MicroLYNX

Part II MicroLYNX

- *The MicroLYNX System*
- *Getting Started*
- *Installing and Mounting the MicroLYNX*
- *Powering the MicroLYNX System*
- *Motor Requirements*
- *Controlling the Output Current*
- *The Communications Interface*
- *Configuring and Isolated Digital I/O*
- *Configuring the Using the Expansion Modules*

IMS™ intelligent motion systems, inc.
Excellence in Motion™

370 North Main St., P.O. Box 457, Marlborough, CT 06447 U.S.A.
Phone: 860/295-6102, Fax: 860/295-6107
Internet: www.imshome.com, E-Mail: info@imshome.com

Table of Contents

Section 1: The MicroLYNX System 2-7

Section Overview 2-7
Introduction 2-7
Electrical Specifications 2-8
    Power Supply Requirements 2-8
    Motor Drive 2-8
    General Purpose I/O 2-8
Communication Specifications 2-8
    Asynchronous 2-8
    Controller Area Network (CAN) 2-9
Mechanical Specifications 2-9
Environmental Specifications 2-9
Motion Specifications 2-10
Software Specifications 2-10
Connection Overview 2-11
Fault Input 2-12

Section 2: Getting Started 2-14

Section Overview 2-14
Getting Started 2-14
    Included in the Package 2-14
    User Provided Tools and Equipment Needed 2-15
    Connecting the Power Supply 2-15
    Motor Connections 2-15
    Communications Wiring 2-15
    Establishing Communications using the IMS LYNX Terminal 2-15
    Testing the MicroLYNX Setup 2-16

Section 3: Installing and Mounting the MicroLYNX 2-18

Section Overview 2-18
Dimensional Information 2-18
Installation and Removal of the Optional Expansion Modules 2-18
Mounting the MicroLYNX System to a Panel 2-20

Section 4: Powering the MicroLYNX System 2-21

Section Overview 2-21
Selecting a Power Supply 2-21
    Selecting a Motor Supply (+V) 2-21
Wiring and Shielding 2-22
    Rules of Wiring 2-22
    Rules of Shielding 2-22
Power Supply Connection & Specification 2-23
Recommended IMS Power Supplies 2-23

Section 5: Motor Requirements 2-24

Section Overview 2-24
Selecting a Motor 2-24
    Types and Construction of Stepping Motors 2-24
    Sizing a Motor for Your System 2-24
    Recommended IMS Motors 2-26
Motor Wiring 2-27
Connecting the Motor 2-28
    8 Lead Motors 2-28
    6 Lead Motors 2-29
        Half Coil Configuration 2-29
    4 Lead Motors 2-29
        Full Coil Configuration 2-29

Section 6: Controlling the Output Current and Resolution 2-30

Section Overview 2-30
Current Control Variables 2-30
Determining the Output Current 2-31
Setting the Output Current 2-33
Setting the Motor Resolution 2-34

Section 7: The Communications Interface 2-35

Section Overview 2-35
Connecting the RS-232 Interface 2-35
Connecting the RS-485 Interface 2-39
MicroLYNX Modes of Operation 2-42
    Immediate Mode 2-42
    Program Mode 2-42
    EXEC Mode 2-42
MicroLYNX Communication Modes 2-42
    ASCII 2-42
    Binary 2-43

Section 8: CAN Communications 2-44
Section Overview 2-44
Connecting to the CAN Bus 2-44
Using the CAN Module 2-45
Configuring the CAN Module 2-46
CAN Configuration Command Summary 2-46
To Initialize the CAN Module. 2-47
To Set the CAN Bit Timing Registers 2-48
To Set The Global Mask Registers 2-49
To Setup Message Frames 2-50
Set Message Frame Arbitration Registers 2-50
Defining the MicroLYNX Mode (Single or Party) 2-51
Setting the MicroLYNX Party Address 2-52
MicroLYNX Prompt 2-52
MicroLYNX Baud Rate 2-52
The CAN Communication Dongle 2-53
Connecting the CAN Dongle 2-53
Setting up Communications with MicroLYNX CAN 2-54
Setup Procedure 2-55
Module Setup 2-55
Dongle Setup 2-57
Establishing Communications between MicroLYNX CAN and the CAN Dongle 2-58
Upgrading MicroLYNX via CAN 2-58
Section 9: DeviceNet 2-59
Section Overview 2-59
MicroLYNX DeviceNet Features 2-59
Connector Locations and Pin Descriptions 2-60
Power and Motor Connections 2-61
Isolated Digital Input 2-61
Encoder Inputs 2-61
Attribute Maps 2-62
I/O Messaging and Response 2-65
I/O Response Supported 2-66
Poll I/O Command Format 2-66
Poll I/O Response Format 2-69
Poll I/O Message Example 2-72
Encoder Configuration Example 2-74
The DeviceNet Programmer Cable 2-75
Section 10: Configuring the Isolated Digital I/O 2-76
Section Overview 2-76
Electrical Characteristics 2-76
The Isolated Digital I/O 2-76
Uses of the Isolated Digital I/O 2-77
The IOS Variable 2-77
Configuring an Input 2-78
Configuring the Digital Filtering 2-79
Configuring an Output 2-80
The IO Variable 2-80
Read/Write an I/O Group 2-81
Section 11: Configuring and Using the Expansion Modules 2-82
Section Overview 2-82
MicroLYNX Expansion Modules 2-82
Additional Isolated Digital I/O 2-82
High-Speed Differential I/O Module 2-82
Analog Input/Joystick Module 2-82
Isolated CommunicationsModule 2-82
Analog Output Module 2-83
12 Channel Isolated Ditital I/O 2-83
Choosing the Expansion Modules for Your Application 2-83
Explanation of Expansion Slot Usage 2-84
Thermal and Environmental Specifications 2-83
Isolated Digital I/O Module 2-85
Electrical Specifications 2-85
I/O Configuration 2-86
Installing The Isolated Digital I/O Module 2-86
Using the Isolated Digital I/O 2-87
High-Speed Differential I/O Module 2-88
Electrical Specifications 2-88
Installing the High-Speed Differential I/O Module 2-89
The Four Clocks Explained 2-89
Clock Types Defined 2-89

Configuring the Differential I/O - The IOS Variable 2-90
Configuring the High Speed I/O a Non-Clock Function 2-91
Configuring an Input 2-91
Setting the Digital Input Filtering for the Differential I/O 2-92
Configuring an Output 2-92
Typical Functions of the Differential I/O 2-93
Connecting and Using and Encoder 2-93
Testing Your Encoder Setup 2-93
Introducing the EUNIT (Encoder UNITS) Variable 2-94
Following and External Clock (Electronic Gearing) 2-95
Analog Input/Joystick Module 2-98
Electrical Specifications 2-98
Installing the Analog Input/Joystick Module 2-99
Instructions and Variables Specific to the Analog Input/Joystick Module 2-99
Error Codes 2-100
The ADS Variable (A to D Setup) 2-100
Typical Functions of the Analog Input/Joystick Module 2-101
Isolated Communications Modules 2-104
Electrical Specifications 2-104
The RS-232 Communications Module 2-105
The RS-485 Communications Module 2-106
Installing the Isolated Communications Module 2-107
Analog Output Module 2-108
Electrical Specifications 2-108
Installing the Analog Output Module 2-109
Analog Output Commands 2-110
Absolute Type Examples (For 0 to 5 Volt Output) 2-111
Plus or Minus Type Examples 2-111
12 Channel Isolated Digital I/O Module 2-112
Electrical Specifications 2-112
I/O Configuration 2-113
Installing the 12 Channel I/O MOdule 2-113
Pull-up Switches 2-114

List of Tables

Table 1.1: MicroLYNX Switches 2-13
Table 6.1: Motor Current Control Variables 2-31
Table 6.2: Microstep Resolution Settings 2-34
Table 7.1: Wiring Connections, RS-232 Interface, Single MicroLYNX System 2-36
Table 7.2: Party Mode Address Configuration Switch Settings 2-37
Table 7.3: Connections and Settings, RS-232 Interface, Multiple MicroLYNX System 2-38
Table 7.4: RS-485 Interface Connections 2-39
Table 7.5: Party Mode Address Configuration Switch Settings 2-40
Table 7.6: RS-485 Interface Connections and Settings, Multiple MicroLYNX System 2-41
Table 7.7: ASCII Mode Special Command Characters 2-43
Table 7.8: Binary Hex Codes 2-43
Table 8.1: CAN Pin Configuration 2-45
Table 8.2: CAN Configuration Command Summary 2-46
Table 8.3: CAN Bit Timing Registers 2-48
Table 8.4: CAN Bit Time Definition 2-48
Table 8.5: Sample Bit Timing Register 2-48
Table 8.6: Global Mask Registers 2-49
Table 8.7: Message Frame Arbitration Registers 2-51
Table 9.1: Motor Power Connections 2-61
Table 9.2: Isolated Digital Inputs 2-61
Table 9.3: Encoder Input Connections 2-61
Table 9.4: MicroLYNX DeviceNet Attribute Map (Part 1) 2-62
Table 9.5: MicroLYNX DeviceNet Attribute Map (Part 2) 2-63
Table 9.6: MicroLYNX DeviceNet Attribute Map (Part 3) 2-64
Table 9.7: Target Position Command Message 2-66
Table 9.8: Target Velocity Command Message 2-66
Table 9.9: Acceleration Command Message 2-67
Table 9.10: Deceleration Command Message 2-67
Table 9.11: Continuous Velocity Command Message 2-67
Table 9.12: Start Homing Command Message 2-68
Table 9.13: Position Controller Supervisor Attribute Command Message 2-68
Table 9.14: Example of Position Controller Supervisor Attribute 2-68
Table 9.15: Position Controller Attribute Command Message 2-69
Table 9.16: Example of Position Controller Attribute 2-69
Table 9.17 Actual Position Response Message 2-69
Table 9.18: Commanded Position Response Message 2-70

Table 9.19: Actual Velocity Response Message 2-70
Table 9.20: Command/Response Error 2-70
Table 9.21: Position Controller Supervisor Attribute Response Message 2-71
Table 9.22: Example of Position Controller Supervisor Attribute 2-71
Table 9.23 Position Controller Attribute Response Message 2-71
Table 9.24: Example of Position Controller Attribute 2-71
Table 9.25: Example of Set Variables and Flags 2-72
Table 9.26: Start Homing Sequence to Home Switch 2-72
Table 9.27: Continuous Velocity in CCW Direction 2-72
Table 9.28: Smooth Stop, Restore Mode 1 2-72
Table 9.29: Target Position = 1 Revolution 2-72
Table 9.30: Continuous Velocity in CW Direction 2-72
Table 9.31: Smooth Stop, Restore Mode 2 2-72
Table 9.32: Alarm Codes 2-73
Table 9.33: Encoder Configuration 2-74
Table 10.1: IOS Variable Settings 2-78
Table 10.2: Digital Filter Settings for the Isolated I/O 2-79
Table 10.3: Binary State of Outputs 2-81
Table 11.1: MicroLYNX Expansion Module Slot Usage 2-83
Table 11.2: Isolated Digital I/O Group and Line Locations (by Connector and Slot) 2-85
Table 11.3: High Speed Differential I/O Expansion Module Pinout by Connector Style and Slot 2-88
Table 11.4: The Four Clocks and their Default Line Placement 2-90
Table 11.5: Digital Input Filter Settings for the Differential I/O 2-92
Table 11.6: Expansion Slot 2 Encoder Connections 2-93
Table 11.7: Analog Input/Joystick Module Pin Configuration 2-98
Table 11.8: Analog Input/Joystick Module Software Command Summary 2-100
Table 11.9: Isolated Communications Module RS-232 Pinout 2-104
Table 11.10: Isolated Communications Module RS-485 Pinout 2-104
Table 11.11: Analog Output Module Pinout 2-108
Table 11.12: 12 Channel Isolated I/O Module Pinout 2-112

List of Figures

Figure 1.1: Dimensional Information 2-9
Figure 1.2: MicroLYNX Connection Overview 2-11
Figure 1.3: Fault Input Diagram 2-12
Figure 1.4: Fault Input (Sourcing Example) 2-12
Figure 1.5: Fault Input (Sinking Example) 2-12
Figure 1.6 MicroLYNX Switches 2-13
Figure 2.1: Basic Setup Configuration (RS-232 Interface) 2-14
Figure 3.1: Dimensional Information 2-18
Figure 3.2: MicroLYNX System with Isolated Digital I/O Expansion Module Installed 2-19
Figure 3.3: Installing the Optional Expansion Modules 2-19
Figure 3.4: Panel Mounting the MicroLYNX 2-20
Figure 4.1: MicroLYNX Power Connection 2-23
Figure 5.1: Per Phase Winding Inductance 2-25
Figure 5.2: 8 Lead Motor, Series Connection 2-28
Figure 5.3: 8 Lead Motor, Parallel Connection 2-28
Figure 5.4: 6 Lead Motor, Half Coil Connection 2-29
Figure 5.5: 6 Lead Motor, Full Coil Connection 2-29
Figure 5.6: 4 Lead Motor 2-29
Figure 6.1: Motor Current Control Variables 2-30
Figure 7.1: Connecting the RS-232 Interface, Single MicroLYNX System 2-36
Figure 7.2: Connecting the RS-232 Interface, Multiple MicroLYNX System 2-38
Figure 7.3: RS-485 Interface, Single MicroLYNX System 2-39
Figure 7.4: Half Duplex Set, Single MicroLYNX System 2-40
Figure 7.5: RS-485 Interface, Multiple MicroLYNX System 2-41
Figure 8.1: Devices on a CAN Bus 2-44
Figure 8.2: Connecting to the CAN Bus 2-45
Figure 8.3: Bit Register Configuration Dialog from IMS Terminal 2-49
Figure 8.4: Setup Dialog for Global Mask Registers in IMS Terminal 2-50
Figure 8.5: Message Frame Setup Dialog From IMS Terminal 2-51
Figure 8.6: MicroLYNX CAN Setup Dialog from IMS Terminal 2-52
Figure 8.7: MicroLYNX MX-CC500-000 CAN Dongle Details 2-53
Figure 8.8: Connecting the CAN Dongle (With MicroLYNX as Power Source) 2-54
Figure 8.9: CAN Communications Diagram 2-54
Figure 8.10: Connect Procedure (CAN Module Setup) 2-55

Figure 8.11: Baud Rate (CAN Module Setup) 2-55
Figure 8.12: Mask Registers (CAN Module Setup) 2-55
Figure 8.13: RS-232 Setup (CAN Module Setup) 2-55
Figure 8.14: Message Frame 1 (CAN Module Setup) 2-56
Figure 8.15: Message Frame 2 (CAN Module Setup) 2-56
Figure 8.16: Message Frame 3 (CAN Module Setup) 2-56
Figure 8.17: CAN Module Setup Complete 2-56
Figure 8.18: Dongle Baud Rate (CAN Dongle Setup) 2-57
Figure 8.19: Mask Registers (CAN Dongle Setup) 2-57
Figure 8.20: Dongle Baud Rate (CAN Dongle Setup) 2-57
Figure 8.21: Message Frame 1 (CAN Dongle Setup) 2-57
Figure 8.22: Message Frame 2 (CAN Dongle Setup) 2-57
Figure 9.1: MicroLYNX DeviceNet Port Pin Configuration 2-60
Figure 9.2: DeviceNet Port Pin Configuration 2-60
Figure 9.3: Motor Power Terminals 2-61
Figure 9.4: Isolated Digital Input Terminals 2-61
Figure 9.5: Encoder Connect (8 Pin Pheonix) 2-61
Figure 9.6: Encoder Connect (10 Pin Header) 2-61
Figure 9.7: DeviceNet Programmer 2-75
Figure 9.8: DeviceNet MicroLYNX 2-75
Figure 9.9: MX-CC600-000 DeviceNet Programmer Details 2-75
Figure 10.1: Isolated I/O Applications 2-77
Figure 10.2: Typical Sinking Isolated Digital I/O Input Equivalent Circuit 2-79
Figure 10.3: Typical Sourcing Isolated Digital I/O Input Equivalent Circuit 2-79
Figure 10.4: Isolated Digital I/O Output Equivalent Circuit 2-80
Figure 11.1: Installing the Isolated Digital I/O Expansion Module 2-86
Figure 11.2: The Isolated Digital I/O Expansion Module, Bottom View 2-87
Figure 11.3: Powering Multiple Isolated Digital I/O Modules 2-87
Figure 11.4: Installing the High-Speed Differential I/O Expansion Module 2-89
Figure 11.5: Clock Functions 2-90
Figure 11.6: IOS Variable Settings for the High Speed Differential I/O 2-91
Figure 11.7: Differential I/O Input Equivalent Circuit 2-91
Figure 11.8: Differential I/O Output Equivalent Circuit 2-92
Figure 11.9: Differential Encoder Connection 2-94
Figure 11.10: Differential I/O Connections for Following an External Quadrature Input 2-96
Figure 11.11: One and a Half Axis Operation 2-97
Figure 11.12: Installing the Analog Input/Joystick Module 2-99
Figure 11.13: Analog Input Module Exercise Connection 2-101
Figure 11.14: Connecting the RS-232 Expansion Module 2-105
Figure 11.15: Connecting the RS-485 Expansion Module 2-106
Figure 11.16: Installing the Isolated Communications Module 2-107
Figure 11.17: Installing the Analog Output Module 2-109
Figure 11.18: Installing the 12 Channel Isolated I/O Module 2-113
Figure 11.19: 12 Channel I/O Module Pull-up Switches 2-114
Figure 11.20: Powering Multiple Isolated Digital I/O Modules (12 Channel Module Included) 2-114

Section 1

The MicroLYNX System

Section Overview

This section summarizes the specifications for the basic MicroLYNX system. It contains the following:

- Introduction
- Electrical Specifications
- Mechanical Specifications

Introduction

The MicroLYNX is a Motion Control System integrating a bipolar stepper motor microstepping drive and a programmable indexer with expandable I/O and communication capability into a compact panel mounted assembly. The basic system is available with either 3 Amp (MicroLYNX-4) or 5 Amp (MicroLYNX-7) RMS motor drive capability. The basic MicroLYNX System can also be purchased with either the standard dual communications or Controller Area Network (CAN) interface. A DeviceNet version is also available.

The MicroLYNX, developed from the LYNX 1.5 Axis Modular Motion System, has inherited all of the LYNX capabilities but in a smaller package that fits in the palm of your hand. The MicroLYNX also has enhanced features and additional software commands to make use of these features and control the motor drive parameters.

The integration of the drive and the small size of the MicroLYNX are the most obvious accomplishments in its development. The ability to customize the I/O suite to the application in smaller increments is another. The basic MicroLYNX System comes standard with six (6) +5 to +24VDC isolated digital programmable I/O lines. This is expandable to a total of twenty-four (24) lines using optional expansion modules. This section summarizes the specifications for the basic MicroLYNX system. The expansion modules available for the MicroLYNX are:

- Isolated Digital +5 to +24VDC I/O
- High Speed Differential I/O
- Analog Input / Joystick Interface
- Isolated Communications Modules
  - RS-232 Module (for use with the CAN version only)
  - RS-485 Module (for use with the CAN version only)
- Analog Output
- 12 Channel Digital I/O

These modules and their applications are covered in detail in *Section 11: Configuring and Using the Expansion Modules*.

A more subtle enhancement is the provision of two fully independent communication ports for the Micro-LYNX system. While Modular LYNX provided both RS-232 and RS-485 ports, these ports shared the same UART on the LYNX CPU. This limited communications on these ports to sequential usage.

Adding a fully independent second UART allows simultaneous usage. Software has been updated to keep this system fully compatible with the Modular LYNX. The MicroLYNX will accept commands from either COMM port and can now direct output to either port regardless of the state of the HOST flag. Of course, compatibility means that HOST mode is still supported. A motion system architecture might use one COMM port for connection to a host PC or PLC while using the other for communication with an operator interface or status display. Another use for the second port could be to pass data between MicroLYNX Systems in a multi-axis system while maintaining a communications link to a host.

Electrical Specifications

Power Supply Requirements

Voltage

-4 Version (P/N MX-CS100-401)	+12 to +48VDC
-7 Version (P/N MX-CS100-701)	+24 to +75VDC

Current

Actual requirements depend on application and programmable current setting.

-4 Version (P/N MX-CS100-401)	2A typical, 4A peak
-7 Version (P/N MX-CS100-701)	3A typical, 6A peak

Motor Drive

Motor type	2/4 phase bipolar stepper
Motor Current (software programmable)	
-4 Version (P/N MX-CS100-401)	to 4A peak
-7 Version (P/N MX-CS100-701)	to 7A peak
MicroStep Resolution (# of settings)	14

Steps per Revolution (1.8° Motor)

400, 800, 1000, 1600, 2000, 3200, 5000, 6400, 10000, 12800, 25000, 25600, 50000, 51200

General Purpose I/O

Number of I/O	6
Input Voltage	+5 to +24VDC
Output Current Sink	350mA
Input Filter Range	215Hz to 21.5kHz (Programmable)
Pull-ups	7.5kOhm individually switchable
Pull-up Voltage	
Internal	+5VDC
External	+24 VDC
Protection	Over temp, short circuit, inductive clamp
Isolated Ground	Common to the 6 I/O

Communication Specifications

Asynchronous

Interface Type:	
COMM 1	RS-232
COMM 2	RS-485
# of Bits / Character	8
Parity	None
Handshake	None
BAUD Rate	4800 to 38.8kbps
Error Checking	16 bit CRC (binary mode)
Communication Modes	ASCII text or binary
Isolated Ground	Common to COMM 1 and COMM 2

Controller Area Network (CAN)

CAN replaces Asynchronous Communications in the MicroLYNX Base System. (Uses COMM 1 internally.)

CAN Compliance	Version 2.0B Active
Message Frames	
Receive	2
Transmit	1
Isolated Ground	Common to COMM 1 and COMM 2

Mechanical Specifications

Dimensions	See Figure 1.1
# of Expansion Modules	3
Cooling	Built in fan
Mounting	2 #6 (or M3.5) machine screws
Mounting Screw Torque	5 to 7 lb-in

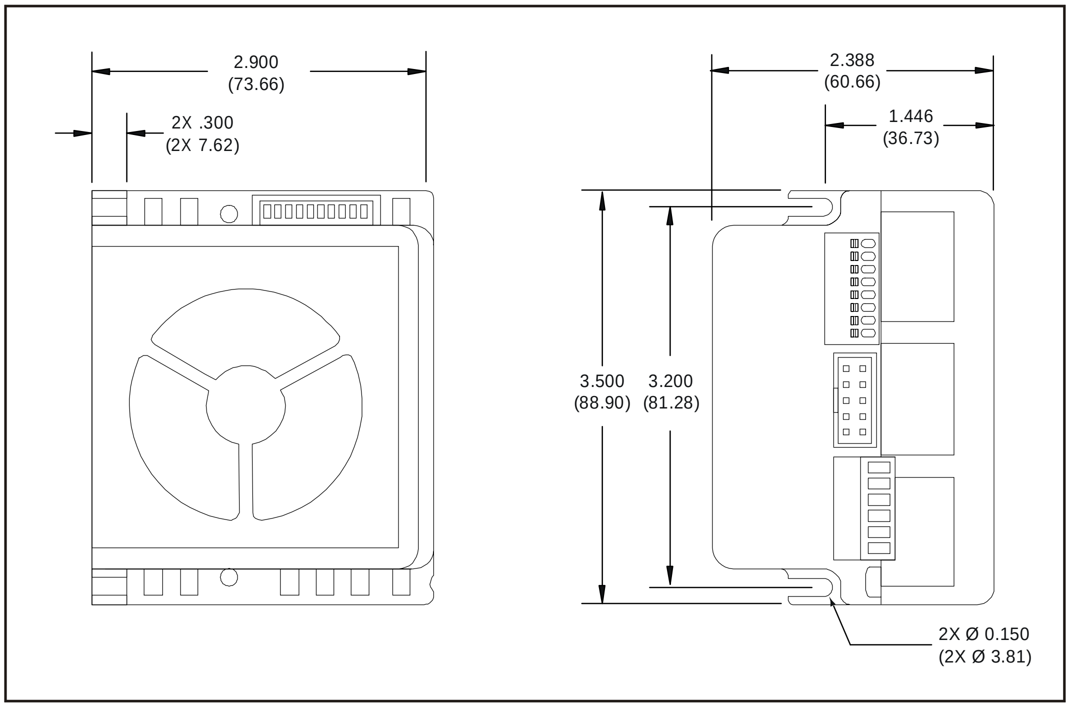


Figure 1.1: Dimensional Information

Environmental Specifications

Ambient Temperature (Operating)	0 to +50°C*
Storage Temperature	-20 to +70°C
Humidity	0 to 90% non-condensing

* Can be duty cycle dependent.

Motion Specifications

Counters

Type	Position, encoder #1, encoder #2
Resolution	32 bits
Edge Rate (Max)	5 MHz

Electronic Gearing

Use of Electronic Gearing requires the Differential I/O Expansion Module.

Range*	-1 to 1 (external clock in)
Resolution	32 bits
Range*	-2 to 2 (secondary clock out)
Resolution	16 bits

*The range may be increased by adjusting the microstep resolution of the drive.

Velocity

Range	±5,000,000 steps/sec
Resolution	0.005 steps/sec
Update Period	25.6 microseconds

Acceleration/Deceleration

Range	±1,530,000,000 steps/sec^2
Resolution	0.711 steps/sec^2

Types:

Linear, triangle s-curve, parabolic, sinusoidal s-curve, user-defined.

Software Specifications

User Program Space .... 8175 bytes
Number of User Definable Labels 291
Program and Data Storage .... Flash
Math, Logic, and Conditional Functions
(32 Bit Floating Point Math IEEE Format):

Add, Subtract, Multiply, Divide, Sine, Cosine, Tangent, Arc Sine, Arc Cosine, Arc Tangent, AND, OR, XOR, NOT, Less Than, Greater Than, Equal, Square Root, Absolute, Integer Part, Fractional Part

Acceleration & Deceleration:

Separate Variables and Flags. 4 Pre-defined Types and 1 User-defined

Limit Switch .... Definable: Deceleration & Type
Isolated I/O .... Programmable as Dedicated or General Purpose
Predefined I/O Functions .... 25 (Limit, Home, Soft Stop, etc.)
Program Trip Functions 13

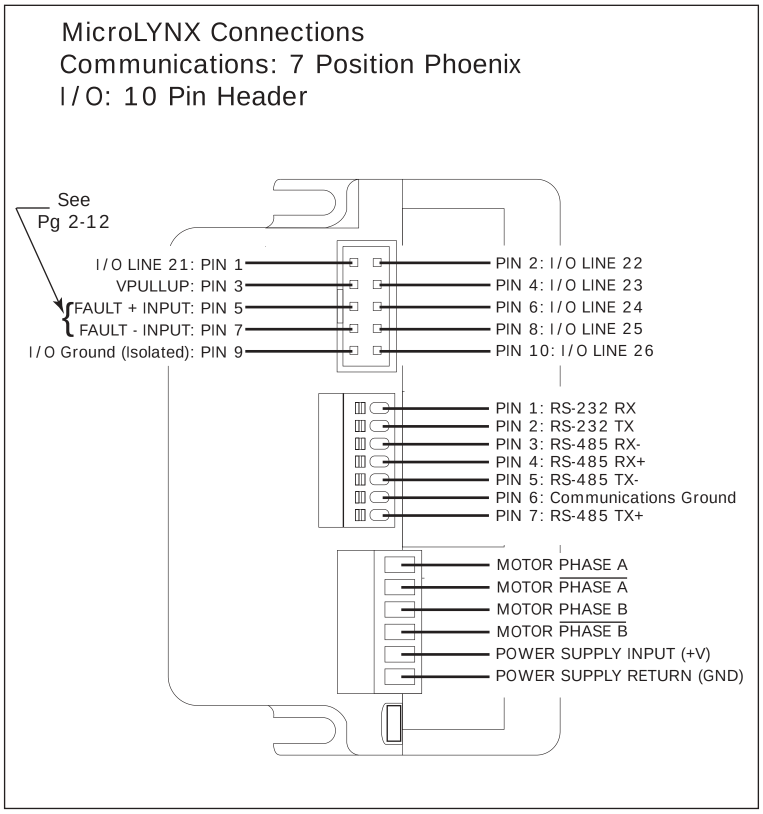
(4 I/O Input Trips, 4 Timer Trips, 4 Position Trips, 1 Velocity Trip)

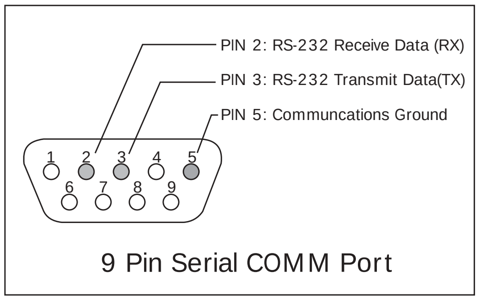
User Programs:

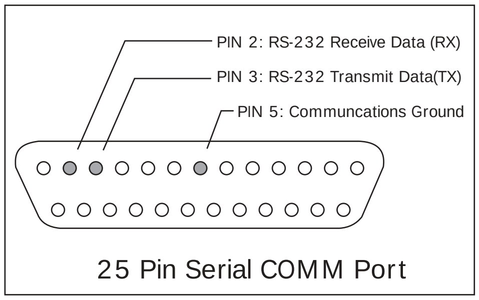
2 Executed simultaneously: 1 Foreground, 1 Background.

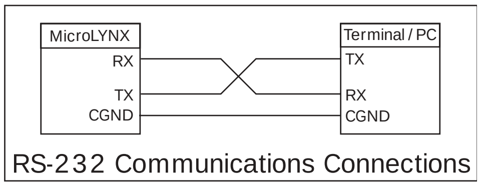
Party Mode Names 62
Communication Modes .... 2: ASCII, Binary
Mechanical Compensation .... Backlash
Encoder Functions .... Stall Detection and Position Maintenance

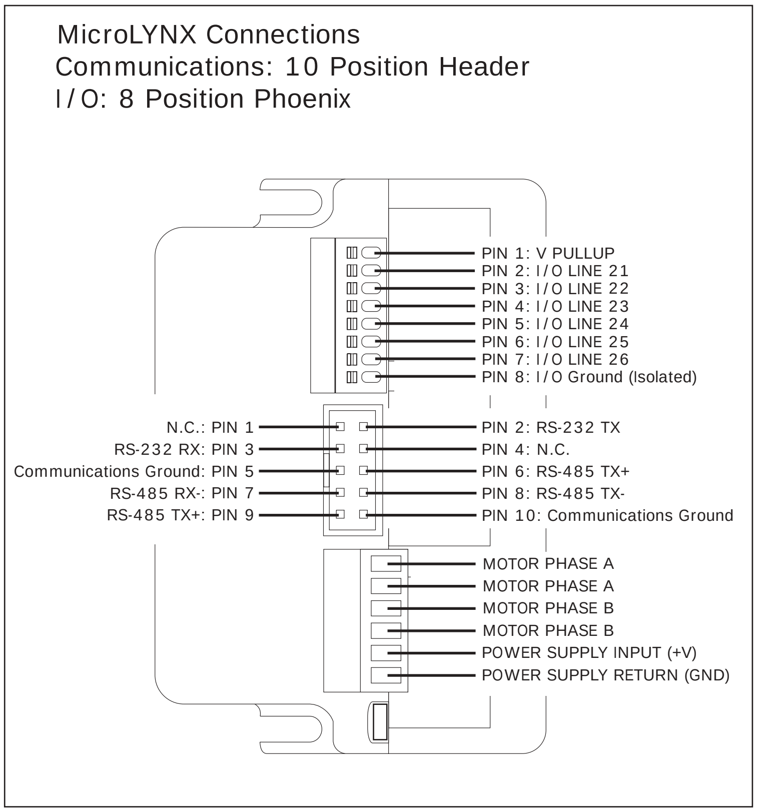
Connection Overview

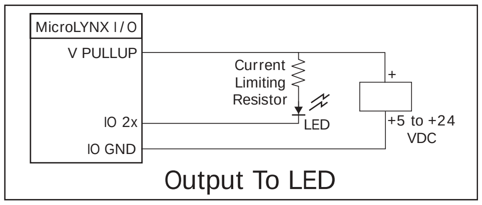


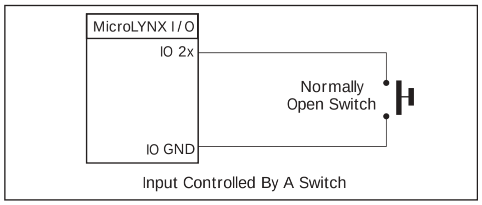

PIN 2: RS-232 Receive Data (RX)
PIN 3: RS-232 Transmit Data(TX)
PIN 5: Communcations Ground
1 2 3 4 5
6 7 8 9
9 Pin Serial COMM Port


PIN 2: RS-232 Receive Data (RX)
PIN 3: RS-232 Transmit Data(TX)
PIN 5: Communcations Ground
25 Pin Serial COMM Port









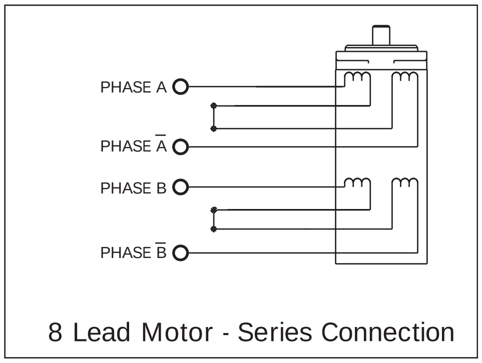


Figure 1.2: MicroLYNX Connection Overview

Fault Input

The Fault Input is only available with the 10 Pin, Group 20 header. The Fault + Input on Pin 5 and the Fault - Input on Pin 7 are not standard I/O. The differential, optically isolated input constitutes a single Fault Input Signal. When the Fault Input is activated the Controller disables the drive, ALL motion commands, and sets and latches error 1100 (Fault Detected). The fault must be eliminated first and then power must be cycled to clear this fault. NOTE: When the drive is disabled there is no holding current.

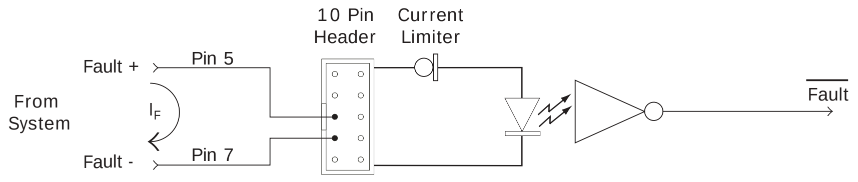


Figure 1.3: Fault Input Diagram

Typical Application

The Fault Input can be the summary of other faults within the system or a dedicated input for a specific purpose.

The Fault Input can be configured as Sourcing or Sinking depending on the user's requirements. (See Examples)

Sourcing Example

Connect Pin 7 to the isolated logic ground and connect an isolated, switched +VDC (such as from the Opto Supply) to the Input Signal on Pin 5.

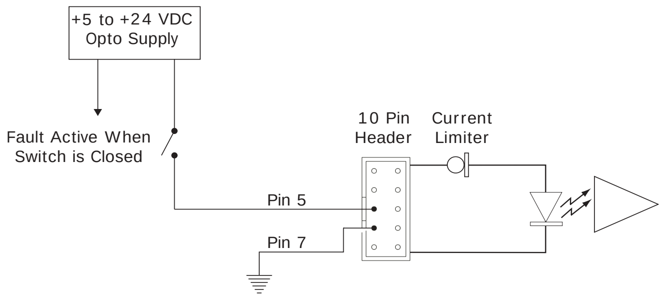


Figure 1.4: Fault Input (Sourcing Example)

Sinking Example

Connect Pin 5 to an isolated +VDC supply (such as the Opto Supply) and connect a switched, isolated logic ground signal to Pin 7.

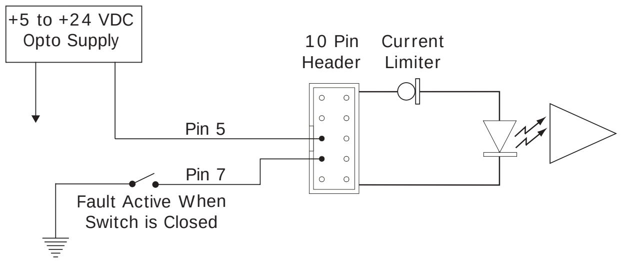


Figure 1.5: Fault Input (Sinking Example)

Fault Level

If (Fault+) - (Fault -) ≥ 4.2 VDC [I_F= 5mA typ], then Fault is set and Error 1100 (Fault Detected) is latched.

If (Fault +) - (Fault -) ≤ 1.2 VDC [I_F= ≤ .05mA], then Fault not set.

NOTE: I_F is self limiting up to 12mA, with (Fault +) - (Fault -) = 24 VDC.
I_F (typ) = 6.5 mA @ [(Fault +) - (Fault -)] = 5.0 VDC
Max [(Fault +) - (Fault -)] = 25 VDC.

