered but the *Enable* signal is inactive (high) this LED will blink. If the unit is not powered this LED is off.

The red LED is called the status LED and is normally off indicating no errors. There are three additional states of this LED consisting of fast blink (three times per second), slow blink (once per second), and solid on. The table below summarizes the above:

Green (Power) LED	Red (Status) LED	Condition
off	off	Unit not powered
solid on	off	Unit Enabled (signal level low)
blinking slow	off	Unit Disabled (Enable signal level high)
solid on	blinking fast	Unit experienced an overvoltage or undervoltage condition
solid on	blinking slow	Unit experienced an overtemperature or I ² t overcurrent condition
solid on	solid on	Unit experienced a hard fault condition

For more information on overvoltage, undervoltage, overtemperature, I²t overcurrent, and overcurrent conditions including suggested recovery procedures refer to [Section 3.8, "Drive Safety Functions."](#) Note that the LED states for these conditions show the latched condition. That is, the LED status will not return to normal until the Magellan IC receives a **ResetEventStatus** command with the condition no longer present.

If operating in its default LED display mode any activation of the red LED indicates a potentially serious fault in the drive or in the system being controlled. It is the responsibility of the user to determine the correct and safe recovery procedure for such a condition.



3.10.5.1 Manual Control of LEDs

The function of the LEDs described above is the default functionality, however control of the LEDs can be taken over manually by the user if desired. This is accomplished by writing to either the 0x412 (for control of the green LED) or 0x414 (for the control of the red LED) PIO register. A read to this same register indicates the current setting of the manual LED control register.

The table below shows the available programmable states for each LED, and the PIO register settings:

LED Behavior	PIO Register Setting	Comments
Solid off	0	
Solid on	1	
Slow 50% blink	2	LED blinks approximately once per second with 50% on/off duty cycle.

LED Behavior	PIO Register Setting	Comments
Fast 50% blink	3	LED blinks approximately three times per second with 50% on/off duty cycle.
Slow 25% blink	4	LED blinks approximately once per second with 25% on/75% off duty cycle.
Fast 33% blink	5	LED blinks approximately three times per second with 33% on/66% off duty cycle.
Slow 8% blink	6	LED blinks approximately once per second with 8% on/92% off duty cycle.
Very fast 50% blink	7	LED blinks approximately six times per second with 50% on/off duty cycle.
Default behavior	-1	Sets the specified LED to the default behavior described in the previous section. This is the default setting for both the red and green LEDs.

These manual LED settings can be applied after power up in RAM, or can be stored in the N-Series ION's NVRAM, thereby becoming active automatically from power up.

3.10.5.2 C-Motion Commands

Manual control of the LEDs is provided via the PIO (Peripheral I/O Space) registers. Refer to [Section 3.10.1, "Peripheral I/O Space,"](#) for more information on how to access these registers via C-Motion commands.

3.10.5.3 Connection & Associated Signals

There are no signals directly associated with the LEDs.

3.10.6 Real Time Clock

N-Series IONs support a real time clock and an associated low current *Battery* pin input to provide just ION's real time clock circuitry with power. This is generally connected to a battery which is located nearby the N-Series ION unit. The provided voltage must be in the range 1.5V - 3.3V. The current draw of the battery signal in power backup mode is between 2 μ A and 8 μ A depending on the input voltage with lower draw occurring with lower voltages.

The user can set and read back the real time clock using C-Motion functions, and the time resolution of the settable and readable time is one millisecond. The absolute worst case stability of the real time clock is 50 ppm which translates to roughly 5 seconds per day.

The time format is 24 hour format and is not connected to a particular time zone.

A C structure is used to read and set the time. This structure has the same format as the Windows SYSTEMTIME structure, with sub fields of year, month, day-of-week, day, hour, minute, second, and millisecond.

3.10.6.1 C-Motion Commands

To set the real time clock the C-Motion command **DeviceSetSystemTime** is used. To read back the current real time the command **DeviceGetSystemTime** is used. For more information on these commands refer to the *C-Motion PRP/II Programming Reference*.

3.10.6.2 Connections & Associated Signals

The table below shows the signals associated with the real time clock function. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
Battery*	33	Battery connection to provide power to real time clock when HV Aux power is off.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

* These signals are located on pins with multiple programmable functions. The pin function applies only if so programmed.

For these signals to function properly one or more of the digital grounds must be connected.

3.10.7 Setting Default Values

There are numerous user-settable parameters that are saved by the N-Series IONs in non-volatile RAM and are utilized after a powerup, or after a unit reset. These include communication parameter settings, pin mux settings, PIO register values, and other quantities. The list of default variables is fixed, and the default values are stored into NVRAM during factory unit production.

These default values can be re-programmed by the user so that application specific settings, rather than the factory default settings, are automatically utilized at powerup and become the operational values. Doing so may be useful simply as a convenience so that application-specific parameters do not always need to be updated after power on, or it may be required for proper communication, peripheral initialization, or for other reasons.

As a convenience to the user the various settable NVRAM default values are summarized in [Section A.2, "N-Series ION NVRAM Default Settings."](#)

If programmed by the user default values will not take affect until the next unit power up or reset. Default values are stored in the N-Series ION's Device NVRAM area. As such, writing and reading operations to the default value area involves special considerations. For more information refer to [Section 3.11, "Device NVRAM."](#)

Note that in addition to the N-Series ION's Device NVRAM area the Magellan IC has its own NVRAM area that may be used to store application specific parameters such as servo gains, PWM frequency, and other settings managed by the motion control IC. For more information refer to [Section 3.2.2.2, "Magellan IC NVRAM."](#)

The typical write time to the non-volatile RAM is 30 μ Sec, however it may take as long as several 100 mSec. If other portions of the user application code, or any other PRP-connected device, depends on these values having been written, it is recommended that you ensure that the write operation has been completed by adding code that explicitly checks the value, or by waiting a fixed period of time after the NVRAM write operation.



3.10.7.1 C-Motion Commands

The C-Motion command **DeviceSetDefault** is used to set new default values. The command **DeviceGetDefault** is used to read these values back. For detailed information on these commands along with a complete list of programmable default parameters and their factory default settings refer to the *C-Motion PRP II Programming Reference*.

3.10.8 Unit Resource Information

There are various resources on the N-Series ION that contain programmable or otherwise changeable logic or firmware. It may be beneficial to the user to be aware of revision information associated with these logic or firmware entities.

In general, once the user has developed his motion application they may want to query and record the resource revision information so that they can be aware of any changes to these revisions in future production units purchased from PMD.

Along these lines, the N-Series ION resource information that may be queried is the device logic version, the device firmware version, the Magellan IC version, the product part number, and the product serial number. The device logic, device firmware, and Magellan IC version are provided in a numerical major/minor revision format. The N-Series ION unit part number and serial number are provided as UTF-16 strings, for example "DD441S0056/06".

The version info is accessed via C-Motion commands as described in the next section. The product part number and serial number are stored within the Magellan IC NVRAM and are accessed using PSF (PMD Structured Data Format). These parameters are contained in segment ID 1 and the parameter names are "P#" for the part number and "S#" for the serial number.

For more information on PSF storage in the Magellan NVRAM refer to [Section 3.2.2.2, "Magellan IC NVRAM."](#) For more information on storing and retrieving information in PSF format refer to the *Magellan Motion Control IC Programming Reference*.

Although it is possible to write software code to execute the C-Motion commands needed to retrieve the above product information, most users will instead utilize Pro-Motion for this purpose, using it to conveniently retrieve and display the product info for a given ION unit.

3.10.8.1 C-Motion Commands

To query the device logic version and the device firmware version the C-motion command **DeviceGetInfo** is used. For more information on this command refer to the *C-Motion PRP II Programming Reference*.

To query the Magellan IC firmware version the Magellan IC C-Motion command **GetProductInfo** is used. For more information on this command refer to the *Magellan Motion Control IC Programming Reference*.

3.10.9 5V Output

N-Series IONs provide a 5V output on pin 29 of the Signal Connector for powering external encoders, Hall sensors, or other circuitry. The rated drive capacity of this output is 50 mA.

This signal is monitored to detect overloading and will trigger a drive exception event in the Magellan IC if the output current exceeds 100 mA. A drive exception event will disable motor output. At higher current levels or if a short circuit is present the N-Series ION unit will shut down entirely.

Whether resulting in a drive exception or full shutdown, if an overcurrent condition exists on the +5V output the user should immediately remove power from the N-Series ION, and only re-power once the condition has been corrected by reducing the load on the 5V output to 50 mA or below.



The maximum output rating of the 5V output pin is 50 mA. Connecting this output to circuitry that exceeds this rating may result in a drive exception event or unit shutdown and may result in damage to the N-Series ION unit.

3.10.9.1 C-Motion Commands

To query the Magellan IC Event Status register the command **GetEventStatus** is used. To query the drive fault status register the command **GetDriveFaultStatus** is used. For more information on these commands refer to the *Magellan Motion Control IC Programming Reference*.

3.10.9.2 Connections & Associated Signals

The table below shows the signals associated with the 5V output pin. All of these signals are located on the N-Series ION's Signal Connector.

Pin Name	Pin #	Description
5V	29	+ 5V output for powering external encoders, hall sensors, or other circuitry.
GND	30	Ground. Note that any of the N-Series ION's ground signal pins can be used including pins 10, 30, or 38.

3.11 Device NVRAM

The N-Series ION has a general purpose 128 Kb memory that retains its contents after a module power down or reset known as the Device NVRAM. This memory is useful for storing parameters that are set only occasionally and stay with the module.

When writing to this memory, a typical write takes 30 μ Secs, however read operations take just a few nanoseconds. Write operations to the Device NVRAM must be made in blocks of 256 bits. Read operations may be byte, 16-bit word, or 32-bit word oriented depending on how access is set up via the C-Motion memory commands.

The non-volatile memory can be rewritten a limited number of times. The worst case write limit cycle is 100,000 times for a given memory address, but in typical operation, the limit is much higher. As a general guideline, to avoid erase/ write cycle limit problems, the non-volatile RAM should not be used for general purpose scratch RAM, and should only be used to store permanent or semi-permanent parameters.

3.11.1 C-Motion Commands

To read or write to the NVRAM, a resource address must first be obtained by sending the Open action to the **Memory** resource. This resource address is used for all further access to the NVRAM. To write data to the NVRAM, the action Write is sent to the **Memory** resource. To read the contents of the NVRAM, the action Read is used.

For complete information on these commands refer to the *C-Motion PRP II Programming Reference*.

3.11.2 Connections & Associated Signals

There are no signals associated with the NVRAM function.

3.12 N-Series ION Powerup & Initialization

N-Series IONs undergo an automatic powerup & initialization sequence when they go from being unpowered to receiving power at the HVAux pin, or when they are powered and receive a reset command. See [Section 3.10.3, "Reset,"](#) for information on resetting the N-Series ION unit.

Here is a summary of the sequence that occurs during initialization:

- Internal hardware checks are performed.
- The Magellan IC undergoes its initialization which may include execution of stored NVRAM sequences. For more information on Magellan IC initialization and executing stored Magellan NVRAM sequences refer to [Section 3.2.2, "Magellan Initialization."](#)
- Host communication interfaces and expansion network interfaces are configured with their programmed settings and initialized.
- Signals which may be assigned multiple functions are loaded with the user-programmed pin MUX settings. For more information on these pin settings refer to [Section 3.10.2, "Pin Multiplexing Control."](#)

3.12.1 Checking Initialization Status

Two different approaches may be used to check the initialization status of an N-Series ION after reset or powerup from an external device connected by one of the host communication interfaces.

A typical initialization sequence where no anomalies occur takes about 1,100 mSec. The first method is therefore to wait 1,100 ms before attempting to communicate. The first communication sent can be any PRP command however a **PMDDDeviceNoOperation** is recommended. This first PRP command will not be processed and will return a **PMD_ERR_RP_Reset**, error. After this first PRP command has been sent regular communication are now possible.

Once the above sequence is completed, to check whether an error occurred during initialization the command **PMDDDeviceGetFaultCode** should be sent. This command will indicate whether the completed initialization occurred normally or whether a fault occurred and if so which specific fault.

An alternative approach to checking when the unit is initialized is to send a sequence of **PMDDDeviceNoOperation** commands followed by a brief delay, until a non-error return code is received. A non-error return code indicates that the N-Series ION unit is initialized. As before, once initialization is completed the status of the initialization sequence should be checked with the **PMDDDeviceGetFaultCode** command to determine whether errors occurred during initialization.

If a fault occurs during initialization the Magellan IC motor drive output will be disabled, however a limited palette of commands useful for checking status and diagnosing the Magellan IC are still available. User code, if it has been previously downloaded, will be executed if it has been set up for automatic startup, however if the type of fault that occurred was related to the user's application code auto-execution will not occur. Refer to the *C-Motion PRP II Programming Reference* for specifics.



A fault during initialization represents a serious condition. It is the responsibility of the user to determine the nature of the fault and undertake corrective measures, if appropriate, before re-initializing the unit.

3.12.2 Magellan IC Initialization

If no NVRAM sequences are stored in the Magellan IC initialization typically completes at the same time that other elements of the N-Series ION initialization complete, within approximately 500 mSec. If NVRAM sequences are stored in the Magellan however, in particular if they include delay commands or conditional execution based on the status of external signals, the time to complete Magellan initialization can be significantly longer.

In this situation it is therefore advisable for the host, or for internally loaded C-Motion code, to wait until Magellan initialization has completed and to check the status of this initialization. To determine whether initialization has completed the **GetDriveStatus** command is used. To determine whether a fault has occurred the **GetEventStatus** command is used, and the **GetInstructionError** command can be used to read the fault condition should one have occurred.

The Magellan IC checks for numerous motion system conditions including over current, over/under voltage, over temperature, as well as others. If a fault has occurred it is up to the user to determine whether a recovery sequence can safely be attempted through the host interface or via C-Motion user code, or whether the ION unit should be disabled and the problem diagnosed by a technician.

For more information on Magellan IC safety features and operation refer to the *Magellan Motion Control IC User Guide*.

3.12.3 C-Motion Engine Initialization

If user code has been downloaded into the C-Motion Engine and if auto execute has been enabled, after initialization this code will begin execution automatically.

This user code will typically undergo its own initialization process which may include communicating with and checking the status of whatever external devices are connected to the N-Series ION unit. In addition to external devices connected to the ION unit this code should also check that the Magellan has completed its initialization and that the Magellan IC is ready for operation.

This C-Motion startup user code typically contains sequences that can respond to various recoverable conditions depending on the application, as well as identifying non-recoverable faults that require unit shutdown or manual operator intervention.

3.12.4 Steady State Operation

If all initialization checks and associated recovery sequences have been completed successfully the ION unit is considered to be in a normal operating condition and will operate indefinitely until one of the following occurs:

- Power to the HVAux pin is removed
- A hardware pin reset or commanded reset is received
- A C-Motion Engine user code invalid memory access or other violation occurs resulting in a reset.
- An internal 'hard fault' error occurs such as logic undervoltage or a failed internal circuit integrity check.

Except for the first condition, where power is removed to HVAux, after reset the unit will once again attempt initialization. Depending on the cause of the reset that aborted normal operation, another reset may again eventually occur, or the unit may initialize and operate normally.

It is strongly advisable to always investigate the cause of an unexpected unit reset. See [Section 3.10.4, "Reset Monitor,"](#) for information on the N-Series' ION reset monitor which allows the user to determine the cause of a reset, and provides suggestions on resolving the reset source.

Note that during unit operation various potentially serious Magellan IC anomalies may occur such as over current, over temperature, etc... however these conditions do not automatically result in an ION unit reset or shutdown. It is up to the host (or the user's downloaded CME code), to check the Magellan IC during operation and determine whether a particular motion system fault can be recovered from safely, or requires a unit shutdown.

3.12.5 Default Pin States

During initialization N-Series ION's Signal Connector pins are restored to default states and interpretation. The following table shows the default signal states or signal state interpretation for selected N-Series ION signals, along with whether or not these states are user programmable.