Switches

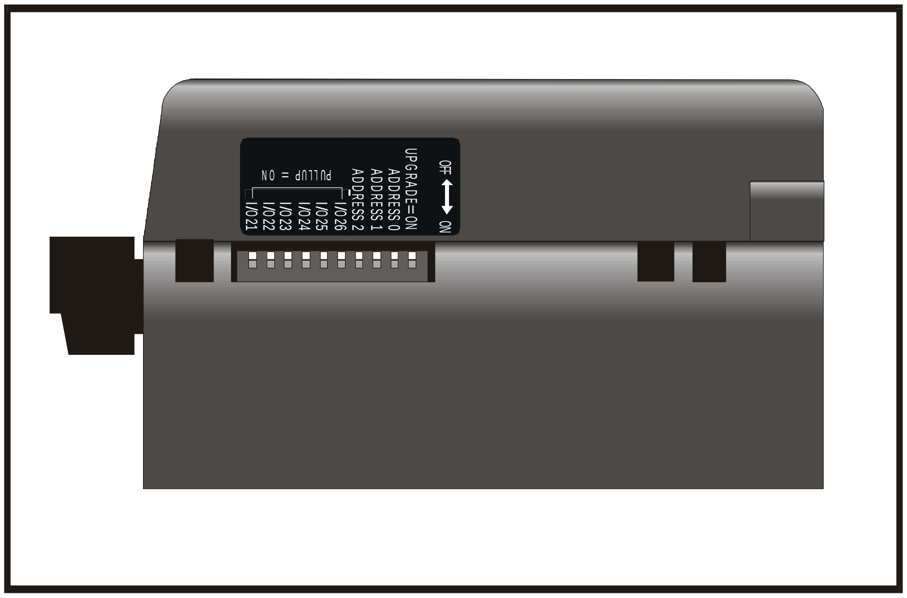


Figure 1.6: MicroLYNX Switches

MICROLYNX SWITCHES		
SWITCH #	**Switch Name**	**Function**
1-6	I/O 26	Pull-up ON/OFF Switches for I/O Lines 21-26
7-9	Address 2-0	Multi-drop Communications Address. (Also can be configured in software.)
10	Upgrade	Firmware Upgrade

Table 1.1: MicroLYNX Switches

Section 2

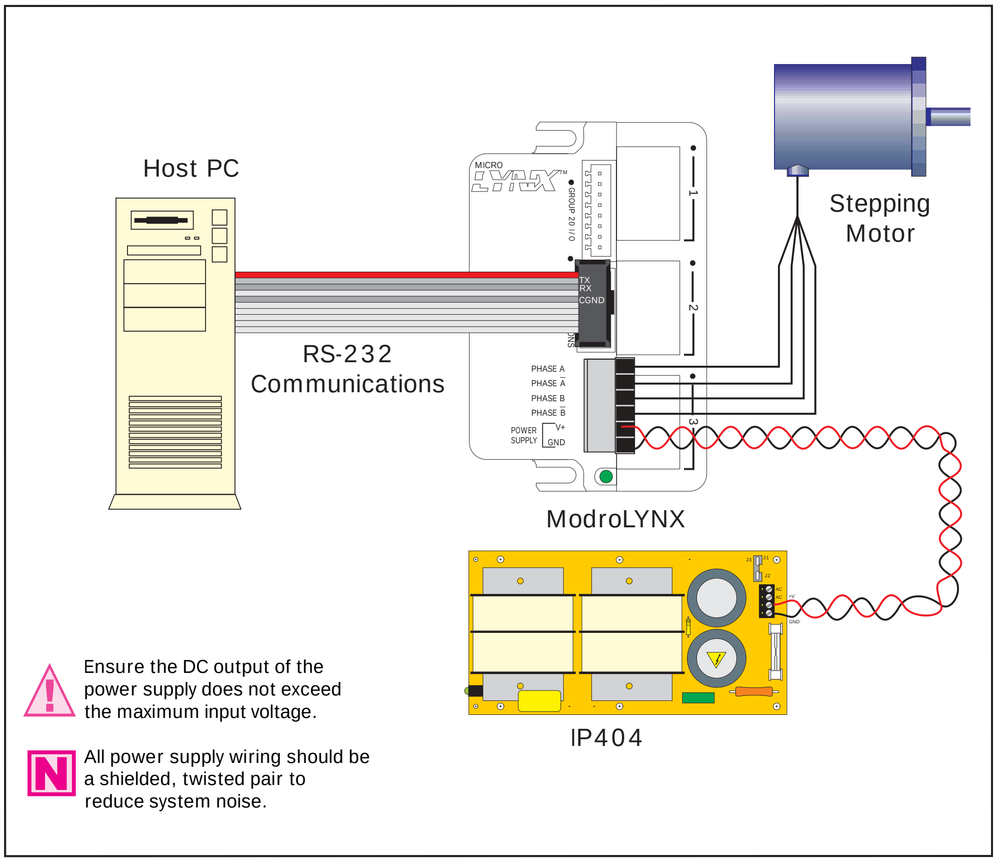
Getting Started

Section Overview

The purpose of this section is to get you up and running quickly. This section will help you do the following:

- Connect power to the MicroLYNX Control System.
- Connect and establish communications in single mode.
- Write a simple test program.

Getting Started



Ensure the DC output of the power supply does not exceed the maximum input voltage.

All power supply wiring should be a shielded, twisted pair to reduce system noise.

Figure 2.1: Basic Setup Configuration, RS-232 Interface

Included in the Package

(1) MicroLYNX Controller	IMS P/N MX-CS100-401 or MX-CS100-701
(1) IMS Compact Disc	IMS P/N IMS-CD100-000
(1) Quick Manual	IMS P/N MX-OM300-000

User Provided Tools and Equipment Needed

- Serial Cable.
- IP404 or equivalent power supply.
- M-22XX or equivalent stepping motor.
- Wire Cutters/Strippers.
- 22 gauge wire for logic level signals.
- 18 gauge wire for power supply and motor wiring.
- PC with a free serial port (COM 1 or 2).

Connecting the Power Supply

1. Using the 18 gauge wire, connect the DC output of your power supply to V+ on your MicroLYNX Control System. See *Figure 2.1: Basic Setup Configuration* for details.
2. Connect the Power Supply Return (GND) to GND on the MicroLYNX Controller.
3. Connect the AC Line cord to your power supply in accordance with any user documentation. **DO NOT PLUG IN AT THIS TIME!**

Motor Connections

Connect the motor to the MicroLYNX System in accordance with *Figure 2.1*.

Communications Wiring

Connect the Host PC to the MicroLYNX in accordance with *Figure 2.1*. This is needed to program the MicroLYNX. If the MicroLYNX has a Terminal Block Connector, connect in the following manner:

PC(25 Pin Serial Port)	PC(9-Pin Serial Port)	MicroLYNX Comm Connector
Pin 7 (GND)	Pin 5 (GND)	Pin 6 (C GND)
Pin 2 (TX)	Pin 3 (TX)	Pin 1 (RS-232 RX)
Pin 3 (RX)	Pin 2 (RX)	Pin 2 (RS-232 TX)

Establishing Communications using the IMS Terminal

Included in the MicroLYNX shipping package is a CD with the IMS Terminal software. This is a programming/communications interface created by IMS to simplify the use of the MicroLYNX. There is a 32 bit version for Windows 9x/NT4/2000 located on the CD. The IMS Terminal is also necessary to upgrade the software in your LYNX Control Module. These updates will be posted to the IMS website at http://www.imshome.com/ as they are made available.

To install the IMS Terminal to your hard drive, insert the CD into your CD-ROM Drive. The CD should autostart to the IMS Main Index Page. If the CD does not autostart, click "Start > Run" and type "x:\IMS.exe" in the "Open" box and then click OK. **NOTE:** "x" is your CD ROM drive letter.

1) The IMS Main Index Page will be displayed.
2) Click the MicroLYNX icon in the upper right corner. This opens the LYNX Family Index Page.
3) Select IMS Terminal (Win9x) or IMS Terminal (WinNT).
4) Click SETUP in the Setup dialog box and follow the on-screen instructions.

Once the IMS Terminal is installed you may run the Setup.

1) Open the IMS Term by clicking Start>Programs>IMS Terminal>IMS Term.
2) Select or verify the Communications Port that you will be using with your MicroLYNX.
   a) Click in the Terminal Window to activate it.
   b) Right click in the Terminal Window.

c) Click “Preferences” in the dialog box.

d) Click the “Comm Settings” tab at the top of the dialog box.

e) Under “Device” near the bottom of the box verify “LYNX” is selected. The BAUD rate is already set to the MicroLYNX default. Do not change this setting until you have established communications with the MicroLYNX Controller.

f) The “Window Size” settings are strictly optional. You may set these to whatever size is comfortable to you.

g) Click “OK”. The setting will be saved automatically.

3) Apply power to the MicroLYNX Controller. The following sign-on message should appear in the Terminal window:

```
Program Copyright © 1996-2002 by:
Intelligent Motion Systems, Inc.
Marlborough, CT 06447
VER = xxxxx   SER = Axxxxx
```

NOTE: If the sign-on message does not appear, check the “Connected/Disconnected” tab at the bottom of the Terminal Window. If “Disconnected” is displayed, double click it to “Connect”.

Detailed instructions for the IMS LYNX Terminal software can be located in Part III “Software Reference” of this manual.

Testing the MicroLYNX Setup

Two basic instructions for communicating with a MicroLYNX are SET and PRINT. The SET instruction is assumed and can be left off when communicating in ASCII mode. (You are in ASCII mode whenever you are using a text based terminal.) It is used to set variables and flags that define MicroLYNX operation. The Software automatically recognizes the SET instruction whenever the name of the variable or flag is typed into the terminal. Here we will set the motor units variable (MUNIT) to 51200 by typing the following at the prompt (>):

```
MUNIT = 51200
```

The PRINT instruction is used to report the values of variables and flags. Now double-check the value of MUNIT by typing the following at the prompt (>):

```
PRINT MUNIT
```

The return from your terminal should be 51200. Note that the case is not important for instructions, variables and flags. They may be typed in upper or lower case.

The next step is to turn the motor. Before doing so, type in the following lines to configure the driver at the prompt. (Note: Command descriptions are found in the Software Reference section of the manual.)

```
MSEL=256
MAC=80
MRC=75
```

Use the SLEW instruction to move the motor at a constant velocity. Be sure that the velocity provided is a reasonable value for your motor and drive and try to move the motor. For instance, at the prompt type:

```
SLEW 10
```

This will move the motor at a speed of 10 munits per second. If the motor does not move, verify that the wiring is in accordance with *Figure 2.1*. If the wiring is determined to be correct, type:

```
PRINT ERROR
```

An error number other than zero (0) will be displayed. See *Appendix B* for more information.

Once you have been able to move the motor, the next step is to write a simple program to illustrate one of the dynamic features of the MicroLYNX: the ability to convert motor steps to a dimension of linear or rotary distance. Let’s begin by discussing the relationship between the MUNIT variable and user units. Typically, when we perform a move, we want to know the distance of that move in a familiar unit of measurement. That means translating motor steps to the desired unit of measurement. The MicroLYNX Controller has the capability of doing this for you. You have already set the motor units variable (MUNIT) to a value of 51200.

With the driver set to a resolution of 256 microsteps per step (MicroLYNX factory default) and a 1.8° step motor that will be equal to 1 revolution of the motor, or one USER UNIT. A user unit can be any unit of measure. At this point, by entering the instruction MOVR 1, the motor will turn one complete revolution relative to its current position. Therefore, 1 User Unit = 1 Motor Revolution. For the exercise below we will use degrees for our user unit. The calculation required to select degrees as our user unit in this example is:

51200 Microsteps per rev ÷ 360 degrees = 142.222 Microsteps per degree

By setting the MUNIT variable to 51200/360, the MicroLYNX will perform the calculation to convert the user unit to degrees.

Now, when a relative motion instruction “MOVR 90” is issued the motor will turn 90 degrees.

Let’s enter a sample program that will convert motor steps to degrees, execute a 90° move and report that move every 100 milliseconds while the motor is moving. Type the following bold commands:

```
‘Enter Program Mode, start program at Location 2000.
PGM 2000
‘Label the program TSTPGM.
LBL TSTPGM
‘Set the user units to degrees.
MUNIT = 51200/360
‘Set the max. velocity to 25 degrees per second.
VM = 25
‘Execute a relative move of 90 degrees.
MOVR 90
‘Report the position every 100 ms while moving.
LBL PRINTPOS
DELAY 100
PRINT “Axis position is”, POS, “Degrees.”
BR PRINTPOS, MVG
‘End the program.
END
PGM
```

Now Type **TSTPGM** to run program.
This sample program will be stored starting at location 2000. It sets the conversion factor for the user units, sets the maximum velocity and then starts a motion. While the motion is occurring, the position is reported every 100 milliseconds.

At this point you may desire to restore the settings to their factory default as you may not wish to use degrees as your user unit. To do this, you will use the CP, DVF and IP instructions.

CP - Clear Program.
To clear the program, type CP 1, 1. This will completely clear program memory space. Should you desire to only remove one program, the instruction “CP [Program Label]”, i.e. “CP TSTPGM”, would clear only the specified program. In this exercise only one program was entered, “CP TSTPGM” will clear it.

DVF - Delete User-Defined Variables and Flags.
By entering DVF, all of the user defined variables will be removed. Although no flags were set in this exercise, this command would clear them were they used.

IP - Initialize Parameters.
This instruction will restore all of the parameters to their factory default state.

After entering these instructions, a SAVE instruction should be entered.

Section 3

Installing and Mounting the MicroLYNX

Section Overview

This section covers the installation of the optional expansion modules and panel mount procedures for the MicroLYNX System.

- Dimensional information.
- Installation and removal of the optional expansion modules.
- Mounting the MicroLYNX System to a panel.

Dimensional Information

Dimensions in Inches (mm)

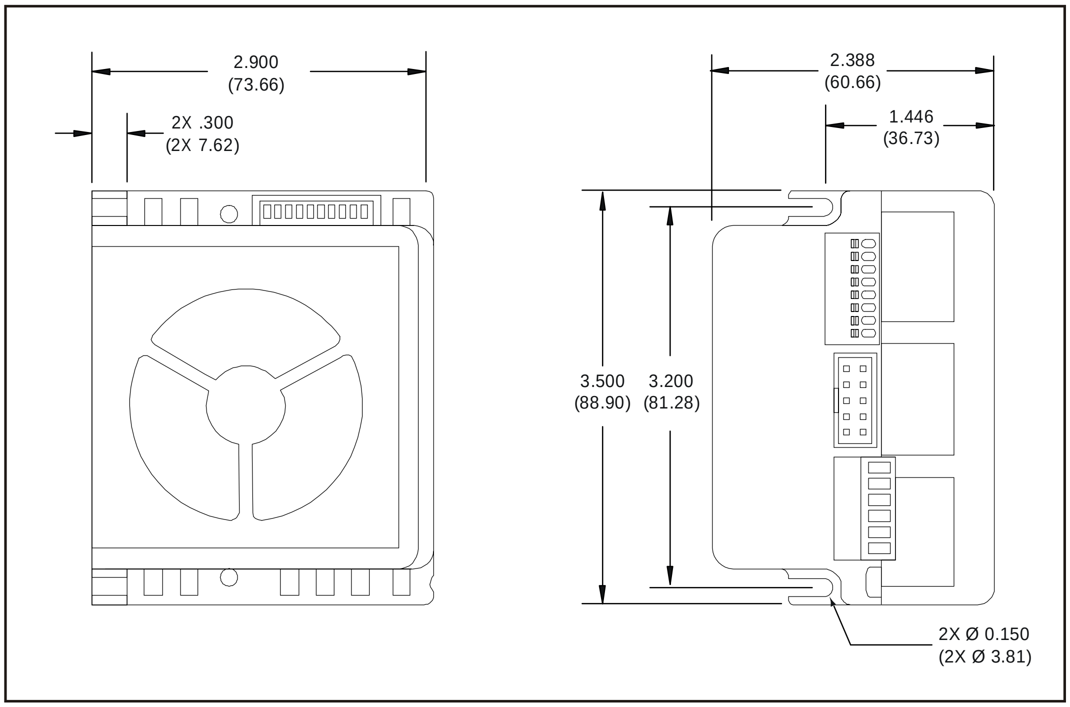


Figure 3.1: Dimensional Information

Installation and Removal of the Optional Expansion Modules

One of the powerful features of the MicroLYNX System is the extreme ease by which it can be configured and installed. There are three (3) bays in which expansion modules can be installed. The expansion modules can be plugged into any available slot, with some exceptions. (See Section 11, Table 11.1 for details.) For ease of configuration, ensure that the pull-up switches on the Isolated Digital I/O expansion module are in the desired position prior to closing and mounting the MicroLYNX System. See *Section 11: Configuring and Using the Optional Expansion Modules* for more information on this topic.

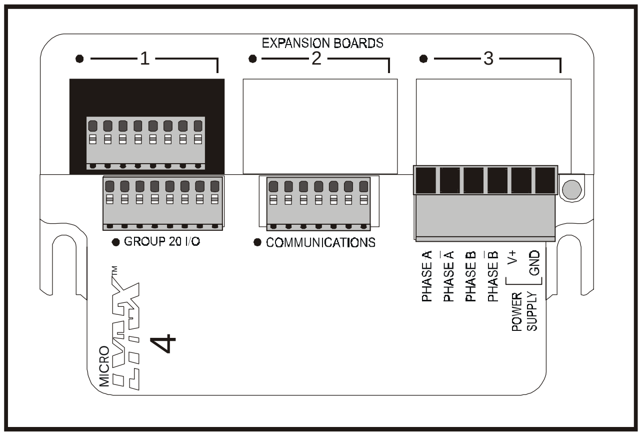


Figure 3.2: MicroLYNX System with the Isolated Digital I/O Expansion Module installed in Slot #1

To install the expansion modules, the only tool required is a phillips head screwdriver. The installation steps follow:

1) Remove the two screws [A] from the MicroLYNX case.
2) Remove the side of the case. (*See Figure 3.3*)
3) Remove the cover from the slot you will be using.
   NOTE: Some Expansion Modules may only be placed in certain slots. Please refer to Section 11 (Configuring and Using Expansion Modules).
4) Insert Expansion Module into open slot, seating until module snaps into place. (*See Figure 3.3*)
5) Replace the side of the case.
6) Insert and tighten screws.

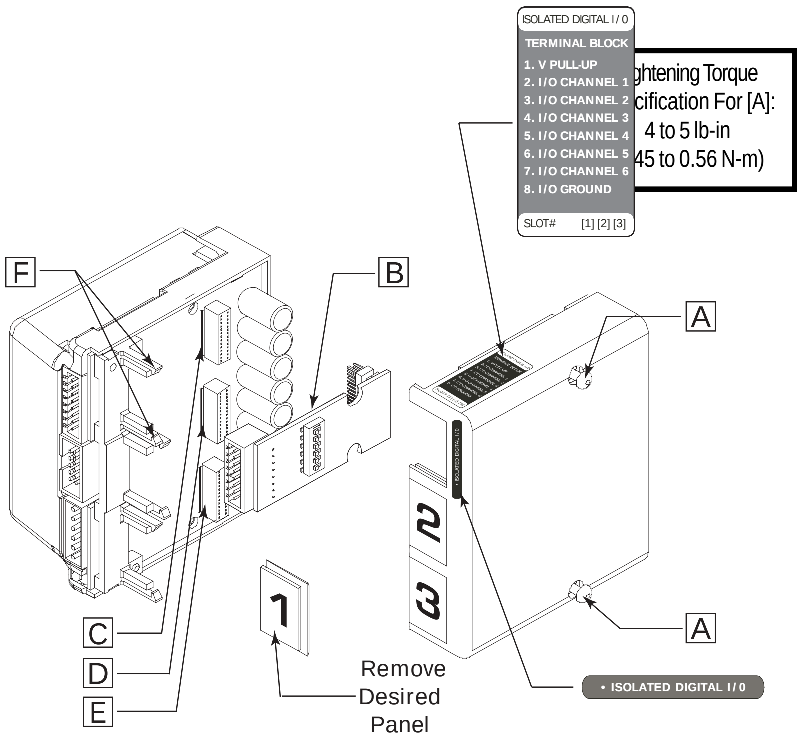


Figure 3.3: Installing the Optional Expansion Modules

Mounting the MicroLYNX System to a Panel

The MicroLYNX System can be mounted to a panel by using standard #6 hardware. As the system has a built-in cooling fan, no heat sinking is necessary. When mounting the MicroLYNX System in an enclosure, ensure that adequate space is available for air flow on the fan side of the MicroLYNX case. Mounting screws should be tightend to 5-7 lb-in torque.

WARNING! Ensure that there is a minimum 1.5" clearance on the fan side of the case for air flow.

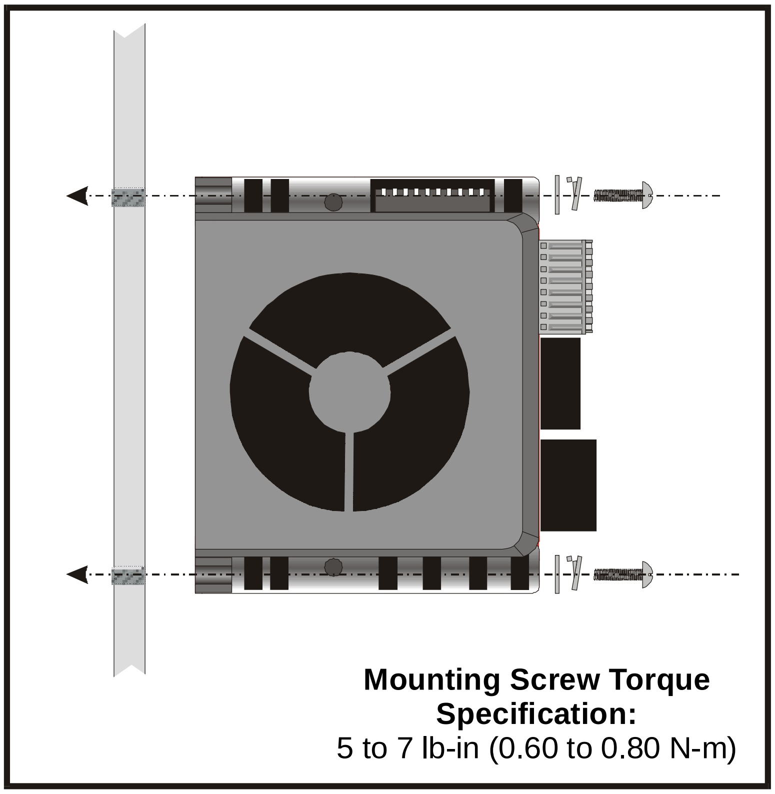

Mounting Screw Torque Specification:
5 to 7 lb-in (0.60 to 0.80 N-m)

Figure 3.4: Panel Mounting the MicroLYNX

Section 4

Powering the MicroLYNX System

Section Overview

This section covers the power requirements for your MicroLYNX System.

- Selecting a power supply.
- Basic rules of wiring and shielding.
- Power supply connection and requirements.
- Recommended power supplies.

Selecting a Power Supply

Selecting a Motor Supply (+V)

Proper selection of a power supply to be used in a motion system is as important as selecting the drive itself. When choosing a power supply for a stepping motor driver, there are several performance issues that must be addressed. An undersized power supply can lead to poor performance, and possibly even damage, to your MicroLYNX System.

The Power Supply - Motor Relationship

Motor windings can be basically viewed as inductors. Winding resistance and inductance result in an L/R time constant that resists the change in current. To effectively manipulate the rate of charge, the voltage applied is increased. When traveling at high speeds there is less time between steps to reach current. The point where the rate of commutation does not allow the driver to reach full current is referred to as Voltage Mode. Ideally you want to be in Current Mode, which is when the drive is achieving the desired current between steps. Simply stated, a higher voltage will decrease the time it takes to charge the coil and, therefore, will allow for higher torque at higher speeds.

Another characteristic of all motors is back EMF. Back EMF is a source of current that can push the output of a power supply beyond the maximum operating voltage of the driver and, as a result, could damage the stepper driver over a period of time.

The Power Supply - Driver Relationship

The MicroLYNX System is very current efficient as far as the power supply is concerned. Once the motor has charged one or both windings of the motor, all the power supply has to do is replace losses in the system. The charged winding acts as an energy storage in that the current will recirculate within the bridge, and in and out of each phase reservoir. This results in a less than expected current draw on the supply. Stepping motor drivers are designed with the intention that a user's power supply output will ramp up to greater or equal to the minimum operating voltage. The initial current surge is quite substantial and could damage the driver if the supply is undersized. The output of the power supply could fall below the operating range of the driver upon a current surge if it is undersized. This could cause the power supply to start oscillating in and out of the voltage range of the driver and result in damage to either the supply, the driver or both. There are two types of supplies commonly used, regulated and unregulated, both of which can be switching or linear. All have their advantages and disadvantages.

Regulated vs. Unregulated

An unregulated linear supply is less expensive and more resilient to current surges; however, the voltage decreases with increasing current draw. This can cause problems if the voltage drops below the working range of the drive. Also of concern are the fluctuations in line voltage. This can cause the unregulated linear supply to be above or below the anticipated or acceptable voltage.

A regulated supply maintains a stable output voltage, which is good for high speed performance. They are also not bothered by line fluctuations; however, they are more expensive. Depending on the current regulation, a regulated supply may crowbar or current clamp and lead to an oscillation that, as previously stated, can cause damage to the driver and/or supply. Back EMF can cause problems for regulated supplies as well. The current regeneration may be too large for the regulated supply to absorb. This could lead to an over voltage condition which could damage the output circuitry of the MicroLYNX System. Non IMS switching power supplies and regulated linear supplies with overcurrent protection are not recommended because of their inability to handle the surge currents inherent in stepping motor systems.

Wiring and Shielding

Noise is always present in a system that involves high power and small signal circuitry. Regardless of the power configuration that you use in your system, there are some wiring and shielding rules that you should follow to keep your noise-to-signal ratio as small as possible.

Rules of Wiring

- Power Supply and Motor wiring should be shielded twisted pair run separately from signal carrying wires.
- A minimum of 1 twist per inch is recommended.
- Motors wiring should be shielded twisted pairs using 20 gauge wire, or 18 gauge or better for distance greater than 5 feet.
- Power ground return should be as short as possible to established ground.
- Power Supply wiring should be shielded twisted pairs. Use 18 gauge wire if load is less than 4 amps, or 16 gauge for more than 4 amps.
- Do not “Daisy-Chain” power wiring to system components.

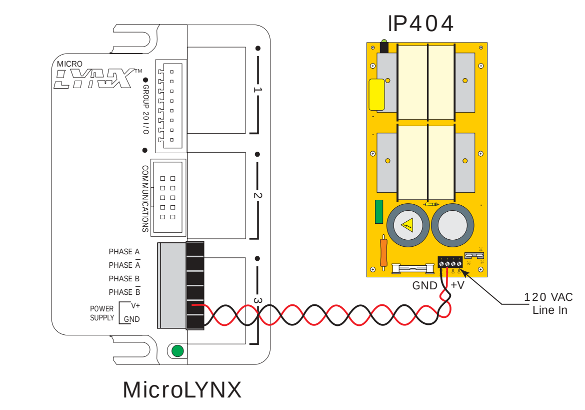
Rules of Shielding

- The shield must be tied to zero-signal reference potential. In order for shielding to be effective, it is necessary for the signal to be earthed or grounded.
- Do not assume that earth ground is true earth ground. Depending on the distance to the main power cabinet, it may be necessary to sink a ground rod at a critical location.
- The shield must be connected so that shield currents drain to signal-earth connections.
- The number of separate shields required in a system is equal to the number of independent signals being processed plus one for each power entrance.
- The shield should be tied to a single point to prevent ground loops.
- A second shield can be used over the primary shield; however, the second shield is tied to ground at both ends.

WARNING! When using an unregulated supply, ensure that the output voltage does not exceed the maximum driver input voltage due to variations in line voltage! It is recommended that an input line filter be used on power supply to limit voltage spikes to the system!

Power Supply Connection & Specification

Power is connected to the MicroLYNX via connector P1. All optional expansion boards are then powered from the MicroLYNX.



Ensure the DC output of the power supply does not exceed the maximum input voltage.

All power supply wiring should be a shielded, twisted pair to reduce system noise.

Figure 4.1: MicroLYNX Power Connection

Power Supply Specifications

Recommended Type	Unregulated DC
Ripple Voltage	±10%
MicroLYNX - 4	
Output Voltage	+12 to +48VDC
*Output Current	2 Amps (Typical)
MicroLYNX - 7	
Output Voltage	+24 to +75VDC
*Output Current	3.5 Amps (Typical)

** The output current needed is dependant on the supply voltage, motor selection and load.*

Recommended IMS Power Supplies

The IP404 and IP804 are low-cost non-regulated linear power supplies which can handle varying load conditions. They are available in either 120 or 240 VAC configuration.

IP404 (MicroLYNX-4)	
Input Range	
120 VAC Versions	102-132 VAC
240 VAC Versions	204-264 VAC
Output	
No Load Output Voltage*	43 VDC @ 0 Amps
Continuous Output Rating*	32 VDC @ 2 Amps
Peak Output Rating*	26 VDC @ 4 Amps
IP804 (MicroLYNX-7)	
Input Range	
120 VAC Versions	102-132 VAC
240 VAC Versions	204-264 VAC
Output	
No Load Output Voltage*	76 VDC @ 0 Amps
Continuous Output Rating*	65 VDC @ 2 Amps
Peak Output Rating*	58 VDC @ 4 Amps

** All measurements were taken at 25°C, 120 VAC, 60 Hz.*

Section 5

Motor Requirements

Section Overview

This section covers the motor configurations for the MicroLYNX-4/7.

- Selecting a motor.
- Motor wiring.
- Connecting the motor.

Selecting a Motor

When selecting a stepper motor for your application, there are several factors that need to be taken into consideration.

- How will the motor be coupled to the load?
- How much torque is required to move the load?
- How fast does the load need to move or accelerate?
- What degree of accuracy is required when positioning the load?

While determining the answers to these and other questions is beyond the scope of this document, they are details that you must know in order to select a motor that is appropriate for your application. These details will effect everything from the power supply voltage to the type and wiring configuration of your stepper motor, as well as the current and microstepping settings of your MicroLYNX System.

Types and Construction of Stepping Motors

The stepping motor, while classed as a DC motor, is actually an AC motor that is operated by trains of pulses. Though it is called a “stepping motor”, it is in reality a Polyphase Synchronous Motor. This means it has multiple phases wound in the stator and the rotor is dragged along in synchronism with the rotating magnetic field. The MicroLYNX System is designed to work with the following types of stepping motors:

1) Permanent Magnet (PM)
2) Hybrid Stepping Motors

Hybrid Stepping Motors combine the features of the PM Stepping Motors with the features of another type of stepping motor called a Variable Reluctance Motor (VR), which is a low torque and load capacity motor that is typically used in instrumentation. The MicroLYNX System cannot be used with VR motors as they have no permanent magnet.

On Hybrid motors, the phases are wound on toothed segments of the stator assembly. The rotor consists of a permanent magnet with a toothed outer surface which allows precision motion accurate to within ± 3 percent. Hybrid Stepping Motors are available with step angles varying from 0.45° to 15°, with 1.8° being the most commonly used. Torque capacity in hybrid steppers ranges from 5 - 8000 ounce-inches. Because of their smaller step angles, Hybrid motors have a higher degree of suitability in applications where precise load positioning and smooth motion is required.

Sizing a Motor for Your System

The MicroLYNX System contains a bipolar driver which works equally well with both bipolar and unipolar motors (i.e. 8 and 4 lead motors, and 6 lead center tapped motors).

To maintain a given set motor current, the MicroLYNX System chops the voltage using a constant 20kHz chopping frequency and a varying duty cycle. Duty cycles that exceed 50% can cause unstable chopping.

This characteristic is directly related to the motor's winding inductance. In order to avoid this situation, it is necessary to choose a motor with a low winding inductance. The lower the winding inductance, the higher the step rate possible.

Winding Inductance

Since the driver integrated into the MicroLYNX System is a constant current source, it is not necessary to use a motor that is rated at the same voltage as the supply voltage. What is important is that the MicroLYNX System is set to the motor's rated current. See *Section 6: Controlling The Output Current* for more details. As was discussed in the previous section, *Power Supply Requirements,* the higher the voltage used the faster the current can flow through the motor windings. This, in turn, means a higher step rate or motor speed. Care should be taken not to exceed the maximum voltage of the driver. Therefore, in choosing a motor for a system design, the best performance for a specified torque is a motor with the lowest possible winding inductance used in conjunction with highest possible driver voltage. The winding inductance will determine the motor type and wiring configuration best suited for your system. While the equation used to size a motor for your system is quite simple, several factors fall into play at this point. The winding inductance of a motor is rated in milliHenrys (mH) per Phase. The amount of inductance will depend on the wiring configuration of the motor.

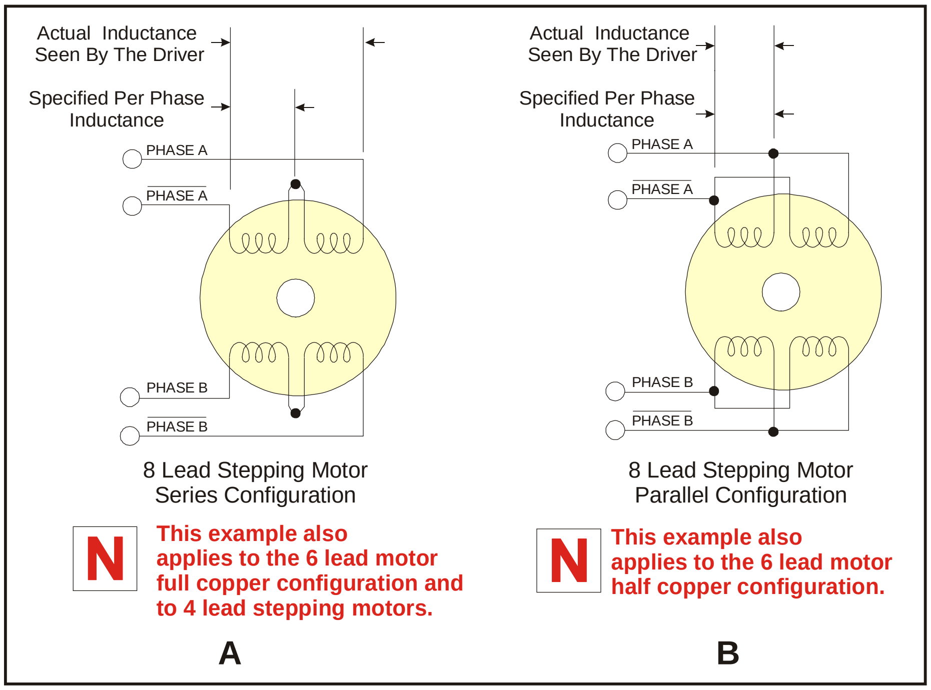
The per phase winding inductance specified may be different than the per phase inductance seen by your MicroLYNX System depending on the wiring configuration used. Your calculations must allow for the actual inductance that the driver will see based upon the motor's wiring configuration.

Figure 5.1A shows a stepper motor in a series configuration. In this configuration, the per phase inductance will be 4 times that specified. For example: a stepping motor has a specified per phase inductance of 1.47mH. In this configuration the driver will see 5.88 mH per phase.

Figure 5.1B shows an 8 lead motor wired in parallel. Using this configuration, the per phase inductance seen by the driver will be as specified. Using the following equation, we will show an example of sizing a motor for a MicroLYNX-4 used with an unregulated power supply with a minimum voltage (+V) of 18 VDC:

$$.2 \times 18 = 3.6 \text{ mH}$$

The maximum per phase winding inductance we can use is 3.6 mH.


Actual Inductance Seen By The Driver
Specified Per Phase Inductance
PHASE A
$\overline{\text{PHASE A}}$
PHASE B
$\overline{\text{PHASE B}}$
8 Lead Stepping Motor Series Configuration
N
A
Actual Inductance Seen By The Driver
Specified Per Phase Inductance
PHASE A
$\overline{\text{PHASE A}}$
PHASE B
$\overline{\text{PHASE B}}$
8 Lead Stepping Motor Parallel Configuration
N
B

This example also applies to the 6 lead motor full copper configuration and to 4 lead stepping motors.

This example also applies to the 6 lead motor half copper configuration.

Figure 5.1 A & B: Per Phase Winding Inductance

Maximum Motor Inductance (mH per Phase) =
.2 X Minimum Supply Voltage

NOTE: In calculating the maximum phase inductance, the minimum supply output voltage should be used when using an unregulated supply.

Recommended IMS Motors

IMS stocks the following Enhanced 1.8° Hybrid Stepping Motors that are recommended for the MicroLYNX System. These motors use a unique relationship between the rotor and stator to generate more torque per frame size while ensuring more precise positioning and increased accuracy. The special design allows the motors to provide higher torque than standard stepping motors while maintaining a steadier torque and reducing torque drop-off. The motors are available in 3 stack sizes, single or double shaft, with or without encoders. For more detailed information on these motors, please see the Product Data Sheets on the IMD CD or the IMS website at *http//:www.imshome.com/.*

17 Frame (MicroLYNX - 4)

Single Shaft	**Double Shaft**
M-1713-1.5S	M-1713-1.5D
M-1715-1.5S	M-1715-1.5D
M-1719-1.5S	M-1719-1.5D

23 Frame (MicroLYNX - 4)

Single Shaft	
M-2218-2.4S	
M-2222-2.4S	
M-2231-2.4S	
	Double Shaft
M-2218-3.0S	M-2218-3.0D
M-2222-3.0S	M-2222-3.0D
M-2231-3.0S	M-2231-3.0D

23 Frame (MicroLYNX - 7)

Single Shaft	**Double Shaft**
M-2218-6.0S	M-2218-6.0D
M-2222-6.0S	M-2222-6.0D
M-2231-6.0S	M-2231-6.0D

34 Frame (MicroLYNX - 7)

Single Shaft	**Double Shaft**
M-3424-6.3S	M-3424-6.3D
M-3431-6.3S	M-3417-6.3D
M-3447-6.3S	M-3447-6.3D

IMS Inside Out Stepper Motors

The new Inside Out Stepper (IOS) Motors were designed by IMS to bring versatility to small motors using a unique multi-functional, hollow-core design.
These versatile new motors can be converted to a ball screw linear actuator by mounting a miniature ball screw to the front shaft face. Ball screw linear actuators offer long life, high efficiency and can be field retrofitted. There is no need to throw the motor away due to wear of the nut or screw.

The IOS motors offer the following features:

- The shaft face diameter offers a wide choice of threaded hole patterns for coupling.
- The IOS motor can be direct coupled in applications within the torque range of the motor, eliminating couplings and increasing system efficiency.
- The IOS motor can replace gearboxes in applications where gearboxes are used for inertia dampening between the motor and the load. The induced backlash from the gearbox is eliminated providing improved bi-directional position accuracy.
- Electrical or pnuematic lines can be directed through the center of the motor enabling the motors to be stacked end-to-end or applied in robotic end effector applications. The through hole is stationary preventing cables from being chaffed by a moving hollow shaft.
- Light beams can be directed through the motor for refraction by a mirror or filter wheel mounted on the shaft mounting face.
- The IOS motor is adaptable to valves enabling the valve stem to protrude above the motor frame. The stem can be retrofitted with a dial indicator showing valve position.
- The motor is compatible with IMS bipolar drivers, keeping the system cost low.
- The IOS motor can operate up to 3000 rpm's.

The IOS motor is available in the following frames:

MicroLYNX-4/-7	
	IMS P/N
17 Frame	M3-1713-IOS
MicroLYNX-7	
23 Frame	M3-2220-IOS
34 Frame	M3-3424-IOS
42 Frame	M3-4247-IOS

Motor Wiring

As with the power supply wiring, motor wiring should be run separately from logic wiring to minimize noise coupled onto the logic signals. Motor cabling exceeding 1' in length should be shielded twisted pairs to reduce the transmission of EMI (Electromagnetic Interference) which can lead to rough motor operation and poor system performance overall. For more information on wiring and shielding, please refer to *Rules of Wiring and Shielding* in Section 4 of this manual.

NOTE: The physical direction of the motor with respect to the direction input will depend upon the connection of the motor windings. To switch the direction of the motor with respect to the direction input, switch the wires on either phase A or phase B outputs.

WARNING! Do not connect or disconnect motor or power leads with power applied!

WARNING! Motor rotation will be reversed when switching from a MicroLYNX-4 to a MicroLYNX-7. Make connections accordingly.

Following are the recommended motor cables:

Dual Twisted Pair Shielded (Separate Shields)

< 5 feet	Belden Part# 9402 or equivalent 20 Gauge
> 5 feet	Belden Part# 9368 or equivalent 18 Gauge

When using a bipolar motor, the motor must be within 100 feet of the drive.

Connecting the Motor

The motor leads are connected to the following connector pins, which are clearly labeled for ease of use:

Phase	**Pin**
Phase B\	4
Phase B	3
Phase A\	2
Phase A	1

8 Lead Motors

For the systen designer, 8 lead motors offer a high degree of flexibility in that they may be connected in series or parallel, thus satisfying a wide range of applications.

Series Connection

A series motor configuration would typically be used in applications where a higher torque at low speeds is needed. Because this configuration has the most inductance, the performance will start to degrade at higher speeds. Use the per phase (or unipolar) current rating as the peak output current, or multiply the bipolar current rating by 1.4 to determine the peak output current.

Parallel Connection

An 8 lead motor in a parallel configuration offers more stability but lower torque at lower speeds, but because of the lower inductance there will be higher torque at higher speeds. Multiply the per phase (or unipolar) current rating by 1.96, or the bipolar current rating by 1.4 to determine the peak output current.

WARNING! Motor rotation will be reversed when switching from a MicroLYNX-4 to a MicroLYNX-7. Make connections accordingly

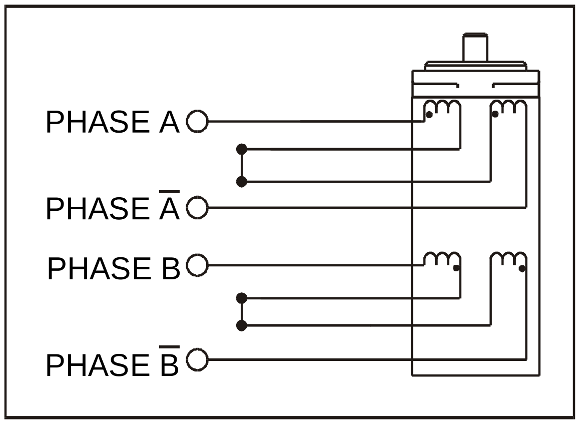


Figure 5.2: 8 Lead Motor, Series Connection

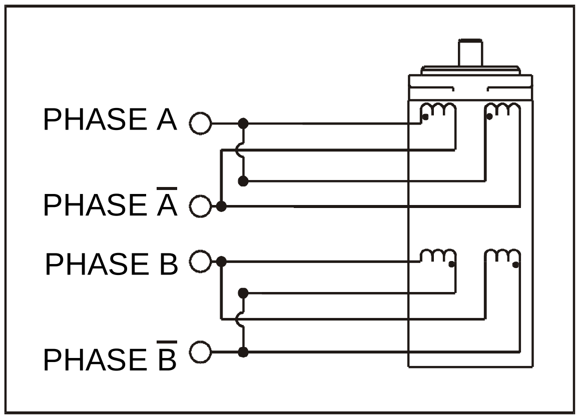


Figure 5.3: 8 Lead Motor, Parallel Connection

6 Lead Motors

As with 8 lead stepping motors, 6 lead motors have two configurations available for high speed or high torque operation. The higher speed configuration, or *half coil, is* so described because it uses one half of the motor's inductor windings. The higher torque configuration, or *full coil,* uses the full windings of the phases.

Half Coil Configuration

As previously stated, the half coil configuration uses 50% of the motor phase windings. This gives lower inductance, hence, lower torque output. As with the parallel connection of 8 lead motor, the torque output will be more stable at higher speeds. This configuration is also referred to as *half copper*. In setting the driver output current, multiply the specified per phase (or unipolar) current rating by 1.4 to determine the peak output current.

Full Coil Configuration

The full coil configuration on a 6 lead motor should be used in applications where higher torque at lower speeds is desired. This configuration is also referred to as *full copper.* Use the per phase (or unipolar) current rating as the peak output current.

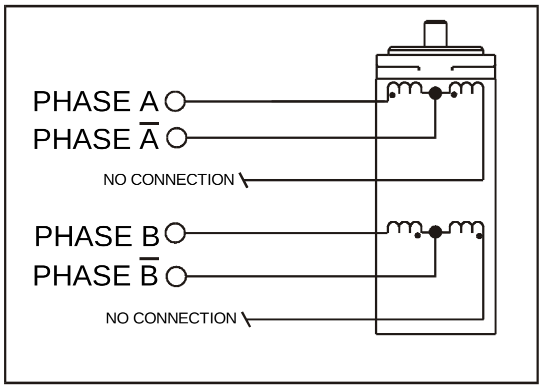


Figure 5.4: 6 Lead Motor, Half Coil Connection

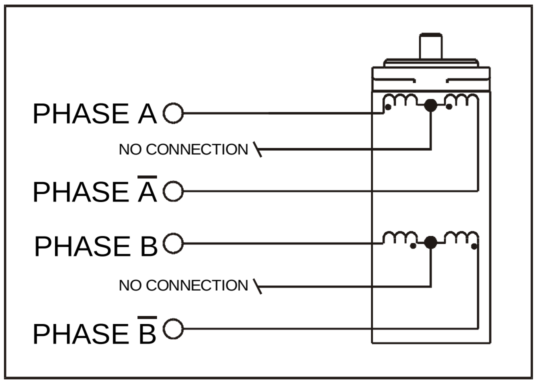


Figure 5.5: 6 Lead Motor, Full Coil Connection

4 Lead Motors

4 lead motors are the least flexible but easiest to wire. Speed and torque will depend on winding inductance. In setting the driver output current, multiply the specified phase current by 1.4 to determine the peak output current.

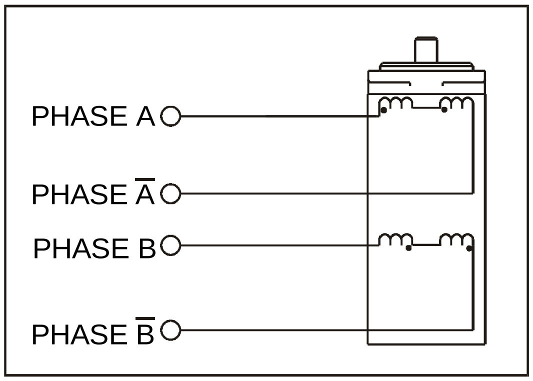


Figure 5.6: 4 Lead Motor

Section 6

Controlling the Output Current and Resolution

Section Overview

This section covers the following current control features of the MicroLYNX System:

- Current control variables.
- Determining the output current.
- Setting the output current.
- Setting the motor resolution.

Current Control Variables

One of the unique and powerful features of the MicroLYNX is the precision current control available through the instruction set. Unlike most stepper drives, which only offer the capability of controlling run current and hold current, the MicroLYNX also has the capability of setting the acceleration current. By setting the acceleration current to a higher value, the system designer can deliver more power to the system at the time when it is needed the most: when system inertia must be overcome. Afterwards, when the motor has reached peak velocity, the run current can be set to a lower value, thus reducing motor heating and improving system power efficiency. See *Figure 6.1 and Table 6.1* for the current control variables.

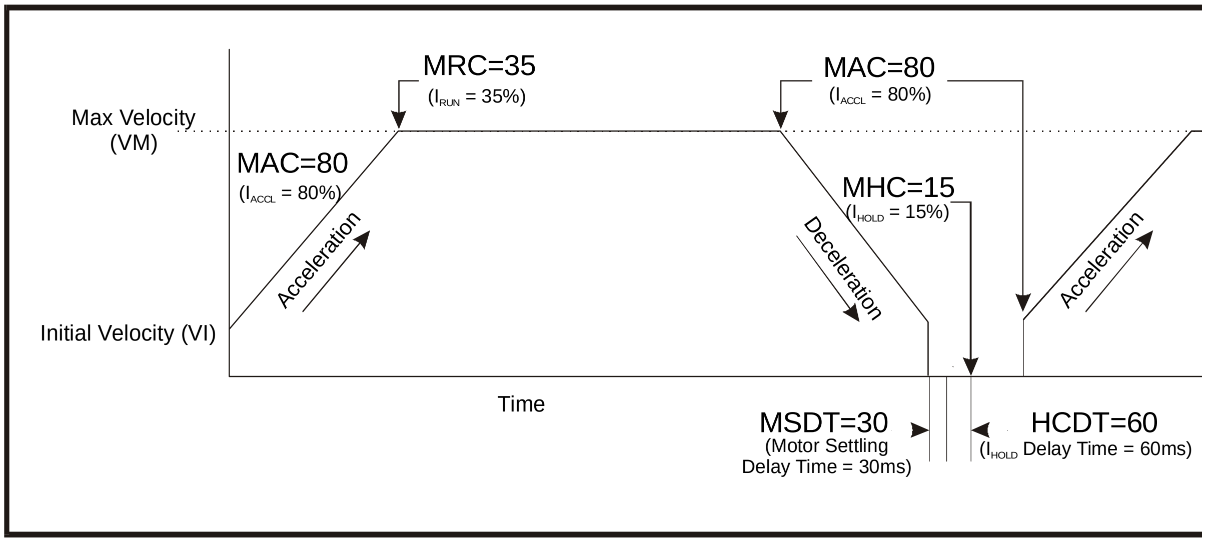


Figure 6.1: Motor Current Control Variables (Values set are for illustration purposes only)

Current Control Variable Summary					
Variable	**Function**	**Usage**	**Unit**	**Range**	**Default Value**
MAC	Motor Acceleration Current Setting	MAC=<num>	Percent	0 - 100	25
MRC	Motor Run Current Setting	MRC=<num>	Percent	0 - 100	25
MHC	Motor Hold Current Setting	MHC=<num>	Percent	0 - 100	5
HCDT	Hold Current Delay Time	HCDT=<num>	Time in milliseconds	0 - 65535	0
MSDT	Motor Settling Delay Time	MSDT=<num>	Time in milliseconds	0 - 65535	0
PMHCC	Position Maintenance Hold Current Change	PMHCC=<num>	Percent	0 - MHC	0

Table 6.1: Motor Current Control Variables

Determining the Output Current

Stepper motors can be configured as 4, 6 or 8 leads. Each configuration requires different currents. Shown below are the different lead configurations and the procedures to determine the peak per phase output current setting that would be used with different motor/lead configurations.

4 Lead Motors

Multiply the specified phase current by 1.4 to determine the peak output current.

Example:
A 4 Lead motor has a specified phase current of 2.0A:

2.0A X 1.4 = 2.8 Amps Peak

6 Lead Motors

A 6 lead motor can be configured two ways: in either the Half Coil Configuration (high speed) or the Full Coil Configuration (higher torque). The current calculation is different for each configuration.

Half Coil Configuration

When configuring a 6 lead motor in the half coil configuration (connected from one end of the coil to the center-tap), multiply the specified per phase (or unipolar) current rating by 1.4 to determine the peak output current.

Example:
A 6 lead motor in half coil configuration has a specified phase current of 3.0A.

3.0A X 1.4 = 4.2 Amps Peak

Full Coil Configuration

When configuring the motor so that full coil is used (connected from end-to-end with the center-tap floating) use the per phase (or unipolar) current rating as the peak output current.

Example:
A 6 lead motor in full coil configuration has a specified phase current of 3.0A.

3.0A per phase = 3.0 Amps Peak

8 Lead Motors

Series Configuration

When configuring the motor windings in series, use the per phase (or unipolar) current rating as the peak output current, or multiply the bipolar current rating by 1.4 to determine the peak output current.

Example #1:
An 8 lead motor in series configuration has a specified unipolar current of 3.0A.

3.0A per phase = 3.0 Amps Peak

Example #2:
An 8 lead motor in series configuration with a specified bipolar current of 2.8A.

2.8 X 1.4 = 3.92 Amps Peak

Parallel Configuration

When configuring the motor windings in parallel, multiply the per phase (or unipolar) current rating by 2.0, or the bipolar current rating by 1.4 to determine the peak output current.

Example #1:
An 8 lead motor in parallel configuration has a specified unipolar current of 2.0A.

2.0A per phase X 2.0 = 4.0 Amps Peak

Example #2:
An 8 lead motor in parallel configuration with a specified bipolar current of 2.8A.

2.8 X 1.4 = 3.92 Amps Peak

Setting the Output Current

The output current on the MicroLYNX is set in software. As previously mentioned, the MicroLYNX differs from other step motor drivers in that the acceleration current can also be set in addition to the run current and holding current.

There are 3 variables in the MicroLYNX instruction set to set these current values:

MAC: Motor Acceleration Current
This value will be used by the MicroLYNX whenever velocity is changing, therefore it will also be the value used when the motor is decelerating.

MRC: Motor Run Current
This value will be used by the MicroLYNX whenever the motor is at peak velocity.

MHC: Motor Holding Current
This value will be used by the MicroLYNX when motion has ceased. The MicroLYNX will change to the hold current setting AFTER the time specified by the MSDT and HCDT variables.

(See *Figure 6.1 and Table 6.1* in the beginning of this section for more detail on these variables and their setup.)

Example Current Setting

For purpose of example we will set the acceleration current to 80%, the run current to 45%, and the holding current to 15%. We will allow the motor 2 seconds to settle into place and delay .5 seconds before reducing the current to the holding value.

```
MAC=80
MRC=45
MHC=15
MSDT=2000
HCDT=500
```

Setting the Motor Resolution

The output resolution of the drive section of the MicroLYNX is set by the MSEL variable. By viewing the table on the right, you can see that there are fourteen (14) resolution settings available with the MicroLYNX. These settings may be changed on-the-fly in either immediate mode or in a program. The operation of this variable is illustrated in the following exercise.
In this excercise we will write a short program that will simply slew the motor and cycle through a few of the binary microstep resolution settings. The lower the resolution is, the higher the speed of the motor.
Enter the following program into the text editor window of the IMS Terminal:

Microstep Resolution Settings	
MSEL Parameter (Microsteps/Step)	Microsteps/Rev
Binary Microstep Resolution Settings (1.8° Motor)	
2	400
4	800
8	1,600
16	3,200
32	6,400
64	12,800
128	25,600
256	51,200
Decimal Microstep Resolution Settings (1.8° Motor)	
5	1,000
10	2,000
25	5,000
50	10,000
125	25,000
250	50,000

Table 6.2: Microstep Resolution Settings

```
MAC=100'set acceleration current to 75%
MRC=100'set run current to 75%

PGM 200'start program at address 200
 SLEW 8000     'slew the motor at 4000 munits/sec
 HOLD 1        'suspend prog. until velocity change
completes
 MSEL=128      'set resolution to 128 msteps/step
 DELAY 1000    'delay program 1 sec.
 MSEL=64       'set resolution to 64 msteps/step
 DELAY 1000    'delay program 1 sec.
 MSEL=32       'set resolution to 32 msteps/step
 DELAY 1000    'delay program 1 sec.
 MSEL=16       'set resolution to 16 msteps/step
 DELAY 1000    'delay program 1 sec.
 MSEL=8        'set resolution to 8 msteps/step
 DELAY 10000
END
PGM
```

Transfer the program to the MicroLYNX by clicking the menu item “Transfer > Download” and selecting “Edit window” as the source. Run the program by typing “EXEC 200” in the terminal. The motor should speed up as it cycles through the resolution setting.

Section 7

The Communications Interface

Section Overview

The basic MicroLYNX features two communication interfaces: RS-232 and RS-485. For both channels the BAUD rate is software configured using the BAUD variable to 4800, 9600, 19200 or 38400 bits/sec. The factory default is set to 9600 bits/sec. Default data settings are 8 data bits, 1 stop bit and no parity.

A host computer can be connected to either interface to provide commands to the MicroLYNX System or to multiple MicroLYNX nodes in a system. Since most personal computers are equipped with an RS-232 serial port, it is most common to use the RS-232 interface for communications from the host computer to the MicroLYNX. You will typically want to use this interface option if your Host PC will be within 50 feet of your system. Should your system design place the MicroLYNX at a distance greater than 50 feet, it will be necessary for you to use the RS-485 interface option. You can accomplish this by using either an RS-232 to RS-485 converter, such as the converter sold by IMS (Part # CV-3222), or installing an RS-485 board in an open slot in your host PC.

Covered in detail in this section are:

- RS-232 interface, single MicroLYNX System.
- RS-232 interface, multiple MicroLYNX System.
- RS-485 interface, single MicroLYNX Interface.
- RS-485 interface, multiple MicroLYNX System.
- MicroLYNX modes of operation.
- MicroLYNX module communication modes.

Connecting the RS-232 Interface

Single MicroLYNX System

In systems with a single MicroLYNX, also referred to as Single Mode, the MicroLYNX is connected directly to a free serial port of the Host PC. Wiring and connection should be performed in accordance with the following table and diagram. In this mode the PARTY ADDRESS switches will be in the OFF position and the PARTY Flag will be set to 0 in software. This is the factory default setting. Please be aware that you cannot communicate with the MicroLYNX in single mode unless those conditions exist.

WARNING! Failure to connect communications ground as shown may result in damage to the Control Module and/or Host!

NOTE! If using the RS-232 Interface Option, the Host PC MUST be less than 50 feet from the Control Module. If your system will be greater than 50 feet from the Host PC you must use the RS-485 interface.

RS-232 MicroLYNX Connection			
MicroLYNX		PC	
10 Pin Header	7 Pin Phoenix	25 Pin Serial Port	9 Pin Serial Port
Pin 3 Receive Data (RX)	Pin 1 Receive Data (RX)	Pin 2 Transmit Data (TX)	Pin 3 Transmit Data (TX)
Pin 2 Transmit Data (TX)	Pin 2 Transmit Data (TX)	Pin 3 Receive Data (RX)	Pin 2 Receive Data (RX)
Pin 5 CGND	Pin 6 CGND	Pin 7 CGND	Pin 5 CGND

Table 7.1: Wiring Connections, RS-232 Interface, Single MicroLYNX System

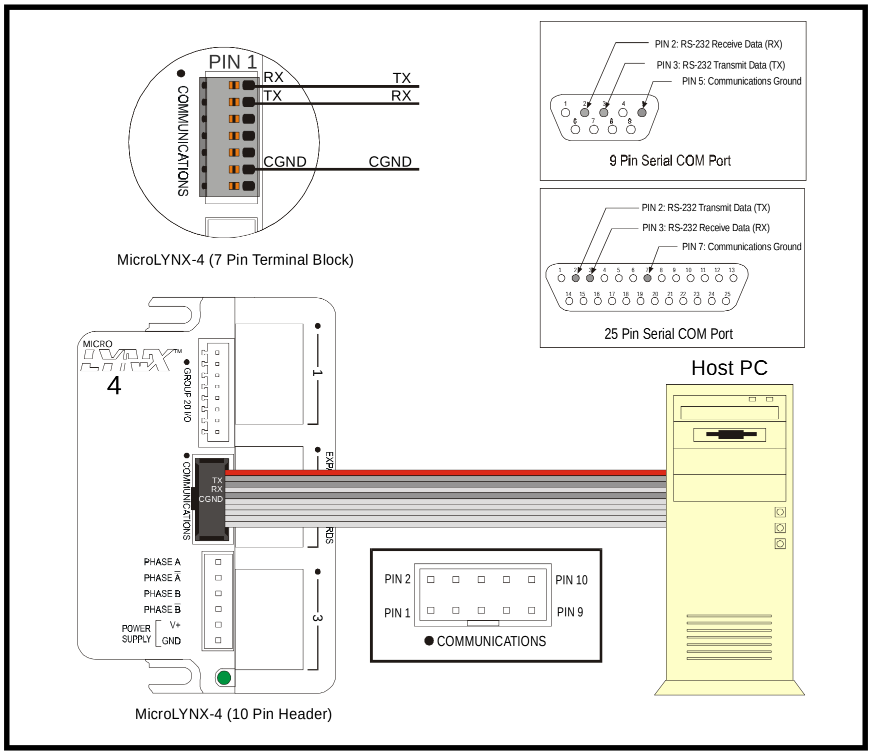

PIN 1
COMMUNICATIONS
RX
TX
CGND
TX
RX
CGND
MicroLYNX-4 (7 Pin Terminal Block)
PIN 2: RS-232 Receive Data (RX)
PIN 3: RS-232 Transmit Data (TX)
PIN 5: Communications Ground
9 Pin Serial COM Port
PIN 2: RS-232 Transmit Data (TX)
PIN 3: RS-232 Receive Data (RX)
PIN 7: Communications Ground
25 Pin Serial COM Port
Host PC
MICRO LYNX 4
GROUP 20 I/O
COMMUNICATIONS
TX
RX
CGND
PHASE A
PHASE A
PHASE B
PHASE B
POWER SUPPLY
V+
GND
PIN 2
PIN 10
PIN 1
PIN 9
COMMUNICATIONS
MicroLYNX-4 (10 Pin Header)

Figure 7.1: Connecting the RS-232 Interface, Single MicroLYNX System

Multiple MicroLYNX Systems

When connecting multiple MicroLYNX nodes in a system using the RS-232 interface it is necessary to establish one MicroLYNX as the HOST. This MicroLYNX will be connected to the Host PC exactly as the system using a single MicroLYNX. The system HOST is established by setting the HOST Flag to True (1) in software. The remaining MicroLYNX nodes in the system must then be connected to the HOST MicroLYNX using the RS-485 interface and will have their HOST Flag set to 0.

In this interface configuration, Host PC communications will be received by the Host MicroLYNX via RS-232 and forwarded to all of the other MicroLYNX nodes in the system via the RS-485 channel. Responses from the individual nodes in the system will be routed back to the Host MicroLYNX via the RS-485 channel, then internally converted to RS-232 before being forwarded back to the Host PC.

In systems with multiple MicroLYNX nodes, it is necessary to communicate with the Host MicroLYNX using PARTY Mode of operation. The MicroLYNX nodes in the system are configured for this mode of operation by setting a unique party address using the address switches, or setting the PARTY Flag to True (1) in software. It is necessary for all of the system nodes to have this configuration selected. When operating in

PARTY Mode, each MicroLYNX in the system will need a unique address to identify it in the system. This can be done using configuration switches A0-A2, or by using the software command SET DN. For example, to set the name of a controller to "A" you would use the following command: SET DN = "A". The factory default name is "!". To set the address with the configuration switches use the following table:

Party Mode Address Configuration Switches			
Address	**A2**	**A1**	**A0**
None	OFF	OFF	OFF
A	OFF	OFF	ON
B	OFF	ON	OFF
C	OFF	ON	ON
D	ON	OFF	OFF
E	ON	OFF	ON
F	ON	ON	OFF
G	ON	ON	ON

Table 7.2: Party Mode Address Configuration Switch Settings

The party address switches provide the simplest means for setting up PARTY operation for up to seven (7) MicroLYNX. In setting up your system for PARTY operation via software, the most practical approach would be to observe the following steps:

1. Connect the Host MicroLYNX to the Host PC configured for single mode operation.
2. Establish communications with the HOST MicroLYNX. (For help in doing this, see Software Reference: Using the IMS Terminal.) Using the Command: SET DN or the configuration switches; give the controller a unique name. If using the software command, this can be any upper or lower case ASCII character or number 0-9. Save the name using the command SAVE.
3. Set the appropriate HOST and PARTY configuration in accordance with Table 7.3 and Figure 7.2. Remove power.
4. Connect the next MicroLYNX in the system in accordance with Table 7.3 and Figure 7.2, setting the Binary switches A0-A2 to select an address A-G. (See Table 7.2)
5. Apply power to the system and establish communications with this module using its unique I.D. according to Table 7.2.
6. Repeat the last two steps for each additional MicroLYNX in the system. Ensure a different address is used for each additional unit.

WARNING! Failure to connect communications ground as shown may result in damage to the Control Module and/or Host!

NOTE! If using the RS-232 Interface Option, the Host PC MUST be less than 50 feet from the Control Module. If your system will be greater than 50 feet from the Host PC you must use the RS-485/RS-485 Interface.

Data Cable Termination Resistors

Data Cable lengths greater than 15 feet (4.5 meters) are susceptible to signal reflection and/or noise. IMS recommends 120Ω termination resistors at both ends of the Data Cables. An example of resistor placement is shown in Figures 7.2 and 7.5. For systems with Data Cables 15 feet (4.5 meters) or less, the termination resistors are generally not required. (For more information and other RS485 termination techniques, search the Internet for "RS485 Application Notes".)

Multiple MicroLYNX System								
Host MicroLYNX			MicroLYNX #1			MicroLYNX #*n*		
10 Pin Header	7 Pin Phoenix	Signal	10 Pin Header	7 Pin Phoenix	Signal	10 Pin Header	7 Pin Phoenix	Signal
6	4	RX+	9	7	TX+	9	7	TX+
7	3	RX-	8	5	TX-	8	5	TX-
8	5	TX-	7	3	RX-	7	3	RX-
9	7	TX+	6	4	RX+	6	4	RX+
10, 5	6	CGND	10, 5	6	CGND	10, 5	6	CGND
Software/Switch Setting			Software/Switch Setting			Software/Switch Setting		
HOST Flag = 1			HOST Flag = 0			HOST Flag = 0		
PARTY Flag = 1 A2, A1, A0 Address Set OR DN=<char>			PARTY Flag = 1 A2, A1, A0 Address Set OR DN=<char>			PARTY Flag = 1 A2, A1, A0 Address Set OR DN=<char>		

Table 7.3: Connections and Settings, RS-232 Interface, Multiple MicroLYNX System

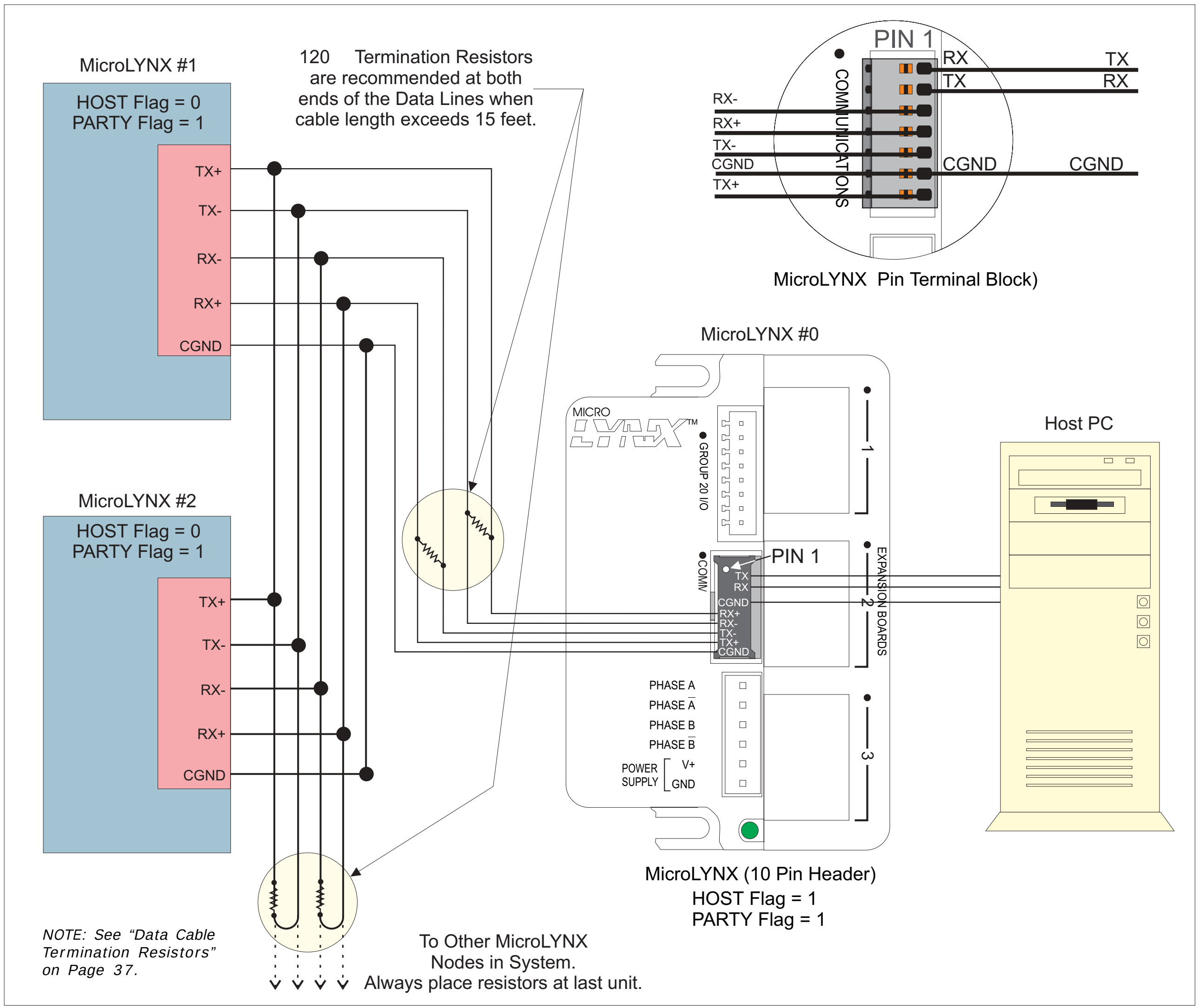


Figure 7.2: RS-232 Interface, Multiple MicroLYNX System

NOTE: See "Data Cable Termination Resistors" on Page 2-37 for details.

Connecting the RS-485 Interface

Single MicroLYNX System

In a Single Controller System, the RS-485 interface option would be used if the Micro-LYNX is located at a distance greater than 50 feet from the Host PC. Since most PC's do not come with an RS-485 board pre-installed, you will have to install an RS-485 board in an open slot in your PC, or purchase an RS-232 to RS-485 converter such as the CV-3222 sold by IMS, to use this connection interface. For wiring and connection information, please use the following table and diagram:

RS-485 Interface Single MicroLYNX System			
RS-232 to RS-485 Converter	MicroLYNX		
Signal	Signal	10 Pin Header	7 Pin Terminal
TX-	RX-	7	3
TX+	RX+	6	4
RX-	TX-	8	5
RX+	TX+	9	7
CGND	CGND	10, 5	6

Table 7.4: RS-485 Interface Connections

NOTE! The HOST Flag MUST be 0 to communicate with the MicroLYNX System in a Single MicroLYNX System using the RS-485 Interface.

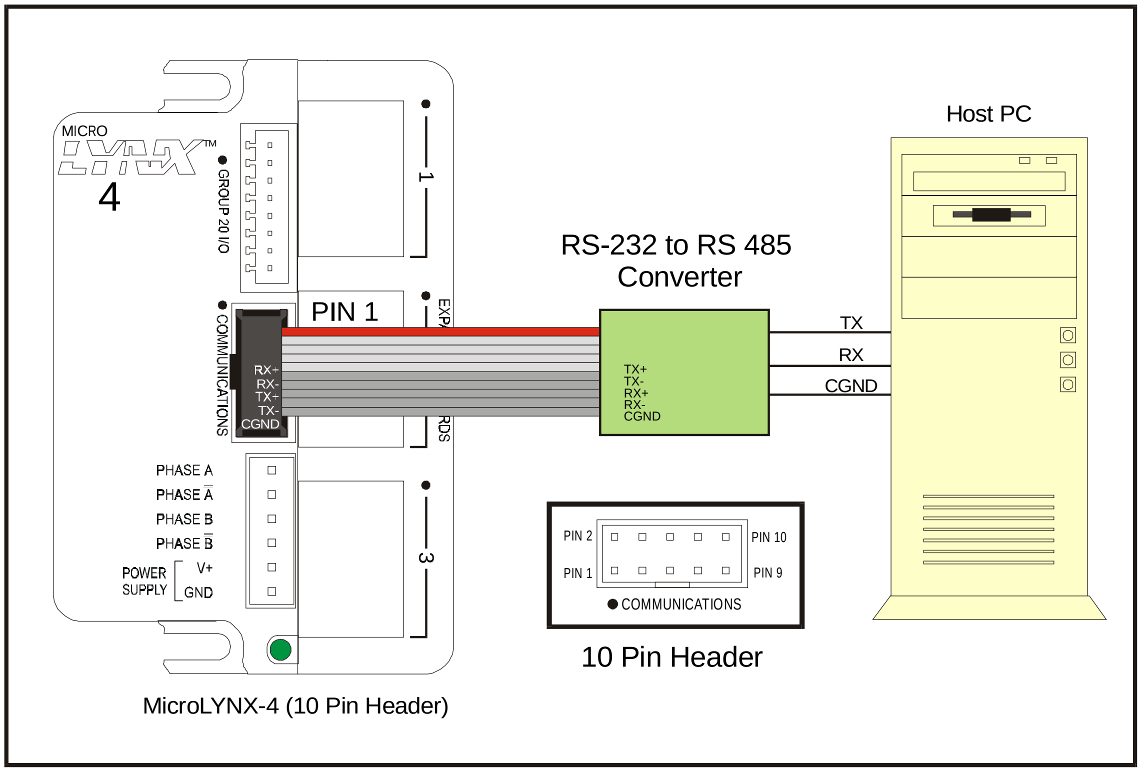


Figure 7.3: RS-485 Interface, Single MicroLYNX System

Half Duplex Set

If you are using a two wire RS-485 system, the MicroLYNX may be configured as shown in Figure 7.4.

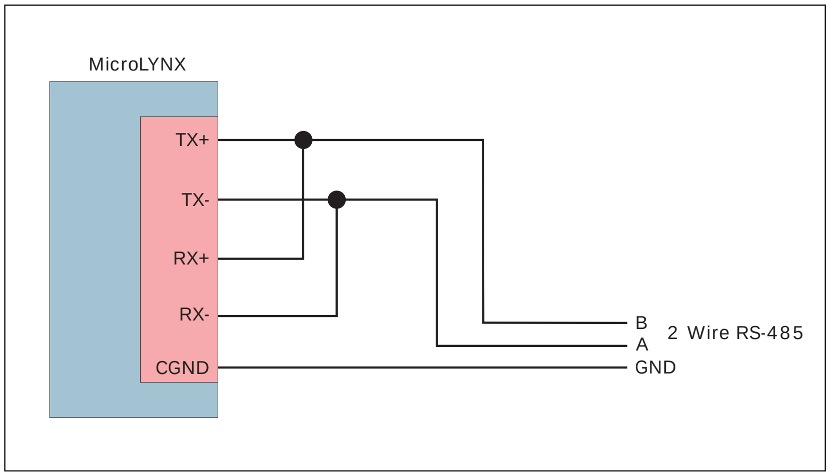


Figure 7.4: Half Duplex Set, Single MicroLYNX System

Multiple MicroLYNX System

When using the RS-485 interface in a Multiple MicroLYNX System, it is not necessary for a MicroLYNX to be set as the Host when connected to a PC. All MicroLYNX nodes in the system should have their HOST flag set to False (0) (Factory Default). The Host PC will be equipped with an RS-485 board or RS-232 to RS-485 converter. In systems with multiple MicroLYNX nodes, it is necessary to communicate with the system nodes using PARTY Mode of operation. The MicroLYNX nodes in the system are configured for this mode of operation by setting the Party Address Switches as shown in Table 7.5. It is necessary for all of the nodes in a system to have this configuration selected. When operating in PARTY Mode, each MicroLYNX node in the system requires a unique address (name) to identify it in the system. This can be done using configuration switches A0-A2, or by using the software command SET DN. For example, to set the name of a controller to “A” you would use the following command: SET DN = “A”. The factory default name is “!”. To set the address of the controller using the configuration switches use Table 7.5.

Party Mode Address Configuration Switches			
Address	**A2**	**A1**	**A0**
None	OFF	OFF	OFF
A	OFF	OFF	ON
B	OFF	ON	OFF
C	OFF	ON	ON
D	ON	OFF	OFF
E	ON	OFF	ON
F	ON	ON	OFF
G	ON	ON	ON

Table 7.5: Party Mode Address Configuration Switch Settings

Multiple MicroLYNX System RS-485 Interface						
RS-232 to RS-485 Converter or RS-485 Board	MicroLYNX #1			MicroLYNX #*n*		
Signal	10 Pin Header	7 Pin Phoenix	Signal	10 Pin Header	7 Pin Phoenix	Signal
RX+	9	7	TX+	9	7	TX+
RX-	8	5	TX-	8	5	TX-
TX-	7	3	RX-	7	3	RX-
TX+	6	4	RX+	6	4	RX+
CGND	10, 5	6	CGND	10, 5	6	CGND
	Software/Switch Setting			Software/Switch Setting		
	HOST Flag = 0			HOST Flag = 0		
	PARTY Flag = 1 A2, A1, A0 Address Set OR DN=<char>			PARTY Flag = 1 A2, A1, A0 Address Set OR DN=<char>		

Table 7.6: RS-485 Interface Connections and Settings, Multiple MicroLYNX System

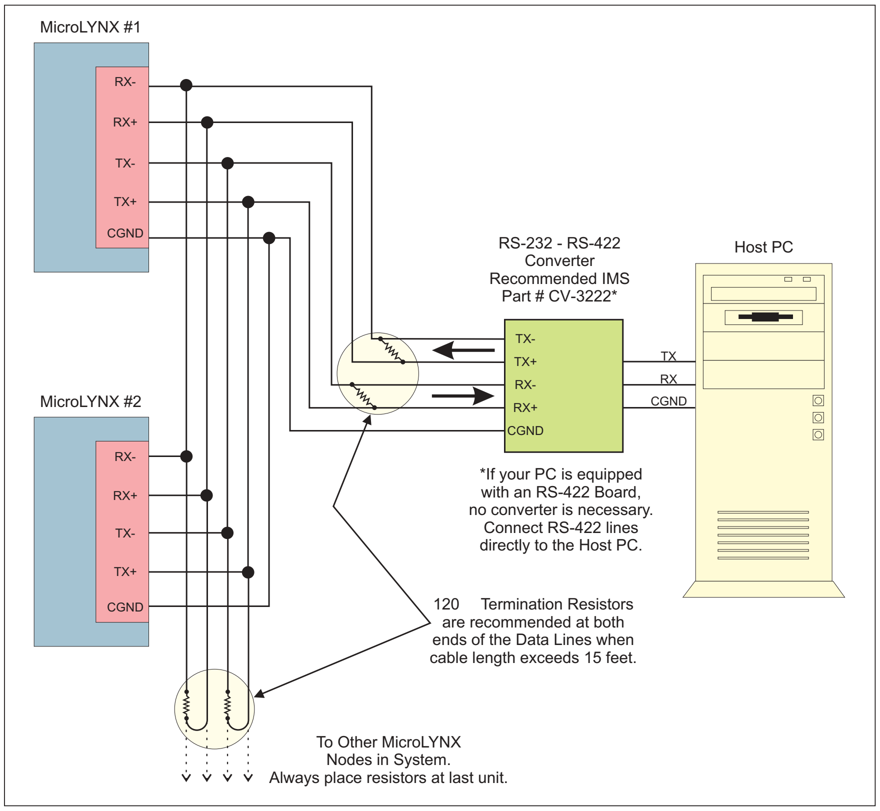

MicroLYNX #1
RX-
RX+
TX-
TX+
CGND
RS-232 - RS-422
Converter
Recommended IMS
Part # CV-3222*
Host PC
TX-
TX+
RX-
RX+
CGND
TX
RX
CGND
MicroLYNX #2
RX-
RX+
TX-
TX+
CGND
*If your PC is equipped with an RS-422 Board, no converter is necessary. Connect RS-422 lines directly to the Host PC.
120 Termination Resistors are recommended at both ends of the Data Lines when cable length exceeds 15 feet.
To Other MicroLYNX Nodes in System.
Always place resistors at last unit.