If user programmable, the interpretation can be temporarily changed in RAM or can overwrite the factory default settings by storage into the ION's NVRAM. For more information refer to [Section 3.10.7, "Setting Default Values."](#)

Signal Connector Pin #	Signal Name*	Input, output, or bidirectional	Sense interpretation	Sense User programmable?
9	RS485Sel	Input	Low = RS485 High = RS232	No
11, 12	QuadA2 +, QuadA2-	Input	N/A	Yes
13, 14	QuadB2 +, QuadB2-	Input	N/A	Yes
15, 16	Index1 +, Index1-	Input	Low, high = in index High, low = not in index	Yes
17, 18	Index2 +, Index2-	Input	High, low = in index Low, high = not in index	Yes
19	HallA	Input	High = on	Yes
20	HallB	Input	High = on	Yes
21	HallC	Input	High = on	Yes
22	PosLim	Input	Low = In limit	Yes
23	NegLim	Input	Low = In limit	Yes
24	Home	Input	Low = in home	Yes
25	Enable	Input	Low = enabled	No
26	FaultOut	Output	High = fault active	No

Signal Connector Pin #	Signal Name*	Input, output, or bidirectional	Sense interpretation	Sense User programmable?
27	Reset	Input	Low = reset	No
28	Brake	Input	Low = brake active	No
33	AxisIn	Input	N/A	Yes
34	SynchIn	Input	High to low = synch pulse	No
35	AxisOut	Output	N/A	Yes
36	HostInterrupt**	Output	Low = interrupt active	No
36	EthLinkLED**	Output	N/A	No
37	SynchOut	Output	High to low = synch pulse	No
39, 40	QuadB1 + , QuadB1-	Input	N/A	Yes
41, 42	QuadA1 + , QuadA1-	Input	N/A	Yes

* For multi-function pins the above table only shows the default pin function. If these pins are programmed to a function other than the default function, the above listed interpretations may change. Refer to the corresponding section of this manual for the programmed pin function for more information.

** The default pin function for pin 36 when the unit host type is serial or CAN/SPI is HostInterrupt. When the unit host type is Ethernet the default function is EthLinkLED.

4. Installation and Production-Related

4

In This Chapter

- ▶ Mechanical Mounting
- ▶ Production Unit Configuration

4.1 Mechanical Mounting

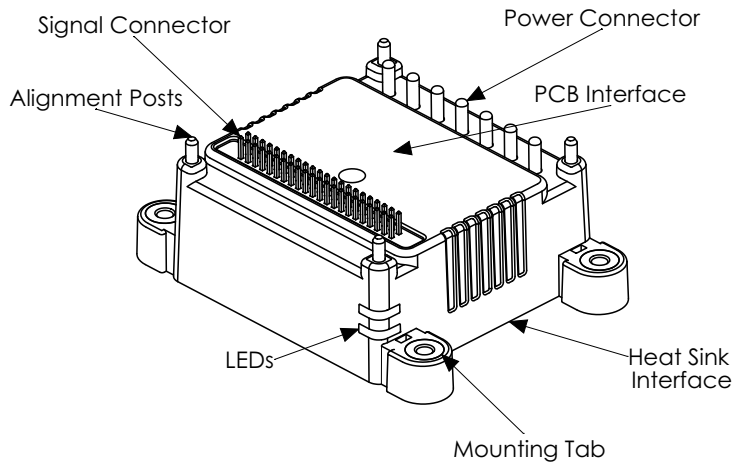


Figure 4-1:
Mechanical
Elements of N-
Series ION
Drive

The N-Series ION Drives are designed to be used in a wide variety of operating conditions. As shown in [Figure 4-1](#) they have a robust mechanical design that allows rigid mechanical attachment at the two primary interface points; the interface between the N-Series ION and the user PCB, and the interface between the N-Series ION and the heat sink or cold plate also called a supporting plate.

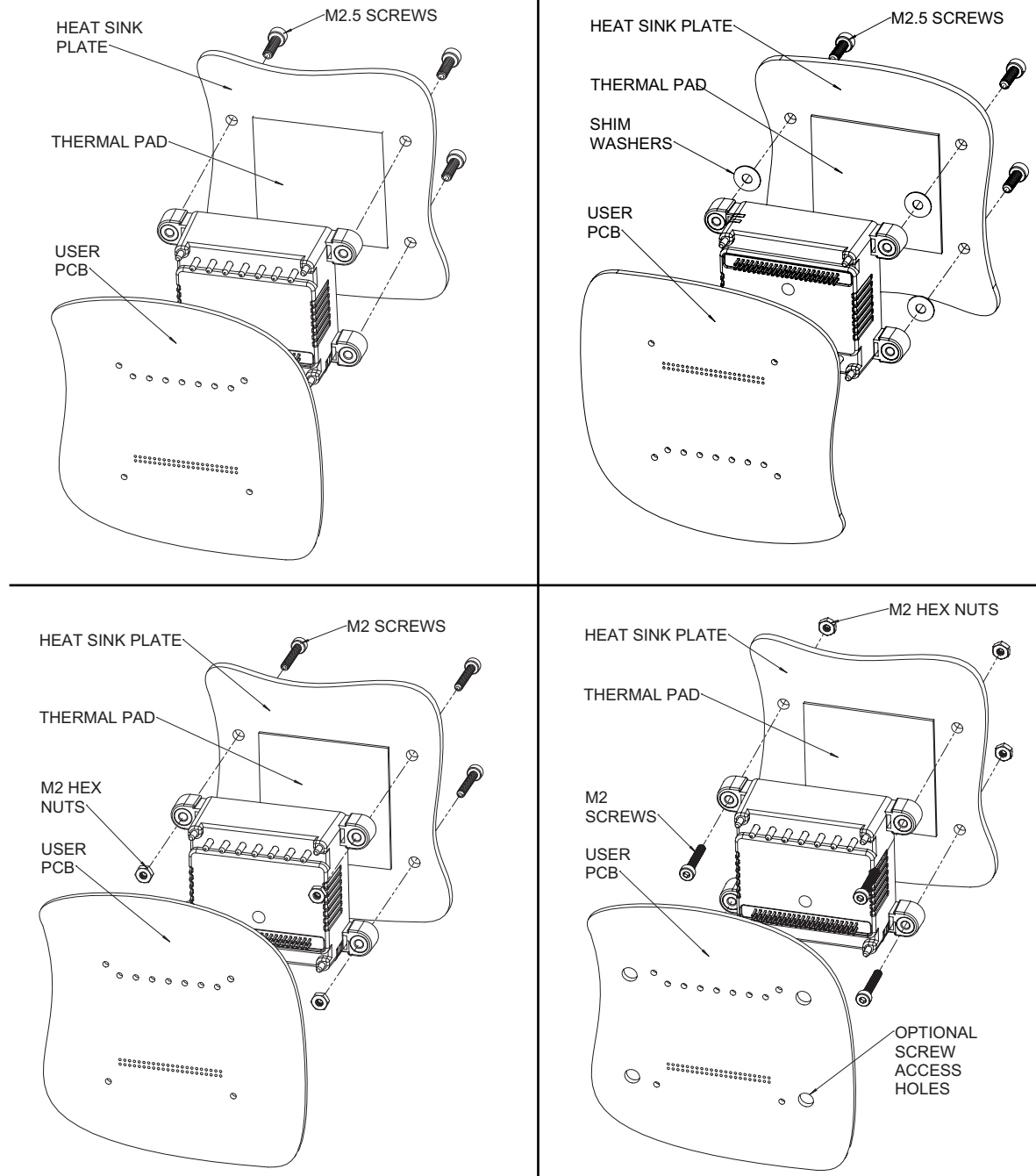
The following sections describe how mechanical attachment at both these interfaces occurs.

4.1.1 Attachment to the User PCB

Attachment at the PCB is accomplished by soldering the PCB to the N-Series ION's four mechanical alignment posts. While these posts do not serve an electrical function, they provide a rigid attachment of the PCB to the N-Series ION, and thereby reduce strain that may occur in the solder connections between the user PCB and the Power and Signal Connectors of the N-Series ION. Use of the four PCB alignment posts are recommended for all applications.

4.1.2 Attachment to the Heat Sink

**Figure 4-2:
N-Series ION
Mounting
Options**



Attachment to the heat sink or supporting plate is accomplished via screws at the N-Series ION's four mounting tabs. As shown in [Figure 4-2](#) there are four typical approaches to mechanically attaching to the heat sink.

The first is mounting from the heat sink side. In this approach M2.5 screws are installed directly into the ION's tab which is threaded for a M2.5 screw. A variation of this option may be used when greater mounting rigidity is desired, typically because the application environment has higher g forces. This mounting method locates shim washers under the tabs to limit tab deflection and to increase the amount of torque that can be applied to tighten down the tabs.

When using this method the thickness of the shim washer should be determined by the type and thickness of thermal transfer material used to create a vibration resistant mount, while maintaining optimal thermal transfer.

Note that in this approach, the N-Series ION's mounting tab mechanism, which normally acts to limit the maximum force that can be applied via the tabs, is effectively bypassed. Therefore it is up to the user to ensure that the force applied to the ION unit's heat sink interface, the limits for which are provided in the table below, are not exceeded. For more information on PMD's mounting tab design see [Section 4.1.2.2, "N-Series ION Mounting Tab Design."](#)

To achieve the shim function using washers, the recommended dimensions are .280" (7.0 mm) outer diameter and .105" (2.5 mm) inner diameter. Representative suppliers for washers such as this include Bokers, Superior Washer, and Phoenix Specialty.

When using shim washers or other approaches that raise the mounting interface under the N-Series ION tabs, the ION unit's force-limiting mounting tab design is effectively disabled. It is therefore up to the user to ensure that the mounting force limit on the ION unit's heat sink interface plate is not exceeded.



Two other mounting options use a nut and a screw. Most commonly M2 screws are used which go through the tab's M2.5 thread without engaging the thread, with the nut used to capture the M2 screw. The two possible orientations of the nut and screw represent these two different mounting options. Note that if the preferred mounting orientation is to have the screw heads facing the user PCB, it may be desirable to have holes in the PCB to give access to the ION screw hardware. However if the ION is mechanically mounted first, and then soldered onto the user PCB, these access holes may not be necessary.

4.1.2.1 Installation & Torque Limits

Before mating the N-Series ION to the heat sink, in a typical installation a thermal transfer material will first be installed to enhance thermal contact. Refer to [Section 4.1.2.3, "Thermal Transfer Materials,"](#) for more information on selection of these materials.

Once this material is in place the tabs can be tightened under careful torque limit control. The table below shows the recommended maximum torque for both M2 and M2.5 screws along with equivalent total force applied to the ION unit's heat sink interface if an alternate attachment method such as a clamp is used, or if shim washers are used as described above.

Type	Minimum	Recommended	Maximum
M2 screw torque	0.082 N-m (0.722 in-lb)	0.094 N-m (0.836 in-lb)	0.113 N-m (1.00 in-lb)
M2.5 screw torque	0.113 N-m (1.00 in-lb)	0.130 N-m (1.15 in-lb)	0.147 N-m (1.30 in-lb)
Force	25 kgs (55 lbs)	29 kgs (68 lbs)	33 kgs (71 lbs)

Exceeding the N-Series ION's mechanical attachment specifications indicated above may cause a failure of the drive unit at the time of installation, or at a later time in the field.



4.1.2.2 N-Series ION Mounting Tab Design

N-Series IONs utilize a mounting tab design that reduces the probability that overtightening of the tab attachment hardware will cause damage to the N-Series ION unit.

This is achieved by the mounting tab deliberately breaking (snapping) under conditions of over tightening. Such a protective 'fuse' break event (snap) of the tab should be audible to the operator during tightening. While an N-Series ION unit that has undergone a fuse break event during tightening is no longer usable, it serves as a signal to



the operator to check and revise their torque control procedure so that the tab mounting torque limits are not exceeded during future attachment procedures.

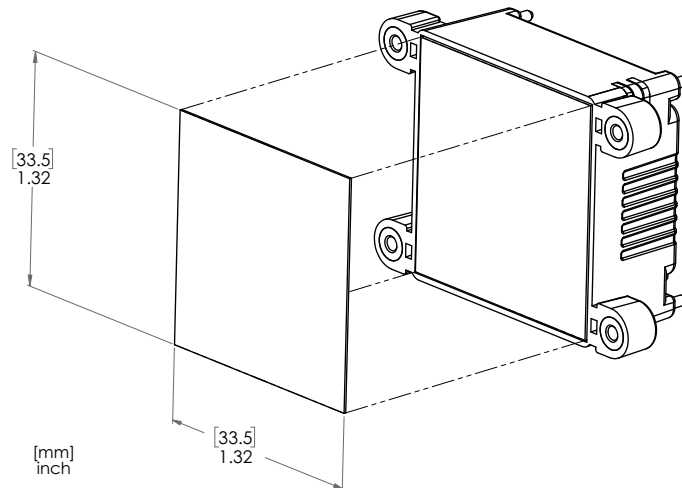
An ION/CME N-Series Digital Drive which has undergone a mounting tab fuse break event during installation must be de-installed and discarded. It is not possible to repair or to otherwise use an N-Series ION unit which has undergone a fuse break.



The N-Series ION's mounting tab fuse break functions when there is an air gap between the bottom of the tab and the surface that it is mounted to. If there is no air gap, or if the air gap is too small then the fuse/snap function may not occur even though the recommended mounting torque limit may have been exceeded. In all cases it is the responsibility of the user to determine that the recommended mounting torque limits or force indicated in [Section 4.1.2.1, "Installation & Torque Limits,"](#) are not exceeded.

4.1.2.3 Thermal Transfer Materials

Figure 4-3:
Recommended
N-Series ION
Thermal
Transfer
Material
Dimensions



Thermal transfer materials in the form of thermal tape, pads, paste, or epoxy may be used to improve thermal transfer between the N-Series ION's metal plate and an attached heat sink or supporting plate. These materials improve thermal conductivity by filling in air gaps that form when two metallic surfaces are mated.

[Figure 4-3](#) shows a typical application of a thermal transfer material between the drive unit and a heat-removing metal surface. The following guidelines may be helpful in selecting and sizing the thermal transfer material best-suited to your application.

The capacity of thermal transfer materials to transfer heat (known as the bulk conductivity) is much lower than that of metals such as aluminum or copper. Therefore, in general, the thinner the transfer material the better. Thickness of the material is only precisely controllable for thermal pads and thermal tapes, with thermal pads providing the thinnest available interfaces beginning at 5 mils (.127 mm) or even less. For use with N-Series IONs thermal transfer materials that are thicker than 10 mils (0.25 mm) are not recommended regardless of the material used.

When using thermal paste or thermal epoxy glue the thickness should be carefully controlled via a silk screen or other wet film application process. The N-Series unit itself should not be used to squeeze non-uniformly applied paste or epoxy flat during installation. Doing so may result in damage to the drive unit.

Whether using tape, pads, paste, or epoxy, as shown in [Figure 4-3](#), the thermal transfer material that is used as the interface should not extend to the area under the N-Series ION's tabs because this may reduce the amount of compression that occurs in the thermal transfer area. The following table provides dimensions for the applied thermal transfer material for N-Series IONs:

Parameter	Maximum Pad Dimensions
Value	33.5 mm x 33.5 mm (1.32" x 1.32")

4.1.2.4 Additional Mounting & Production Considerations

Below are recommended soldering and mounting procedures that should be followed to maintain the electrical and mechanical integrity of the N-Series ION unit during installation.

Mounting surface flat and clean. Thermal performance as well as safe operation of the N-Series ION requires that the surface that it is mounted to be flat and clean, free of dust, grease, or foreign objects. The recommended maximum deviation of the mating surface flatness is 3 mils (.076 mm).

Progressive tightening. N-Series ION units that are mated to a heat sink or supporting plate should be attached by progressively tightening each of the unit's four tabs. This means that one screw may be tightened, followed by the others, then back to the first etc. until the desired torque at each screw has been achieved. Following this procedure is particularly important when installing N-Series ION units over paste or epoxy, where the subsurface layer will undergo compression and movement before settling to a final installed position.

Soldering N-Series ION units in place. Applications that involve N-Series ION units mated to a supporting plate should take special care to insure that the solder joints are not stressed by the supporting plate once installed. The recommended method to achieve this is to mechanically mate the unit to the supporting plate before soldering into the PCB. If, for whatever reason, this is not possible, then special care should be taken to insure that the N-Series ION is aligned with the supporting plate after soldering and before mechanical attachment so that upon mechanical attachment no stress is placed on the ION unit, the solder contacts, or the PCB.

Soldering Process. Either a manual or reflow soldering process can be used. The recommended reflow soldering temperature specification is 300 deg C (1.5mm for 10 sec).

No-wash. N-Series IONs should not be put through a wash process during PCB manufacturing.

It is the responsibility of the user to ensure that all N-Series ION units have been installed within the prescribed mechanical stress limits and following the above described procedures. Failure to observe any of the above recommended procedures and limits may result in incorrect operation or failure of the unit during operation.



4.2 Production Unit Configuration

Each N-Series ION, before undertaking motion control, must be programmed with control and communication parameters appropriate for the application that it will be used in. These parameters include quantities such as PWM (Pulse Width Modulation) frequency, current gains, safety thresholds, node IDs, and much more. In addition, if C-Motion Engine user-code is to be downloaded into the N-Series ION unit this code memory image must be downloaded into the ION unit prior to operation in the production application.

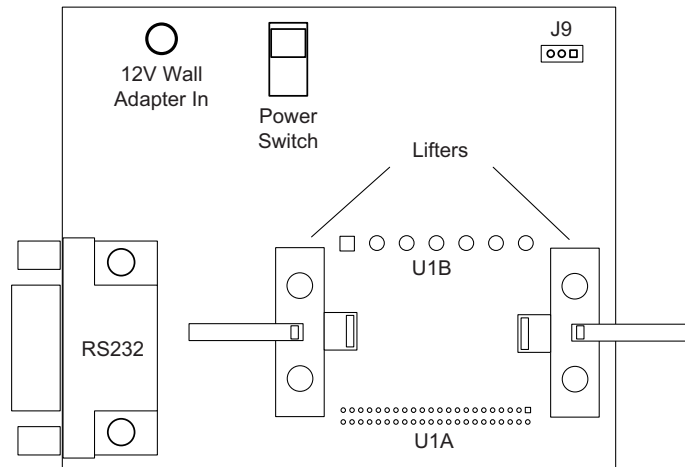
The user has several choices for how control parameters and/or C-Motion user code can be loaded into the N-Series ION, however they broadly fall into two categories; loading parameters into ION units prior to installation in the application board (also called pre-loading the ION unit), and loading parameters into ION units already installed on the user's production application board.

4.2.1 Pre-loading Parameters

There are two options for pre-loading N-Series IONs as detailed in the following sections.

**Figure 4-4:
N-Series ION
Programming
Board**

4.2.1.1 N-Series ION Programming Board



PMD provides a special programming board, similar in size and appearance to the N-Series ION DKs, to program factory default N-Series ION Devices prior to soldering onto the production PCB. This board can program all N-Series ION unit types regardless of the host interface, power level, or motor type.

This programming board, P/N Prog-NION-A, is shown in [Figure 4-4](#) and can be ordered from your local PMD representative. It connects to a host computer via a standard RS232 DB9 serial connection, includes a 12V wall plug power adapter, a power switch, and utilizes a socket to electrically connect to the N-Series ION unit.

The programming sequence is to turn the power switch off, install the ION unit into the socket, turn the power switch on, and then use Pro-Motion or a dedicate command-line executable available from PMD to program the ION unit. Once the unit is programmed, depending on the application the user may consider applying a label to identify its contents. When ready, the programmed ION unit is then soldered into the application PCB and will be ready for operation.

Pro-Motion as well as the command-line executable available from PMD utilize script files to program two different NVRAM memories on the ION unit, one referred to as the Magellan IC NVRAM, and the other the Device NVRAM. For more information on these script files refer to [Section 4.2.3, "Loading NVRAM Contents & Script Files."](#) In addition to NVRAM script files, if user code is to be loaded into the C-Motion Engine this can also be accomplished with these programs. See [Section 4.2.3.3, "Loading CME User Programs,"](#) for details.

4.2.1.2 Purchasing Pre-Loaded ION Units

Depending on the anticipated volume of your application it may be possible to order ION units already programmed for your application. This service is provided by PMD for selected customers and may also be provided by your local PMD representative. Contact your PMD representative for details.

As was the case with the programming board described above, application specific script files and user code bin files created by the customer are utilized by the programming service provider to program the units.

4.2.2 Loading Parameters After Installation onto the Application PCB

Some users will not find it convenient to pre-program the ION unit before soldering onto the user PCB and will instead program factory default IONs unit after they have been soldered onto the application PCB. There are several ways to do this as indicated in the following sections.

4.2.2.1 Programming Via a Communications Port

One approach to programming the ION unit is via a communications port. How best to communicate via a communications port to a factory default ION unit depends on the interface type and whether the host communication connections on the PCB are to a single ION unit, or to multiple units connected on a network.

The following table shows this:

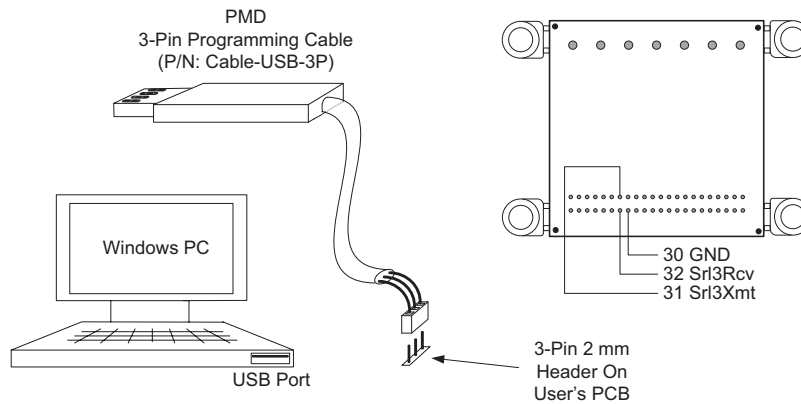
Communications Port	Single Unit	Multiple Unit Network
RS232 (Serial1)	Match comm parameters to Serial1 default values of 57,600, none, 8, 1.	N/A
RS422	Tie pin #9 of Signal Connector low to select RS422/RS485. Match comm parameters to Serial1 default values of 57,600, none, 8, 1.	N/A
RS485	N/A	There are two overall approaches, manipulation of the reset signals and daisy chain Node ID assignment. For reset-based setup tie pin #9 of Signal Connector low to select RS422/RS485. Tie all reset signals low (Signal Connector #29) except unit being configured. Use default comm parameters as for single unit. For daisy chain Node ID based setup refer to Section 3.6.1.3, "Daisy-Chain Node ID Assignment."
Host CAN	Match comm parameters to host CAN default values of 1,000,000 baud, send and receive addresses of 0x580, 0x600 respectively.	There are two overall approaches, manipulation of the reset signals and daisy chain Node ID assignment. For reset-based setup tie all reset signals low (Signal Connector #29) except unit being configured. Use default comm parameters as for single unit. For daisy chain Node ID based setup refer to Section 3.6.1.3, "Daisy-Chain Node ID Assignment."
Expansion CAN	Match comm parameters to expansion CAN default values of 1,000,000 baud with send and receive addresses of 0x580 and 0x600 respectively.	Match comm parameters to expansion CAN default values of 1,000,000 baud with send and receive addresses of 0x580 and 0x600 respectively.
Ethernet	Match comm parameters to Ethernet default values of 192.168.2.2 IP Address, 255.255.255.0 net mask, 0.0.0.0 gateway.	Tie all reset signals low (Signal Connector #29) except unit being configured. Use default comm parameters as for single unit.
Host SPI	Match comm parameters to Host SPI default values of idle low and second edge for clock polarity and clock phase respectively.	Match comm parameters to Host SPI default values of idle low and second edge for clock polarity and clock phase respectively.

** To tie reset signals low jumpers could be used, or if there is a microprocessor or FPGA on the board these can also be used to control the unit reset signals and achieve this ION isolating function.*

For multiple unit networks the connection method detailed above is typically only used to download initial control and comm settings into the unit. Those initial settings will include new network addresses, for example new node IDs for RS485 communications and IP Addresses for Ethernet, to thereafter allow each ION unit to be addressed on the network in a normal way without manipulation of reset signals.

**Figure 4-5:
3-Pin
Programming
Interface
Connection
Diagram**

4.2.2.2 Programming Via 3-Pin Connector



An alternative approach to programming the unit is to utilize the 3-pin Programming Cable (PMD P/N Cable-USB-3P). This is shown in [Figure 4-5](#). The 3-pin programming cable plugs into a PC USB port on one end and into each 3-pin connector located on the application PCB on the other. A technician plugs into the connector and performs the programming/configuration function for each ION unit on the PCB.

This approach requires that each installed N-Series ION have a 3-pin connector installed on the production board. This mating connector is a male 3-pin 2 mm header. A representative component P/N is Samtec MTMM-103-04-x-S-150. This connector should be wired as follows:

Signal Name	Header Pin #	ION Signal Connector Pin #
Srl3Xmt	1	31
Srl3Rcv	2	32
GND	3	30

4.2.2.3 Programming Via Microcontroller

An alternative to cable-connected programming via Serial3 is to design a small microcontroller on the PCB for each or all installed N-Series IONs. This microcontroller connects through a UART interface to the ION's Serial3 interface. There are numerous microcontroller products that can serve this function, for example various types of PIC processors, many costing less than \$1 and taking up as little as 10 mm² of board space.

For this scheme to work the microcontroller, whose sole function is programming the N-Series ION unit, must contain a software program that can transfer the ION configuration data, also stored inside the microcontroller to the ION unit.

This microcontroller is programmed using standard IC programming equipment and then soldered onto the board. The microcontroller is set for the Serial 3 default comm parameters of 57,600, none, 8, 1 and as soon as power is provided to both the microcontroller and the N-Series ION the microcontroller can begin downloading the configuration information into the ION unit.

4.2.3 Loading NVRAM Contents & Script Files

When initializing N-Series ION units from a microcontroller, or from a host interface that doesn't require reset signal jumper changes, it is possible to send the active control parameters at each power up. If this is the case then there may be no need to store control parameters into the ION unit's NVRAM buffers.

For most applications however there will be a first-time setup, as described in the previous sections, which stores control parameter settings along with a user code CME block if being used, into NVRAM, thereby allowing the ION unit to initialize and operate on its own.

In total there are three separate memory areas that may be loaded with such NVRAM startup information as summarized in the table below:

NVRAM Area	Typical Programming Format	Description
Magellan IC NVRAM	PSF Script	The Magellan IC contains its own NVRAM which may be used to initialize Magellan IC control parameters, and control the startup timing of Magellan motor output.
Device SetDefaults NVRAM	Pro-Motion Command Window Script	The Device SetDefaults NVRAM stores parameter settings not directly related to Magellan IC functions. These include host communication settings, pin mux settings, LED mode settings, and more.
CME Program Storage	bin file	The CME program NVRAM holds the memory image of compiled user code which will run on the ION's C-Motion Engine.

In the following sections we will provide more information on loading each of these memory areas on the ION unit.

4.2.3.1 Loading the Magellan IC NVRAM

Magellan IC NVRAM command sequences use a storage format known as PSF (PMD Structured data Format). The associated PSF script files are ASCII, and can be edited by any standard text editor. In addition to holding the stored command sequences, PSF can record information to help identify the contents of the stored data. This is convenient when retrieving PSF content from the Magellan IC to keep track of multiple versions of particular command sequence. An example PSF script file is shown below:

```
#ScriptVersion 1
:DESC "Motor 2 settings"
:CVER 1.3
SetMotorType 0
SetPhaseCounts 2000
SetPhaseInitializeMode 1      ' Hall based
SetSignalSense 0x0001
SetCurrentControlMode 1      ' Field oriented control
SetFOC 512 680                ' Kp for both d & q loops
SetFOC 513 28                 ' Ki for both d & q loops
SetFOC 514 100 0x7FFF         ' integral limit for both d & q loops
SetOperatingMode 0x0007      ' motor output and current loop on
InitializePhase
```

Figure 4-6:
Example PSF
Script File
Content

The heart of PSF scripts are the C-Motion command mnemonics and corresponding argument values representing the commands to execute during initialization. For detailed information on Magellan IC commands refer to the *Magellan Motion Control IC Programming Reference*.

Some additional information on PSF script files that may be useful:

- PSF script files contain ASCII characters and terminate each line with a <CR> <LF> sequence (carriage return and line feed).
- Each line of the script file contains a single complete command mnemonic.
- An apostrophe ' is used to indicate that the characters from the apostrophe to the end of the line are a comment.

- Pro-Motion always inserts as the first entry “#ScriptVersion” indicating the script file version format that the file was stored in. This entry is not stored into the Magellan IC but allows past and future versions of Pro-Motion to interpret the script file correctly.
- Numbers default to decimal interpretation but can be forced for hexadecimal interpretation by preceding with “0x”.
- Script file lines that begin with a colon “:” record PSF identifiers such as the version #, user-provided content description, creation date, and other information. Users may edit these lines directly but should only do so if they are familiar with PSF.

For additional information on the Magellan IC NVRAM refer to [Section 3.2.2.2, “Magellan IC NVRAM.”](#)

For additional information on PMD Structured data Format refer to the *Juno Velocity & Torque Control IC Programming Reference*.

4.2.3.2 Loading the Device NVRAM

In addition to the Magellan IC’s NVRAM, each N-Series ION has an NVRAM known as the Device NVRAM. A portion of this NVRAM is dedicated to storing **DeviceSetDefault** command values, which hold the default settings for various parameters. For more information refer to [Section 3.10.7, “Setting Default Values.”](#)

While **DeviceSetDefault** commands can be sent through the host interface or even executed by on-board CME user code, for the purpose of production configuration use of a script file is the most convenient approach for setting these ION unit default values. This type of script file, known as a Pro-Motion command window script, has a somewhat different format than a PSF script file but is similar in that both script file types use ASCII characters. Command window scripts however do not allow information about the script version or content to be retrieved from the ION unit’s NVRAM.

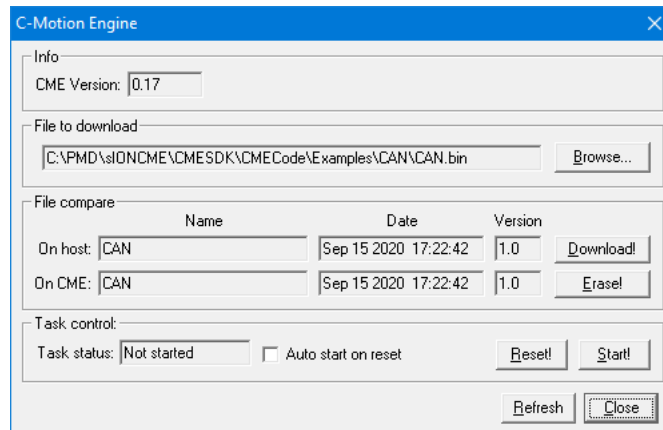
Command window scripts may be used to record any sequence of C-Motion commands processable by the ION unit, not just **DeviceSetDefault** commands.

Some additional information on Command Window script files that may be useful:

- Pro-Motion script files contain ASCII characters and terminate each line with a <CR> <LF> sequence (carriage return and line feed).
- Each line of the script file contains a single complete command mnemonic.
- An apostrophe ' is used to indicate that the characters from the apostrophe to the end of the line are a comment.
- Numbers default to decimal interpretation but can be forced for hexadecimal interpretation by preceding with “0x”.

In the readme.txt file that comes with all PMD SDKs a file name is provided which contains a pre-formatted script for N-Series IONs containing the factory default settings for every settable **DeviceSetDefault** parameter. To change the SetDefaults NVRAM content edit this file with new values and store with a unique file name representing the specific configuration label you would like to use. Then go to Pro-Motion’s Command Window and type “<” and the file name to execute the script. You will see the command contents of the script output in the command window. Alternatively you can go to the menu item File/LoadScript.