Figure 7.5: RS-485 Interface, Multiple MicroLYNX System

NOTE: See “Data Cable Termination Resistors” on Page 2-37 for details.

It is also possible to communicate with a MicroLYNX in the system in single mode by sending it a command (with address) to clear the party flag and then communicate with it as in single mode (no line feed terminator) then reset the PARTY Flag when done.

MicroLYNX Modes of Operation

There are three modes of operation for the MicroLYNX. These are Immediate Mode, Program Mode and EXEC Mode.

Immediate Mode

In this mode the MicroLYNX responds to instructions from the user that may be a result of the user typing instructions directly into a host terminal, or of a user program running on the host which communicates with the MicroLYNX.

Program Mode

The second mode of operation of the MicroLYNX is Program Mode. All user programs are entered in this mode. Unlike the other modes of operation, no commands or instructions can be issued to the MicroLYNX in Immediate Mode. This mode is exclusively for entering programs for the MicroLYNX. The command to enter Program Mode is PGM <address>. When starting Program Mode, you must specify at what address to enter the program instructions in the program space. Simply type PGM again when you have finished entering your program commands to go back to Immediate Mode.

EXEC Mode

In EXEC Mode a program is executed either in response to the EXEC instruction from the user in Immediate Mode, or in response to a specified input. While the MicroLYNX is running a program, the user may still communicate with it in Immediate Mode. As part of a user program, the MicroLYNX may start a second task using the RUN instruction. Thus, there can be two tasks running on the MicroLYNX at the same time, a foreground task (started by the EXEC instruction in Immediate Mode) and a background task (started by the RUN instruction in Immediate Mode or EXEC Mode).

MicroLYNX Communication Modes

When the MicroLYNX is operating in Immediate Mode, there are two methods of communicating. The first is ASCII where the instructions are communicated to the MicroLYNX in the form of ASCII mnemonics and data is also given in ASCII format. The second is binary where the instruction is in the form of an OpCode and numeric data is given in IEEE floating point hex format. In binary mode, there is also the option of including a checksum to ensure that information is received properly at the MicroLYNX. The BIO flag controls the method of communication. When it is True (1) the binary method should be used, and when it is False (0) the ASCII method should be used.

ASCII

ASCII is the most common mode of communicating with the MicroLYNX System. It allows the use of readily available terminal programs such as HyperTerminal, ProComm and the new LYNX Terminal.

When using the ASCII method of communications, the MicroLYNX tests for four special characters each time a character is received. These characters are given in the following table along with an explanation of what occurs when the character is received.

The command format in ASCII mode when the MicroLYNX is in Single Mode (PARTY = FALSE) is:

<Mnemonic><white space><ASCII data for 1st parameter>, <ASCII data for 2nd parameter>, … , <ASCII data for nth parameter><CR/LF>

The mnemonics for MicroLYNX instructions, variables, flags and keywords are given in Section 16 of this document. White space is at least one space or tab character. CR/LF represent the carriage return line feed characters that are transmitted in response to the Enter key on the keyboard provided the ASCII setup specifies "Send line feeds with line ends". Note that there need not be a space between the data for the last parameter and the CR/LF. Also note that if there is only one parameter, the CR/LF would immediately follow the data for that parameter.

The command format in ASCII mode, when the MicroLYNX is in Party Mode (PARTY = TRUE), would be identical to that in Single Mode with the exception that the entire command would be preceded by the MicroLYNX's address character (stored in DN) and terminated by a CTRL-J rather than ENTER:

<Address character><Mnemonic><white space><ASCII data for 1st parameter>, <ASCII data for 2nd parameter>, … , <ASCII data for nth parameter><CTRL-J>

ASCII Mode Special Command Characters	
Character	**Action at MicroLYNX**
<esc> Escape Key	Terminates all active operations and all running programs.
<^C> Ctrl+C Keys	Terminates all active operations and all running programs, forces a reset of the MicroLYNX.
<BKSP> Backspace Key	Moves the cursor back one in the buffer to correct a typing error.
<CR> or <LF> Carriage Return or Line Feed	Depending on the mode, either Single or Party. <LF> is not necessary in Single Mode communications. <CTRL+J> is the same as <LF> (0A Hex)

Table 7.7: ASCII Mode Special Command Characters

Binary

Binary mode communications is faster than ASCII and would most likely be used in a system design where the communication speed is critical to system operation. This mode cannot be used with standard terminal software.

The command format in binary mode when the MicroLYNX is in Single Mode (PARTY = FALSE) is:

<20H><character count><opcode><Field type for 1st parameter><IEEE hex data for 1st parameter><0EH><Field type for 2nd parameter><IEEE hex data for 2nd parameter><0EH> … <Field type for nth parameter><IEEE hex data for nth parameter><optional checksum>

Note that <20H> is 20 hex, the character count is the number of characters to follow the character count not including the checksum if one is being used. The OpCodes for MicroLYNX instructions, variables, flags and keywords are given in Sections 15 and 16 of this document. The Field type byte will be one of the following based on the type of data that is expected for the specific parameter:

Binary Hex Codes	
Hex Code	**Data Type**
01	Label Text
02	ASCII Text
03	1 byte unsigned
04	2 byte signed
05	2 byte unsigned
06	4 byte signed
07	4 byte unsigned
08	4 byte float

Table 7.8: Binary Hex Codes

<0EH> is 0E hex, which is a separator character in this mode. Finally, the optional checksum will be included if CSE is TRUE and excluded if it is FALSE. If included, the checksum is the low eight bits of the complemented sixteen-bit sum of the address field (20H here), character count, OpCode, all data fields and separators (0E hex).

SECTION 8

CAN Communications

Section Overview

The CAN (Controller Area Network) bus is a high-integrity serial data communications bus for real-time applications originally developed for the automotive industry. Because of its high speed, reliability and robustness, CAN is now being used in many other automation and industrial applications. Using the CAN bus to network controllers, sensors, actuators, etc., allows the designer to reduce design time and improve reliability because of readily available components and fewer connections.

Please refer to "CAN In Automation" (CIA) for an overview, details, and CAN standards at: http://www.can-cia.de/can/

The MicroLYNX System can be purchased with the capabiltiy to connect to a Controller Area Network (CAN) bus in place of the standard 2 Port RS-232/RS-485 interface. The MicroLYNX with this option conforms to the CAN2.0B Active protocol. CAN2.0B is fully backwards compatible with CAN2.0A, therefore the MicroLYNX can be used on a network with CAN 2.0A devices. There are two receive message frames and one transmit message frame. The CAN version of the MicroLYNX can also be optionally outfitted with RS-232 or RS-485 expansion modules for asynchronous communications.

Covered in this section are:

- Connecting and configuring the optional Controller Area Network (CAN) bus.
- Configuring the CAN Module.
- The CAN Communication Dongle
- Setting up Communications with the MicroLYNX CAN.
- Upgrading the MicroLYNX via CAN.

Connecting to the CAN Bus

To connect to the CAN bus, the only necessary connections are CAN H & CAN L. Since the majority of CAN cabling consists of shielded twisted pair cable, the shield can be connected to the SHIELD connection of the MicroLYNX communications connector. (*See Table 8.1 for pin configuration.*)

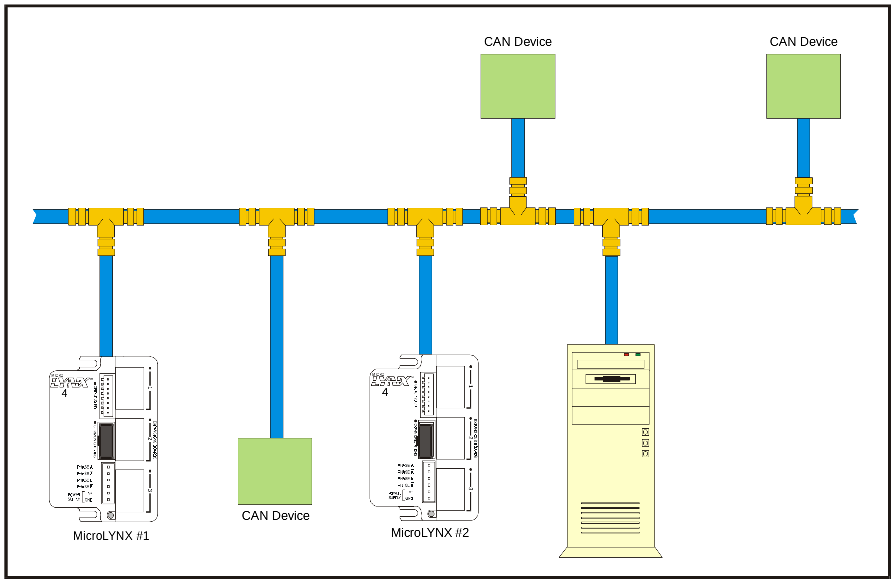


Figure 8.1: Devices on a CAN Bus

Controller Area Network (CAN) Version		
	Connector Option	
Pin #	8 Position Phoenix	10 Pin Header
1	V- (CGND)	N.C.
2	CAN_L	CAN_L
3	SHIELD	V- (CGND)
4	CAN_H	SHIELD
5	N.C.	SHIELD
6	/CONFIG	N.C.
7	N.C.	CAN_H
8		N.C.
9		N.C.
10		/CONFIG

Table 8.1 CAN Pin Configuration

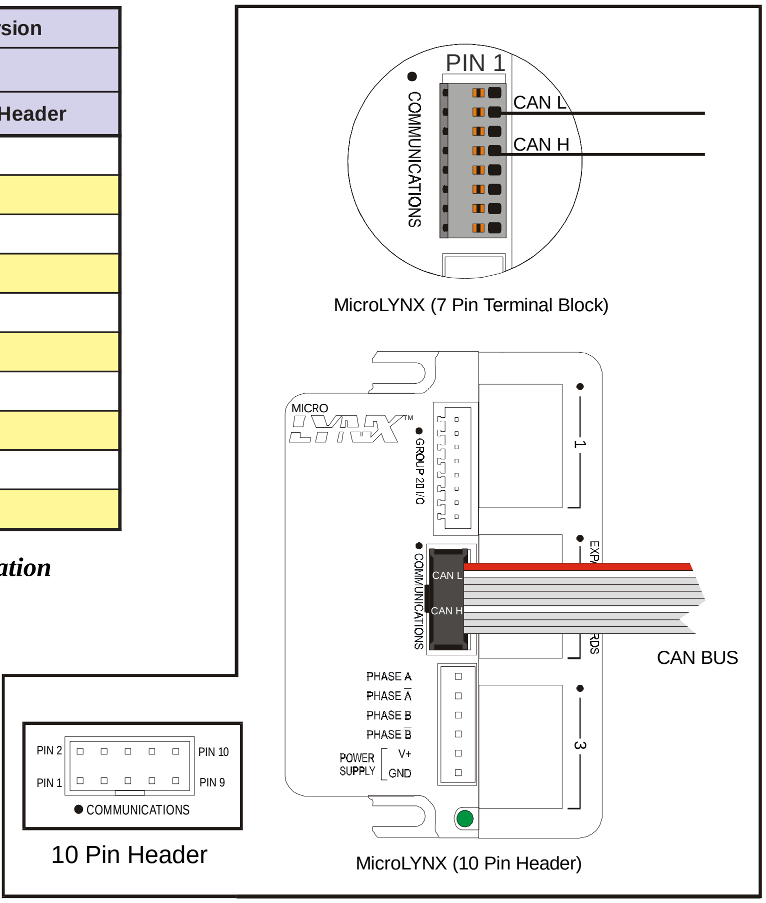


Figure 8.2: Connecting to the CAN Bus

Using the CAN Module

The format of all MicroLYNX commands remain unchanged when using the MicroLYNX CAN Module. The CAN Protocol limits the amount of data to be transmitted in a message frame to 8 bytes. Because Micro-LYNX commands can be longer than 8 bytes, the MicroLYNX CAN module employs a double buffer scheme. Each enabled receive message frame will buffer a maximum of 64 bytes of data. Once the CAN Module detects a complete MicroLYNX command, the complete command is queued to a 128-byte buffer for transfer to the MicroLYNX.

Any response from the MicroLYNX is queued to a 256-byte buffer and is transferred on the CAN bus when the transmit message frame is enabled. The system designer must be careful not to generate MicroLYNX code that will overflow the 128-byte and 256-byte buffers. All buffers are circular and no checks are made for overflow.

IMPORTANT: If the CAN module detects CAN errors and initiates “Bus Off”, the CAN controller is shut down and must be power cycled.

The IMS Terminal communications software, which ships with the MicroLYNX System, contains a CAN configuration utility to aid in configuring the CAN module. This utility can be accessed via the “View” menu item on the IMS Terminal.

NOTE: The MicroLYNX must be in ASCII Communications Mode. Once configured to issue commands in ASCII, terminate with a carriage return <cr> in Single Mode or a line feed <lf> in Party Mode.

Configuring the CAN Module

The CAN module is placed in configuration mode by holding the CONFIG input LOW on power-up. The module can then be configured using the configuration commands. Care must be taken to ensure proper initialization as no syntax checking is performed on the commands. The CAN module powers up as follows when the config input is held LOW:

BAUD Rate	500 kbps
Time Quanta (t_q) before sample point	5
Time Quanta (t_q) after sample point	4
Time Quanta (t_q) before (re) synchronization jump width	2
Identifier	Standard 11 bit
Global Mask	FFFFh
CAN Receive Identifier (UARØ = FF, UAR1 = ØØ)	7F8h
CAN Transmit Identifier (UARØ = FF, UAR1 = 2Ø)	7F9h

CAN Configuration Command Summary

CAN Configuration Command Summary		
Description	**Command / Usage**	**Usage Example**
Initialize CAN Registers	INIT	INIT

Set CAN Bit Timing Registers	BTR0=<hex digit><hex digit>
	BTR1=<hex digit><hex digit>
		BTR0=49
		BTR1=34

Set Global Mask Registers	GMS0=<hex digit><hex digit>
	GMS1=<hex digit><hex digit>
	UGML0=<hex digit><hex digit>
	UGML1=<hex digit><hex digit>
	LGML0=<hex digit><hex digit>
	LGML1=<hex digit><hex digit>
		GMS0=FF
		GMS1=FF
		UGML0=FF
		UGML1=FF
		LGML0=FF
		LGML1=F8

Setup Message Frames	FRM=<frame #><valid flag><extended ID flag>	FRM=210

Set Message Frame Arbitration Registers	UAR0=<frame#><hex digit><hex digit>
	UAR1=<frame#><hex digit><hex digit>
	LAR0=<frame#><hex digit><hex digit>
	LAR1=<frame#><hex digit><hex digit>
		UAR0=2A3
		UAR1=200
		LAR0=200
		LAR1=200

MicroLYNX Mode	LMODE=<flag>	LMODE=0
MicroLYNX Party Address	LADDR=<address>	LADDR=X
MicroLYNX Prompt	LPRMPT=<char>	LPRMPT=>
MicroLYNX BAUD Rate	LBAUD=<baud#><baud#>	LBAUD=38
Get CAN Module firmware version number.	VER	VER

Table 8.2: CAN Configuration Command Summary

NOTE: All configuration commands must be terminated with a Carriage Return <cr> or a Line Feed <lf>.

To Initialize the CAN Module.

Command : **INIT**

The factory default settings for the CAN module are detailed below:

BAUD Rate	500 kbps
Time Quanta (t_q) before sample point	5
Time Quanta (t_q) after sample point	4
Time Quanta (t_q) before (re) synchronization jump width	2
Global Mask	FFFFh
CAN Receive Identifier (UARØ = FF, UAR1 = ØØ)	7F8h
CAN Transmit Identifier (UARØ = FF, UAR1 = 2Ø)	7F9h
BTR0	41h
BTR1	34h
GMS0	FFh
GMS1	FFh
UGML0	FFh
UGML1	FFh
LGML0	FFh
LGML1	F8h
UMLM0	FFh
UMLM1	FFh
LMLM0	FFH
LMLM1	FFh
Message Frame 1	valid
Message Frame 2	not valid
Message Frame 3	valid
MicroLYNX Mode	single
MicroLYNX Party Address	!
MicroLYNX Prompt	>
MicroLYNX BAUD Rate (Fixed Value)	19.2 K

The use of the "INIT" instruction will restore these defaults in the CAN module. There are several new enhancements to the MicroLYNX instruction set which add the functions of the CAN module while maintaining backward compatibility with the MicroLYNX system. The following instructions and variables are specific to the CAN Module. These are introduced here and covered in more detail in the Part III "Software Reference". *Table 8.2* contains a summary of the configuration commands for the CAN Module.

The MicroLYNX must be in ASCII communications mode for use with the CAN Module.

To Set the CAN Bit Timing Registers

Command:
BTR0=<hex digit><hex digit>
BTR1=<hex digit><hex digit>

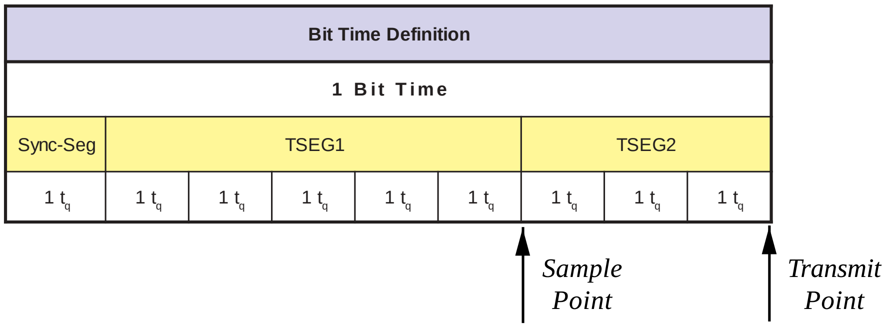
Usage Example:
BTR0=49
BTR1=34

This sets 100 Kbaud, 5 time quanta before the sample point, 4 time quanta after the sample point, and 2 time quanta for (re)synchronization jump width.

CAN Bit Timing Registers								
	7	6	5	4	3	2	1	0
BTR0	SJW		BRP					
BTR1	0	TSEG2			TSEG1			

Table 8.3: CAN Bit Timing Registers

A "bit time" is subdivided into four segments. Each segment is a multiple of the Time Quantum (t_q). The synchronization segment (Sync-Seg) is always 1 Time Quantum. The propagation time segment and the phase buffer segment1 are combined into (TSEG1). TSEG1 defines the time before the sample point. The phase buffer segment2 (TSEG2), defines the time after the sample point. See Tables 8.3 and 8.4.



Bit Time Definition								
1 Bit Time								
Sync-Seg	TSEG1					TSEG2		
1 t_q	1 t_q	1 t_q	1 t_q	1 t_q	1 t_q	1 t_q	1 t_q	1 t_q

Table 8.4: CAN Bit Time Definition

The bit time is determined from the following formulae:

$t_q = (BRP + 1)(50 \times 10^{-9}$

$t_{seg2} = (TSEG2 + 1)(t_q)$

$t_{seg1} = (TSEG1 + 1)(t_q)$

$t_{sync\text{-}seg} = t_q$

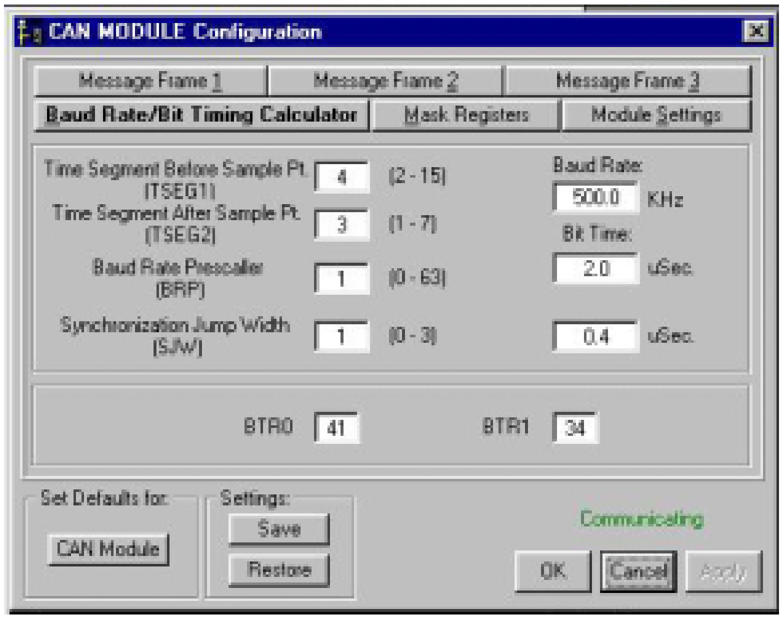
bit time $= t_{sync\text{-}seg} + t_{seg1} + t_{seg2}$

Sample Bit Timing Register Settings			
Default Time Quanta Settings 5 time quanta before sample point. 4 time quanta after sample point. 2 time quanta for (re)synchronization jump width.			
t_q	BAUD (kbps)	BTR0	BTR1
2 us	50	53h	34h
1 us	100	49h	34h
0.4 us	250	43h	34h
0.2 us	500	41h	34h
0.1 us	1000	40h	34h

Table 8.5: Sample Bit Timing Register Settings

NOTES:

The values of BRP, TSEG2 and TSEG1 are encoded in BTRØ and BTR1. See Table 8.3 above.
TSEG2 Valid Values: Min = 1 / Max = 7
TSEG1 Valid Values: Min = 2 / Max = 15
SJW Valid Values: Min = 0 / Max = 3
(Re) Synchronization Jump Time $t_{sjw} = (SJW + 1)(t_q)$



Time Segment Before Sample Pt. (TSEG1) 4 (2 - 15)
Time Segment After Sample Pt. (TSEG2) 3 (1 - 7)
Baud Rate Prescaler (BRP) 1 (0 - 63)
Synchronization Jump Width (SJW) 1 (0 - 3)
Baud Rate: 500.0 KHz
Bit Time: 2.0 uSec.
0.4 uSec.
BTR0 41
BTR1 34

Figure 8.3: Bit Register Configuration Dialog from IMS Terminal

The Bit Timing Registers can also be set using the Bit Rate/Bit Timing Calculator utility in the IMS Terminal software that comes with the MicroLYNX *(see Figure 8.3)*. This utility can be accessed from the Setup > Configure CAN menu item on the main IMS Terminal window.

To Set The Global Mask Registers

Command:	Usage Example:
GMS0=<hex digit><hex digit>	**GMS0=FF**
GMS1=<hex digit><hex digit>	**GMS1=FF**
UGML0=<hex digit><hex digit>	**UGML0=FF**
UGML1=<hex digit><hex digit>	**UGML1=FF**
LGML0=<hex digit><hex digit>	**LGML0=FF**
LGML1=<hex digit><hex digit>	**LGML1=F8**

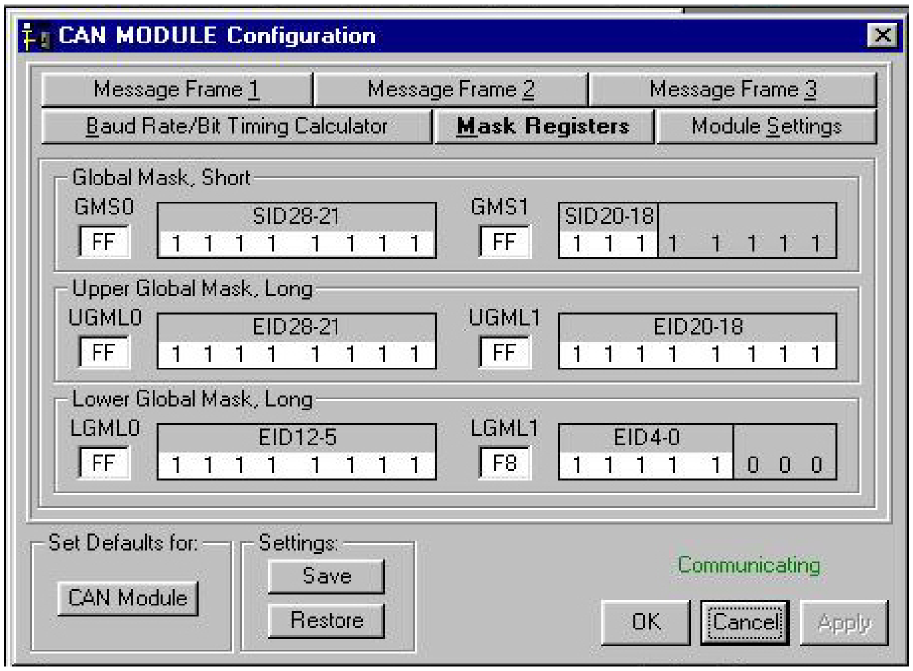
SID28-18 – Standard Identifier (11-bit)
EID28-0 – Extended Identifier (29-bit)

Incoming message frames are masked with the appropriate global mask. If the bit position in the global mask register is 0 (don't care), then the bit position will not be compared with the incoming message's identifier.

IT IS IMPORTANT THAT YOU NOTE THE BIT POSITIONS IN THE GLOBAL MASK REGISTERS.

Global Mask Registers								
	7	6	5	4	3	2	1	0
GMS0	SID 28-21							
GMS1	SID 20-18				1	1	1	1
UGML0	EID 28-21							
UGML1	EID 20-13							
LGML0	EID 12 -5							
LGML1	EID 4-0					0	0	0

Table 8.6: Global Mask Registers



Global Mask, Short
GMS0 FF SID28-21 1 1 1 1 1 1 1 1
GMS1 FF SID20-18 1 1 1 1 1 1 1 1

Upper Global Mask, Long
UGML0 FF EID28-21 1 1 1 1 1 1 1 1
UGML1 FF EID20-18 1 1 1 1 1 1 1 1

Lower Global Mask, Long
LGML0 FF EID12-5 1 1 1 1 1 1 1 1
LGML1 F8 EID4-0 1 1 1 1 1 0 0 0

Figure 8.4: Setup Dialog for Global Mask Registers in IMS Terminal

To Setup Message Frames

Command: **FRM=<frame#><valid flag><extended ID flag>**

<frame#>	= frame number (1-3). Frames 1 and 2 are fixed as receive frames. Frame 3 is fixed as a transmit frame.
<valid flag>	= 1 frame valid
<valid flag>	= 0 frame not valid
<extended ID flag>	= 1 extended identifier
<extended ID flag>	= 0 standard identifier

The CAN module will only operate on valid message objects.

Example:
FRM=210
This sets message 2 valid, using the standard identifier.

Set Message Frame Arbitration Registers

Command:	Usage Example:
UAR0=< frame#><hex digit><hex digit>	**UAR0=2A3**
UAR1=< frame#><hex digit><hex digit>	**UAR1=200**
LAR0=< frame#><hex digit><hex digit>	**LAR0=200**
LAR1=< frame#><hex digit><hex digit>	**LAR1=200**
< frame#> = frame number	(1-3)This sets message 2 arbitration registers to A30h.

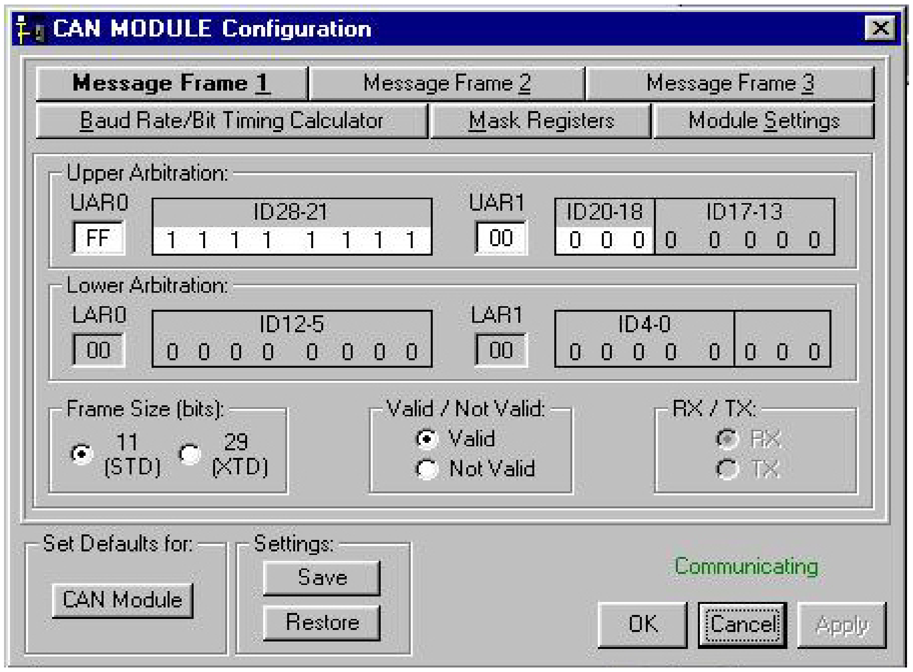
ID28-18 – Identifier of a standard message. ID17-0 set to 0 for a standard message.
ID28-0 – Identifier of an extended message.

The arbitration registers are used for acceptance filtering of incoming messages and to define the identifier of outgoing messages. **(IMPORTANT!** THERE MUST NOT BE MORE THAN ONE VALID MESSAGE OBJECT WITH A PARTICULAR IDENTIFIER AT ANY TIME.) If some bits are masked by the global mask registers, then the identifiers of the valid message objects must differ in the remaining bits which are used for acceptance filtering.

IT IS IMPORTANT THAT YOU NOTE THE BIT POSITIONS IN THE GLOBAL MASK REGISTERS.

Message Frame Arbitration Registers								
	7	6	5	4	3	2	1	0
UAR0	ID28 - 21							
UAR1	ID20 - 18			ID17 - 13				
LAR0	ID12 - 5							
LAR1	ID4 - 0					0	0	0

Table 8.7: Message Frame Arbitration Registers


CAN MODULE Configuration

Upper Arbitration: UAR0 FF, ID28-21: 1 1 1 1 1 1 1 1; UAR1 00, ID20-18: 0 0 0, ID17-13: 0 0 0 0 0

Lower Arbitration: LAR0 00, ID12-5: 0 0 0 0 0 0 0 0; LAR1 00, ID4-0: 0 0 0 0 0, 0 0 0

Frame Size (bits): [✓ 11 (STD)] 29 (XTD)

Valid / Not Valid: [✓ Valid] Not Valid

RX / TX: RX TX

Communicating

Figure 8.5: Message Frame Setup Dialog from IMS Terminal

Defining the MicroLYNX Mode (Single or Party)

This command identifies to the CAN module the MicroLYNX mode.

Command:
LMODE=<flag>

<flag> = 0	single mode
<flag> = 1	party mode

Example:
LMODE=0
This indicates to the CAN module that the MicroLYNX is operating in single mode.

Setting the MicroLYNX Party Address

Command:	Usage Example:
LADDR=<address>	**LADDR=X**
<address> = any valid MicroLYNX address.	This indicates to the CAN module that the MicroLYNX party address is X.

This command identifies to the CAN module the MicroLYNX address when party mode is enabled in the MicroLYNX.

MicroLYNX Prompt

Command:	Usage Example:
LPRMPT=<char>	**LPRMPT>**
	This indicates to the CAN module that the MicroLYNX prompt is the > character.

<char> = any valid MicroLYNX prompt character.

This command identifies to the CAN module the MicroLYNX prompt character.

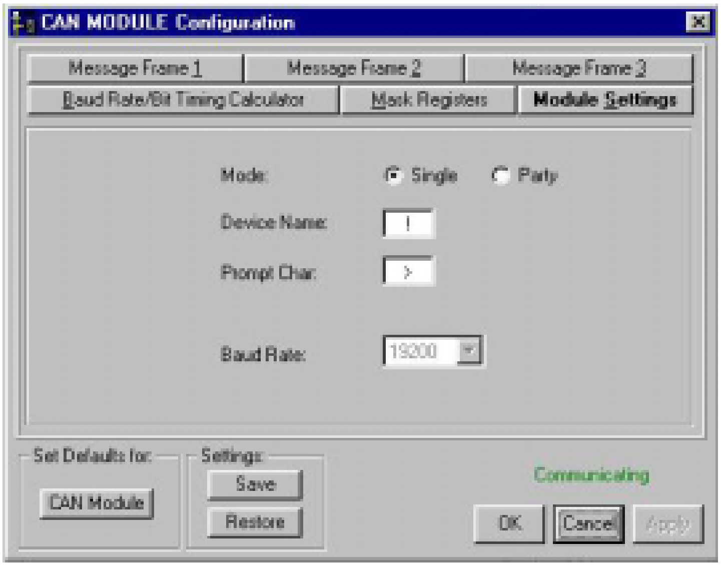
MicroLYNX Baud Rate

Command:	Usage Example:
LBAUD=<baud#><baud#>	**LBAUD=38**
	This indicates to the CAN module that the MicroLYNX baud rate is 38400 baud.

<baud#><baud#> = 48	4800 baud
<baud#><baud#> = 96	9600 baud
<baud#><baud#> = 19	19200 baud
<baud#><baud#> = 38	38400 baud

This command identifies to the CAN module the MicroLYNX baud rate.

NOTE: The Baud Rate is fixed at 19.2K. The command to change it is disabled in the GUI. However, the commands remain in the CAN Module Firmware to maintain compatibility with earlier systems.

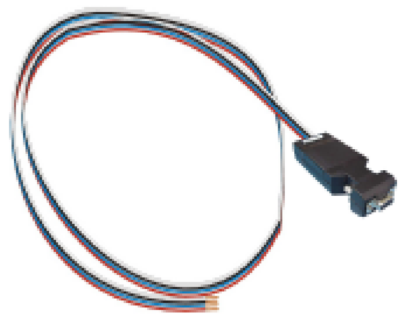

CAN MODULE Configuration

Mode: [✓ Single] Party
Device Name: !
Prompt Char: >
Baud Rate: 19200

Communicating

Figure 8.6: MicroLYNX CAN Setup Dialog from IMS Terminal

The CAN Communication Dongle



IMS offers an optional CAN Dongle which allows communication between an RS-232 port and the MicroLYNX CAN Module. The CAN Dongle can be used as a temporary communication and configuration device as well as a permanent interface connection for the MicroLYNX CAN.

The CAN Dongle consists of a powered converter with a DB-9F plug to connect to the customer PC and flying leads which connect to the MicroLYNX communications terminal block and the MicroLYNX CAN power source. The length of the flying leads is variable. The CAN Dongle also has an on-board power jack to allow the user to connect an external power source. This is necessary when the MicroLYNX CAN power source is greater than +60 VDC. The CAN Dongle is also equipped with Power and Fault LEDs. The CAN Dongle may be ordered from IMS under Part Number **MX-CC500-000.**

Dimensions in Inches (mm)

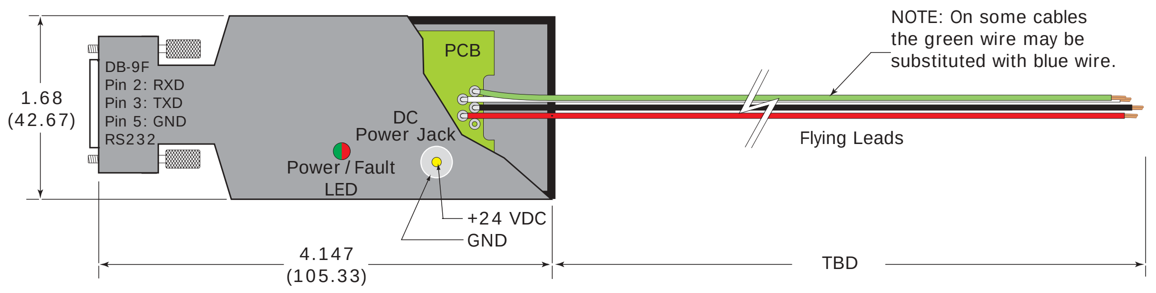


Figure 8.7: MicroLYNX MX-CC500-000 CAN Dongle Details

Connecting the CAN Dongle

The CAN Dongle connects easily. However, the connections and usage must be carried out as described or failure and possible damage to the CAN Dongle and/or MicroLYNX CAN may occur.

The CAN Dongle is fitted with a DB-9F connector which plugs into the COMM port of your PC. It is also equipped with “flying leads” to connect to the MicroLYNX CAN Module. The flying leads are color coded. The connections are as follows:

Green:	CAN-L Signal (Note: On some cables the green wire may be substituted with blue wire).
White:	CAN-H Signal
Red:	+ VDC (Max +60 VDC)
Black:	Ground

The CAN Dongle must be powered with DC voltage. The recommended voltage is +24 VDC but it will operate on a voltage range from +7 VDC to a maximum of +60 VDC. On many systems the power from the MicroLYNX CAN may be used. However, if the power from the MicroLYNX CAN is greater than +60 VDC, an external power source must be used.

The illustration below shows the required and optional connections for the CAN Dongle.

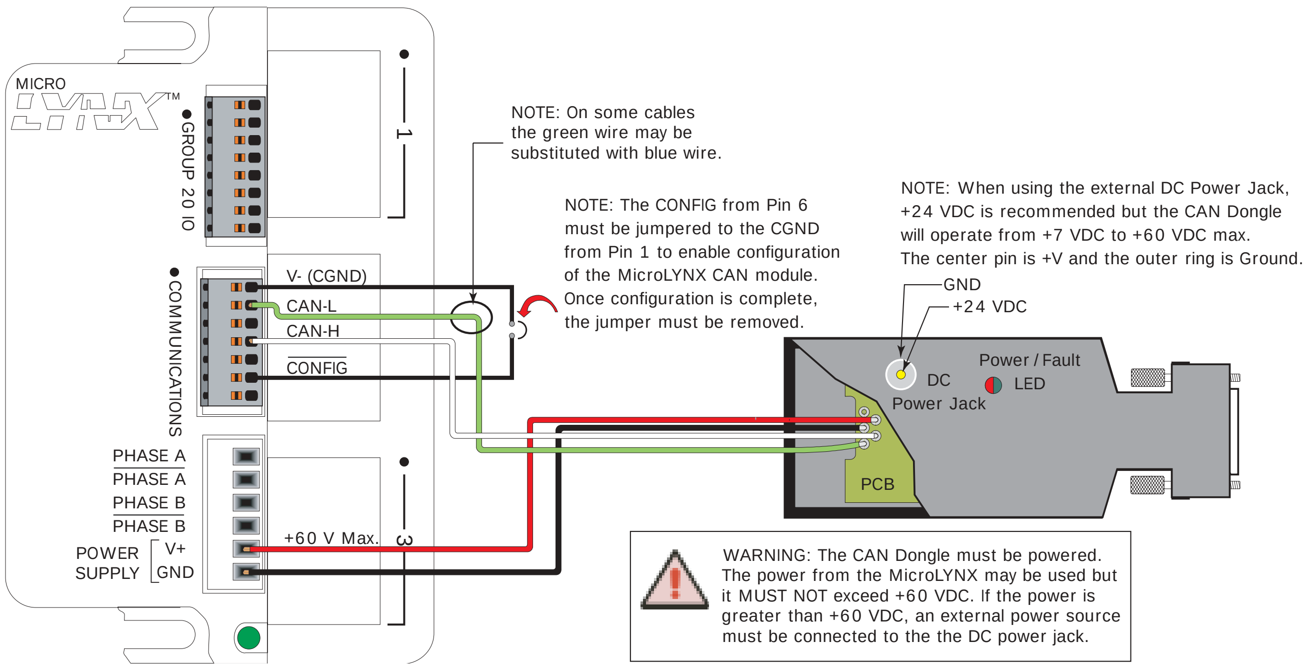

MICRO
LYNX™
GROUP 20 IO
COMMUNICATIONS
V- (CGND)
CAN-L
CAN-H
CONFIG
PHASE A
PHASE A
PHASE B
PHASE B
POWER SUPPLY V+
GND
+60 V Max.
1
3
NOTE: On some cables the green wire may be substituted with blue wire.
NOTE: The CONFIG from Pin 6 must be jumpered to the CGND from Pin 1 to enable configuration of the MicroLYNX CAN module. Once configuration is complete, the jumper must be removed.
NOTE: When using the external DC Power Jack, +24 VDC is recommended but the CAN Dongle will operate from +7 VDC to +60 VDC max. The center pin is +V and the outer ring is Ground.
GND
+24 VDC
DC Power Jack
Power / Fault LED
PCB
WARNING: The CAN Dongle must be powered. The power from the MicroLYNX may be used but it MUST NOT exceed +60 VDC. If the power is greater than +60 VDC, an external power source must be connected to the the DC power jack.