4.2.3.3 Loading CME User Programs



C-Motion Engine User programs that have been developed and are ready to load into the ION unit can be sent to the ION unit from the Pro-Motion "C-Motion Engine" window.

This window allows you to specify the program file name you want to send, the version # of the program to be specified, as well as whether the code should auto-execute or not. Once programmed into the ION these parameters can be read back from a programmed ION unit to determine whether the units' user code content matches what was expected.

This page intentionally left blank.

Appendix A. Scale Factors, Defaults and Limits

A

In This Appendix

- ▶ Power Level Related
- ▶ N-Series ION NVRAM Default Settings

A.1 Power Level Related

To correctly control various power level related N-Series ION features it is helpful to know certain drive-specific scale factors, defaults and limits. The following tables summarize these values.

A.1.1 Conversion Factors, Low Power Units

The following table provides electrical conversion factors for low power level N-Series ION units (P/Ns: DD4X1X0056/02).

These factors convert various integer Magellan IC command arguments (referred to as having units of counts) to physical quantities such as amperage, volts, etc... For more information on the Magellan Motion Control IC refer to the *Magellan Motion Control IC User Guide*. For more information on C-Motion commands refer to the *C-Motion PRP II Programming Reference*.

Unit	Example C-Motion Commands	Scaling	Example usage
Amps	GetCurrentLoopValue	.231 mA/count *	A command request to read the ActualCurrent parameter returns a value 12,345. This corresponds to a current of 12,345 counts * 0.231 mA/count = 2.851A.
Volts	SetDriveFaultParameter GetBusVoltage	10.0 mV/count	To set an overvoltage threshold of 50V, the command value should be 50 V * 1,000 mV/V / 10.0 mV/count = 5,000.
Temperature	SetDriveFaultParameter GetTemperature	.00391 °C/count	To set an overtemperature threshold of 65 °C, the command value should be 65 °C / .0039 °C/count = 16,624.
I ² t continuous current	SetCurrent	.231 mA/count	To set an I ² t continuous current of 1.0A, the command value should be 1.0A * 1,000 mA/A / 0.231 mA/count = 4,329.
I ² t energy	SetCurrent	.00588 A ² Sec/ count	To set an I ² t energy of 5.0 A ² Sec, the command value should be 5.0 A ² Sec / 0.00588 A ² Sec/count = 850.

* Some C-Motion commands specifying a current may have a scaling of two times this value. Refer to the *Magellan Motion Control IC User Guide* for more information.

A.1.2 Defaults & Limits, Low Power Units

The following table provides default values, low limits and high limits for various specifiable drive-related parameters for low power level N-Series ION units (P/Ns: DD4X1X0056/02).

Setting	Default Setting	Low Limit	High Limit
Overtemperature limit	75.0 °C	0 °C	75.0 °C
Overtemperature hysteresis	5.0 °C	0 °C	25.0 °C
Overvoltage limit	60.0 V	10.0 V	60.0 V
Undervoltage limit	10.0 V	10.0 V	56.0 V
I^2t continuous current limit, Brushless DC motor	2.1 A	0.0 A	2.1 A
I^2t continuous current limit, DC Brush motor	1.5 A	0.0 A	1.5 A
I^2t continuous current limit, step motor	2.1 A	0.0 A	2.1 A
I^2t energy limit, Brushless DC Motor	2.95 A ² sec	0.0 A ² sec	2.95 A ² sec
I^2t energy limit, DC Brush Motor	3.63 A ² sec	0.0 A ² sec	3.63 A ² sec
I^2t energy limit, step motor	2.95 A ² sec	0.0 A ² sec	2.95 A ² sec
Commandable current	N/A	-3.8 A	+3.8 A
Maximum readable current	N/A	N/A	6.0 A

A.1.3 Conversion Factors, Medium Power Units

The following table provides electrical conversion factors for medium power level N-Series ION units (P/Ns: DD4X1X0056/06).

These factors convert various integer Magellan IC command arguments (referred to as having units of counts) to physical quantities such as amperage, volts, etc... For more information on the Magellan motion control IC refer to the *Magellan Motion Control IC User Guide*. For more information on C-Motion commands refer to the *C-Motion PRP II Programming Reference*.

Unit	Example C-Motion Commands	Scaling	Example usage
Amps	GetCurrentLoopValue	.733 mA/count*	A command request to read the ActualCurrent parameter returns a value 12,345. This corresponds to a current of 12,345 counts * 0.733 mA/count = 9.049.
Volts	SetDriveFaultParameter GetBusVoltage	10.0 mV/count	To set an overvoltage threshold of 50V, the command value should be $50 \text{ V} * 1,000 \text{ mV/V} / 10.0 \text{ mV/count} = 5,000$.
Temperature	SetDriveFaultParameter GetTemperature	.0039 °C/count	To set an overtemperature threshold of 65 °C, the command value should be $65 \text{ °C} / .0039 \text{ °C/count} = 16,667$.
I ² t continuous current	SetCurrent	.733 mA/count	To set an I ² t continuous current of 4.0A, the command value should be $4.0\text{A} * 1,000 \text{ mA/A} / 0.733 \text{ mA/count} = 5,457$.
I ² t energy	SetCurrent	.0590 A ² Sec/count	To set an I ² t energy of 20.0 A ² Sec, the command value should be $20.0 \text{ A}^2\text{Sec} / 0.0590 \text{ A}^2\text{Sec/count} = 339$.

* Some C-Motion commands specifying a current may have a scaling of two times this value. Refer to the *Magellan Motion Control IC User Guide* for more information.

A.1.4 Defaults & Limits, Medium Power Units

The following table provides default values, medium limits and high limits for various specifiable drive-related parameters for low power level N-Series ION units (P/Ns: DD4X1X0056/06).

Setting	Default Setting	Low Limit	High Limit
Overtemperature limit	75.0 °C	0 °C	75.0 °C
Overtemperature hysteresis	5.0 °C	0 °C	25.0 °C
Overvoltage limit	60.0 V	10.0 V	60.0 V
Undervoltage limit	10.0 V	10.0 V	56.0 V
I^2t continuous current limit, Brushless DC motor	7.8 A	0.0 A	7.8 A
I^2t continuous current limit, DC Brush motor	7.1 A	0.0 A	7.1 A
I^2t continuous current limit, step motor	7.1 A	0.0 A	7.1 A
I^2t energy limit, Brushless DC Motor	25.1 A ² sec	0.0 A ² sec	25.1 A ² sec
I^2t energy limit, DC Brush Motor	28.1 A ² sec	0.0 A ² sec	28.1 A ² sec
I^2t energy limit, step motor	28.2 A ² sec	0.0 A ² sec	28.2 A ² sec
Commandable current	N/A	-12.0 A	+12.0 A
Maximum readable current	N/A	N/A	20.0 A

A.1.5 Conversion Factors, High Power Units

The following table provides electrical conversion factors for high power level N-Series ION units (P/Ns: DD4X1X0056/18).

These factors convert various integer Magellan IC command arguments (referred to as having units of counts) to physical quantities such as amperage, volts, etc... For more information on the Magellan Motion Control IC refer to the *Magellan Motion Control IC User Guide*. For more information on C-Motion commands refer to the *C-Motion PRP II Programming Reference*.

Unit	Example C-Motion Commands	Scaling	Example usage
Amps	GetCurrentLoopValue	2.198 mA/count *	A command request to read the ActualCurrent parameter returns a value 12,345. This corresponds to a current of 12,345 counts * 2.198 mA/count = 27.13A.
Volts	SetDriveFaultParameter GetBusVoltage	10.0 mV/count	To set an overvoltage threshold of 50V, the command value should be 50 V * 1,000 mV/V / 10.0 mV/count = 5,000.
Temperature	SetDriveFaultParameter GetTemperature	.0039 °C/count	To set an overtemperature threshold of 65 °C, the command value should be 65 °C / .0039 °C/count = 16,667.
I ² t continuous current	SetCurrent	2.198 mA/count	To set an I ² t continuous current of 15.0A, the command value should be 15.0A * 1,000 mA/A / 2.198 mA/count = 6,824.
I ² t energy	SetCurrent	.531 A ² Sec/count	To set an I ² t energy of 100.0 A ² Sec, the command value should be 100 A ² Sec / .531 A ² Sec/count = 188.

* Some C-Motion commands specifying a current may have a scaling of two times this value. Refer to the *Magellan Motion Control IC User Guide* for more information.

A.1.6 Defaults & Limits, High Power Units

The following table provides default values, high limits and high limits for various specifiable drive-related parameters for low power level N-Series ION units (P/Ns: DD4X1X0056/18).

Setting	Default Setting	Low Limit	High Limit
Overtemperature limit	75.0 °C	0 °C	75.0 °C
Overtemperature hysteresis	5.0 °C	0 °C	25.0 °C
Overvoltage limit	60.0 V	10.0 V	60.0 V
Undervoltage limit	10.0 V	10.0 V	56.0 V
I ² t continuous current limit, Brushless DC motor	20.9 A	0.0 A	20.9 A
I ² t continuous current limit, DC Brush motor	19.0 A	0.0 A	19.0 A
I ² t continuous current limit, step motor	18.9 A	0.0 A	18.9 A
I ² t energy limit, Brushless DC Motor	257.4 A ² sec	0.0 A ² sec	257.4 A ² sec
I ² t energy limit, DC Brush Motor	280.5 A ² sec	0.0 A ² sec	280.5 A ² sec
I ² t energy limit, step motor	281.0 A ² sec	0.0 A ² sec	281.0 A ² sec

Setting	Default Setting	Low Limit	High Limit
Commandable current	N/A	-36.0 A	+ 36.0 A
Maximum readable current	N/A	N/A	60.1 A

For the N-Series ION, default values and limits for the I^2t continuous current limit and I^2t energy limit are designed to be safe for operation in the drive's highest output mounting option, namely, using a cold plate for heat sinking. See [Section 4.1, "Mechanical Mounting,"](#) for information on N-Series ION mounting options.

If the N-Series ION drive is being operated at a lower voltage it may be possible to specify values for I^2t continuous current limit and I^2t energy limit that are higher than the default, but lower than or equal to the limit, since the continuous output current rating of the N-Series ION drive is higher for lower input voltages. See [Section 2.3, "Drive Ratings,"](#) for drive output specifications.

For other mounting configurations, or for use with motors that have lower current and energy limits, it may be useful to set these parameters to values lower than the default values.



It is the responsibility of the user to set the I^2t continuous current and I^2t energy limit parameters to values that are safe for the specific N-Series ION mounting configuration and motor setup being used.

A.2 N-Series ION NVRAM Default Settings

There are various parameters that are stored in the N-Series ION's non-volatile RAM and which become active after a power-on or a reset. These parameters are referred to as default values.

Default values come factory programmed to specific values but can also be re-programmed by the user. If re-programmed by the user default values will not take affect until the next unit power up or reset. Default values are stored in the N-Series ION's Device NVRAM area. As such, writing and reading operations to the default value area involves special considerations. For more information refer to [Section 3.11, "Device NVRAM."](#)



In addition to the N-Series ION's Device NVRAM area the Magellan IC has its own NVRAM area that may be used to store application specific parameters such as servo gains, PWM frequency, and other settings managed by the motion control IC. For more information refer to [Section 3.2.2.2, "Magellan IC NVRAM."](#)

The **DeviceSetDefault** command is used to program NVRAM parameter values. This command takes arguments of DefaultCode which identifies which parameter is being programmed, DefaultValue which specifies the value to program, and the DefaultValue size in bytes. For detailed information on this command refer to the *C-Motion PRP II Programming Reference*.

A.2.1 PRP Communication Port Settings

The following table lists settable NVRAM parameters, factory default values, and parameter formats related to PRP communication settings for various N-Series ION ports.

Parameter Name	Default Code	Factory Value	Parameter Format
DefaultIPAddress	0x303	0xC0A80202	Four-byte word encoding the N-Series ION's Ethernet IP address. Factory default value corresponds to IP address 192.168.2.2.
DefaultNetMask	0x304	0xFFFFFFFF00	Four-byte word encoding the N-Series ION's net-mask, which is a bitmask that defines which IP addresses are directly accessible to the local subnet.
DefaultGateway	0x305	0x00000000	Four-byte word encoding the IP address of the router to be used for all non-local IP addresses. Factory default value corresponds to IP address 0.0.0.0.
DefaultTCPPort	0x106	40100 (dec)	Two-byte word encoding the base TCP port used for accepting host commands.
DefaultDHCPAttempts	0x0120	2	Two-byte word which specifies the maximum number of attempts that will occur to obtain the IP configuration from the DHCP server. The maximum allowed value for this setting is 65,535.
DefaultSerial1Mode	0x10E	4	Two-byte bit-encoded word which specifies the operating settings for Serial1 port when used in RS232 mode. The factory default value of 4 corresponds to 57,600 baud, no parity, 1 stop bit, and point-to-point protocol. For a detailed description of the encoding of this word and the other serial mode default settings in this table refer to the Open Serial Device PRP command in the <i>C-Motion PRP II Programming Reference</i> .
DefaultSerial1ModeRS485	0x124	0x0084	Two-byte bit-encoded word which specifies the operating settings for Serial1 port when used in RS485 mode. The factory default value of 0x0084 corresponds to 57,600 baud, no parity, 1 stop bit, multi-drop protocol, and a multi-drop address setting of 0.
DefaultSerial2Mode	0x10F	5	Two-byte bit-encoded word which specifies the operating settings for Serial2 port. The factory default value of 5 corresponds to 115,200 baud, no parity, 1 stop bit, point-to-point protocol, and a multi-drop address setting of 0.
DefaultSerial3Mode	0x121	4	Two-byte bit-encoded word which specifies the operating settings for Serial3 port. The factory default value of 4 corresponds to 57,600 baud, no parity, 1 stop bit, point-to-point protocol, and a multi-drop address setting of 0.

Parameter Name	Default Code	Factory Value	Parameter Format
DefaultRS485Duplex	0x110	0	Two-byte bit-encoded word which specifies RS485 modes. The factory setting corresponds to full duplex, 120 ohm not enabled, RS232 mode, and DIO8 pin not overridden. The encoding of this word is as follows: Bit 0 specifies half-duplex (2-wire) enable. A value of 0 specifies full-duplex and a value of 1 specifies half-duplex. Bit 4 is RS485/422 120 ohm termination enable. A value of 1 enables the 120 ohm terminator resistor across the Tx and Rx differential pairs. Bit 8 sets RS485 or RS232 mode. A value of 1 sets the serial port to RS485 mode, otherwise RS232. This value is only applicable when bit 12 is set to 1. Bit 12 is DIO8 pin override. A value of 1 forces the serial port to the mode set by the PMD_RS485Mode_RS485_232 bit on power up regardless of the state of DIO8 pin.
DefaultHostCANMode	0x112	0	Two-byte bit-encoded word which specifies the operating settings for the ION's Host CAN port parameters. The factory setting corresponds to a NodeID of 0 and a baud rate of 1,000,000. The encoding of this word is as follows: bits 0-6 hold a 7-bit integer representing the NodeID from 0 to 127, and bits 13-15 specify the baud rate with encoded values of 0 through 5 corresponding to selected rates of 1,000,000, 800,000, 500,000, 250,000, 125,000, and 50,000 baud.
DefaultCANMode	0x111	0	Two-byte bit-encoded word which specifies the operating settings for the ION's Expansion CAN port parameters. The factory setting corresponds to a NodeID of 0 and a baud rate of 1,000,000. See DefaultHostCANMode description command for bit encoding of this word.

A.2.2 C-Motion Engine Settings

The following table lists settable NVRAM parameters, factory default values, and parameter formats related to the N-Series ION's C-Motion Engine settings.

Parameter Name	Default Code	Factory Value	Parameter Format
DefaultAutoStartMode	0x114	0	Two-byte word specifying whether a user program loaded into the C-Motion Engine will be run automatically after coming out of reset. The factory default value is not run automatically. A 0 setting of this parameter indicates not to start a loaded user program automatically while a 1 indicates to start automatically.
DefaultConsoleIntfType	0x118	0	Two-byte word specifying what port type will be used for console traffic. The factory default value is none. The allowed settings for this parameter are 0, 4, 7, and 9 specifying none, serial, UDP, and PRP respectively.

Parameter Name	Default Code	Factory Value	Parameter Format
DefaultConsoleIntfAddr	0x119	0xC0A80201	Four-byte word specifying the address to be used with the specified port console type. The factory setting specifies a UDP IP address setting of 192.168.2.1. The allowed settings for this parameter when the console type is serial are 0 and 1 specifying Serial1 or Serial2 respectively. The allowed settings for this parameter when the console type is UDP is a four-byte IP address in 'dotted quad' notation.
DefaultConsoleIntfPort	0x11A	0	Two-byte word specifying the port settings to be used with the specified port console type. The factory setting specifies no port settings. The allowed settings for this parameter when the console type is serial are formatted the same as for the DefaultSerial1Mode parameter settings described in Section A.2.1, "PRP Communication Port Settings." The allowed settings when the console type is UDP is a UDPPort which can have a value of 0 to 0xffff. See the Open UDP Device PRP command description in the <i>C-Motion PRP II Programming Reference</i> for more information.
DefaultTaskParam	0x31E	0	Four-byte word which specifies the parameter to pass to the main user task upon startup.

A.2.3 Digital I/O Settings

The following table lists settable NVRAM parameters, factory default values, and parameter formats related to the NSeries ION's Digital I/O signal settings.

Parameter Name	Default Code	Factory Value	Parameter Format
DefaultDIOMux	0x0125	2	Two-byte word which specifies the value loaded into the 0x228 address PIO register which determines the N-Series ION's Digital I/O pin MUX signal settings.
DefaultDIODir	0x0126	2	Two-byte word which specifies the value loaded into the 0x222 address PIO register which determines the direction of Digital I/O signal pins.
DefaultDIOOut	0x0127	2	Two-byte word which specifies the value loaded into the 0x212 address PIO register which determines the output state settings of Digital I/O signal pins.

A.2.4 BiSS-C & SSI Encoder Settings

The following table lists settable NVRAM parameters, factory default values, and parameter formats related to the NSeries ION's BiSS-C & SSI encoder settings.

Parameter Name	Default Code	Factory Value	Parameter Format
DefaultBiSSConfig	0x0129	2	Two-byte word which specifies the value loaded into the 0x100 address BiSS-C & SSI configuration PIO register.
DefaultBiSSResolution	0x012A	2	Two-byte word which specifies the value loaded into the 0x102 address BiSS-C & SSI resolution PIO register.

Parameter Name	Default Code	Factory Value	Parameter Format
DefaultBiSSFrequency	0x012B	2	Two-byte word which specifies the value loaded into the 0x104 address BiSS-C & SSI frequency PIO register.
DefaultBiSSEnable	0x012C	2	Two-byte word which specifies the value loaded into the 0x106 address BiSS-C & SSI Enable PIO register.
DefaultBiSSSingleTurn	0x012D	2	Two-byte word which specifies the value loaded into the 0x108 address BiSS-C single turn resolution and width PIO register.
DefaultBiSSMultiTurn	0x012E	2	Two-byte word which specifies the value loaded into the 0x10A address BiSS-C multi turn resolution and width PIO register.
DefaultBiSSRightBitShift	0x012F	2	Two-byte word which specifies the value loaded into the 0x10C address BiSS-C position shift amount PIO register.

Appendix B. BiSS-C Encoder Settings

B

In This Appendix

- ▶ Renishaw Resolute Absolute Encoder, Linear
- ▶ Renishaw Resolute Absolute Encoder, Angular
- ▶ Parker Hannifin P-Series AC Servo Motor with BiSS-C Feedback

To facilitate using specific encoders or motor /encoder combinations with the N-Series ION Digital Drive this appendix provides register settings for several popular BiSS-C Encoders or motor/encoder combinations. For detailed information on the user settable BiSS-C registers refer to [Section 3.3.4.2, "Interface Control Registers."](#)

B.1 Renishaw Resolute Absolute Encoder, Linear

P/N: RL32BBS050C30A

Here are the BiSS-C register settings that should be used with this encoder type:

Parameter	PIO Register Address	Setting	Comments
CRC bits	0x100	6	
Encoder Resolution	0x102	32	
Clock Frequency	0x104	3	Corresponds to 10.0 MHz
Single-turn resolution	0x108	32	
Single-turn width	0x108	34	
Multi-turn resolution	0x10A	0	
Multi turn width	0x10A	0	
Shift	0x10C	2	

B.2 Renishaw Resolute Absolute Encoder, Angular

P/N: RA26BBA150B10A

Here are the BiSS-C register settings that should be used with this encoder type:

Parameter	PIO Register Address	Setting	Comments
CRC bits	0x100	6	
Encoder Resolution	0x102	26	
Clock Frequency	0x104	3	Corresponds to 10.0 MHz
Single-turn resolution	0x108	26	
Single-turn width	0x108	26	
Multi-turn resolution	0x10A	0	
Multi turn width	0x10A	0	
Shift	0x10C	2	

B.3 Parker Hannifin P-Series AC Servo Motor with BiSS-C Feedback

P/N: PM-FBL02AMK

Here are the BiSS-C register settings that should be used with this motor/encoder type:

Parameter	PIO Register Address	Setting	Comments
CRC bits	0x100	6	
Encoder Resolution	0x102	32	
Clock Frequency	0x104	3	Corresponds to 10.0 MHz
Single-turn resolution	0x108	19	
Single-turn width	0x108	21	
Multi-turn resolution	0x10A	16	
Multi turn width	0x10A	16	
Shift	0x10C	2	

Appendix C. Application Notes & Example Schematics

C

In This Appendix

- ▶ General Design Notes
- ▶ Power Supplies
- ▶ RS232 Host Interface and Programming Port
- ▶ RS485 Host Interface
- ▶ CAN Host Interface
- ▶ SPI Host Interface
- ▶ Ethernet Host Interface
- ▶ Expansion CAN Network and Motor Connection Setup
- ▶ Expansion SPI Network
- ▶ Connecting to an MC58113 IC
- ▶ Digital IO Connection Schemes
- ▶ Analog Input

C.1 General Design Notes

This section provides general information intended to be useful for the design and development of a dedicated N-Series ION interconnect PCB or a full-fledged PCB with active circuitry elements beyond just wiring connections for N-Series IONs.

Power, Signal, and Alignment Pins

The N-series ION has 7 power pins, 44 signal pins, and 4 alignment pins. The alignment pins have no electrical connection to any internal or external element of the N-Series ION and therefore can not be used as a case ground.

The following table provides the pin dimensions and recommended PCB drill hole and pad dimensions for the power, signal, and alignment pins. For detailed mechanical drawings of the N-Series ION unit refer to [Section 2.1, "Physical Dimensions."](#)

Pin Type	PCB Via Drill Diameter	PCB Pad Shape	PCB Pad Dimensions
Power	0.067" (1.7 mm)	Circle	0.110" (2.8 mm)
Signal	0.025" (0.64 mm)	Ellipse*	0.042" (1.1 mm) x 0.035" (0.89 mm)
Alignment	0.067" (1.7 mm)	Circle	0.110" (2.8 mm)

* The 40 Signal pins are arranged as 2 rows of 20 pins. Recommended pad shape is oblong with long axis of the pad shape oriented perpendicular to the long axis of the 2x20 connector. This allows multiple traces to be routed between signal connector pins. Refer to [Appendix D, Example Interconnect Board Layout](#), for examples of this.

PCB Layers & Traces

For PCBs that function only as an interconnect board for N-Series IONs without additional active circuitry four layers is recommended. For PCBs that also have active circuitry on the board, particularly high density parts such as microcontrollers, FPGAs, etc... six, eight, or even more layers is common.

Use of four layers versus two is recommended for a number of reasons, but especially because it allows greater use of ground planes. Ground planes on one or more PCB layers reduce EMI and minimize cross talk between the high current, high voltage portions of the PCB (the traces connected to the Power Connector of the N-Series ION) and the lower voltage, more noise sensitive portions of the PCB (the traces connected to the Signal Connector of the N-Series ION). For additional comments on grounding recommendations see [Section C.2, "Power Supplies."](#)

In addition to ground planes, especially for the Power Connector signals, consideration should be given to the trace dimensions. Current carrying capacity of traces and associated issues such as thermal rise in the trace are determined by a combination of the PCB copper thickness, the trace width, and the trace length. Although beyond the scope of this document it is possible to calculate the thermal rise and increase in resistance that occurs in each trace based on the dimensions and current flow through it.

For a complete example four layer interconnect board layout which demonstrates use of ground planes, PCB layer thickness, and trace dimensions refer to [Appendix D, Example Interconnect Board Layout](#).

C.2 Power Supplies

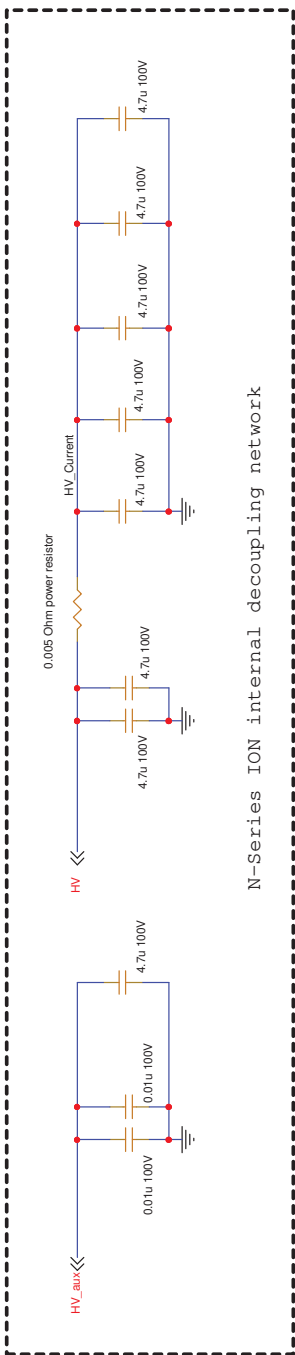
The N-Series ION is powered by two external connections; the motor supply HV which powers the N-Series ION's switching motor amplifier and the auxiliary supply HV Aux which powers the unit's internal control circuitry. Many applications will use a single DC supply by connecting HV Aux to HV. If powering separately, HV Aux may be powered with HV not powered, but HV should not be powered unless HV Aux is also powered.

The DC supply voltage range is 12-56V.

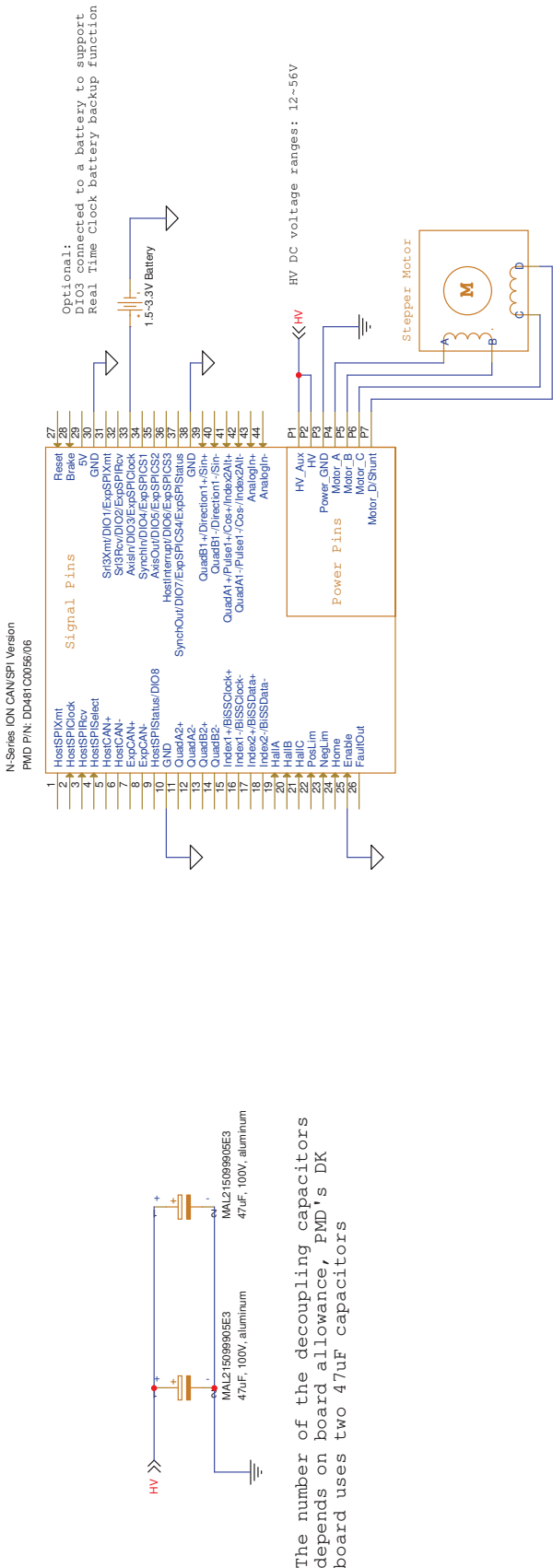
In the schematic on the facing page the ground pin on the N-series power connector has been given a unique signal name (Power_GND) to distinguish it from the signal GND pins. Power_GND and GND are connected inside the N-Series ION but should be segregated at a layout level. Power_GND, the current return path for the motor power train, is paired with HV and may therefore be noisy. GND is the reference for digital control signals and these signals require a quiet ground reference. Typically the layout should isolate Power_GND and GND as shown in the DK board layout below. Doing so will keep the noise on the power train from the rest of the digital circuits and improve noise immunity. Certain scenarios may benefit from connecting Power_GND and GND which is at the discretion of the designer.

This schematic also shows the N-Series ION power connection and internal decoupling schematic. There are two 4.7 μ F decoupling capacitors for HV, and five 4.7 μ F capacitors for HV_Current for a total DC bus decoupling capacitance of 32.9 μ F. To have better decoupling performance, extra capacitance may be added depending on the application. For example two 47 μ F 100V capacitors were used on the N-Series ION developer kit PCB for HV decoupling.

Finally, N-Series ION supports a real time clock battery backup function by connecting pin 33 of the signal connector (DigitalIO3) to a battery. The battery will power ION's real time clock circuitry when HV power supply is turned off. For applications that do not use real time clock battery backup DigitalIO3 can be used for one of its other available functions.



N-Series ION internal decoupling network



The number of the decoupling capacitors depends on board allowance, PMD's DK board uses two 47uF capacitors

Performance Motion Devices, Inc

N-Series ION Power Supplies

Size Document Number

Revision 1

Date: Thursday, December 08, 2022 Sheet 1 of 14

Figure C-1: N-Series ION Power Supply Connections

C.3 RS232 Host Interface and Programming Port

The Serial interface in the serial version N-Series ION supports either two RS232 interfaces or one RS485 interface. The RS485 or RS232 physical mode is selected by the RS485Sel pin state or by the PMD_Default_RS485Mode setting. To select RS232 mode, RS485Sel is tied high or left unconnected.

In RS232 mode there are two serial ports. Serial port 1 processes PRP communications on power-up but can be repurposed by user code. The second serial port (Serial port 2) can be used for debug console output or general-purpose by the user code.

All N-Series ION units, regardless of host interface type, support a serial connection known as the programming port on Serial3. Serial3 provides UART functionality and has TTL level signals only. The Serial3 interface consists of two signals: Srl3Xmt and Srl3Rcv.