Figure 8.8: Connecting the CAN Dongle (With MicroLYNX as Power Source)

WARNING: DO NOT connect the Red and Black flying leads to the MicroLYNX CAN power and the external power source to the DC Power Jack at the same time. Use one or the other.

Setting Up Communications with MicroLYNX CAN

If the following procedures are followed properly, Communication with MicroLYNX CAN Version can be accomplished using IMS Terminal or any other terminal program.

Note: The commands will not be echoed until the terminator <CR> is pressed.

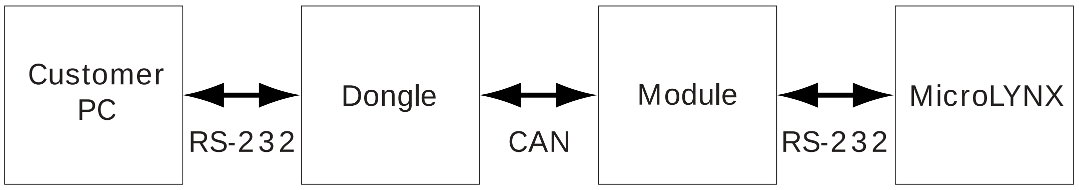


Figure 8.9: Communications Diagram

Communications

Communications must be setup between:

1) The PC and the CAN Dongle.
2) The CAN Dongle and the CAN Module.
3) The CAN Module and the MicroLYNX.

NOTE: The RS-232 baud rate between the PC and the CAN Dongle does not have to be the same as between the CAN Module and the MicroLYNX.

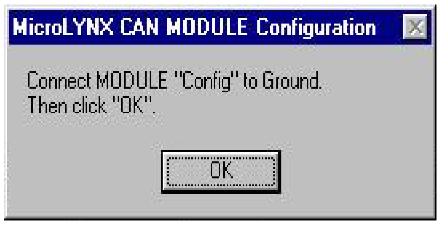
Setup Procedure

1) First setup the CAN Dongle to match the MicroLYNX CAN Module
2) Next setup the MicroLYNX CAN Module
3) Optionally, UPGRADING via CAN

Module Setup

When setting up the MicroLYNX CAN Module, the Dongle will be reconfigured to allow communication with the Module in the Setup Mode. The “Config” IO line must be connected to Ground.

This Procedure **<u>must</u>** be followed exactly. The reason for this is that the previous settings in the CAN Dongle are saved and will be reset after the Module is finished.



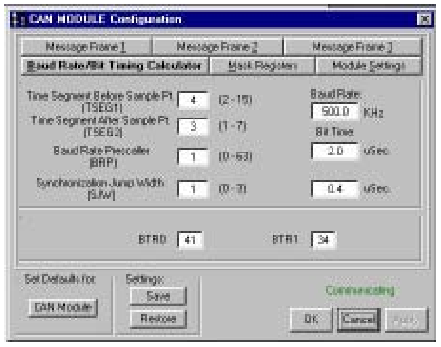
MicroLYNX CAN MODULE Configuration

Connect MODULE "Config" to Ground.
Then click "OK".

Figure 8.10: Connect Procedure

The Module is setup using the following screens.

This screen sets up the CAN Module Communication Baud Rate. It should be set to match the system into which it is to be installed.



CAN MODULE Configuration

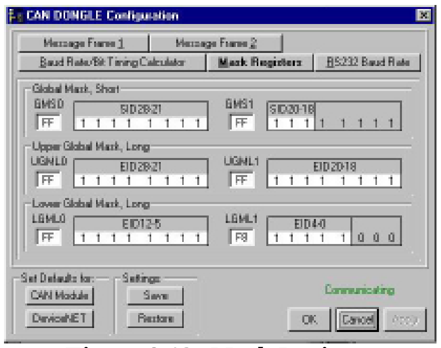
Time Segment Before Sample Pt (TSEG1)	4	(2 - 15)	Baud Rate:	500.0 KHz
Time Segment After Sample Pt (TSEG2)	3	(1 - 7)	Bit Time:	2.0 uSec
Baud Rate Prescaller (BRP)	1	(0 - 63)		
Synchronization Jump Width (SJW)	1	(0 - 3)		0.4 uSec

BTR0 41 BTR1 34

Communicating

Figure 8.11: CAN Module Baud Rate

This screen sets the Mask Registers. The factory settings are shown. In most cases they will not change.



CAN DONGLE Configuration

Global Mask, Short

GMS0	FF	SID28-21	1 1 1 1 1 1 1 1
GMS1	FF	SID20-18	1 1 1 1 1 1 1 1

Upper Global Mask, Long

UGML0	FF	EID28-21	1 1 1 1 1 1 1 1
UGML1	FF	EID20-18	1 1 1 1 1 1 1 1

Lower Global Mask, Long

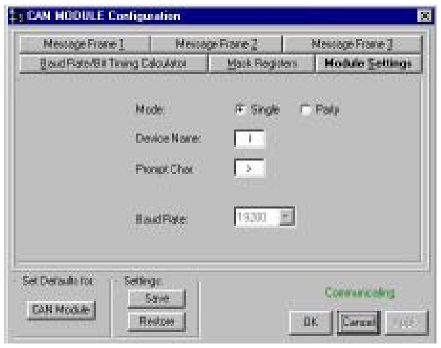
LGML0	FF	EID12-5	1 1 1 1 1 1 1 1
LGML1	F8	EID4-0	1 1 1 1 1 0 0 0

Communicating

Figure 8.12: Mask Registers

This screen is to set the RS-232 communications between the CAN Module and the MicroLYNX. It must be set to reflect the settings in the MicroLYNX.

NOTE: The Baud Rate is fixed at 19.2K.



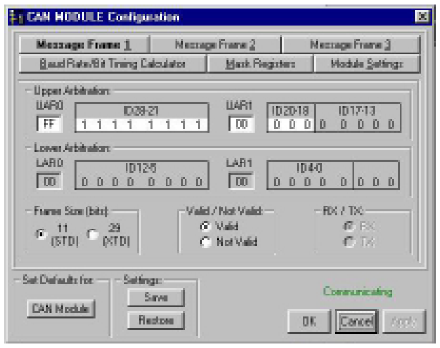
CAN MODULE Configuration

Mode: [✓ Single] Party
Device Name: !
Prompt Char: >
Baud Rate: 19200

Communicating

Figure 8.13: RS-232 Setup

The following three screens illustrate the CAN setup for communications to and from the MicroLYNX CAN Module.


CAN MODULE Configuration

Upper Arbitration: UAR0 FF; ID28-21 1 1 1 1 1 1 1 1; UAR1 00; ID20-18 0 0 0; ID17-13 0 0 0 0 0

Lower Arbitration: LAR0 00; ID12-5 0 0 0 0 0 0 0 0; LAR1 00; ID4-0 0 0 0 0 0 0 0 0

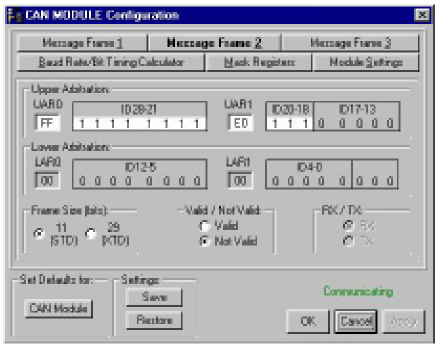
Frame Size (bits): [✓ 11 (STD)] 29 (XTD)

Valid / Not Valid: [✓ Valid] Not Valid

RX / TX: [✓ RX] TX

Communicating

Figure 8.14: Message Frame 1


CAN MODULE Configuration

Upper Arbitration: UAR0 FF; ID28-21 1 1 1 1 1 1 1 1; UAR1 E0; ID20-18 1 1 1; ID17-13 0 0 0 0 0

Lower Arbitration: LAR0 00; ID12-5 0 0 0 0 0 0 0 0; LAR1 00; ID4-0 0 0 0 0 0 0 0 0

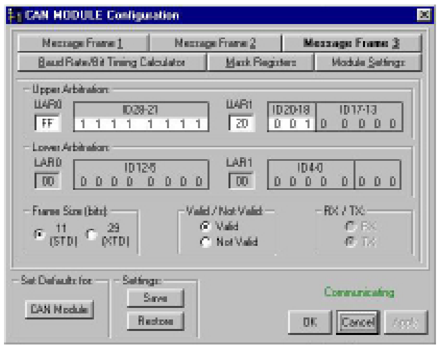
Frame Size (bits): [✓ 11 (STD)] 29 (XTD)

Valid / Not Valid: Valid [✓ Not Valid]

RX / TX: [✓ RX] TX

Communicating

Figure 8.15: Message Frame 2


CAN MODULE Configuration

Upper Arbitration: UAR0 FF; ID28-21 1 1 1 1 1 1 1 1; UAR1 20; ID20-18 0 0 1; ID17-13 0 0 0 0 0

Lower Arbitration: LAR0 00; ID12-5 0 0 0 0 0 0 0 0; LAR1 00; ID4-0 0 0 0 0 0 0 0 0

Frame Size (bits): [✓ 11 (STD)] 29 (XTD)

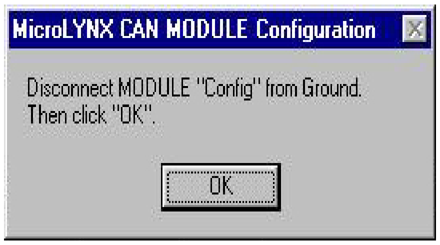
Valid / Not Valid: [✓ Valid] Not Valid

RX / TX: RX [✓ TX]

Communicating

Figure 8.16: Message Frame 3

This screen completes the MicroLYNX CAN Module setup. The steps must be performed precisely.


MicroLYNX CAN MODULE Configuration

Disconnect MODULE "Config" from Ground.
Then click "OK".

OK

Figure 8.17: Module Setup Complete

Dongle Setup

The Dongle is setup using the following screens.

This screen illustrates the settings from the factory.
The CAN Baud Rate must match the settings in the Module.
These settings will be set to match the CAN system in which the MicroLYNX CAN is installed.

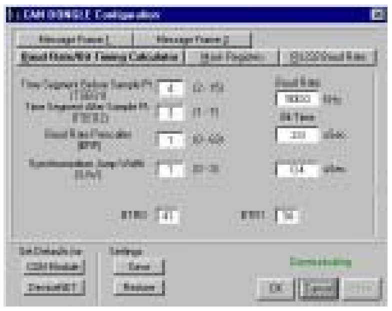


Figure 8.18: Dongle Baud Rate

This screen sets the Mask Registers.
The factory settings are shown. Most likely, they will not change.

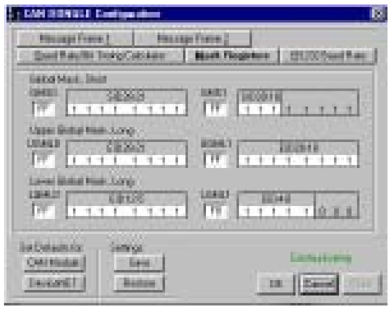


Figure 8.19: Mask Registers

This screen sets the RS-232 Baud Rate between the PC and the CAN Dongle. This does not have to match the setting between the Module and the MicroLYNX CAN. The IMS Terminal program will search for the baud rate setting of the CAN Dongle while it is establishing communications.

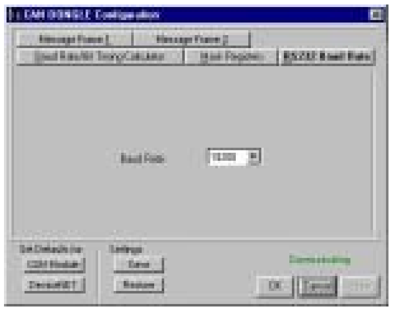


Figure 8.20: Dongle Baud Rate

Message Frame 1: The CAN Dongle should be set to match the "Message Frame 3" settings from the MicroLYNX CAN Module.

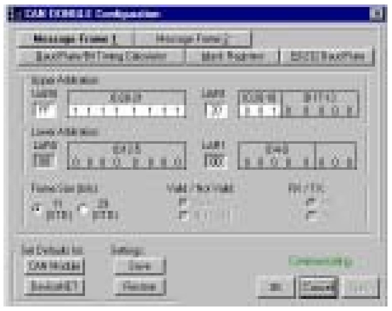


Figure 8.21: Message Frame 1

Message Frame 2: The CAN Dongle should be set to match the "Message Frame 1" settings from the MicroLYNX CAN Module.

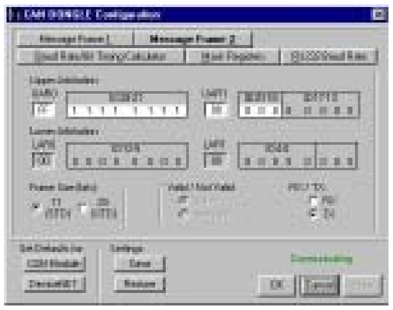


Figure 8.22: Message Frame 2

MicroLYNX CAN / CAN Dongle Communications

Communications Between MicroLYNX CAN and the Dongle

The following procedure can be followed to establish communications with the MIcroLYNX CAN and the CAN Dongle when the CAN Baud Rate and Message IDs are unknown.

1) Connect the CAN Dongle to the PC only.
2) Apply power to the CAN Dongle. Power can be supplied via the red and black flying leads or with an external power source plugged into the +24VDC jack on the CAN Dongle.

   (See Figure 8.8)
3) Configure the CAN Dongle with IMS Term.
   a) View
   b) CAN Configuration
   c) Dongle
   d) Select Set Defaults for CAN Module
4) Click OK
5) Power Down the CAN Module.
6) Connect the Config input on the CAN Module to GND and power the MicroLYNX and the CAN Dongle.
7) Configure the CAN Module with IMS Term.
   a) View
   b) CAN Configuration
   c) Module A dialog box will open “Connect Module Config to Ground”. This has already been done.
8) Click OK.
9) Select “Set Defaults for CAN Module” and click OK. A dialog box will open “Disconnect Module Config from Ground.
10) Disconnect the Module Config Input fromt he GND and click OK.
11) Communication has now been established at the factory default settings.

Upgrading the MicroLYNX Firmware Via CAN

1) Establish communications between the PC, CAN Dongle, and the MicroLYNX. (The above procedure may be used.)

   NOTE: The CAN Module Message Frames 1 and 3 must be valid. The recommended CAN Baud Rate is 500K.
2) Configure communications between the PC and the CAN Dongle at 19.2K.
3) Confirm that LYNX and CAN are selected and click OK.
   a) Select “Preferences”
   b) Select “Comm Settings”
   c) Select “Device”
4) Select “Upgrade” in IMS Term and follow the instructions on the screen.
5) After Upgrading, the IMS Term will remain at the 19.2K Baud Rate.

NOTE: Communications between the PC and the MicroLYNX will be fully synchronized after the Upgrade is completed.

SECTION 9

DeviceNet

Section Overview

This IMS MicroLYNX DeviceNet version is specifically designed per the ODVA Volume II, Release 2.0, Errata 3. Included in this section are:

- MicroLYNX DeviceNet features.
- Connector locations and pin descriptions.
- Attribute tables.
- I/O messaging and response.
- DeviceNet Programmer.

MicroLYNX DeviceNet Features

ODVA DeviceNet MicroLYNX

- Conforms to the Predefined Master/Slave Connection Set as a Group 2 Slave.
- Supports Poll IO and Explicit Messaging only.
- No support for UCMM.
- Device Type: Position Controller (16)

Graphical User Interface (GUI) Software

NOTE: GUI to be developed.

The GUI Software provided with the ODVA DeviceNet MicroLYNX will ONLY work with the following DeviceNet cards:

SST (a Woodhead Industries Inc. company) series 5136-DNP Pro Series format cards.

Part Numbers:	5136-DNP-PCI	PCI
	5136-DNP-PCM-ST	PCMCIA
5-pin DeviceNet Conn.		
	5136-DNP-PCM-SM	PCMCIA
With Sealed Micro Conn.		
	5136-DNP-ISA ISA format	

Setup

To setup the MicroLYNX DeviceNet Configuration Utility, perform the following steps.

1. Install the program to your PC’s hard drive.
2. Run the program.
3. Click the button labeled “Load” to load the SST DeviceNet card you are using.
4. Select the MACID (0-63) of the MicroLYNX being configured from the pull-down, click the button labeled “Select”, or click the button labeled “Scan” to scan the DeviceNet buss for connected MicroLYNX.

Specific Features

The following listed features are specific to the MicroLYNX DeviceNet MX-CS30X-X01

1. Dedicated Isolated Input/Output Functions
   - IO 21 = Home Input
   - IO 22 = CW Limit Input
   - IO 23 = CCW Limit Input
   - IO 24 = Fault Input
   - IO 25 = Brake Output
   - IO 26 = General Purpose IO
2. Dedicated Encoder Functions
   - IO 13+ = Encoder A+
   - IO13- = Encoder A -
   - IO 14+ = Encoder B+
   - IO 14- = Encoder B-
   - IO 17 = Encoder Index (Z+)
   - IO 17- = Encoder Index (Z-)
3. Node Address MSD and LSD switch-selectable on front panel
4. Data Rate switch-selectable on front panel

Connector Locations and Pin Descriptions

Connector Locations

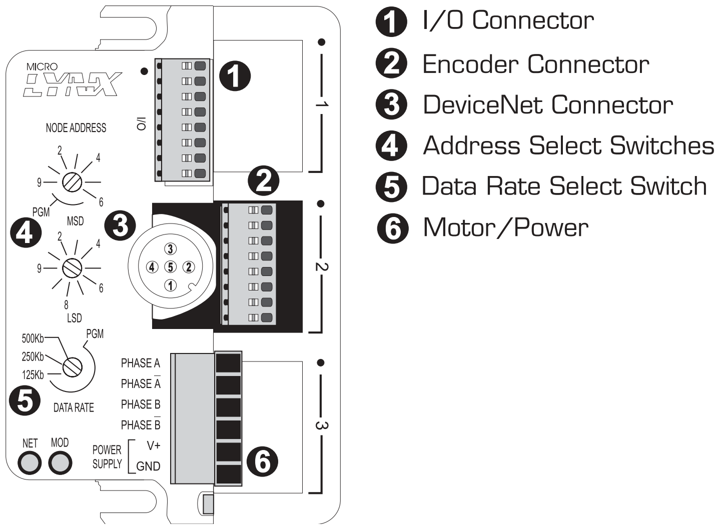


Figure 9.1: MicroLYNX DeviceNet Port Pin Configuration

Connector Pin Configuration

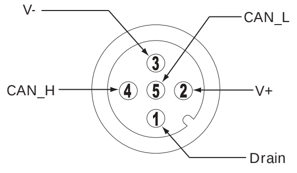


Figure 9.2: DeviceNet Port Pin Configuration

Power and Motor Connections

Pin Number	Pin Function
1	Motor Phase A
2	Motor Phase $\overline{A}$
3	Motor Phase B
4	Motor Phase $\overline{B}$
5	+12 to +48 VDC (MicroLYNX 4) +24 to +75 VDC (MicroLYNX 7)
6	Power Ground Supply

Table 9.1: Motor Power Connections

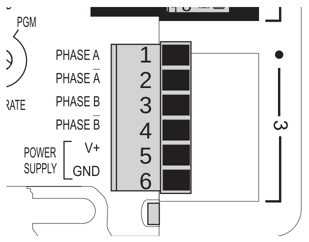


Figure 9.3: Motor Power Terminals

Isolated Digital Input

Pin Number	Pin Function
1	V Pull-Up
2	Home Input (level active)
3	CW Limit Input (disabled)
4	CCW Limit Input (disabled)
5	Fault Input (level active)
6	Brake Output
7	General Purpose IO
8	Isolated Ground

Table 9.2: Isolated Digital Inputs

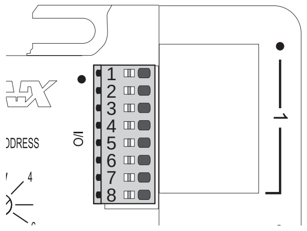


Figure 9.4: Isolated Digital Input Terminals

Encoder Inputs

Single Encoder	Differential Encoder	8 Pin Pheonix	10 Pin Header	Pin Function
Channel A	Channel A+	5	6	Channel A / A+
	Channel A-		5	Channel A-
Channel B	Channel B+	6	8	Channel B / B+
	Channel B-		4 or 7	Channel B-
Index	Index +	7	10	Index / Index+
	Index -		9	Index-
+5 VDC	+ 5 VDC	3	2	+ 5 VDC
GND	GND	2	3	Ground

Table 9.3: Encoder Input Connections

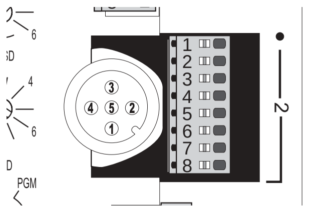


Figure 9.5: Encoder Connect - 8 Pin Pheonix

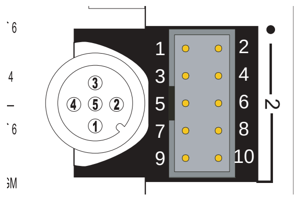


Figure 9.6: Encoder Connect - 10 Pin Header

Attribute Tables

Attribute Map - Part 1 of 3

Object	Attribute	Access Rule	Name	Data Type	Semantics / Description	Factory Default	Stored in NVM
0x24 - Position Controller Supervisor Object							
0x24	1 (class)	Get	Revision	UINT	Rev = 2		
0x24	3	Get	Axis Instance	USINT	1 = axis 1 other values = error	1	
0x24	5	Get	General Fault	BOOL	0 = no alarms 1 = fault	0	
0x24	6	Get / Set	Command Assembly Type	USINT		0	
0x24	7	Get / Set	Response Assembly Type	USINT		0	
0x24	8	Get	Fault Input State	BOOL	1 = active 0 = inactive	0	
0x24	9	Get / Set	Fault Input Action	USINT	0 = disable 1 = hard stop 2 = smooth stop 3 = no action (ignore fault)	3	X
0x24	11	Get / Set	Home Active Level	BOOL	0 = active low 1 = active high	0	X
0x24	12	Get / Set	Home Arm (Start Find Home)	BOOL	1 = find home 0 = home complete	0	
0x24	16	Get	Home Input State	BOOL	1 = active 0 = inactive	0	
0x24	101	Get / Set	Homing Type	USINT	1 = home to switch 2 = home to index mark	1	X
0x24	102	Get / Set	General Fault Action	USINT	0 = Stop and Disable Drive when General Fault = 1 1 = Process Commands if General Fault = 1	1	X
0x24	103	Get / Set	Alarm Clear	USINT	1 = General Fault (0x25-5) 2 = General Fault (0x24-5) and Following Error Fault (0x25-47)	0	
0x24	104	Get	Alarm Code	DINT		0	
0x24	110	Get / Set	Fault Input Logic		0 = active low 1 = active high	0	X
0x24	111	Get / Set	Enable NVM Storage	BOOL	Attributes labeled as stored in NVM will behave as follows 0 = NVM disabled, attributes not stored 1 = NVM enabled, attributes stored Note: Once the NVM is enabled, it will remain enabled until disabled or power is cycled	0	
0x24	112	Get / Set	Restore NVM to Factory Defaults	BOOL	0 = Restore Complete / Ready 1 = Restore to Factory Defaults / In Progress Will restore on next power cycle	0	
0x24	113	Get	MicroLYNX Firmware Version	REAL			
0x24	114	Get	MicroLYNX DeviceNet Code Version	REAL			

Table 9.4: MicroLYNX DeviceNet Attribute Map (Part 1)

Attribute Map - Part 2 of 3

Object	Attribute	Access Rule	Name	Data Type	Semantics / Description	Factory Default	Stored in NVM
0x25 - Position Controller Object							
0x25	1 (class)	Get	Revision	UINT	Rev = 2		
0x25	1	Get	Number of Attributes	USINT			
0x25	2	Get	Attribute List	Array of USINT			
0x25	3	Get / Set	Mode	USINT	0 = Position Mode 1 = Velocity Mode	0	X
0x25	6	Get / Set	Target Position	DINT	Range 0x80000001 to 0x7FFFFFFF	Undefined	
0x25	7	Get / Set	Target Velocity	DINT	Positive Number ≥ 0	768,000	X
0x25	8	Get / Set	Acceleration	DINT	Positive Number > 0	1,000,000	X
0x25	9	Get / Set	Deceleration	DINT	Positive Number > 0	1,000,000	X
0x25	10	Get / Set	Absolute / Incremental	BOOL	0 = Absolute Position Value 1 = Relative Position Value	0	X
0x25	11	Get / Set	Load Data / Profile	BOOL	1 = Trajectory start, in motion 0 = move complete	0	
0x25	12	Get	On Target Position (motor within deadband)	BOOL	1 = On Target / End of Move	1	
0x25	13	Get / Set	Actual Position (absolute)	DINT	Actual position in steps or encoder counts	0	
0x25	14	Get	Actual Velocity	DINT	Actual Velocity	0	
0x25	15	Get	Command Position	DINT	Command Position (echo 0x25-6)	0	
0x25	17	Get / Set	Enable	BOOL	0 = Disable 1 = Enable	1	X
0x25	20	Get / Set	Smooth Stop	BOOL	Bring motor to a controlled stop at the programmed deceleration rate.	0	
0x25	21	Get / Set	Hard Stop	BOOL	Bring motor to an immediate stop.	0	
0x25	23	Get / Set	Direction (V Mode)	BOOL	0 = CCW Direction 1 = CW Direction	1	
0x25	24	Get / Set	Reference Direction	BOOL	1 = CW is positive direction 0 = CCW is positive direction	1	X
0x25	38	Get / Set	Position Deadband	USINT	Range 0 to 255	2	X
0x25	39	Get / Set	Feedback Enable	BOOL	0 = Disable 1 = Enable	0	X
0x25	40	Get / Set	Feedback Resolution	DINT	Encoder Lines X 4 Positive Number > 0	2,000	X
0x25	41	Get / Set	Motor Resolution	DINT	Full motor steps / revolution Positive Number > 0	200	X
0x25	42	Get / Set	Position Tracking Gain	DINT	Range 0 to 100	0	X
0x25	43	Get / Set	Max Correction Velocity	UINT	Range 1 to 65,535	10,240	X
0x25	45	Get / Set	Max Dynamic Following Error	DINT	Positive Number > 0	10	X
0x25	46	Get / Set	Following Error Action	USINT	2 = Stop motor at programmed deceleration 3 = Do not stop motor	2	X
0x25	47	Get / Set	Following Error Fault	BOOL	0 = No Error 1 = Following Error	0	
0x25	49	Get / Set	Hard Limit Action	USINT	1 = Hard Stop 2 = Smooth Stop	1	X
0x25	50	Get	CW Limit Input	BOOL	0 = Inactive 1 = Active	0	
0x25	51	Get	CCW Limit Input	BOOL	0 = Inactive 1 = Active	0	
0x25	52	Get / Set	Soft Limit Enable	BOOL	0 = Disable 1 = Enable	0	X
0x25	53	Get / Set	Soft Limit Action	USINT	1 = Hard Stop 2 = Smooth Stop	1	X
0x25	54	Get / Set	Positive Software Limit Position	DINT	Range 0x80000001 to 0x7FFFFFFF	0x7FFFFFFF	X
0x25	55	Get / Set	Negative Software Limit Position	DINT	Range 0x80000001 to 0x7FFFFFFF	0x80000001	X
0x25	56	Get	Positive Software Limit State	BOOL	1 = Exceeded Limit	0	
0x25	57	Get	Negative Software Limit State	BOOL	1 = Exceeded Limit	0	
0x25	58	Get / Set	Load Data Complete	BOOL		0	
0x25	102	Get / Set	Position Deadband Extended Range	UINT	Range 0 to 65,535	2	X
0x25	103	Get / Set	Hard limit Input Enable	BOOL	0 = Disable 1 = Enable	0	X
0x25	104	Get / Set	Hard Limit Input Logic	BOOL	0 = Active Low 1 = Active High	0	X

Table 9.5: MicroLYNX DeviceNet Attribute Map (Part 2)

Attribute Map - Part 3 of 3

Object	Attribute	Access Rule	Name	Data Type	Semantics / Description	Factory Default	Stored in NVM
0x25	105	Get	Raw Motor Counts	DINT	Clock pulses sent to the motor drive. Note: Not updated during motion.	0	
0x25	106	Get	Raw Encoder Counts	DINT	Clock pulses received from the encoder. Note: Not updated during motion.	0	
0x25	109	Get / Set	Microstep Resolution	UINT	2 = 400 Microsteps / Revolution 4 = 800 Microsteps / Revolution 8 = 1,600 Microsteps / Revolution 16 = 3,200 Microsteps / Revolution 32 = 6,400 Microsteps / Revolution 64 = 12,800 Microsteps / Revolution 128 = 25,600 Microsteps / Revolution 256 = 51,200 Microsteps / Revolution 5 = 1,000 Microsteps / Revolution 10 = 2,000 Microsteps / Revolutionv 25 = 5,000 Microsteps / Revolution 50 = 10,000 Microsteps / Revolution 125 = 25,000 Microsteps / Revolution 250 = 50,000 Microsteps / Revolution All settings based on a 1.8° motor.	256	X
0x25	110	Get / Set	Initial Velocity	DINT	Positive Number ≥ 1	1000	X
0x25	111	Get / Set	Acceleration Profile	USINT	0 = Linear 2 = Parabolic 128 - Triangle S-Curve 129 - Sinusodial S-Curve	0	X
0x25	112	Get / Set	Deceleration Profile	USINT	0 = Linear 2 = Parabolic 128 - Triangle S-Curve 129 - Sinusodial S-Curve	0	X
0x25	113	Get / Set	Run Current	USINT	Range 1 to 100	25	X
0x25	114	Get / Set	Hold Current	USINT	Range 0 to 100	5	X
0x25	115	Get / Set	Acceleration / Deceleration Current	USINT	Range 1 to 100	25	X
0x25	116	Get / Set	Motor Setting Delay Time	UINT	Range 0 to 65,535	0	X
0x25	117	Get / Set	Hold Current Time Delay	UINT	Range 0 to 65,535	500	X
0x25	118	Get / Set	Position Maintenance Enable	BOOL	0 = Disable 1 = Enable	0	X
0x25	119	Get / Set	Isolated IO Filtering	USINT	0 = Frequency Cutoff - 27.5 kHz Min Pulse Width - 18 µsec 1 = Frequency Cutoff - 13.7 kHz Min Pulse Width - 36 µsec 2 = Frequency Cutoff - 6.89 kHz Min Pulse Width - 73 µsec 3 = Frequency Cutoff - 3.44 kHz Min Pulse Width - 145 µsec 4 = Frequency Cutoff - 1.72 kHz Min Pulse Width - 290 µsec 5 = Frequency Cutoff - 860 kHz Min Pulse Width - 581 µsec 6 = Frequency Cutoff - 430 kHz Min Pulse Width - 1,162 µsec 7 = Frequency Cutoff - 215 kHz Min Pulse Width - 2,323 µsec	7	X
0x25	197	Get / Set	Home Direction	BOOL	0 = Negative Direction 1 = Positive Direction	0	X
0x25	198	Get / Set	Home Fast Velocity	DINT	Positive Number ≥ 0	76,800	X
0x25	199	Get / Set	Home Slow Velocity	DINT	Positive Number ≥ 0	1,000	X
0x25	120	Get / Set	Enable Stall Detect	BOOL	0 = Disabled 1 = Enabled	1	X
0x25	121	Get / Set	Brake Control Mode	BOOL	0 = Manual 1 = Automatic	0	X
0x25	122	Get / Set	Brake Output Logic	BOOL	0 = Output LOW when on. 1 = Output HIGH when on.	0	X
0x25	123	Get / Set	Brake On / Off	BOOL	0 = Brake Output OFF 1 = Brake Output ON	0	
0x25	124	Get / Set	IO Configuration	USINT	0 = Input 1 = Output	0	X
0x25	125	Get / Set	IO Logic	USINT	INPUT 0 = Active LOW 1 = Active HIGH OUTPUT 0 = Active Low when ON 1 = Active High when ON	0	X
0x25	126	Get / Set	IO State	BOOL	INPUT 0 = Input Inactive 1 = Input Active OUTPUT 0 = Output OFF 1 = Output ON	0	

Table 9.6: MicroLYNX DeviceNet Attribute Map (Part 3)

I/O Messaging and Response

IO Messaging

Command Message Format

Byte 0
- bit 7 – Enable
- bit 6 – undefined
- bit 5 – Hard Stop
- bit 4 – Smooth Stop
- bit 3 – Direction (Velocity Mode)
- bit 2 – Incremental
- bit 1 - undefined
- bit 0 – Load Data/ Start Profile

Byte 1 – Byte 7 as defined by ODVA

IO Commands Supported

0x01 – Target Position
0x02 – Target Velocity
0x03 – Acceleration
0x04 – Deceleration
0x11 – Continuous Velocity
0x12 – Start Homing
0x1a – Position Controller Supervisor Attribute
0x1b – Position Controller Attribute

Response Message Format

Byte 0
- bit 7 – Enable state
- bit 6 – undefined
- bit 5 – Home Level
- bit 4 – Current Direction
- bit 3 – General Fault
- bit 2 – On Target
- bit 1 – undefined
- bit 0 – Profile in Progress

Byte 1 – as defined by ODVA

Byte 2
- bit 7 – load complete
- bit 6 – undefined
- bit 5 – Following Error
- bit 4 – Negative Software Limit
- bit 3 – Positive Software Limit
- bit 2 – CCW limit
- bit 1 – CW limit
- bit 0 – undefined

Byte 3 – Byte 7 as defined by ODVA, IO Response Supported

IO Response Supported

0x01 – Actual Position
0x02 – Commanded Position
0x03 – Actual Velocity
0x14 – Command/Response Error
0x1a – Position Controller Supervisor Attribute
0x1b – Position Controller Attribute

Poll IO Command Format

Note that all commands **require** a transition from 0 to 1 in order to execute.

Target Position:

This command starts motion if :
Mode (C: 0x25 A:3) = 0

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Block #							
2	Command Axis Number			Command Message Type				
3	Response Axis Number			Response Message Type				
4	Target Position Low Byte							
5	Target Position Low Middle Byte							
6	Target Position High Middle Byte							
7	Target Position High Byte							

Table 9.7: Target Position Command Message (Type 01 Hex)

Target Velocity:

This Command Starts Motion if:
Mode (C: 0x25 A:3) = 1

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Block #							
2	Command Axis Number			Command Message Type				
3	Response Axis Number			Response Message Type				
4	Target Velocity Low Byte							
5	Target Velocity Low Middle Byte							
6	Target Velocity High Middle Byte							
7	Target Velocity High Byte							

Table 9.8: Target Velocity Command Message (Type 02 Hex)

Acceleration:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Block #							
2	Command Axis Number			Command Message Type				
3	Response Axis Number			Response Message Type				
4	Acceleration Low Byte							
5	Acceleration Low Middle Byte							
6	Acceleration High Middle Byte							
7	Acceleration High Byte							

Table 9.9: Acceleration Command Message (Type 03 Hex)

Deceleration:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Block #							
2	Command Axis Number			Command Message Type				
3	Response Axis Number			Response Message Type				
4	Deceleration Low Byte							
5	Deceleration Low Middle Byte							
6	Deceleration High Middle Byte							
7	Deceleration High Byte							

Table 9.10: Deceleration Command Message (Type 04 Hex)

Continuous Velocity:

This command will start a velocity mode profile (slew) without using explicit messaging. The following attributes are modified with this command:

0x25 – 7: Target Velocity

0x25 – 3: Mode

NOTE: A Hard Stop or a Smooth Stop MUST be issued via IO messaging to revert to a prior mode!

The mode will be restored to the mode prior to the use of the Continuous Velocity command.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Block #							
2	Command Axis Number			Command Message Type				
3	Response Axis Number			Response Message Type				
4	Continuous Velocity Low Byte							
5	Continuous Velocity Low Middle Byte							
6	Continuous Velocity High Middle Byte							
7	Continuous Velocity High Byte							

Table 9.11: Continuous Velocity Command Message (Type 11 Hex)

Start Homing:

This command will start a Homing Sequence without using explicit messaging. The following attributes are modified with this command:

0x24 – 12: Home Arm
0x24 – 13: Actual Position
0x24 – 101: Homing Type
0x25 – 197: Home Direction

The transition of attribute 0x25 – 11 (Load Data/Profile) from 0 to 1 will execute the motion profile. When homing is complete, the Actual Position attribute (0x24 – 13) will be reset to logic state 0.