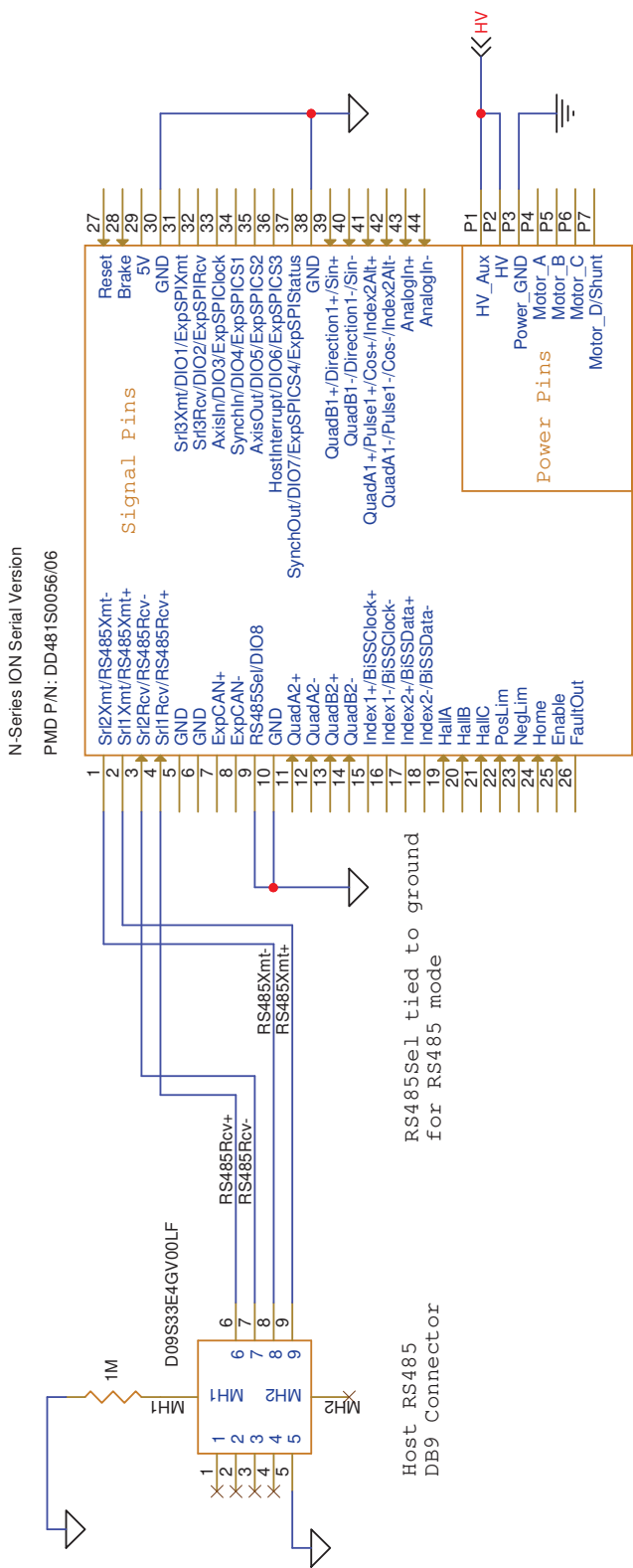
<i>Performance Motion Devices, Inc</i>					
Title	N-Series ION RS232 Host Interface and Programming Port				
Size Custom 1	Size Document Number	Rev A			
		Released by: December 09, 2000			
		Sheet 1 of 1			

**Figure C-2:
N-Series ION
RS232 Host
Interface and
Programming
Port**

C.4 RS485 Host Interface

The schematic in the facing page illustrates the serial version N-Series ION host communication via the RS485 ports.

To select RS485 mode RS485Sel is tied to ground.



Performance Motion Devices, Inc

Title N-Series ION RS485 Host Interface and Connector

Size A Document Number 1

Rev A

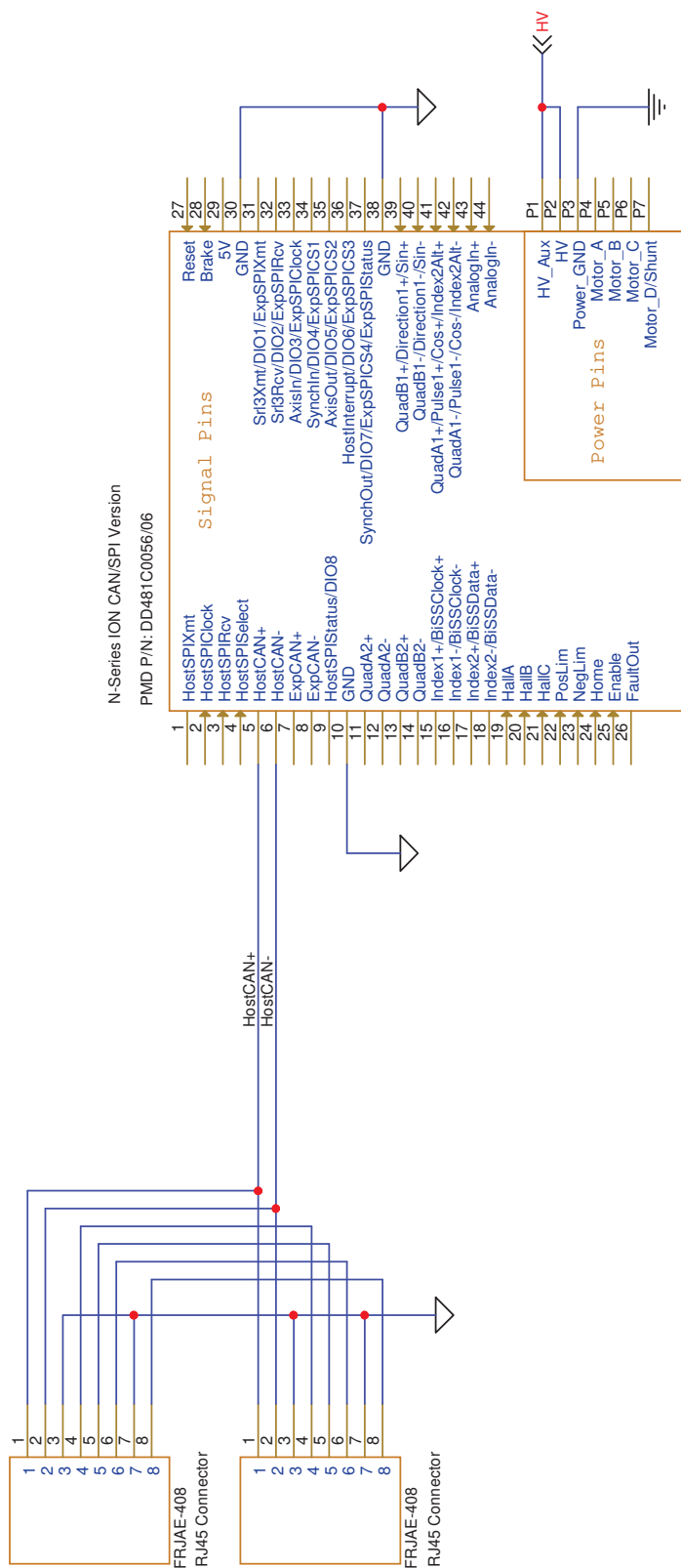
Date: Thursday, December 08, 2022 Sheet 3 of 14

Figure C-3:
N-Series ION
RS485 Host
Interface

C.5 CAN Host Interface

CAN/SPI type host interface N-Series IONs support a host interface CAN FD port. The host CAN interface consists of two signals: HostCAN+ and HostCAN-. The facing page illustrates host communication via the host CAN port.

If the N-Series ION is at the end of CAN bus network, a 120 Ohm termination resistor is required between HostCAN+ and HostCAN-.



A 120 Ohm termination resistor between HostCAN+ and HostCAN- is required if the unit is at the end of CAN bus network

Performance Motion Devices, Inc			
Title N-Series ION CAN Host Interface and Connectors			
Size A	Document Number 1	Rev A	
Date: Thursday, December 08, 2022	Sheet 4	of 14	

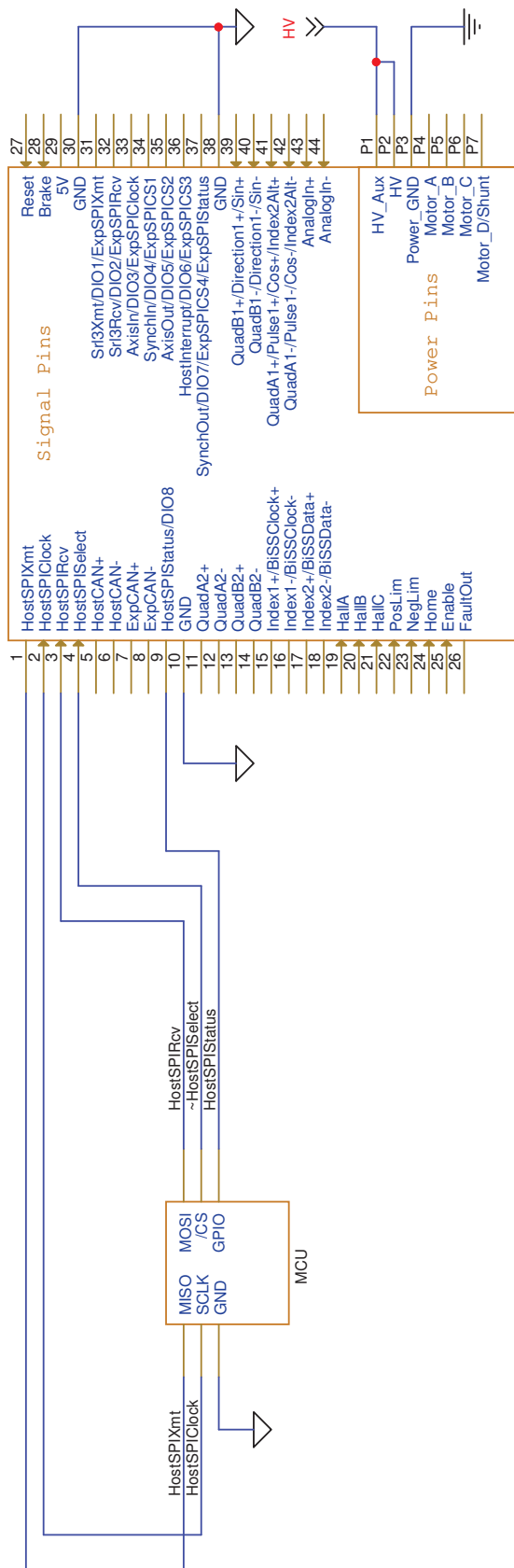
Figure C-4:
N-Series ION
CAN Host
Interface

C.6 SPI Host Interface

N-Series IONs that are of CAN/SPI host interface type provide an SPI (Serial Peripheral Interface) port for host communications. This SPI port operates as an SPI slave, and is most often used as a PRP-formatted connection from another N-Series ION or from an on-board microprocessor.

The host SPI interface consists of four signals: HostSPISelect, HostSPIXmt, HostSPIRcv, and HostSPIClock. When operating as a PRP network an additional signal is used called HostSPIStatus.

PMD P/N: DD481C005606



Performance Motion Devices, Inc			
Title N-Series ION SPI Host Interface and Connector			
Size A	Document Number 1	Rev A	
Date: Thursday, December 08, 2022	Sheet 5	of 14	

Figure C-5:
N-Series ION
SPI Host
Interface

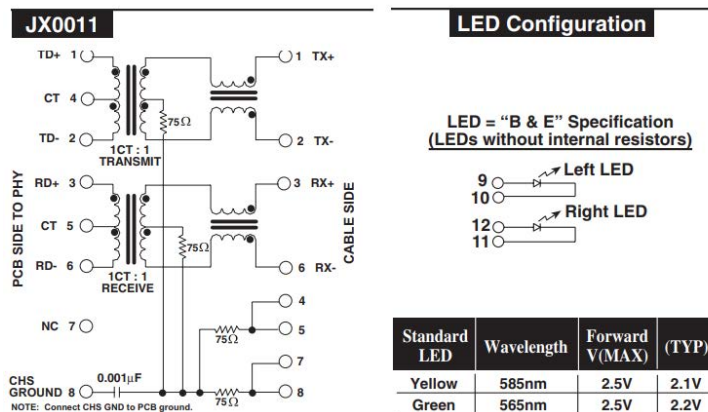
C.7 Ethernet Host Interface

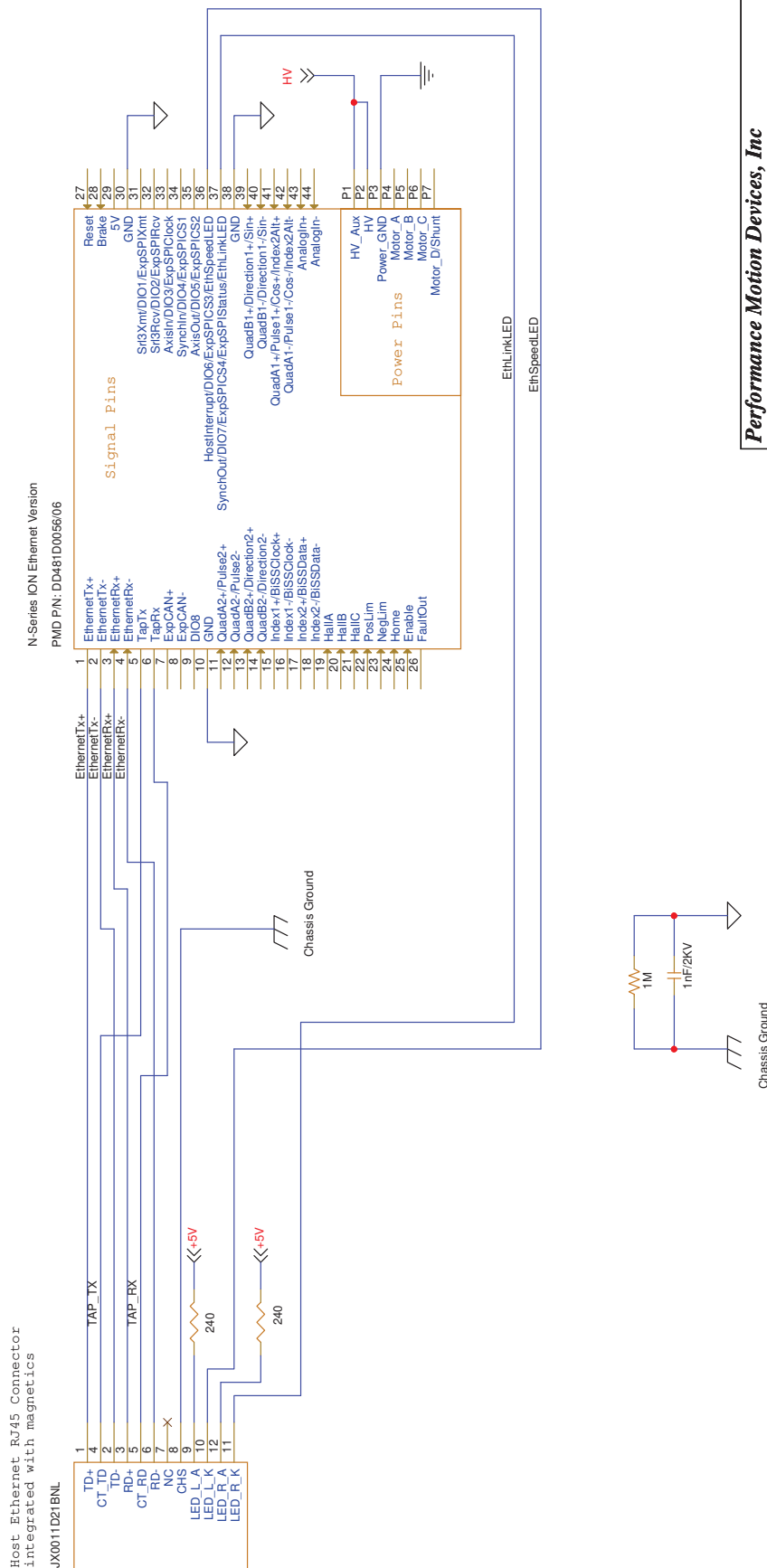
N-Series IONs that are of Ethernet host interface type provide 100 Base-T standard Ethernet communications. The 10/100 Mbps Ethernet interface signals consist of the EthernetTX +, EtherentTX-, EthernetRX +, EtherentRX- analog signals.

The N-Series ION's internal circuitry and signaling provide the electrical functions needed for a direct Ethernet connection, however external magnetics are required for providing galvanic isolation. N-Series ION provides center tap connection for the transformer. The simplest way to accomplish this is to utilize an RJ-45 connector with integrated magnetics (Pulse Electronics Network JX0011D21BNL). This option is convenient and compact, and is the approach used with PMD's Ethernet N-Series ION developer kit. The schematic on the facing page shows how the RJ-45 connector with integrated magnetics is connected to an N-Series ION. The figure below shows the detailed internal schematic of the specific RJ-45 connector used on the developer kit, the Pulse Engineering JX0011D21BNL.