The Direction bit (Byte 0 - bit 3) will determine the homing direction: Bit 3 = 0 - Home in CW Direction, Bit 3 = 1 - Home in CCW Direction

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Block #							
2	Command Axis Number			Command Message Type				
3	Response Axis Number			Response Message Type				
4	Homing Type							
5	0x00							
6	0x00							
7	0x00							

Table 9.12: Start Homing Command Message (Type 12 Hex)

Position Controller Supervisor Attribute:

This command will set the following attributes without using explicit messaging:

0x24-103: Alarm Clear
0x24-111: Enable NVM Storage

All data in the command message must be valid or the response assembly will be the Error Response. The transition of attribute 0x25 – 11 (Load Data/Profile) from 0 to 1 will set the above attributes.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Position Controller Supervisor Attribute to Get							
2	Command Axis Number			Command Message Type				
3	Position Controller Supervisor Attribute to Set							
4	Position Controller Supervisor Attribute Value Low Byte							
5	Position Controller Supervisor Attribute Value Low Middle Byte							
6	Position Controller Supervisor Attribute Value High Middle Byte							
7	Position Controller Supervisor Attribute Value High Byte							

Table 9.13: Position Controller Supervisor Attribute Command Message (Type 1A Hex)

Example:

This command assembly will get attribute 0x24-68 and set attribute 0x24-67. The set occurs when the Load/ Start Profile bit transitions from 0 to 1.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80	68	3A	67	00	00	00	00
81	68	3A	67	00	00	00	00

Table 9.14: Example of Position Controller Supervisor Attribute

Position Controller Attribute:

This command will set the following attributes without using explicit messaging:
0x25-52: Soft Limit Enable
0x25-53: Soft Limit Action
0x25-54: Positive Software Limit Position
0x25-55: Negative Software Limit Position
0x25-110: Initial Velocity

All data in the command message must be valid or the response assembly will be the Error Response. The transition of attribute 0x25 – 11 (Load Data/Profile) from 0 to 1 will set the above attributes.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Hard Stop	Smooth Stop	Direction (Velocity Mode)	Incremental	Undefined	Load Data / Start Profile
1	Position Controller Attribute to Get							
2	Command Axis Number			Command Message Type				
3	Position Controller Attribute to Set							
4	Position Controller Attribute Value Low Byte							
5	Position Controller Attribute Value Low Middle Byte							
6	Position Controller Attribute Value High Middle Byte							
7	Position Controller Attribute Value High Byte							

Table 9.15: Position Controller Attribute Command Message (Type 1B Hex)

Example:

This command assembly will get attribute 0x25-110 and set attribute 0x25-110. The set occurs when the Load/Start Profile bit transitions from 0 to 1.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80	6E	3B	6E	00	00	00	00
81	6E	3B	6E	00	00	00	00

Table 9.16: Example of Position Controller Attribute

Poll IO Response Format

Actual Position:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Home Level	Current Direction	General Fault	On Target Position	Undefined	Profile In Progress
1	Undefined							
2	Load Complete	Undefined	Following Error	Negative Software Limit	Positive Software Limit	CCW Limit	CW Limit	Fault Input
3	Response Axis Number			Response Message Type				
4	Actual Position Low Byte							
5	Actual Position Low Middle Byte							
6	Actual Position High Middle Byte							
7	Actual Position High Byte							

Table 9.17: Actual Position Response Message (Type 01 Hex)

Commanded Position:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Home Level	Current Direction	General Fault	On Target Position	Undefined	Profile In Progress
1	Undefined							
2	Load Complete	Undefined	Following Error	Negative Software Limit	Positive Software Limit	CCW Limit	CW Limit	Fault Input
3	Response Axis Number			Response Message Type				
4	Command Position Low Byte							
5	Command Position Low Middle Byte							
6	Command Position High Middle Byte							
7	Command Position High Byte							

Table 9.18: Commanded Position Response Message (Type 02 Hex)

Actual Velocity:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Home Level	Current Direction	General Fault	On Target Position	Undefined	Profile In Progress
1	Undefined							
2	Load Complete	Undefined	Following Error	Negative Software Limit	Positive Software Limit	CCW Limit	CW Limit	Fault Input
3	Response Axis Number			Response Message Type				
4	Actual Velocity Low Byte							
5	Actual Velocity Low Middle Byte							
6	Actual Velocity High Middle Byte							
7	Actual Velocity High Byte							

Table 9.19: Actual Velocity Response Message (Type 03 Hex)

Command / Response Error:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Home Level	Current Direction	General Fault	On Target Position	Undefined	Profile In Progress
1	Reserved = 0							
2	Load Complete	Undefined	Following Error	Negative Software Limit	Positive Software Limit	CCW Limit	CW Limit	Fault Input
3	Response Axis Number			Response Message Type				
4	General Error Code							
5	Additional Code							
6	Copy of Command Message Byte 2							
7	Copy of Command Message Byte 3							

Table 9.20: Command/Response Error Response Message (Type 14 Hex)

Position Controller Supervisor Attribute:

This command will get the following attributes without using explicit messaging:

0x24-103: Alarm Clear
0x24-104: Alarm Code
0x24-111: Enable NVM Storage

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Home Level	Current Direction	General Fault	On Target Position	Undefined	Profile In Progress
1	Position Controller Supersor Attribute to Get							
2	Load Complete	Undefined	Following Error	Negative Software Limit	Positive Software Limit	CCW Limit	CW Limit	Fault Input
3	Response Axis Number			Response Message Type				
4	Position Controller Supervisor Attribute Value Low Byte							
5	Position Controller Supervisor Attribute Value Low Middle Byte							
6	Position Controller Supervisor Attribute Value High Middle Byte							
7	Position Controller Supervisor Attribute Value High Byte							

Table 9.21: Position Controller Supervisor Attribute Response Message (Type 1A Hex)

Example:

All data in the command message must be valid or the response assembly will be the Error Response. This command assembly will get attribute 0x24-68.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80	68	3A	67	00	00	00	00

Table 9.22: Example of Position Controller Supervisor Attribute

Position Controller Attribute:

This command will get the following attributes without using explicit messaging:

0x25-52: Soft Limit Enable
0x25-53: Soft Limit Action
0x25-54: Positive Software Limit Position
0x25-55: Negative Software Limit Position
0x25-110: Initial Velocity

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Undefined	Home Level	Current Direction	General Fault	On Target Position	Undefined	Profile In Progress
1	Position Controller Attribute to Get							
2	Load Complete	Undefined	Following Error	Negative Software Limit	Positive Software Limit	CCW Limit	CW Limit	Fault Input
3	Response Axis Number			Response Message Type				
4	Position Controller Attribute Value Low Byte							
5	Position Controller Attribute Value Low Middle Byte							
6	Position Controller Attribute Value High Middle Byte							
7	Position Controller Attribute Value High Byte							

Table 9.23: Position Controller Attribute Response Message (Type 1B Hex)

Example:

All data in the command message must be valid or the response assembly will be the Error Response. This command assembly will get attribute 0x25-110.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80	6E	3B	6E	00	00	00	00

Table 9.24: Example of Position Controller Attribute

Poll IO Message Example

Set Variables and Flags:

This example is shown using a 1.8° stepping motor, a 500 line encoder with encoder feedback enabled. Assumes all controller variables and flags are set to factory default.

The following Variables and Flags will be set as shown.

MicroLYNX Command	Attribute
VM = 30,000	0x25 - 7
VI = 39	0x25 - 110
ACCL = 30,000	0x25 - 8
DECL = 30,000	0x25 - 9
Home Fast VEL = 3,000	0x25 - 198
Home Slow VEL = 39	0x25 - 199
Feedback Enable = ON	0x25 - 39
Max. Correction Velocity = 400	0x25 - 43

Table 9.25: Example of Set Variables and Flags

Commands:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80	00	32	21	01	00	00	00
81	00	32	21	01	00	00	00

Table 9.26: Start Homing Sequence to Home Switch, Return to Current Position

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80	00	31	21	00	75	00	00
81	00	31	21	00	75	00	00

Table 9.27: Continuous Velocity in the CCW Direction

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
90	00	31	21	00	75	00	00

Table 9.28: Smooth Stop, Restore Mode 1

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
85	00	21	21	D0	07	00	00

Table 9.29: Target Position = 1 Revolution

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
88	00	31	21	40	9C	00	00
89	00	31	21	40	9C	00	00

Table 9.30: Continuous Velocity in the CW Direction

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
90	00	31	21	40	9C	00	00

Table 9.31: Smooth Stop, Restore Mode 2

Homing

C:0x24 A:101	Home Type
C:0x24 A:11	Home Active Level
C:0x24 A:12	Home Arm
C:0x24 A:16	Home Input Level
C:0x25 A:197	Home Direction
C:0x25 A:198	Home Fast Velocity
C:0x25 A:199	Home Slow Velocity

Homing will move in the direction specified, at the programmed acceleration/deceleration and Home Fast Velocity. When the home switch changes states, direction will reverse and the motor will move off the home switch at the Home Slow Velocity. Once the home switch changes state again, the motor will decelerate to a stop. Homing is now complete.

Description of GO and Moving Bit

The DeviceNet MicroLYNX is designed to conform to ODVA. The GO command will be implemented through Command Message, Byte 0, bit 0 – Load Data/Start Profile. The Moving Bit will be implemented through the Response Message, Byte 0, bit 0 – Profile in Progress.

Motion

When the encoder is not enabled, the MicroLYNX will move in microsteps. When the encoder is enabled, the MicroLYNX will move in encoder steps. This is fixed at (number of lines * 4) / revolution.

Alarm Codes

Alarm Code	MS Led	NS Led	Alarm Description
0x20		Solid Red	Duplicate MAC ID
0x21		Solid Red	Bus Off
0x044C	Solid Red		Drive Error
0x1771	Flash Red		Reached +Limit Switch
0x1772	Flash Red		Reached - Limit Switch
0x1B59	Flash Red		Stall
0x1B5B	Flash Red		Moved Out of Deadband
0x4E21	Flash Red		Reached + Software Limit
0x4E22	Flash Red		Reached - Software Limit

Table 9.32: Alarm Codes

Soft Limits

The soft limits need to be configured prior to being enabled. The soft limit commands must be issued in the following order:

0x25-54, 0x25-55, 0x25-53

Once configured, they may be enabled with 0x25-52.

Encoder Configuration Example

Step 1:

Shown using a 1.8° motor, 500-line encoder, starting with DeviceNet MicroLYNX in the factory default state.

Service	Class	Instance	Data (Hex)	Note
0x10	0x25	1	28 D0 07 00 00	500 X 4 = 2000
0x10	0x25	1	27 01	
0x10	0x25	1	6E 27 00 00 00	
0x10	0x25	1	07 30 75 00 00	
0x10	0x25	1	08 96 98 00 00	
0x10	0x25	1	09 96 98 00 00	
0x10	0x25	1	2D 90 01 00 00	

Table 9.33: Encoder Configuration

Note: C: 0x25, I: 1, A: 41 Default is 200.

Step 2:

Power down the DeviceNet MicroLYNX.

Step 3:

Send IO message (hex): 85 00 21 21 D0 07 00 00
On transition of start trajectory bit from low to high, motor will turn 1 revolution.

Note: All motion is in encoder counts.

DeviceNet Programmer

Introduction and Description

The DeviceNet Programmer is a cable that consists of a powered RS-232 converter with a DB-9F plug on one end to connect to the customer PC Comm port and a 5 Pin DeviceNet Port connector on the other to connect to the DeviceNet MicroLYNX. This cable is used for the sole purpose of communicating with, and configuring the DeviceNet MicroLYNX.

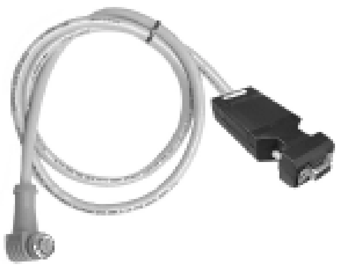


Figure 9.7: DeviceNet Programmer

The length of the cable is approximately 39.37 inches (1 meter). The DeviceNet Programmer has a power jack for the user to connect an external power source. The operating voltage is +24 VDC. You may order the DeviceNet Programmer from IMS under Part Number **MX-CC600-000.**

The DeviceNet Programmer is used to communicate with a single DeviceNet MicroLYNX. It is not designed for, and **must not be used** as an interface cable for the DeviceNet Bus.

Figure 9.8: DeviceNet MicroLYNX

Dimensions in Inches (mm)

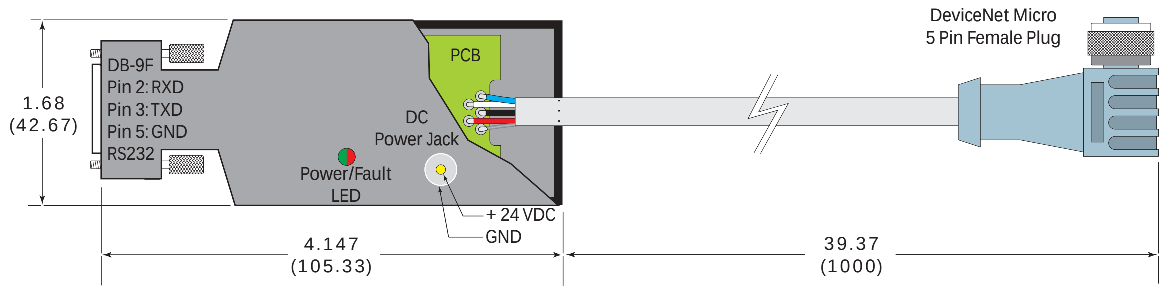


Figure 9.9: MX-CC600-000 DeviceNet Programmer Details

Using the DeviceNet Programmer

- Disconnect the DeviceNet MicroLYNX from the DeviceNet Buss.
- Connect the DeviceNet Programmer to the Comm Port on your PC.
- Connect the 5 Pin Micro Plug to the DeviceNet Jack on the MicroLYNX.
- Connect a +24 VDC supply to the DeviceNet Programmer.
- Establish communication and configure or update the DeviceNet MicroLYNX as required.

WARNING! The DeviceNet Programmer may only be used on a single DeviceNet MicroLYNX for communication purposes. It is **NOT** intended to be used as an interface with DeviceNet. **NEVER** connect the DeviceNet Programmer to the DeviceNet Buss or damage may occur to the DeviceNet system.

Section 10

Configuring the Isolated Digital I/O

Section Overview

This section covers the usage of the Isolated Digital I/O which is available on the MicroLYNX System

- Electrical Characteristics.
- The Isolated Digital I/O:
  - Configuring an Input
  - Setting the Digital Input Filtering for the Isolated I/O
  - Configuring an Output
  - Setting the Binary State of an I/O Group

Electrical Characteristics

Number of I/O	6
Input Voltage	+5 to +24VDC
Output Current Sink	350mA
Input Filter Range	215Hz to 21.5kHz (Programmable)
Pull-ups	7.5kOhm individually switchable
Pull-up Voltage	
Internal	+5VDC
External	+24 VDC
Protection	Over temp, short circuit, inductive clamp
Isolated Ground	Common to the 6 I/O

The Isolated Digital I/O

The MicroLYNX System comes standard with a set of six (6) +5 to +24VDC I/O lines which may be programmed individually as either general purpose or dedicated inputs or outputs, or collectively as a group. The isolated digital I/O may also be expanded to twenty-four (24) lines in groups of six (6).

The I/O groups and lines are numbered in the following fashion:

Group 20 = Lines 21 - 26 (Standard)
Group 30 = Lines 31 - 36 (Optional)
Group 40 = Lines 41 - 46 (Optional)
Group 50 = Lines 51 - 56 (Optional)

The isolated digital I/O may be defined as either active HIGH or active LOW. When the I/O is configured as active HIGH, the level is +5 to +24 VDC and the state will be read as a “1”. If the level is 0 VDC then the state will be read as “0”. Inversely, if configured as active LOW, then the state of the I/O will be read as a “1” when the level is LOW, and a “0” when the level is HIGH. The active HIGH/LOW state is configured by the third parameter of the IOS variable, which is explained further on. The goal of this I/O configuration scheme is to maximize compatibility between the MicroLYNX and standard sensors and switches.
The MicroLYNX I/O scheme is a powerful tool for machine and process control. Because of this power, a level of complexity in setup and use is found that doesn’t exist in controllers with a less capable I/O set.

Uses of the Isolated Digital I/O

The isolated I/O may be utilized to receive input from external devices such as sensors, switches or PLC outputs. When configured as outputs, devices such as relays, solenoids, LED's and PLC inputs may be controlled from the MicroLYNX. Depending on the device connected, the input or output may be pulled-up to either the internal +5VDC supply or an external +5 to +24VDC supply, or the I/O lines may be pulled-down to ground. These features, combined with the programmability and robust construction of the MicroLYNX I/O, open an endless vista of possible uses for the I/O in your application.

Each I/O line may be individually programmed to any one of 8 dedicated input functions, 7 dedicated output functions, or as general purpose inputs or outputs. The I/O may be addressed individually or as a group. The active state of the line or group may also be set. All of these possible functions are accomplished with of the IOS variable.

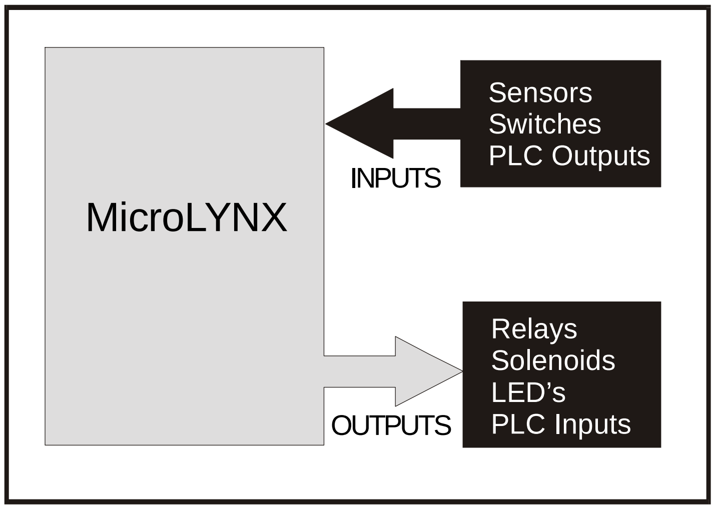


Figure 10.1: Isolated I/O Applications

The IOS Variable

The IOS variable has three parameters when used to configure the isolated digital I/O. These are:

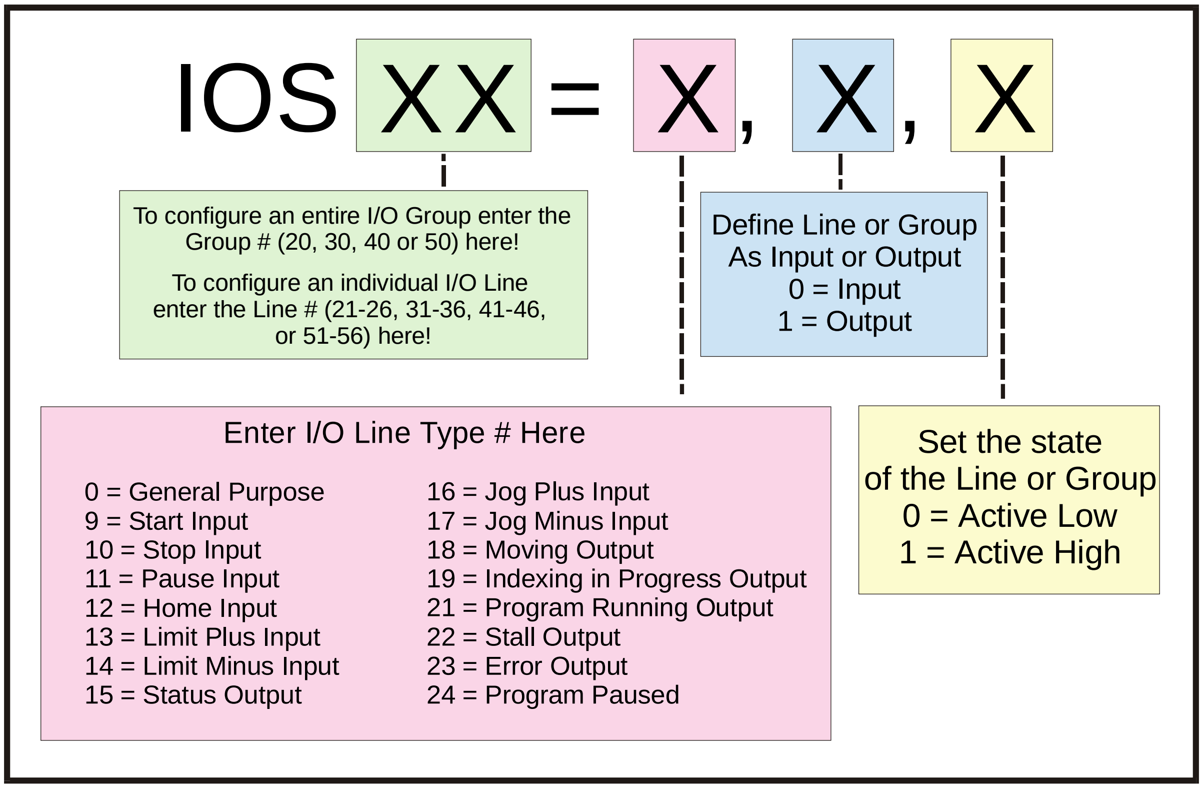
1] **I/O Line Type:** Specifies the the type of I/O that the line or group will be configured as, i.e. general purpose or dedicated function.

2] **I/O Line Function:** Either an input or an output.

3] **Active State:** Specifies whether or not the line will be active HIGH or active LOW.

The default configuration of the standard I/O set is: 0,0,1. This means that by default each line in group 20 is configured to be a General Purpose (0), Input (0), which is active when HIGH (1). On the following page, Table 10.1 and the exercises illustrate possible configurations of the IOS.

NOTE: When configuring a dedicated input or output, the second parameter of the IOS Variable MUST match the function, either input or output, or an error will occur.



IOS XX = X, X, X

To configure an entire I/O Group enter the Group # (20, 30, 40 or 50) here!

To configure an individual I/O Line enter the Line # (21-26, 31-36, 41-46, or 51-56) here!

Define Line or Group As Input or Output
0 = Input
1 = Output

Enter I/O Line Type # Here

0 = General Purpose	16 = Jog Plus Input
9 = Start Input	17 = Jog Minus Input
10 = Stop Input	18 = Moving Output
11 = Pause Input	19 = Indexing in Progress Output
12 = Home Input	21 = Program Running Output
13 = Limit Plus Input	22 = Stall Output
14 = Limit Minus Input	23 = Error Output
15 = Status Output	24 = Program Paused

Set the state of the Line or Group
0 = Active Low
1 = Active High

Table 10.1: IOS Variable Settings

Configuring an Input

On the following page, Figures 10.2 and 10.3 illustrate the Input Equivalent Circuit of the Isolated I/O being used with a mechanical switch and a proximity switch. To illustrate the usage of an input you will go through the steps to configure this switch to start a simple program at Line 1000 to index a motor 200 user units. First you must configure the I/O Line 21 as a "GO" input:

```
IOS 21 = 9, 0, 0
```

To break this command down:

IOS 21 - Identifies the I/O Line we are setting as 21.
9 - Configures the I/O Type to "GO".
0 - Configures I/O as Input.
0 - Configures I/O as Active LOW.

When the Input Type "GO" is selected it will always look to execute a program located at line 1 of program memory. Therefore, to run a program at line 1000 you must do the following:

```
PGM 1        'Records program at line 1 of memory space
EXEC 1000    'Execute program located at line 1000 of memory space
END          'Terminates Program
PGM          'Switches system back to immediate mode

PGM 1000     'Records program at line 1000 of memory space
MOVR 200     'Move relative to current position 200 user units
HOLD 2       'Hold program execution until specified motion is
             'completed
END
PGM
```

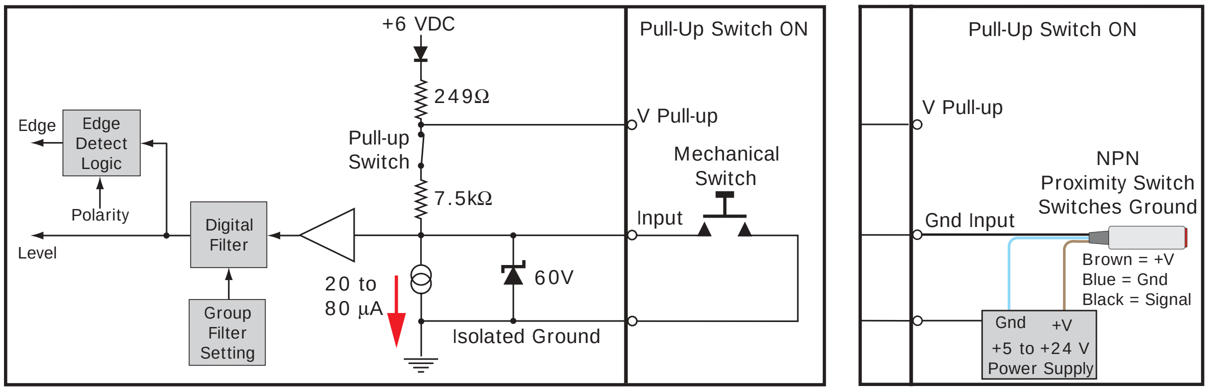


Figure 10.2: Typical Sinking Isolated Input Equivalent Circuit

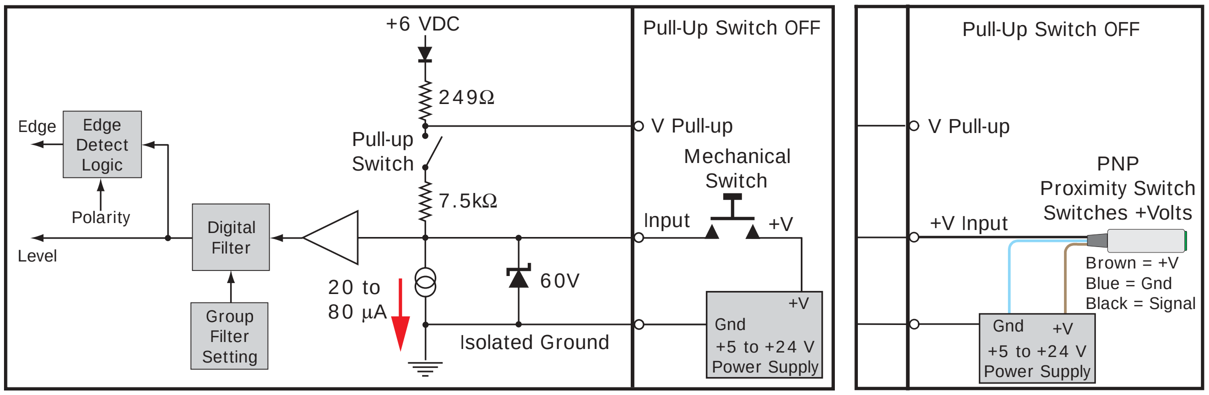


Figure 10.3: Typical Sourcing Isolated Input Equivalent Circuit

Configuring the Digital Filtering

User definable Digital filtering makes the MicroLYNX well suited for noisy industrial environments. The filter setting is software selectable using the IOF Variable with a minimum guaranteed detectable pulse width of 18 microseconds to 2.3 milliseconds.

Table 10.2 illustrates the IOF settings.

The filter setting will reject any frequency above the specified bandwidth. For example:

```
IOF 2 = 3   ‘Set the
Digital Filter for I/O
Group 20 to 3.44kHz
```

IOF Filter Settings for the General Purpose Isolated I/O IOF=<num> (<num> = 0-7)		
Filter Setting	**Cutoff Frequency**	**Minimum Detectable Pulse Width**
0	27.5 kHz	18 microseconds
1	13.7 kHz	36 microseconds
2	6.89 kHz	73 microseconds
3	3.44 kHz	145 microseconds
4	1.72 kHz	290 microseconds
5	860 Hz	581 microseconds
6	430 Hz	1.162 milliseconds
7 (default)	215 Hz	2.323 milliseconds

Table 10.2: Digital Filter Settings for the Isolated I/O

This setting will cause any signal above 3.44 kHz on I/O lines 21-26 to be rejected. The default filter setting for the isolated I/O groups is 7, or 215Hz.

Configuring an Output

Figure 10.4 illustrates the Output equivalent circuit of the Isolated I/O. When used as an output the I/O line is able to sink 350mA continuous for each output, or a total of 1.5A for the entire I/O Group. Refer to *The Isolated Digital I/O Module* in Section 11 for detailed specifications. In the usage example we will use an LED on I/O Line 31 for the load. We will use the same program from the input example, only we will use the output to light the LED while the motor is moving.

```
IOS 31 = 18, 1,  1
```

Using Table 10.2 on the previous page we can break this setting down as follows:
IOS 31 - Identifies that I/O line 31 is being configured.
18 - Configures the I/O Type as “Moving”.
1 - Configures the I/O line as an output.
1 - Configures the Line as “Active HIGH”.

Now when the input program above is executed, the LED will be lit during the move.

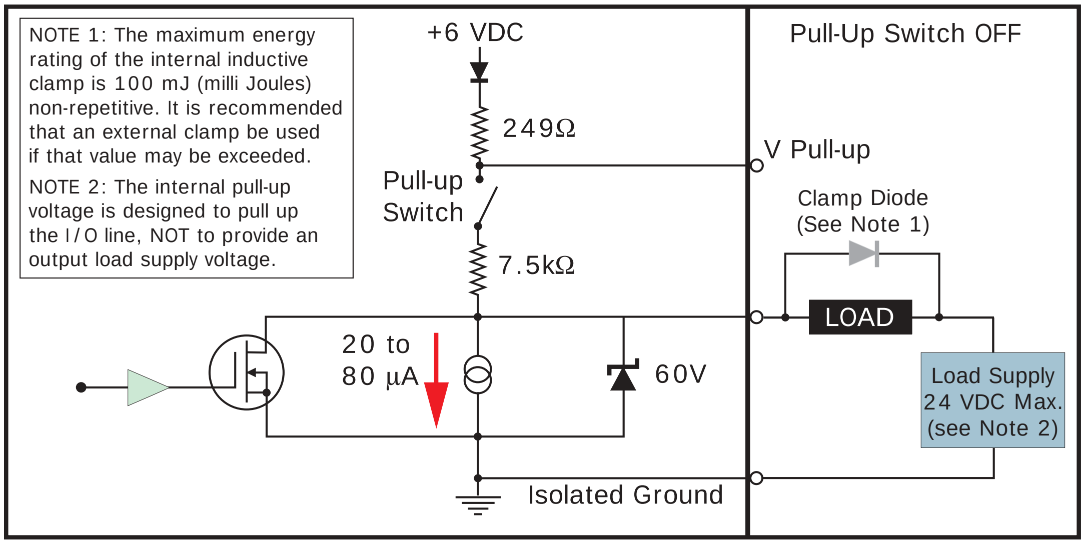

NOTE 1: The maximum energy rating of the internal inductive clamp is 100 mJ (milli Joules) non-repetitive. It is recommended that an external clamp be used if that value may be exceeded.
NOTE 2: The internal pull-up voltage is designed to pull up the I/O line, NOT to provide an output load supply voltage.
+6 VDC
249Ω
Pull-up Switch
7.5kΩ
20 to 80 µA
60V
Isolated Ground
Pull-Up Switch OFF
V Pull-up
Clamp Diode (See Note 1)
LOAD
Load Supply 24 VDC Max. (see Note 2)

Figure 10.4: Isolated Digital I/O Output Equivalent Circuit

The IO Variable

After configuring the I/O by means of the IOS variable, we need to be able to do two things with the I/O.

1] Write to an output, or group of outputs, thus setting or changing its (their) state.
2] Read the states of either inputs or outputs. We can use this information to either display those states to our terminal, or to set up conditions for branches and subroutine calls within a program.

We can also use this command to write or read the state of an entire I/O group.

Read / Write a Single I/O Line

To read the state of a single input or output, the following would be typed into the terminal:

```
PRINT IO 21
```

The response from this would be 1 or 0, depending on the state of the line.
The state of an input or output in a program can be used to direct events within a MicroLYNX program by either calling up a subroutine using the “CALL” instruction, or conditionally branching to another program address using the “BR” instruction. This would be done in the following fashion.

```
CALL MYSUB, IO 22=1
```

This would call up a subroutine labled “MYSUB” when I/O line 21 is active.

```
BR 200, IO 22=0
```

This would branch to address 200 when I/O line 22 is inactive.

Writing to an output is accomplished by entering the following into a terminal or program:

```
IO 21=1
IO 21=0
```

This would change the state of I/O line 21.

Read / Write an I/O Group

When using the IO variable to read the state of a group of inputs/outputs, or write to a group of outputs, you would first want to configure the entire I/O group to be general purpose inputs or outputs using the IOS variable. In this case the response or input won’t be a logic state of 1 or 0, but rather the decimal equivalent (0 to 63) of the 6 bit binary number represented by the entire group.

When addressing the I/O as a group the LSB (*Least Significant Bit*) will be line 1 of the group, (*e.g. 21, 31, 41, 51*). The MSB (*Most Significant Bit*) will be line 6 of the group (*e.g. 26, 36, 46, 56*).

The table on the left shows the bit weight of each I/O line in the group. It also illustrates the state should 6 LED’s be connected to I/O group 20 when entering the IO variables in this exercise.

Configure the IOS variable such that group 20 is all general purpose outputs, active low or:

```
IOS 20 = 0,1,0
```

Enter the following in the terminal:

```
IO 20 = 35
```

As shown in the table I/O lines 26, 22 and 21 should be illuminated, 25, 24 and 23 should be off.

Enter this next:

```
IO 20 = 7
```

Now I/O 21, 22 and 23 should be illuminated.

```
IO 20 = 49
```

I/O 26, 25, and 21 are illuminated.

BIT WEIGHT DISTRIBUTION TABLE FOR GROUP 20 I/O					
I/O 26 MSB	I/O 25	I/O 24	I/O 23	I/O 22	I/O 21 LSB
32	16	8	4	2	1

BINARY STATE OF I/O GROUP 20 IO 20 = 35					
○	●	●	●	○	○
1	0	0	0	1	1
I/O 26 MSB	I/O 25	I/O 24	I/O 23	I/O 22	I/O 21 LSB

BINARY STATE OF I/O GROUP 20 IO 20 = 7					
●	●	●	○	○	○
0	0	0	1	1	1
I/O 26 MSB	I/O 25	I/O 24	I/O 23	I/O 22	I/O 21 LSB

BINARY STATE OF I/O GROUP 20 IO 20 = 49					
○	○	●	●	●	○
1	1	0	0	0	1
I/O 26 MSB	I/O 25	I/O 24	I/O 23	I/O 22	I/O 21 LSB

Table 10.3: Binary State of Outputs

NOTE: You can only write to General Purpose Outputs. If you attempt to write to an input or dedicated output type, an error will occur!

Section 11

Configuring and Using the Expansion Modules

Section Overview

This section covers the configuration and usage of the optional expansion modules available for the MicroLYNX System. The information covered in this section are:

- Isolated Digital I/O Module.
- High-Speed Differential I/O Module.
- Typical Functions of the Differential I/O.
- Analog Input/Joystick Interface Module.
- Typical Functions of the Analog Input/Joystick Module.
- Isolated Communications RS-232 Expansion Module (CAN only).
- Isolated Communications RS-485 Expansion Module (CAN only).
- Analog Output Module.
- 12 Channel Isolated Digital I/O Module.

MicroLYNX Expansion Modules

Additional Isolated Digital I/O Module

The Isolated Digital I/O can be expanded an additional 3 groups (30 - 50) for a total of 24 programmable I/O lines. These Modules may be installed in any available slot. The group number will be determined by the MicroLYNX slot into which they are plugged: slot 1 will be group 30, slot 2 will be group 40, and slot 3 will be group 50. These Expansion Modules are configured and used in the same manner as the Standard I/O on the MicroLYNX. The IMS Part # is MX-DI100-000 (8 Pin Terminal) or MX-DI200-000 (10 Pin Header).

High-Speed Differential I/O Module

If your system requires closed loop motion control and/or ratio functions, such as following or electronic gearing or the ability to sequentially control a second axis, up to two High-Speed Differential I/O Modules can be installed in slots 2 and 3 of the MicroLYNX, giving three channels of high-speed differential (or single) I/O a piece. The IMS Part # is MX-DD100-000 (8 Pin Terminal) or MX-DD200-000 (10 Pin Header).

Analog Input/Joystick Interface Module

The Analog Input/Joystick Interface Module features two 12 bit, 0 to +5 volt input channels which can be used to monitor devices such as temperature and pressure sensors. It can also be used to control an axis with a joystick. It features two voltage outputs: a 5 volt joystick reference, and a precision 4.096 volt calibration reference. This device can be installed in any available MicroLYNX slot. The IMS Part # is MX-AJ100-000 (8 Pin Terminal) or MX-AJ200-000 (10 Pin Header).

Isolated Communications Module

The Isolated Communications Module allows a second, fully independent Communication Port when using a CAN Based MicroLYNX Control System. This Module comes in a choice of RS-232 or RS-485. The IMS Part# is MX-CM102-000 (RS-232 w/8 Pin Terminal), MX-CM202-000 (RS-232 w/10 Pin Header), MX-CM104-000 (RS-485 w/8 Pin Terminal) or MX-CM204-000 (RS-485 w/10 Pin Header).

Analog Output Module

The Analog Output Module provides two 0 to +5 VDC Output Channels (four if two Modules are used). This Module adds the capability to control AC Variable Frequency Drives, Servo Drives and Brush-Type DC Drives. The IMS Part# is MX-DA100-000 (8 Pin Terminal) or MX-DA200-000 (10 Pin Header).

12 Channel Isolated Digital I/O Module

The 12 Channel Isolated Digital I/O Module provides twelve +5 to +24 VDC Isolated I/O Channels. This coupled with a the Six Channel Isolated I/O Module will yield the maximum twenty four Isolated I/O Channels in the MicroLYNX and leave an open slot for an Expansion Module of a different type. The IMS Part# is MX-DI400-000 (16 Pin Header) or MX-DI401-000 (16 Pin and Receptacle).

Choosing the Expansion Modules for Your Application

A powerful feature of the MicroLYNX is the versatility offered by its wide range of configurations available through the expansion modules. The expansion modules listed above may be used singly or in combination to customize your MicroLYNX System to the specific requirements of your application. The table and explanation below outline the application requirements and MicroLYNX Slot usage.

MicroLYNX Expansion Slot Usage				
Expansion Module	**Slot 1**	**Slot 2**	**Slot 3**	**Maximum Allowed**
Isolated Digital I/0	Yes	Yes	Yes	3*
High Speed Differential I/O	No	Yes	Yes	2
Analog Input/Joystick	Yes	Yes	Yes	1
Isolated Communication	No	Yes	No	1
Analog Output	Yes	Yes	Yes	2
12 Channel I/O	Yes	Yes	No	1*

** The MicroLYNX is capable of handling up to 24 I/O signals. If you should opt to use a 12 Channel I/O Module, you can only use one Isolated Digital I/O Module.*

Table 11.1: MicroLYNX Expansion Module Slot Usage

Explanation of Expansion Slot Usage

Isolated Digital I/O Module

The Isolated Digital I/O Module may be used in any MicroLYNX Slot. You may use up to three (3) 6-channel Modules in a MicroLYNX. However, the MicroLYNX is capable of a maximum 24 I/O channels. If you elect to use one 12 Channel I/O Module, you may only use one Isolated Digital I/O Module.

High-Speed Differential I/O Module

Up to 2 High-Speed Differential Modules may be used in Slots 2 or 3.

Analog Input/Joystick Interface Module

The Analog Input/Joystick Interface Module may be used in any Slot of the MicroLYNX. Only 1 Module may be used per MicroLYNX.

Isolated Communications Module

The Isolated Communications Module may be used only in Slot 2 of the MicroLYNX. Only 1 Module may be used and it must be with a CAN MicroLYNX System.

Analog Output Module

The Analog Output Module may be used in any slot of the MicroLYNX. A maximum of 2 Modules may be used in a MicroLYNX.