Chassis ground is connected to signal ground through a 1nF/2KV capacitor and a 1M ohms resistor in parallel. The resistor is a bleeder to avoid chassis ground plane being charged up to a very high voltage due to static charge. A 1nF capacitor has an impedance of 1.6 ohms at 100MHz, this is low enough to provide a good RF ground. The isolation between the chassis ground and signal ground helps to reduce EMI.

Figure C-6:
Internal
Schematic of
RJ-45 with
Integrated
Magnetics





Performance Motion Devices, Inc

Title			
N-Series ION Ethernet Host Interface and Connector			
Size	Document Number	Rev	
Custom 1		A	
Date:	Thursday, December 08, 2022	Sheet	6 of 14

Figure C-7:
N-Series ION
Ethernet host
interface with
integrated
magnetics

C.8 Expansion CAN Network and Motor Connection Setup

All N-Series IONs support an expansion CAN FD port which is useful for networking IONs together and building hierarchical networks that can reduce network traffic bottlenecks. The schematic on the facing page shows the Expansion CAN network connected at daisy-chain connection scheme and different types of motors connection setup.

Above each network-attached device requires a unique Node ID. For example, the master N-Series ION Node ID is 0, the first unit in the chain's Node ID is 1, the second unit in the chain's Node ID is 2. The desired Node ID value is set by **SetNodeID** command. The N-Series ION supports auto Node ID assignment via a daisy-chain Node ID assignment scheme utilizing its SynchIn and SynchOut signals. For more information on daisy-chain Node ID assignment refer to user manual Section 4.6.1.3, "Daisy-Chain Node ID Assignment".

The ISO 11898 standard specifies the interconnect to be a twisted-pair cable (shielded or unshielded) with 120 Ω characteristic impedance. Resistors equal to the characteristic impedance of the line should be used to terminate both ends of the cable to prevent signal reflections.

CAN version N-Series IONs are recommended to be used as slaves in CAN network configuration, but Series and Ethernet version N-Series IONs can be used as slaves too. In this scenario, the master device ExpCAN + and ExpCAN- signals are connected to slaves' ExpCAN + and ExpCAN- respectively.

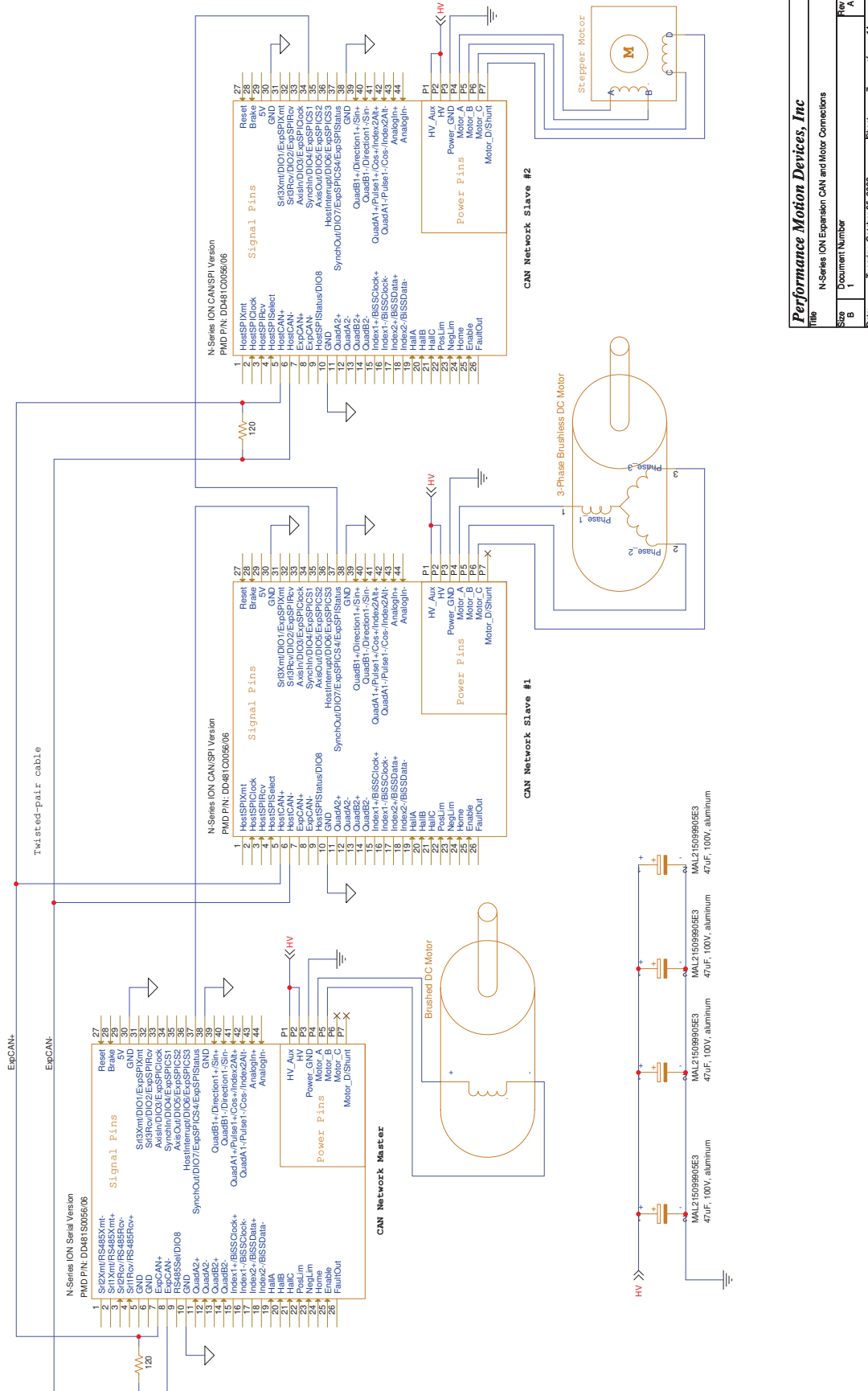


Figure C-8:
N-Series ION
Expansion CAN
network and
Motor
Connections

Performance Motion Devices, Inc

Title: N-Series ION Expansion CAN and Motor Connections

Size: B

Document Number: 1

Rev: A

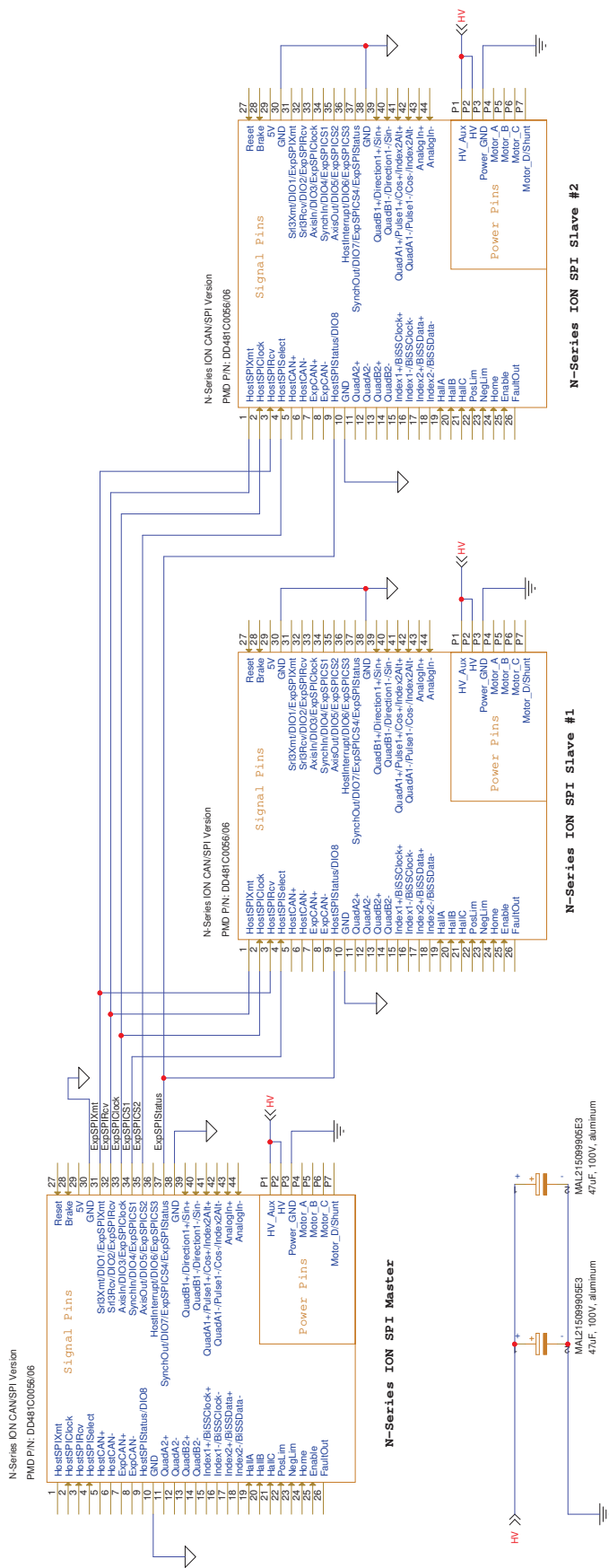
Date: Tuesday, October 25, 2022

Sheet: 7 of 14

C.9 Expansion SPI Network

The schematic on the facing page shows a multi-ION connection with the N-Series ION unit's expansion SPI port operating as an SPI master to command two SPI host interface N-Series ION units operating as slaves.

Separate select signals connect the master ION and each slave device. The SPI clock, data transmit, data receive, and SPI status signals are tied together, forming a bus. Signals output by the slave units (host SPI transmit and SPI status) are asserted only by the slave SPI device selected via the select signal. Non-selected devices leave these signals in a floating condition thereby avoiding signal contention.



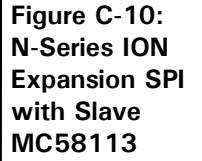
Performance Motion Devices, Inc			
Title	N-Series ION Expansion SPI	Rev	A
Size	Document Number	Sheet	8 of 14
Customer	Customer 1		
Date	Thursday, December 08, 2023		

Figure C-9:
N-Series ION
Expansion SPI
Connection

C.10 Connecting to an MC58113 IC

The schematic on the facing page shows the N-Series ION unit's expansion SPI port operating as an SPI master to command a Magellan MC58113 operating as an SPI slave.

Since the N-Series ION Expansion SPI IO voltage is 5V and the Magellan MC58113 IO voltage is 3.3V, buffers are needed to convert the signal voltage levels between them.



C.11 Digital IO Connection Schemes

The facing schematic shows the N-Series ION digital I/O connection schemes:

- 1 Digital IO2 is set as input accepting 0 ~ 5V signal.
- 2 Digital IO4 is set as output driving a LED. When functioning as output the I/O can provide 0-5V and 8mA . A resistor is connected in series with the LED to limit the current (less than 8mA).
- 3 Digital IO5 is set as output driving a 24V relay. When functioning as an output this signal is an open collector output with 4mA, 5V sourcing and up to 24V, 50mA sinking capability.
- 4 Digital IO7 is set as input accepting 0 ~ 24V signal. When functioning as an input this signal is a TTL compatible input and accepts 0-24V.