12 Channel Isolated Digital I/O Module

The 12 Channel Isolated Digital I/O Module can be used in Slot 1 or 2 of the MicroLYNX. Only 1 (one) 12 Channel Module may be used. You may elect to use 1 standard Isolated Digital I/O Module along with the 12 Channel Module to give the MicroLYNX the maximum number of 24 Isolated I/O channels.

Thermal/Environmental Specifications

All MicroLYNX Expansion Modules have thermal and environmental specifications which must be met. They are:

	Range
Ambient Temperature (Operating)	0 to +50°C
Storage Temperature	-20 to +70°C
Humidity	0 to 90% non-condensing

Isolated Digital I/O Module

The Isolated Digital I/O can be expanded to 24 lines. Expansion to this level would require the use of all three slots. The I/O groups are slot dependent. The slots will yield the following groups as numbered:

Slot 1	Group 30
Slot 2	Group 40
Slot 3	Group 50

The Isolated Digital I/O Module expands the capabilities of the MicroLYNX to include application features such as:

1) Six +5 to +24 VDC Isolated Input Channels
2) I/O Lines Software Configurable as Inputs or Outputs
3) I/O user definable as Dedicated or General Purpose
4) Programmable Digital Filtering for Inputs

Electrical Specifications

Input Voltage Range	0 to +24 VDC
Input Low Level	< 1.5 Volts
Input High Level	> 3.5 Volts
Open Circuit Input Voltage	
Pull-up Switch ON	4.5 Volts
Pull-up Switch OFF	0 Volts
Load Supply Voltage	28 VDC Maximum (Transient protected at 60 volts)
FET On Resistance	2W Maximum (Tj=125°C)
Continuous Sink Current	350 mA max each output (Ta = 25°C)
Maximum Group Sink	1.5 A (Thermally Limited)
Filter Cutoff Frequencies	27.5, 13.7, 6.89, 3.44, 1.72 kHz, 860, 430, 215 Hz

Pin #	Connector Option					
	8 Position Phoenix			10 Pin Header		
	Slot 1	Slot 2	Slot 3	Slot 1	Slot 2	Slot 3
1	V_{pullup}	V_{pullup}	V_{pullup}	IO 31	IO 41	IO 51
2	IO 31	IO 41	IO 51	IO 32	IO 42	IO 52
3	IO 32	IO 42	IO 52	V_{pullup}	V_{pullup}	V_{pullup}
4	IO 33	IO 43	IO 53	IO 33	IO 43	IO 53
5	IO 34	IO 44	IO 54	N.C.	N.C.	N.C.
6	IO 35	IO 45	IO 55	IO 34	IO 44	IO 54
7	IO 36	IO 46	IO 56	N.C.	N.C.	N.C.
8	I/O GND	I/O GND	I/O GND	IO 35	IO 45	IO 55
9				I/O GND	I/O GND	I/O GND
10				IO 36	IO 46	IO 56

Table 11.2: Isolated Digital I/O Group and Line Locations by Connector Option and Slot

I/O Configuration

Inputs and Outputs as well as digital filtering are configured in the same manner as the Standard I/O (Group 20). Please refer to Section 10 “Configuring the Isolated Digital I/O” for details.

The Isolated Digital I/O can be expanded to 24 lines. Expansion to this level would require the use of all three slots. The I/O groups are slot dependent. The slots will yield the following groups as numbered:

Slot 1	Group 30
Slot 2	Group 40
Slot 3	Group 50

Installing The Isolated Digital I/O Module

To install the Isolated Digital I/O Expansion Module in your MicroLYNX perform the following in accordance with Figure 11.1.

To Install the Module:

1) Remove the two retaining screws (A) from the cover.
2) Remove the blank panel (1, 2, or 3) from the desired slot you want to use.
3) Carefully press the Expansion Module (B) into place by plugging the 28 pin connector into the desired receptacle (C, D, or E) and snapping it into place under the retaining clips (F).
4) Reinstall the MicroLYNX cover.
5) Affix the labels supplied with the Module as shown.

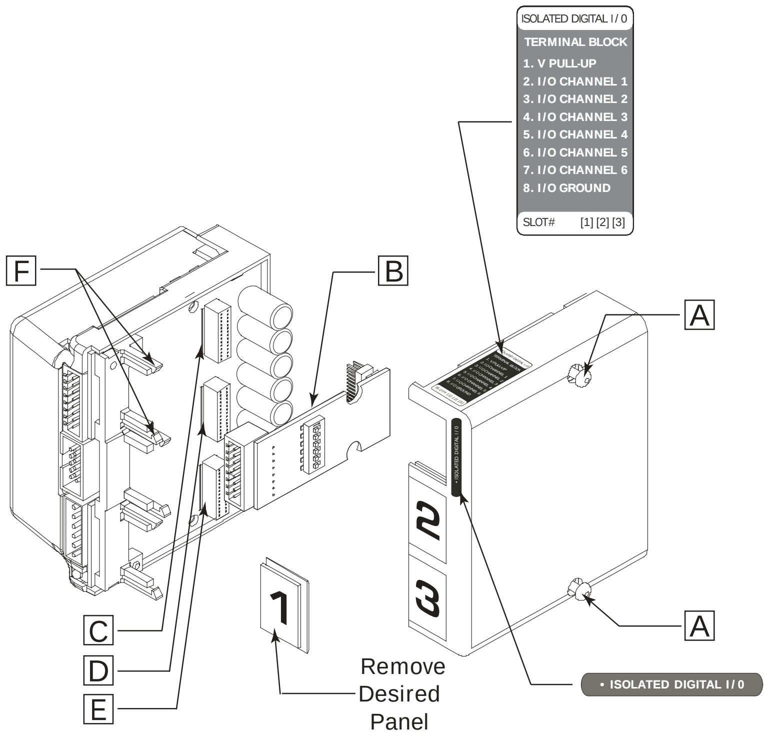


Figure 11.1: Installing the Isolated Digital I/O Expansion Module

Using the Isolated Digital I/O

The Isolated Digital Expansion I/O operates in the very same manner as the standard isolated I/O. The only differences are the location of the pull-up switches, and the method of supplying an external pull-up voltage.

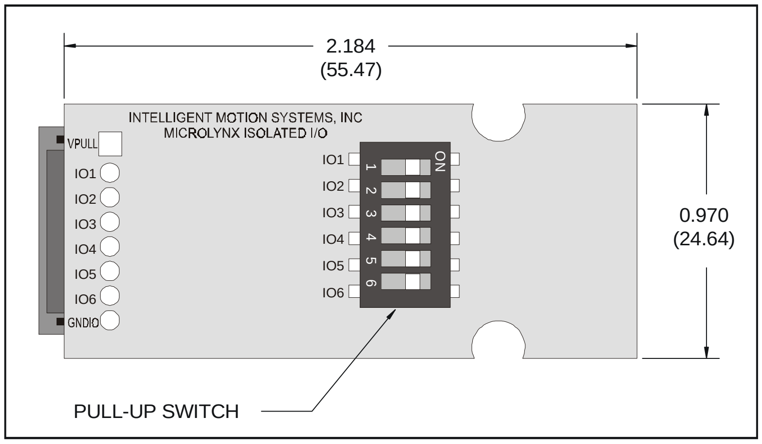


Figure 11.2: The Isolated Digital I/O Module, Bottom View

The pull-up switches are located on the bottom of the expansion board. They operate in the same fashion as the standard I/O set pull-ups. Configuring and using these switches is detailed in Section 10 of this document. Another key difference is the method by which an external pull-up voltage is supplied to the I/O. While the I/O Ground is common to each Isolated Digital I/O Module installed (both the Differential I/O Module and the Analog Input/ Joystick Module have separate, non-isolated grounds) V-PULLUP is **NOT** common. This allows you to power each I/O group independently if you choose.

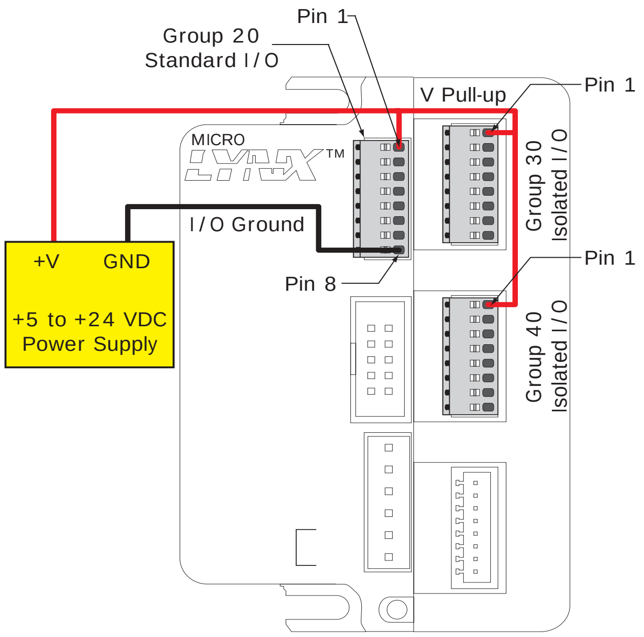


Figure 11.3: Powering Multiple Isolated Digital I/O Modules

The expansion isolated digital I/O is configured and controlled by the IOS variable and the IO instructions in the same manner as the standard I/O set. The only difference is in how the lines and groups are addressed.

See Section 10 for instructions on using the isolated I/O

.

If digital filtering is used (IOF variable) it must be configured for each group separately.

High-Speed Differential I/O Module

The MicroLYNX has the capability of having up to two High-Speed Differential I/O Modules installed in expansion slot numbers 2 and 3. The High-Speed Differential I/O Module expands the capabilities of the MicroLYNX to include application features such as:

1] Closed Loop Motion Control (Encoder Feedback)
2] Electronic Gearing (Ratio Functions)
3] Secondary Clock Output
4] General Purpose High-Speed I/O

The pinout by slot location and connector style is given in Table 11.3.

The high-speed differential I/O is non-isolated, meaning the ground is not common with the isolated I/O ground.

Electrical Specifications

Differential Input Threshold	-0.2 to +0.2 Volts
Input Hysteresis	60 Millivolts Typical
Input Common Mode Range	-6 to +6 Volts
Open Circuit Input Voltage	
Positive Input	4.3 Volts
Negative Input	1.4 Volts
Output Voltage (each output)	No Load/6 Milliamp Load
Logic “0”	0.5 Volts/0.8 Volts
Logic “1”	4.5 Volts/4.2 Volts
Encoder Voltage	+5VDC Output
Current Limit (All Outputs Combined)	150 mA
Short Circuit Current	250 mA Max.
Maximum Clock Frequency	5MHz
Digital Input Filtering	39kHz to 5MHz
Filter Cutoff Frequencies	5.00, 2.50, 1.25 MHz, 625, 313, 156, 78.1, 39.1 kHz

Pin #	Connector Option			
	8 Position Phoenix		10 Pin Header	
	Slot 2	Slot 3	Slot 2	Slot 3
1	I/O 17(-)	I/O 18(-)	N.C.	N.C.
2	GND	GND	+5 VDC	+5 VDC
3	+5 VDC	+5 VDC	GND	GND
4	I/O 14(-)	I/O 16(-)	I/O 14(-)	I/O 16(-)
5	I/O 13(+)	I/O 15(+)	I/O 13(-)	I/O 15(-)
6	I/O 14(+)	I/O 16(+)	I/O 13(+)	I/O 15(+)
7	I/O 17(+)	I/O 18(+)	I/O 14(-)	I/O 16(-)
8	I/O 13(-)	I/O 15(-)	I/O 14(+)	I/O 16(+)
9			I/O 17(-)	I/O 18(-)
10			I/O 17(+)	I/O 18(+)

Table 11.3: High-Speed Differential I/O Expansion Module Pinout by Connector Style and Slot

Installing the High-Speed Differential I/O Module

To install the High-Speed Differential I/O Expansion Module in your MicroLYNX perform the following in accordance with Figure 11.4.

To Install the Module:

1) Remove the two retaining screws (A) from the cover.
2) Remove the blank panel (2 or 3) from the desired slot you want to use.
3) Carefully press the Expansion Module (B) into place by plugging the 28 pin connector into the desired receptacle (D or E) and snapping it into place under the retaining clips (F).
4) Reinstall the MicroLYNX cover.
5) Affix the labels supplied with the Module as shown.

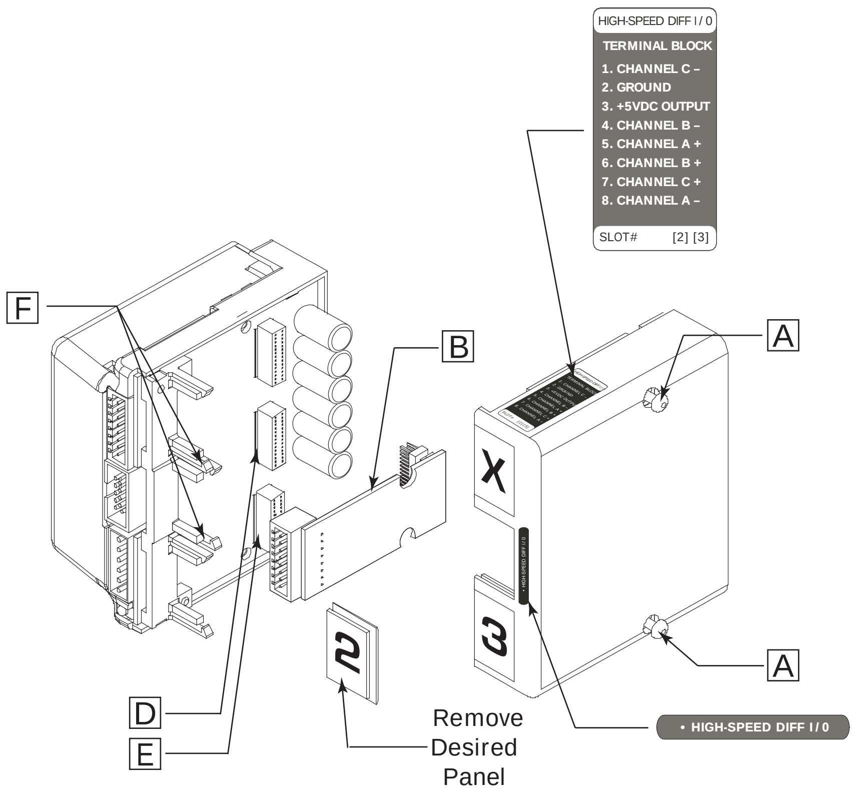


Figure 11.4: Installing the High-Speed Differential I/O Expansion Module

The Four Clocks Explained

The MicroLYNX has four clock pairs that are used by the high-speed I/O. One of these, clock pair 11 and 12, is fixed as an output and is used internally to provide step clock and direction pulses to the driver section of the MicroLYNX. The step clock output increments CTR1 (Counter 1). The user has no physical access to this clock, however, CTR1 may be read from or written to by software instructions in either program or immediate mode. The following table explains the clocks, as well as their default I/O line pair placement.

Clock Types Defined

There are three basic types of clocks that may be configured for the MicroLYNX, they are:

1] Quadrature
2] Step/Direction
3] Up/Down

These clock functions are illustrated in figure 11.5.

Quadrature

The quadrature clock function is the most commonly used input clock function. This is the default setting for each high-speed I/O channel except 11 & 12. This clock function will typically be used for closed loop control (encoder feedback) or for following applications.

Step / Direction

The step/direction clock function would typically be used in an application where a secondary or tertiary clock output is required to sequentially control an additional axis.

NOTE: On clocks configured for Step/Direction, the LOW number of the I/O Line pair will be Direction and the HIGH number of the I/O Line pair will be StepClock.

Up / Down

The up/down clock type would typically be used as an output function where a secondary axis is being driven by a stepper or servo drive with dual-clock direction control circuitry.

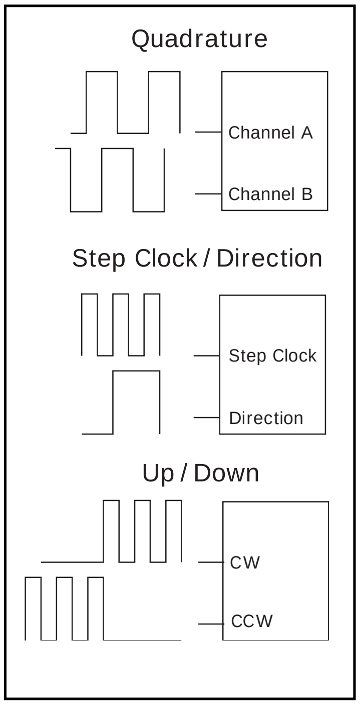


Figure 11.5: Clock Functions

N When using a clock pair as Step and Direction, the lower number or the "A" clock of the pair will always be the Direction. The higher number or the "B" clock of the pair will always be the Step Clock.

The Four Clocks				
Clock #	I/O Line Pair	Slot Position	Counter	Function
1	11 & 12	None	CTR1	This clock is internally generated motion clock. It provides step clock and directional control to the driver section. This clock is not available on any external connector.
2	13 & 14	Slot 2	CTR2	May be configured as an input or output. By default this is configured as a quadrature input. It can be configured as a secondary clock output electronically geared to CLK1.
3	15 & 16	Slot 3	CTR3	May be configured as an input or output. By default this is configured as a quadrature input. It can be configured as a tertiary clock output electronically geared to CLK1.
4	17	Slot 2	None	May be configured as a high speed input or an output. As an output it is a 1MHz reference clock.
	18	Slot 3	None	May be configured as a high speed input or output. As an output it is a 10MHz reference clock.

Table 11.4: The Four Clocks and their Default Line Placement

Configuring the Differential I/O - The IOS Variable

The high-speed differential I/O is configured by means of the IOS variable, and is used in the the same fashion in which the isolated I/O is configured. The main difference is that there are three additional parameters which need to be set in configuring the triggering, clock type and ratio mode settings.

It is important to note that the high-speed differential I/O lines may be used for the same input or output functions as the isolated digital I/O, where the higher speed capabilities of the differential I/O is required. However, for purposes of this example we will only illustrate the clock functions associated with the high-speed differential I/O. Figure 11.6 illustrates the IOS variable settings for the high speed differential I/O.

Configuring the High Speed I/O a Non-Clock Function

Configuring the high speed I/O to clock functions will be covered in depth in the following subsections on configuring encoder and ratio functions. Here we will briefly discuss using the high speed I/O as a general purpose or dedicated I/O function.

Care must be taken when configuring the high speed I/O to a general purpose or dedicated function as the output current sink is 150mA for the entire I/O group 10.

The IOS variable will be configured for the high speed I/O in the same fashion as it is set for the isolated I/O.

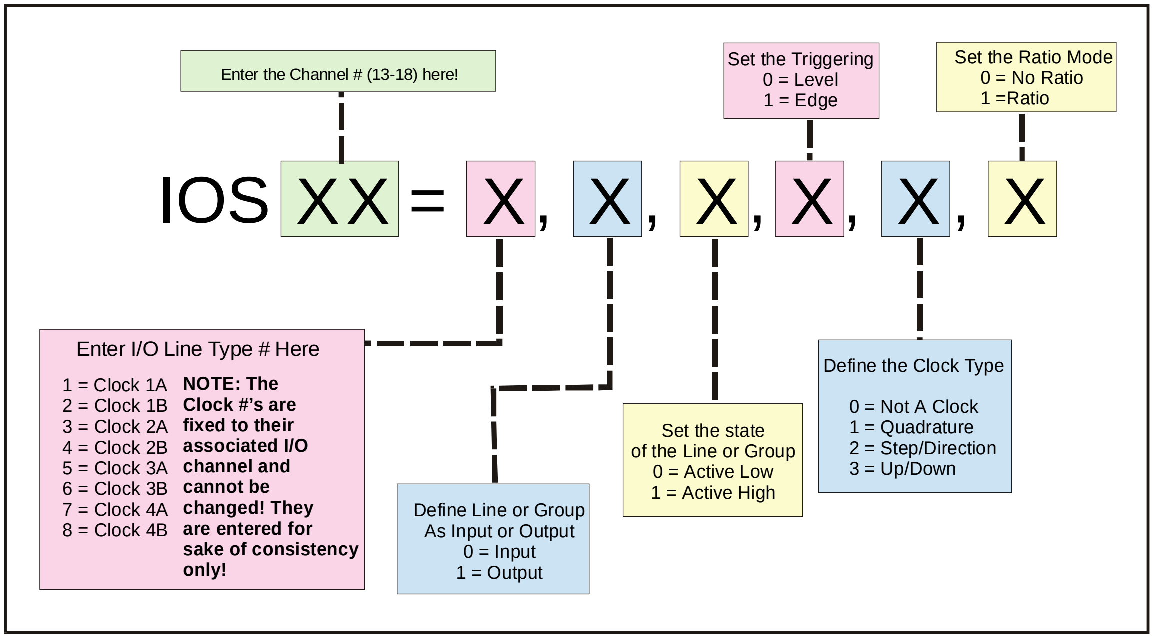


Figure 11.6: IOS Variable Settings for the High-Speed Differential I/O

Configuring an Input

Clocks 2, 3 and 4 can be configured as high speed inputs, or as a general purpose input in the same fashion as the Isolated I/O. In configuring the Differential I/O line as a general purpose input you would typically use the "+" line of the line pair. You cannot use both lines as separate I/O lines. The figure below shows the

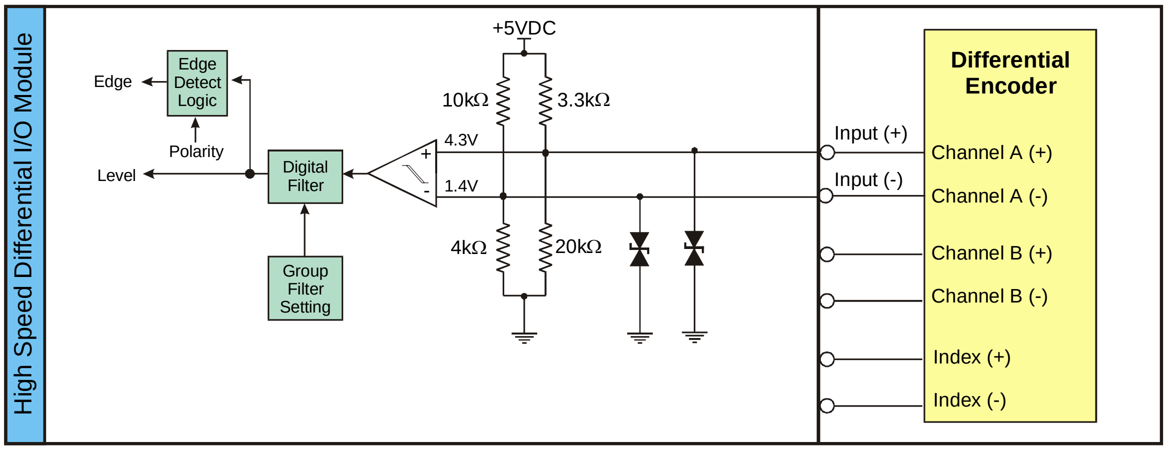


Figure 11.7: Differential I/O Input Equivalent Circuit

Input Equivalent Circuit with the I/O line pair connected to channel A of a differential encoder. This feature is demonstrated in *Typical Functions of the Differential I/O: Connecting and Using an Encoder.* on the following page. Clocks 2, 3 and 4 are set up as Quadrature inputs by default. The defaults for each I/O Line Pair are:

IOS 13 = 3, 0, 1, 0, 1, 0
IOS 14 = 4, 0, 1, 0, 1, 0
IOS 15 = 5, 0, 1, 0, 1, 0
IOS 16 = 6, 0, 1, 0, 1, 0
IOS 17 = 7, 0, 1, 0, 1, 0
IOS 18 = 8, 0, 1, 0, 1, 0

Setting the Digital Input Filtering for the Differential I/O

User-definable digital filtering makes the LYNX well suited for noisy industrial environments. The filter setting is software selectable using the *IOF Variable* with a minimum guaranteed detectable pulse width of 18 microseconds to 2.3 milliseconds. Table 11.5 illustrates the IOF settings.

IOF Filter Settings for the High Speed Differential I/O IOF=<num> (<num> = 0-7)		
Filter Setting	**Cutoff Frequency**	**Minimum Detectable Pulse Width**
0 (default)	5.00 MHz	100 nanoseconds
1	2.50 MHz	200 nanoseconds
2	1.25 MHz	400 nanoseconds
3	625 kHz	800 nanoseconds
4	313 kHz	1.6 microseconds
5	156 kHz	3.2 microseconds
6	78.1 kHz	6.4 microseconds
7	39.1 kHz	12.8 microseconds

Table 11.5: Digital Filter Settings for the Differential I/O

Configuring an Output

The Differential I/O Group 10 has 3 Channels (Line Pairs 13 & 14, 15 & 16, and 17 & 18) that can be configured as an output by the user and 1 Channel (Line Pairs 11 & 12) that is configured as output only. (SCK and DIR on the Control Module.) These outputs can be configured as high speed outputs or 0 to 5VDC general purpose outputs by using the IOS variable. The high speed clock outputs have the following restrictions:

Line Pairs 11/12, 13/14 and 15/16 can be configured to Step Clock/Direction or Up/Down.

Line Pair 17/18 is limited to 1MHz Reference Out (17) and 10MHz Reference Out (18).

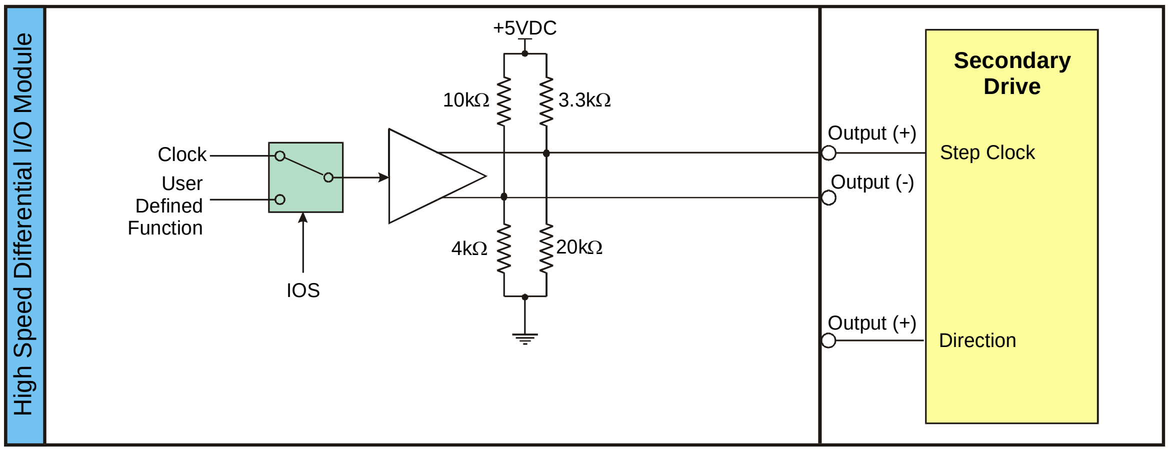


Figure 11.8: Differential I/O Output Equivalent Circuit

In the Equivalent Circuit in Figure 11.8 an Output is being used as Step or Direction on a driver.

For the configuration example, use I/O line 13 for the output. Since by default the line is a quadrature input we must configure it to be a Step/Direction Output by setting the IOS Variable to the following:
IOS 13 = 3, 1, 0, 1, 2, 0
This breaks down as:

IOS 13 - Identifies the line being configured as 13.
3 - Sets the I/O Type to Clock 2A (default).
1 - Sets it as an output.
0 - Sets Logic at Low True.
1 - Edge Triggered.
2 - Sets the Clock Type to Step/Direction.
0 - No Ratio.

Typical Functions of the Differential I/O

Connecting and Using an Encoder

The High-Speed Differential I/O Module may be used for closed loop motion control by receiving quadrature input from a differential or single ended encoder. High-Speed I/O channels 13 and 14 are configured by default for this function, so you would want your expansion module inserted into expansion slot #2.

Connect your encoder as shown in Table 11.6 and Figure 11.9.

Encoder Connections - Expansion Slot #2				
Encoder Signal		MicroLYNX		
Differential	Single	I/O Channel	8 Position Pheonix	10 Pin Header*
Channel A+	Channel A	13+	5	6
Channel A-		13-	8	5
Channel B+	Channel B	14+	6	8
Channel B-		14-	4	4 or 7
Index +	Index	17+	7	10
Index -		17-	1	9
+5 VDC	+5 VDC		3	2
GND	GND		2	3
NOTE: IMS differential encoder follow the Hewlett Packard pin configuration. Thus if your encoder is manufactured by HP or IMS a 10 conductor "straight through wired" ribbon cable with female DIN ribbon cable connectors can be connected directly between the differential encoder and the expansion board (10 Pin Header Version) without wiring modification.				

Table 11.6: Expansion Slot 2 Encoder Connections

Testing Your Encoder Setup

Now that your encoder is connected, it is time to test the setup and verify its operation by typing the following into your terminal:

```
‘set munits to correspond with MSEL=256
MUNIT=51200

‘set the encoder units variable EUNIT to the number = 4 x
‘encoder resolution, ie 500 line encoder x 4 = 2000,
'200 line encoder x 4 = 800 etc.
EUNIT=2000

'Set the stall factor variable to 10% of EUNIT (10% of a
'revolution
STLF=200

'Enable encoder functions
EE=1
POS=0  'set position counter to 0
CTR2=0 'set counter 2 to 0
SAVE   'save the aforementioned settings.
```

Test the encoder setup by entering the following into your terminal:

```
MOVR 10       'the motor moves 10 revolutions (we hope)
PRINT POS     'we read the POS variable, it should say "10.000"
PRINT CTR2    'we read CTR2, it should read 10 X EUNIT, or 20000
```

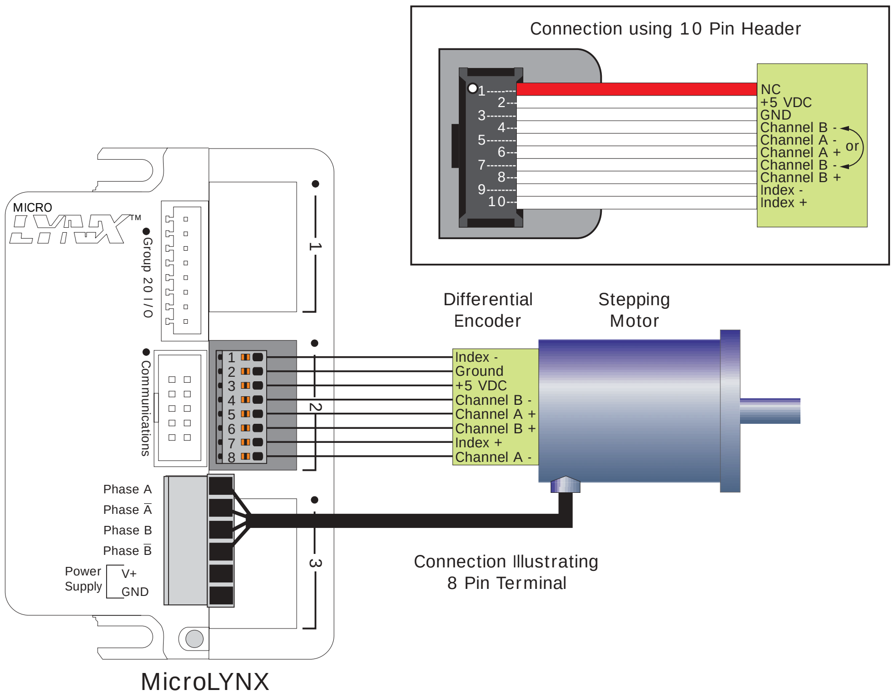


Figure 11.9: Differential Encoder Connection

Introducing The EUNIT (Encoder UNITS) Variable

During open loop operation, the MicroLYNX takes the number of clock pulses registered on CTR1, scales that number using the MUNIT variable and then writes the result to the position variable POS.
For closed loop operation, where the encoder functions are enabled (EE=1), the MicroLYNX takes the number of clock pulses registered on CTR2, scales them by the EUNIT variable and stores them to the POS counter.

The EUNIT variable must be scaled to the same factor as the MUNIT variable. For example, if you were scaling your system to operate in degrees, the MUNIT/EUNIT relationship would be expressed thus:

```
MUNIT=51200/360
EUNIT=2000/360
```

(This assumes MSEL=256 and a 500 line encoder.)
With this configuration, if you performed the following absolute move:

```
MOVA 270
```

the axis would turn 270°. Thus when you enter:

```
PRINT POS
```

the terminal will display “270.00”.
The program that follows will illustrate encoder feedback by making a series of moves while displaying both the raw counts from CTR2 and the scaled POS value.
Enter the program below in the text editor window.

```
'******PARAMETERS*******
MUNIT=51200'motor units = 1/256 resolution
EUNIT=2000 '500 line encoder quad input
EE=1       'enable encoder functions
STLF=200          'stall factor 10% of 1 rev.
STLDE=1           'enable stall detection
STLDM=0           'stop motion if stall is detected
MAC=75            'accel. current to 75%
MRC=50            'run current to 50%
MHC=25            'hold current to 25%


'******PROGRAM********
PGM 200
  CTR2=0
  POS=0
  MOVR 1
  HOLD 2
  DELAY 250
  PRINT "\rEncoder Count= ", CTR2, "  Position Count= ", POS,"\e[K";
  MOVR 10
  HOLD 2
  DELAY 250
  PRINT "\rEncoder Count= ", CTR2, "  Position Count= ", POS,"\e[K";
  MOVR -11
  HOLD 2
  DELAY 250
  PRINT "\rEncoder Count= ", CTR2, "  Position Count= ", POS,"\e[K";
  BR 200
END
PGM
```

Execute the program by entering “EXEC 200” into the terminal.

Following an External Clock (Electronic Gearing)

The High-Speed Differential I/O Module allows you to configure the MicroLYNX’s primary axis to follow an external clock input. The hardware connection *(Figure 11.10)* is almost identical to that shown for closed loop control, only in this instance instead of using a quadrature clock input for position monitoring and maintenance, we will use the encoder input to control the primary axis.

Using this type of application introduces the HAE (Half Axis Enable) flag and the HAS (Half Axis Scaling) variable. In half axis mode the master clock is taken from the CLK2, CLK3 or CLK4 (I/O channels 13 & 14, 15 & 16 or 17 & 18), which have the IOS variable configured as inputs, a clock type, and ratio mode enabled. The primary axis will move as a ratio of this clock based upon the factor entered in the HAS variable.

HAE Half Axis Enable/Disable Flag

This flag (1) enables and (0) disables half axis scaling mode. The default condition is (0) disabled. The HAE flag must be enabled for this mode to function.

HAS Half Axis Scaling Variable

The half axis scaling variable is the factor by which the Follower Input: Primary Axis ratio is scaled. The range of the factor is >-1 to <1. For example, a setting of HAS=.5 will output 1 pulse on the primary axis for every 2 pulses input to the follower input or a 2:1 ratio, HAS=.2 will be 5:1, HAS=.999 will be .999:1 and so on. The default HAS value is 0.000, thus some factor must be entered to make this function.

Configuring the I/O for Half Axis Mode

The parameter setup to make this configuration follows. This assumes a High-Speed Differential I/O Expansion Module installed in slot 2. If your module is installed in slot 3, use I/O channels 15 and 16 (IOS 15=5,0,1,0,1,1 and IOS 16=6,0,1,0,1,1) instead. The raw count of clock pulses will register to CTR3. I/O channels 17 and 18 can be used for this also, only there is no registration of clock pulses:

```
IOS 13=3,0,1,0,1,1   'I/O 13 quad. input, ratio mode
IOS 14=4,0,1,0,1,1   'I/O 14 quad. input, ratio mode
HAE=1                'Enable half-axis scaling mode
HAS=.5               'Half-axis scaling variable to .5 (1 output
                     'pulse on the pri. axis for 2 input pulses)
```

NOTE: In this example a differential encoder is used to illustrate the quadrature input clock pulses.

With this configuration, one (1) step clock pulse will output to the primary axis for every two (2) quadrature input clock pulses. By reading the value of CTR2 and CTR1 you can see the ratio of the pulses.

Try different HAS variable, motor resolution and MUNIT settings to see how the primary axis is effected by different settings.

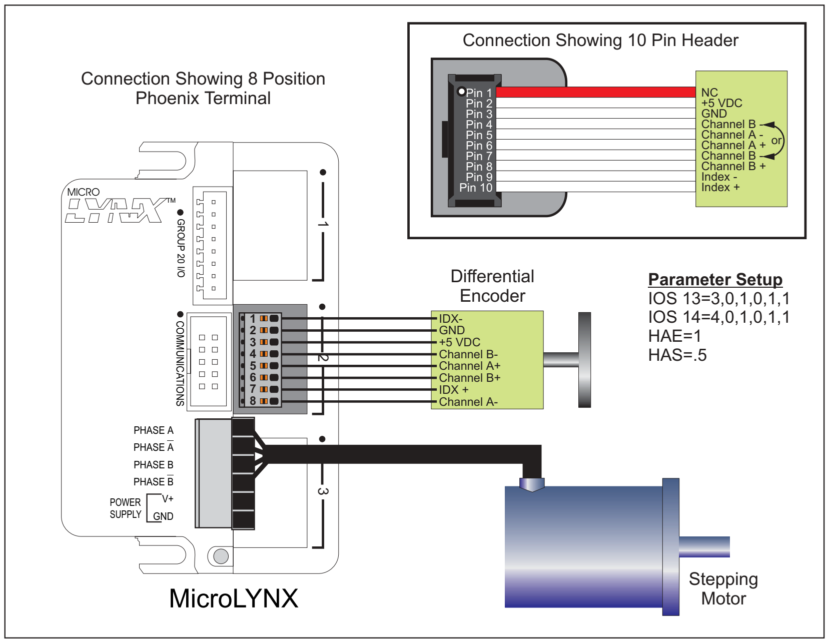


Figure 11.10: Differential I/O Connections for Following an External Quadrature Input

NOTE: The HAS variable must be set to less than 1 or Error Code 9004, "Ratio Out of Range" will occur.