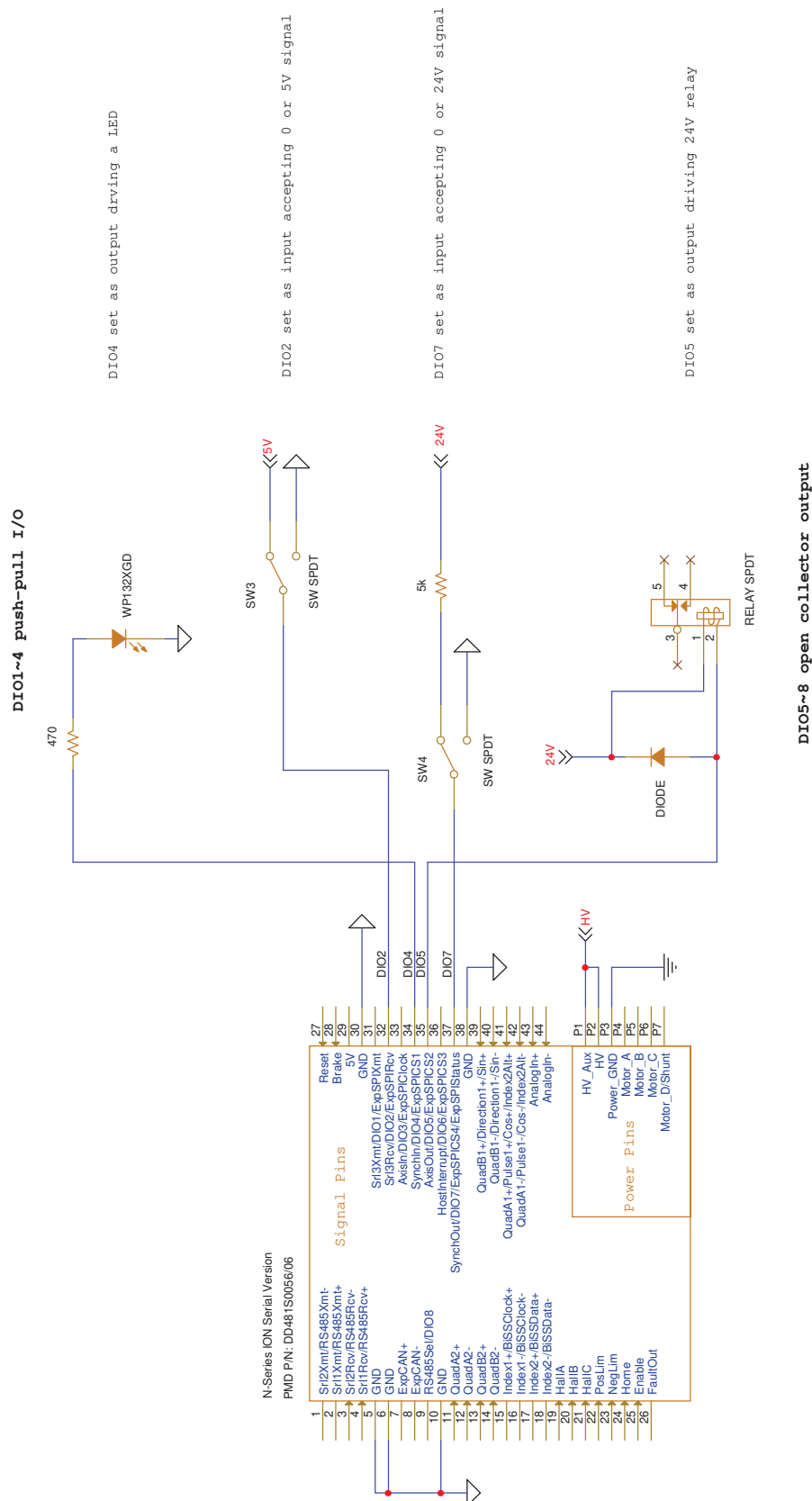


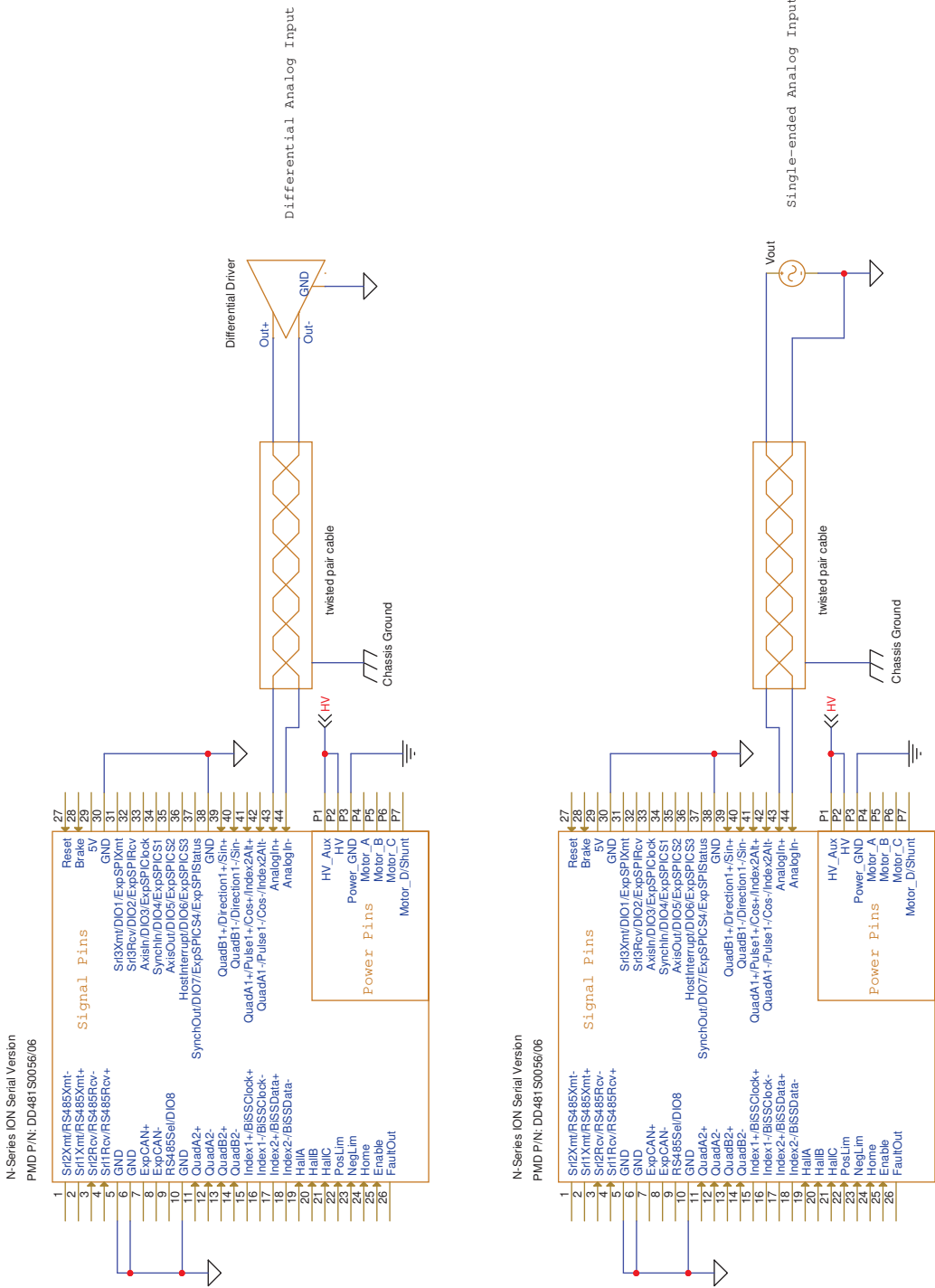
Figure C-11: N-Series ION Digital IO Connection Schemes

Performance Motion Devices, Inc	
title	N-Series ION Digital IO Applications
size	Document Number
custpm 1	Rev A

C.12 Analog Input

The high-resolution Analog Input has a voltage range of $\pm 10\text{V}$. The N-Series ION receives both single-ended and differential analog inputs as shown in the schematic on the facing page.

When used in single ended configuration, AnalogIn- should be connected to Ground of the transmitter. When used in differential configuration, AnalogIn+ and AnalogIn- are connected to the differential output of the transmitter.



Performance Motion Devices, Inc

Title			
N-Series ION Analog Input Connection			
Size	Document Number	Rev	
Custom1		A	
Date:	Thursday, December 08, 2022	Sheet	11 of 14

Figure C-12: N-Series ION Analog Signal Input Connection Schemes

This page intentionally left blank.

Appendix D. Example Interconnect Board Layout

D

In This Appendix

- ▶ Layer 1
- ▶ Layer 2
- ▶ Layer 3
- ▶ Layer 4

This appendix provides design details for an actual four layer N-Series ION interconnect card, the Ethernet N-Series ION DK Interconnect Board, PMD part number: PCB-1047-01. This is the board that is used in the three Ethernet host N-Series ION DKs. This PCB uses FR-4 material and has a finished total thickness of $0.062" \pm 0.007"$.

The sections below provide images and narrative comments for each layer of the PCB.

For more technical information on this board or for similar information for the Serial or CAN/SPI DK interconnect boards including the BOM and electronic versions of the images shown in this appendix contact your local PMD representative.

D.1 Layer 1

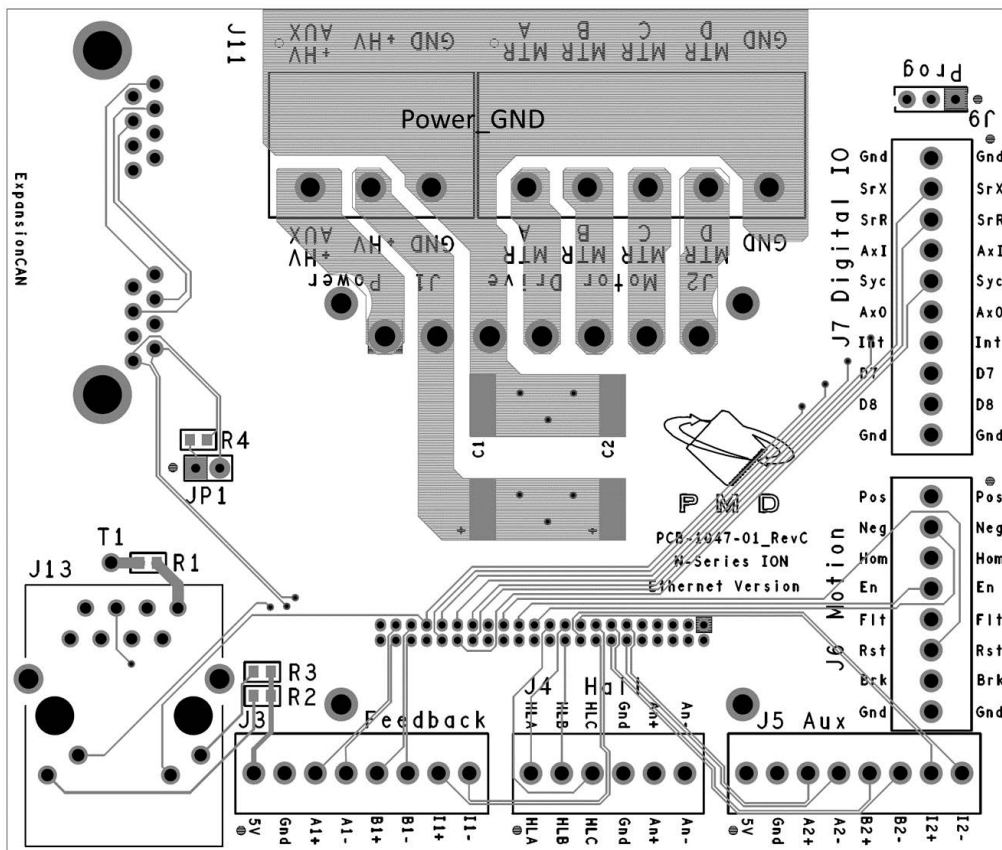


Figure D-1:
PCB-1047-01
Layer 1

Layer 1 contains the power connector traces, signal connector traces and a power ground plane.

The power traces/shapes going from the N-Series ION pins to the connectors are 0.120" in width. These power traces are also present on other layers (more parallel conductors) to reduce the trace resistance as much as possible. Two bulk capacitors (C1 & C2) on HV line are placed on top. HV and HV_Aux are kept isolated. The trace width of the power traces (HV, HV_Aux, Motor A, B, C, D) is approximately 0.125"

Layer 1 is made with 2-oz copper. Note that in PCB design the thickness of copper is defined by oz/ft², but referred to in short as just oz in common practice.

D.2 Layer 2

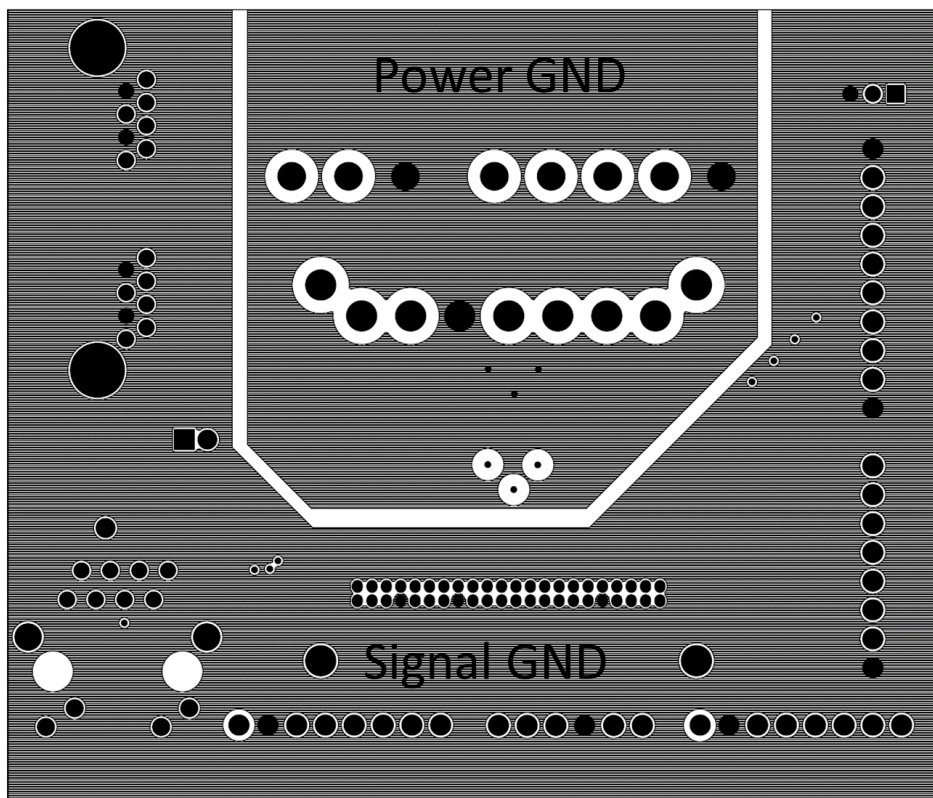


Figure D-2:
PCB-1047-01
Layer 2

Layer 2 is exclusively for ground planes, divided into two isolated sections; one for the power ground and the other for signal ground.

This approach improves the ground plane noise isolation between the power ground and digital ground. The reason this may be important is that the power ground is generally noisier because the magnitude of switching voltages and currents are much higher compared to signal grounds.

Layer 2 is made with 1 oz copper.

D.3 Layer 3

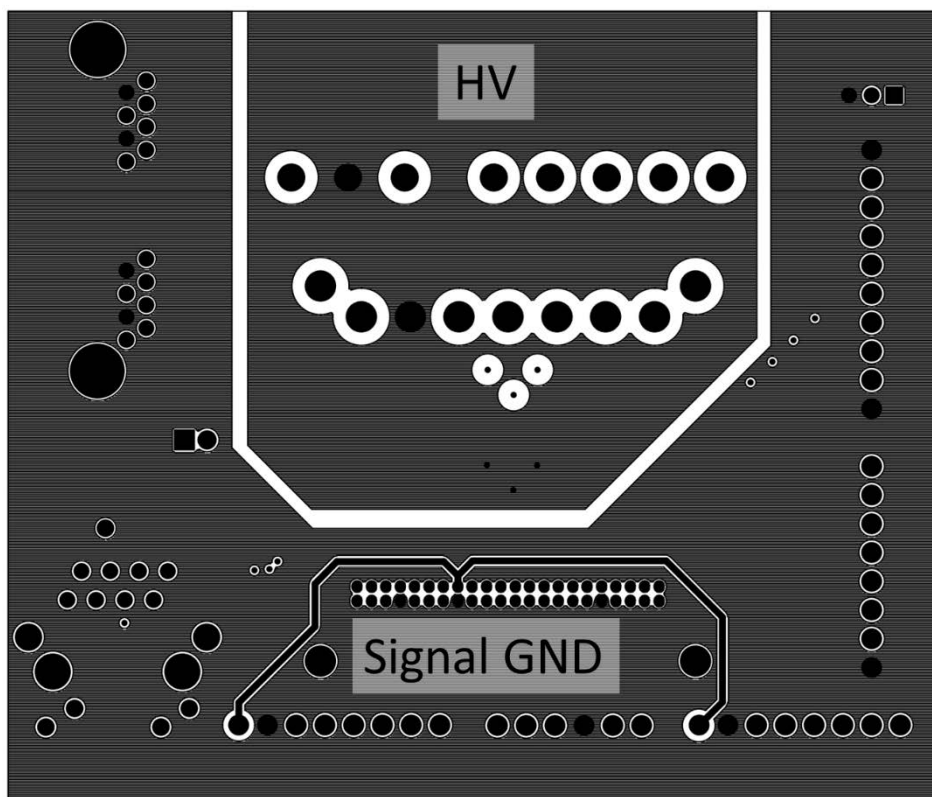


Figure D-3:
PCB-1047-01
Layer 3

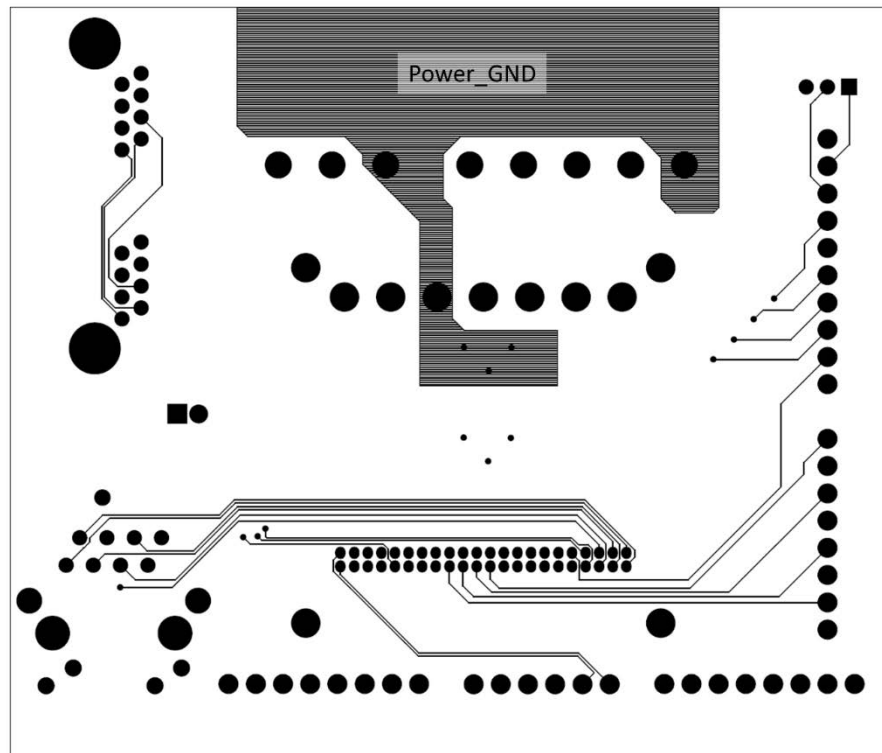
Layer 3 contains another signal ground plane and an HV plane. HV is the DC bus voltage that powers the drive. The HV plane is made as large as possible complementing a parallel HV trace in layer 1.

Note that the clearance between the signal ground plane and the HV plane meet the minimum IPC requirements for levels of isolation based on DC bus voltage. The encoder +5V signal trace can also be seen in this layer.

Layer 3 is made with 1 oz copper.

D.4 Layer 4

Figure D-4:
PCB-1047-01
Layer 4



Layer 4 is the bottom layer of the board. It contains an additional parallel trace/plane for the power ground. In addition, the Ethernet signal traces and a few other signal traces are on this layer.

Layer 4 is made with 2 oz copper.