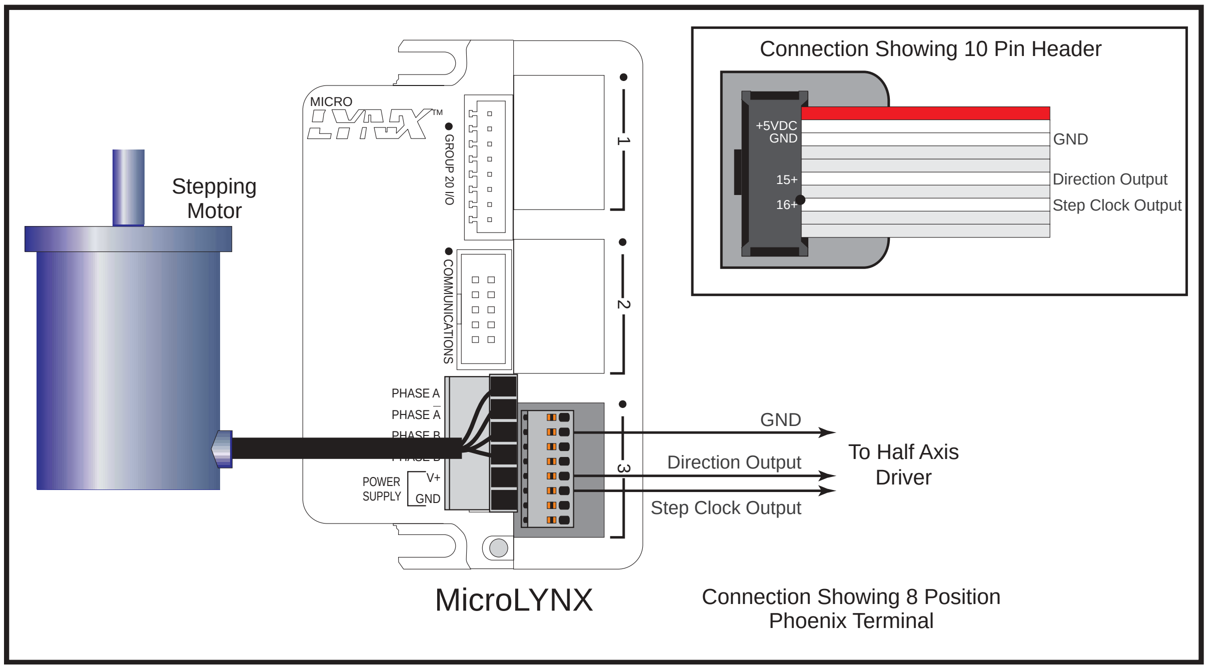


Figure 11.11: One and a Half Axis Operation

One and a Half Axis Operation (RATIOE)

A secondary drive can be connected to a pair of differential outputs. The secondary driver will operate off of the differential output pair 15 and 16 (I/O pair 13 and 14 can also operate in this mode). Setting the ratio mode to TRUE (1) for the differential output clock (IOS) specifies a secondary drive function. Then when ratio mode is enabled (RATIOE); the secondary axis will follow the primary axis with the ratio specified by the RATIO variable.

The sequence of commands used to make this setup function would be as follows:

```
'Set IOS 15 to step/direction clock type, and ratio mode
IOS 15 = 5,0,1,0,2,1
'Set IOS 16 to step/direction clock type, and ratio mode
IOS 16 = 6,0,1,0,2,1
'Set Ratio Mode Enable Flag to TRUE (1)
RATIOE = 1
'Set RATIO variable to .5 for the secondary drive
RATIO = .5
```

With this setup, the motor on the secondary drive will move half the distance of the primary.

N **NOTE:** The HAS variable must be set to less than 1 or Error Code 9004, "Ratio Out of Range" will occur.

Analog Input / Joystick Module

The Analog Input/Joystick Expansion Module adds two 0 to 5 volt analog input channels to the MicroLYNX System. Both channels can be used for data aquisition, or either channel can be used to directly control motion. This offers the user the capability of receiving input from a variety of analog sources such as temperature or pressure sensors, and then controlling events based upon those inputs.

The user-selected Joystick channel can be programmed to set the range, zero, deadband and sensitivity.
Each channel uses a 12 bit D/A converter for better resolution as well as a fixed single pole analog filter with a cutoff frequency of 658 Hz to reduce the electrical noise that can be present in industrial environments.

The Analog Input/Joystick Module can be installed in any free slot, however only one (1) module can be used per MicroLYNX.

Electrical Specifications

Analog Input Voltage Range	0 to +5 Volts
Resolution	12 Bits
Offset	±2 LSB
Integral Linearity Error	±2 LSB
Differential Linearity Error	±3/4 LSB
Absolute Maximum Voltage at Inputs	±24 Volts
Joystick Reference Voltage	+5 Volts
Precision Calibration Reference Voltage (±0.2%)	+4.096 Volts
Calibration Reference Voltage Tolerance	±2 %
Analog Input Filter Cutoff Frequency	658 Hz

	Connector Option	
Pin #	**8 Position Phoenix**	**10 Pin Header**
1	+5V (Joystick Reference)	+5V (Joystick Reference)
2	AIN 1	GND
3	GND	AIN 1
4	+5V (Joystick Reference)	+5V (Joystick Reference)
5	AIN 2	GND
6	GND	AIN 2
7	4.096V (Calib. Reference)	4.096V (Calib. Reference)
8	GND	GND
9		GND
10		N.C.

Table 11.7: Analog Input/Joystick Module Pin Configuration

Installing the Analog Input/Joystick Module

To install the Analog Input/Joystick Expansion Module in your MicroLYNX, perform the following in accordance with Figure 11.12.

1) Remove the two retaining screws (A) from the cover.
2) Remove the blank panel (1, 2, or 3) from the desired slot you want to use.
3) Carefully press the Expansion Module (B) into place by plugging the 28 pin connector into the desired receptacle (C, D, or E) and snapping it into place under the retaining clips (F).
4) Reinstall the MicroLYNX cover.
5) Affix the labels supplied with the Module as shown.

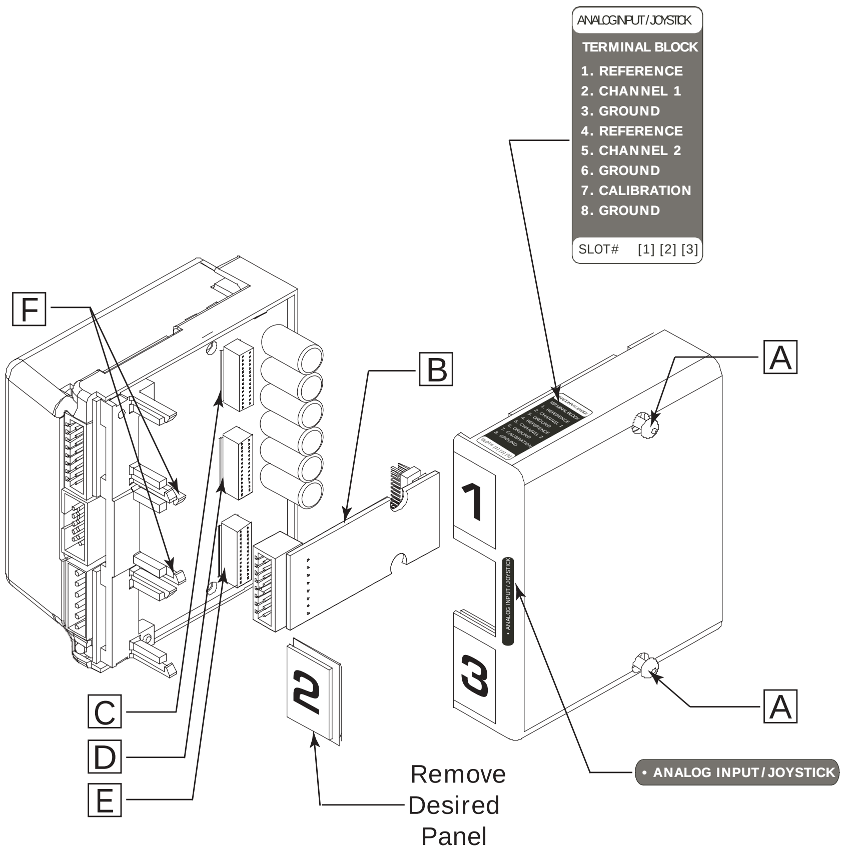


Figure 11.12: Installing the Analog Input/Joystick Module

Instructions & Variables Specific to the Analog Input/Joystick Module

There are several new enhancements to the MicroLYNX instruction set which add the functions of the Analog Input/Joystick Interface Module while maintaining backward compatibility with the modular LYNX System. The following instructions and variables are specific to the Analog Input/Joystick Interface Module. These are introduced here and covered in more detail in Part III, Software Reference.

Instruction	Usage	Description
IJSC	**IJSC**	**CALIBRATE JOYSTICK INSTRUCTION:** Supports the Analog / Joystick Interface Module when operating in joystick mode. Execution of this command followed by moving the connected joystick over its range of motion and back to center, and then pressing the "ENTER" key, or letting it time out for 30 seconds calibrates the joystick. This instruction allows for rapid calibration of the joystick.
Variable	**Usage**	**Description**
ADS	**ADS<chan>=<aunit>,<func>,<law>**	**ANALOG INPUT SETUP VARIABLE:** Supports the Analog Input / Joystick Module. **<chan>** = channel # (1 or 2) **<aunit>** = converts analog units to motor steps (velocity = aunit x munit) **<mode>** = 1 - analog, 2 - joystick **<law>** = Adjusts joystick position to motor velocity transformation. 1 = linear, 2 = square law, 3 = cube law
AIN	**<var>=AIN<chan>**	**READ ANALOG INPUT CHANNEL:** Causes a read of analog input channel <chan>, data is saved as variable <var>.
JSC	**JSC=<num>**	**JOYSTICK CENTER POSITION VARIABLE,** updated by IJSC command or directly as shown.
JSDB	**JSDB=<num>**	**JOYSTICK DEADBAND VARIABLE,** updated by ISJC command or directly as shown.
JSFS	**JSFS=<num>**	**JOYSTICK FULL SCALE VARIABLE,** updated by ISJC command or directly as shown.
Flag	**Usage**	**Description**
JSE	**JSE=<flg>**	**JOYSTICK ENABLE FLAG:** <flg> = 1 enables joystick function, <flg> = 0 (default) disables.

Table 11.8: Analog Input/Joystick Module Software Command Summary

Error Codes

In addition to the instructions and variables, the following error codes have been added to support the inclusion of the Analog Module and aid in troubleshooting the MicroLYNX System:

1201	Selected Analog Board not installed.
1202	Analog channel number not available.
1204	Analog option not installed.
1205	Analog VALUE out of range, possibly defective Board.
2101	Analog RANGE not allowed.
2102	Analog destination/source not allowed.
2103	Analog Destination/Source already used.
2104	Invalid Analog Channel number.
2105	Analog LAW not allowed.
2106	Can't enable Joystick while in motion, or can't exec motion cmd with Joystick enabled.
9014	Analog input not allowed for data.

The ADS Variable (A to D Setup)

The ADS variable is the heart of the MicroLYNX Analog Input/Joystick Interface Module. There are three parameters that control how the module will respond to input. It is used as follows:

```
ADS <chan>=<aunit>,<mode>,<law>
```

<chan>: Is the analog input channel that will be used, either 1 or 2.
<aunit>: This parameter sets the relationship between the analog input and units that are convenient to the user. In analog (User) mode the aunits parameter is the number of user units corresponding to the Analog Module full scale. In Joystick (Velocity) mode the aunits parameter is the number of munits/second corresponding to the Joystick Full Scale (JSFS) parameter.

<mode>: The mode parameter controls whether or not the input is used for velocity control; 1 = analog input, 2 = velocity or joystick mode.

<law>: Controls the sensitivity of the velocity with respect to the analog input. The effect of the analog input can be linear, square or cube. <law> applies to velocity mode only.

Here are two examples that illustrate the ADS variable:

Example 1

A pressure transducer is connected to input 1. The transducer output is 10 psi/volt. Vref represents the voltage at the Input to the Analog Joystick Module corresponding to full scale. Vref as measured at pin 1 on the Analog Joystick Module is 5.05 volts. Thus aunits for channel 1 is 10 psi/volt x 5.05 volts or 50.5. The value returned by an analog read of Channel 1 will be in psi. Note that the full scale output of the transducer does not have to equal the Analog Module full scale. This setup would be expressed thus:
ADS 1=50.5, 1

Example 2

A 1.8 degree (per full step) motor connected to a lead screw with a lead of .1 inches/rev. The step motor drive is set for 32 usteps per full step. A joystick is connected to channel 1. To program speed and motion in inches set munits to (32 pulses/1.8 degrees) x (360 degrees/1 rev) x (1 rev/.1 inches). If a maximum speed of 3 inches/second is desired while in Joystick operation set aunits for channel 1 to 3. For linear Joystick operation the setup command is ADS 1 = 3,2,1

Typical Functions of the Analog Input / Joystick Module

There are three program examples that will illustrate the use of the Analog Input/Joystick Module. In each case a 1kΩ potentiometer is used to emulate a sensor for analog input mode and a joystick for velocity mode.

Use the connection configuration shown in figure 11.13 below, a joystick or a sensor would be connected the same way.

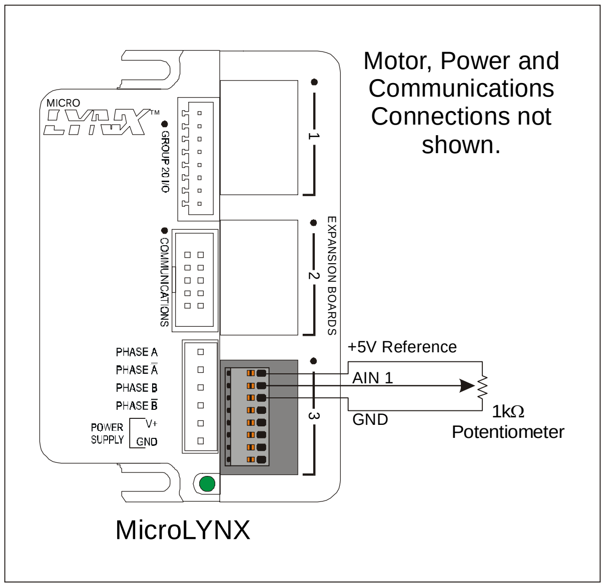


Figure 11.13: Analog Input Module Exercise Connection

Exercise 1: Velocity (Joystick) Mode

Here the potentiometer is emulating a joystick. Enter and execute the following program. When the voltage on AIN 1 is roughly 100mV either side of 2.5 volts it will be in the deadband range of the joystick. When less than 2.4 volts, the axis will accelerate in the minus (-) direction. When more than 2.6 volts, it will accelerate in the positive (+) direction. The velocity will increase as the voltage decreases from 2.4 to 0, or increases from 2.6 to 5.0. This can be watched with a multimeter. In this exercise both the axis velocity and position will display to the terminal screen.

```
'****Parameters****
MSEL=256
MRC=100
MAC=100
MUNIT=51200
JSDB=100         ‘Joystick deadband =100 aunits
VM =10000        ‘max velocity 10,000 munits/sec
ADS 1=1000,2,1   ‘chan. 1,aunits=1000, joystick, linear law
JSE = 1          ‘enable joystick functions

'****Program****

PGM 1
PRINT "\e[2J"
  LBL RUN
  PRINT "\e[1;1HInput Channel =    " , AIN
  PRINT "Axis Velocity =  " , VEL
  PRINT "Axis Position = " , POS
  BR RUN
END
PGM
```

Exercise 2: Sensor Input I

Here we pretend the potentiometer is a pressure transducer and use it to display a pressure value to the screen.

```
ADS 1=50.5,1             ‘set ADS to aunit=50.5,analog input mode

PGM 200
LBL PRNTPSI              ‘name program “PRNTPSI”
PRINT "\e[2J"            ‘ansi esc. sequence to clear display
PRINT "Pressure = ", AIN 1 , " PSI"
BR PRNTPSI       ‘loop to program beginning
END
PGM
```

Exercise 3: Sensor Input II

Once again our potentiometer is pretending to be a sensor. In this exercise the program will call up a subroutine based upon the voltage seen on AIN 1 and position the axis at an absolute position. The best analog to this example might be a flow control application.

```
'****Parameters****
MUNIT=51200               ‘munits=51200
MAC=75                    ‘acceleration current to 75%
MRC=50                    ‘run current to 50%
ADS 1=5,1                 ‘aunits 5, analog input mode
VAR LIMIT=0               ‘declare user var “LIMIT”

'****Program****
PGM 200
LBL AINTST                ‘name program “AINTST”
LIMIT = AIN 1             ‘set user var “LIMIT” = AIN 1
CALL ATEST, LIMIT>3.5     ‘call ATEST if LIMIT is greater than 3.5 aunits
CALL BTEST, LIMIT<3.5     ‘call BTEST if LIMIT is less than 3.5 aunits
BR 200                    ‘loop to beginning of program
END

'****Subroutines****
LBL ATEST                 ‘declare subroutine “ATEST”
VM=20                     ‘max. velocity = 20 munits/sec.
MOVA 10                   ‘index to abs. pos. 10
HOLD 2                    ‘suspend prog. until motion completes
RET                       ‘return from subroutine

LBL BTEST                 ‘declare subroutine “BTEST”
VM=5                      ‘max velocity = 5 munits/sec.
MOVA 22                   ‘index to abs. pos. 22
HOLD 2                    ‘suspend prog. until motion completes
RET                       ‘return from subroutine
```

Isolated Communications Module

The Isolated Communication Expansion Module adds either RS-232 or RS-485 to the CAN based MicroLynx motion control system. This second communication port is fully independent and optically isolated from I/O and input power ground.

This second port could be used to communicate to an operator interface or for system diagnostics while the system is in use.

Electrical Specifications

RS-232 Receiver	
Input Voltage Range	±30 Volts
Input Threshold Low	0.4 Volts
Input Threshold High	2.4 Volts
Input Resistance	3 to 7 k Ω
RS-232 Transmitter	
Output Voltage Swing	±5 Volts
Output Resistance	300 Ω
Output Short Circuit Duration	Continuous
RS-485 Receiver	
Input Voltage Range	-8 to +12.5 Volts
Input Differential Threshold	±200 Millivolts
Input Resistance	96 k Ω
Driver	
Differential Output (R=50 W)	2 Volts
Output Voltage Range	-8 to +12.5 Volts

RS-232 Expansion Module		
	Connector Option	
Pin #	8 Position Phoenix	10 Pin Header
1	CGND	N.C.
2	RS-232 RX	RS-232 TX
3	RS-232 TX	RS-232 RX
4	N.C	N.C.
5	N.C.	CGND
6	N.C.	N.C.
7	N.C.	N.C.
8	N.C.	N.C.
9		N.C.
10		N.C.

Table 11.9: RS-232 Pinout

RS-485 Expansion Module		
	Connector Option	
Pin #	8 Position Phoenix	10 Pin Header
1	N.C.	N.C.
2	N.C.	N.C.
3	N.C.	N.C.
4	RS-485 RX-	N.C.
5	RS-485 RX+	CGND
6	RS-485 TX-	RS-485 RX+
7	CGND	RS-485 RX-
8	RS-485 TX+	RS-485 TX-
9		RS-485 TX+
10		CGND

Table 11.10: RS-485 Pinout

The RS-232 Communications Module

The RS-232 Communications Expansion Module, which allows for use of the RS-232 interface , can only be used with the CAN bus version of the MicroLYNX. This expansion board may only be used in slot 1of the MicroLYNX and is automatically recognized; no configuration is needed.

This expansion board uses MicroLYNX COMM 2 and can be used to simultaneosly communicate with the MicroLYNX via RS-232, while communicating via the CAN bus. This is useful in requesting and displaying system status information from and to a PC or terminal.

NOTE! Since the RS-232 Expansion Module uses MicroLYNX COMM 2, it cannot be used in conjunction with the RS-485 Expansion Module. Only one of the two interfaces can be used with the CAN bus version of the MicroLYNX.

Table 11.9 and Figure 11.14 illustrate the pin configuration and connection of the RS-232 Expansion Module.

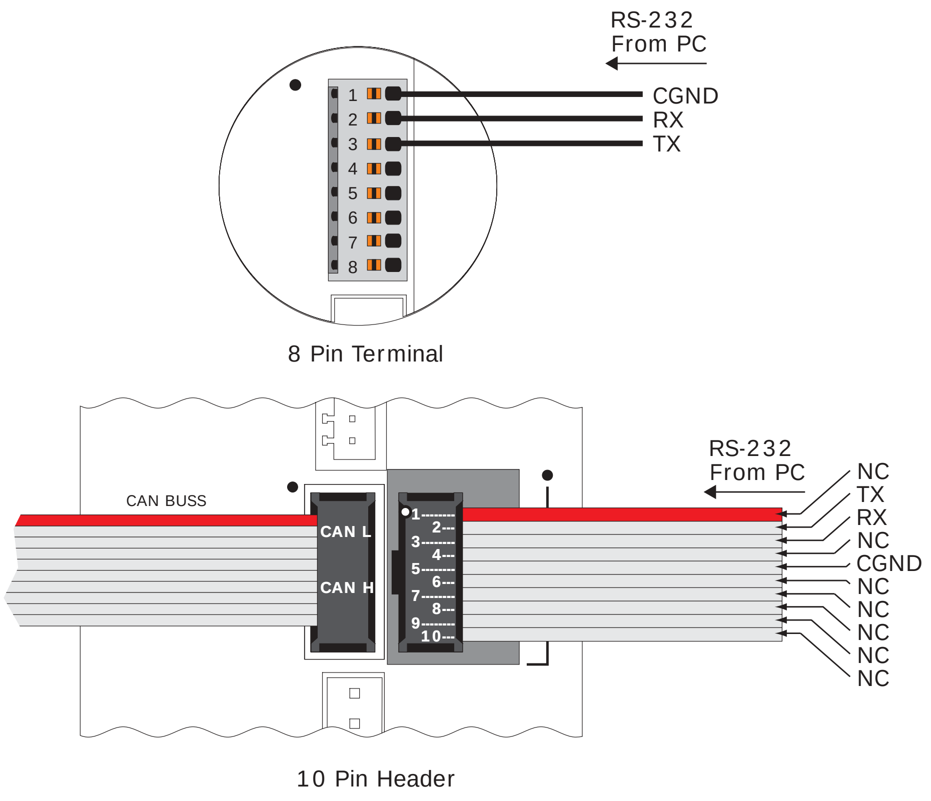


Figure 11.14: Connecting the RS-232 Expansion Module

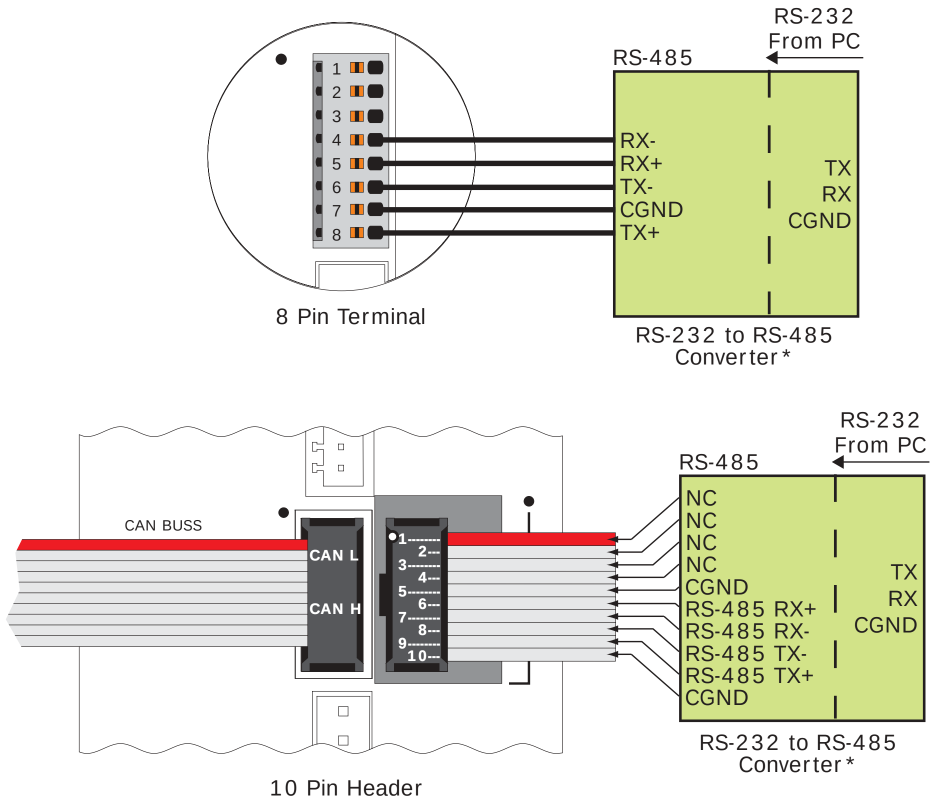
The RS-485 Communications Module

The RS-485 Communications Expansion Module allows for use of the RS-485 interface on the CAN bus version of the MicroLYNX only. This expansion module may only be used in slot 1 of the MicroLYNX and is automatically recognized; no configuration is needed.

This expansion board uses MicroLYNX COMM 2 and can be used to simultaneosly communicate with the MicroLYNX via RS-485, while communicating via the CAN bus. This is useful in requesting and displaying system status information from and to a PC, terminal or machine interface such as the IMS HMI. It also allows for the use of multiple MicroLYNX Systems in a PARTY configuration without using additional CAN bus node positions, if the CAN interface is not required in all the MicroLYNX nodes.

NOTE! Since the RS-485 Expansion Module uses MicroLYNX COMM 2, it cannot be used in conjunction with the RS-232 Expansion Module. Only one of the two interfaces can be used with the CAN bus version of the MicroLYNX.

Table 11.10 and Figure 11.15 illustrate the pin configuration and connection of the RS-485 Expansion Module. For multi-drop connection information see *Section 7: The Communications Interface*.



* *The recommended RS-232 to RS-485 Converter is IMS Part Number CV-3222. If your PC is equipped with an RS-485 board, no converter is necessary. Connect the RS-485 lines from the host PC directly to the MicroLYNX.*

Figure 11.15: Connecting the RS-485 Expansion Module

Installing the Isolated Communications Module

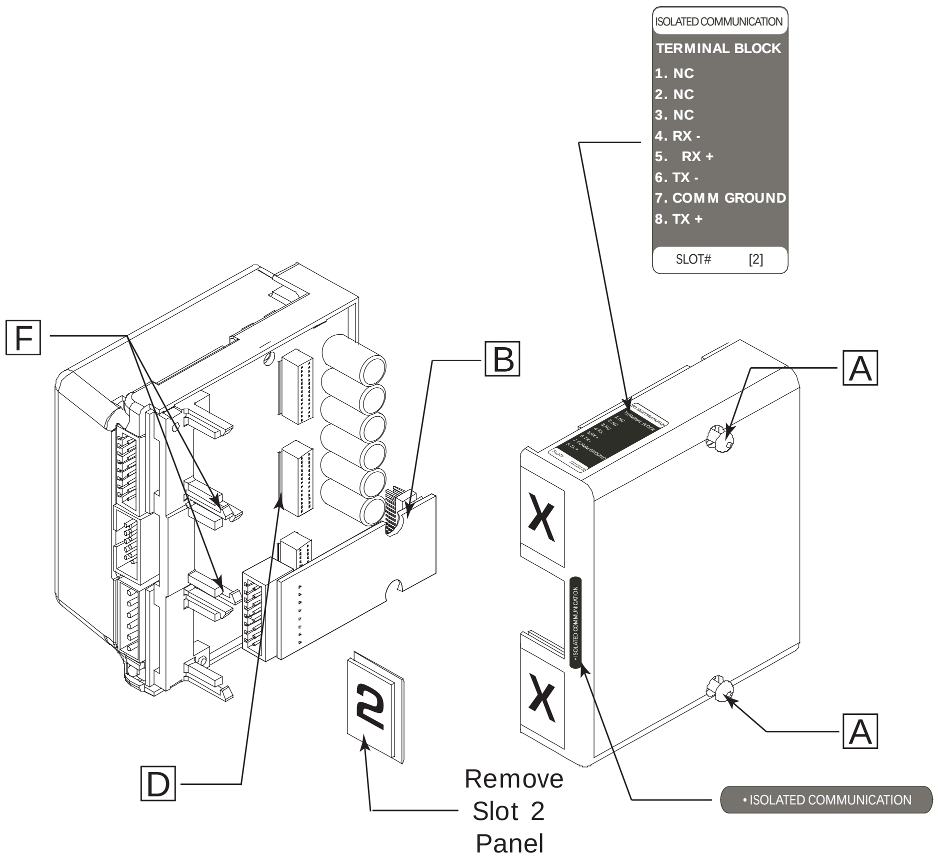


Figure 11.16: Installing the Isolated Communications Module

To Install the Module:

1) Remove the two retaining screws (A) from the cover.
2) Remove the blank panel from the #2 slot
3) Carefully press the Expansion Module (B) into place by plugging the 28 pin connector into the #2 receptacle and snapping it into place under the retaining clips (F).
4) Reinstall the MicroLYNX cover.
5) Affix the labels supplied with the Module as shown.

Analog Output Module

The Analog Output Expansion Module gives the MicroLYNX the ability to control drives that require an analog control signal such as variable frequency drives, servos and brush-type DC motor drives. It adds two 0 to +5 volt output channels to the functionality of the MicroLYNX. Each channel features 12 bit resolution and may be programmed to one of three operational modes: voltage, velocity or position. Each of these modes may be set to a 2.5V centerpoint for plus or minus operation.

In many cases, this module may save the expense and complexity of using a PLC by enabling MicroLYNX to handle applications such as Indexing Conveyor, Material Handling and Feed Rate Override.

Electrical Specifications

Analog Output Voltage	0 to +5 Volts @ 50mA Max.
Resolution	12 bits
Offset	±3 LSB
Integral Linearity Error	±2 LSB
Differential Linearity Error	±3/4 LSB

Pin #	Connector Option	
	8 Pin Terminal Block	10 Pin Header
1	Ground	Ground
2	CH1 (slots 1-2), CH3 (slot 3)	Ground
3	Ground	CH1 (slots 1-2), CH3 (slot 3)
4	Ground	Ground
5	CH2 (slots 1-2), CH4 (slot 3)	Ground
6	Ground	CH2 (slots 1-2), CH4 (slot 3)
7	NC	NC
8	NC	Ground
9		NC
10		NC

Table 11.11: Analog Output MOdule Pinout

Installing the Analog Output Module

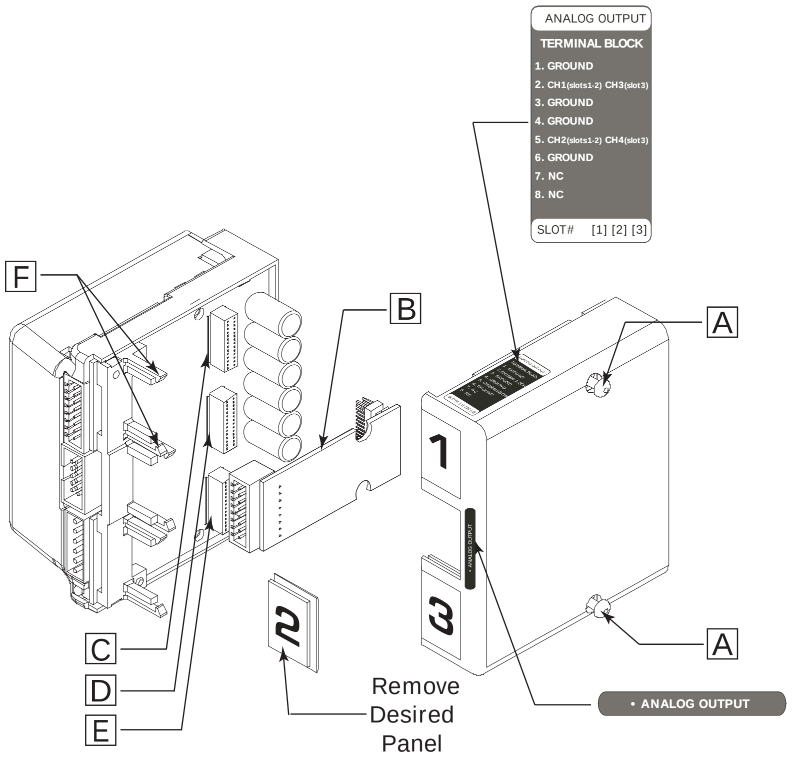


Figure 11.17: Installing the Analog Output Module

To Install the Module:

1) Remove the two retaining screws (A) from the cover.
2) Remove the blank panel (1, 2, or 3) from the desired slot you want to use.
3) Carefully press the Expansion Module (B) into place by plugging the 28 pin connector into the desired receptacle (C, D, or E) and snapping it into place under the retaining clips (F).
4) Reinstall the MicroLYNX cover.
5) Affix the labels supplied with the Module as shown.

Analog Output Commands

The Analog output module adds 3 new commands to the MicroLynx code and is available in Firmware 1.528 or higher.

The commands are AOUT, VAE and DAS.

MNEMONIC	Description	OPCODE
AOUT <chan> = <value>*AUNIT If DAS = 1 or 2	Analog OUT	7Ch (124)
AOUT <chan> = Velocity*AUNIT*MUNIT AOUT <chan> = Position*AUNIT*MUNIT Converts using this formula if DAS = 3,4,5 or 6	Analog OUT	7Ch (124)

Outputs scaled <value> to the Digital to Analog Board. (Converts scaled <value> from D to A) DAS = 1 or 2
or
Converts using this formula if DAS = 3: Velocity*AUNIT*MUNIT = 0 to 4095 counts
Converts using this formula if DAS = 4: Velocity*AUNIT*MUNIT = -2048 to +2047 counts
Converts using this formula if DAS = 5: Position*AUNIT*MUNIT = 0 to 4095 counts
Converts using this formula if DAS = 6: Position*AUNIT*MUNIT = -2048 to +2047 counts

MNEMONIC	Description	OPCODE
VAE = 0 or 1 0 for DAS 1 & 2 1 for DAS 3, 4, 5 or 6	Velocity or position to Analog Enable	C2h (194)

Enables Velocity or Position to be sent to the digital to analog channel that has a DAS type of 3 to 6.

MNEMONIC	Description	OPCODE
DAS chan = AUNIT,type	Digital to Analog Output Setup	B4h (180)

type = 1 to 6 (all types produce 0 to 5 volts)
1 = Volts, absolute <value>.
2 = Volts, plus or minus centered around 2.5 volts
3 = Velocity, absolute Velocity.
4 = Velocity, plus or minus centered around 2.5 volts
5 = Position, absolute Position.
6 = Position, plus or minus centered around 2.5 volts

For setting up Digital to Analog channels:
chan = 1,2 when in slot 2
chan = 3,4 when in slot 3
value = 0 to 4095 when AUNIT = 1
value = 32 bit IEEE floating point when AUNIT not 0 or not 1
AUNIT = 1 or 32 bit IEEE floating point. NOT ZERO
4095 / USER_UNITS; Absolute types (DAS=1, 3, 5)
4095 / (USER_UNITS * 2); plus or minus types (DAS = 2, 4, 6)

Absolute Type Examples (For 0 to 5 Volt Output)

Result: With DAS=1 and AUNIT = 4095/5 and VAE = 0.
AOUT <chan> = <value>
AOUT <chan> = 1 volt when <value> = 1; AOUT = 2.75 volts when <value> = 2.75.

Result: With DAS = 3 and AUNIT = 4095/5*51200 and MSEL=256 and MUNIT = 51200 and VAE = 1.
AOUT <chan> = 1 volt when Velocity = 1; AOUT = 2.75 volts when Velocity = 2.75.

Result: With DAS = 5 and AUNIT = 4095/5*51200 and MSEL=256 and MUNIT = 51200 and VAE = 1.
AOUT <chan> = 1 volt when Position = 1; AOUT = 2.75 volts when Position = 2.75.

Plus or Minus Type Examples

Set an analog value (as if controlling a speed of 0 to 10000 steps/sec in a separate drive) and control direction:
AOUT <chan> = <value>
Result: With DAS = 2 and AUNIT = 4095/20000 and VAE = 0.
AOUT <chan> = 2.5 volts when <value> = 0;
AOUT <chan> = 5 volts when <value> = +10000;
AOUT <chan> = 0 volts when <value> = -10000.

Result: With DAS = 4 and AUNIT = 4095/20000 and MSEL = 2 and MUNIT = 1 and VAE = 1.
AOUT <chan> = 2.5 volt when Velocity = 0;
AOUT <chan> = 5 volts when Velocity = 10000;
AOUT <chan> = 0 volts when Velocity = -10000

Result: With DAS = 6 and AUNIT = 4095/20000 and MSEL = 2 and MUNIT = 1 and VAE = 1.
AOUT <chan> = 2.5 volt when Position = 0;
AOUT <chan> = 5 volts when Position = 10000
AOUT <chan> = 0 volts when Position = -10000

12 Channel Isolated Digital I/O Module

The 12 Channel Isolated Digital I/O Expansion Module adds an additional twelve +5 to 24 VDC isolated I/O channels. All of the I/O channels can be individually programmed as either inputs or outputs, or as dedicated (limit, home, etc.) or general purpose.

When used as inputs, these I/O channels have seven programmable digital filter settings ranging from 215 Hz to 27.5 kHz. As outputs, each channel can sink up to 350 mA. The I/O is isolated from the power supply ground.

A 7.5kOhm switch selectable pull-up resistor is provided for each I/O channel. The twelve I/O channels may be pulled up to either the internal +5 VDC supply or an external voltage provided by the user. Protection circuitry includes over temperature, short circuit and inductive current clamp.

Electrical Specifications

Input Voltage Range	0 to +24 Volts
Input Low Level	< 1.5 Volts
Input High Level	> 3.5 Volts
Open Circuit Input Voltage	
Pull-up Switch ON	4.5 Volts
Pull-up Switch OFF	0 Volts
Load Supply Voltage	28 VDC Maximum (Transient protected at 60 volts)
FET On Resistance	2W Maximum (Tj=125°C)
Continuous Sink Current	350 mA max each output (Ta = 25°C)
Maximum Group Sink	15.A (Thermally Limited)
Filter Cutoff Frequencies	27.5, 13.7, 6.89, 3.44, 1.72 kHz, 860, 430, 215 Hz

	16 Pin Connectors Samtec (S) / Hirose (H)				
Pin #s		Function	Pin #s		Function
1 (S)	15 (H)	V Pull-Up A	9 (S)	7 (H)	V Pull-Up B
2 (S)	16 (H)	I / O Channel 1A	10 (S)	8 (H)	I / O Channel 1B
3 (S)	13 (H)	I / O Channel 2A	11 (S)	5 (H)	I / O Channel 2B
4 (S)	14 (H)	I / O Channel 3A	12 (S)	6 (H)	I / O Channel 3B
5 (S)	11 (H)	I / O Channel 4A	13 (S)	3 (H)	I / O Channel 4B
6 (S)	12 (H)	I / O Channel 5A	14 (S)	4 (H)	I / O Channel 5B
7 (S)	9 (H)	I / O Channel 6A	15 (S)	1 (H)	I / O Channel 6B
8 (S)	10 (H)	I / O Ground A	16 (S)	2 (H)	I / O Ground B

Table 11.12: 12 Channel Isolated I/O Module Pinout

I/O Configuration

Inputs and Outputs as well as digital filtering are configured in the same manner as the Standard I/O (Group 20). Please refer to Section 10 “Configuring the Isolated Digital I/O” for details.

Installing the 12 Channel I/O Module

To Install the Module:

1) Remove the two retaining screws (A) from the cover.
2) Remove the blank panel (1 or 2) from the desired slot you want to use.
3) Carefully press the Expansion Module (B) into place by plugging the 28 pin connector into the desired receptacle (C or D) and snapping it into place under the retaining clips (F).
4) Reinstall the MicroLYNX cover.
5) Affix the labels supplied with the Module as shown.

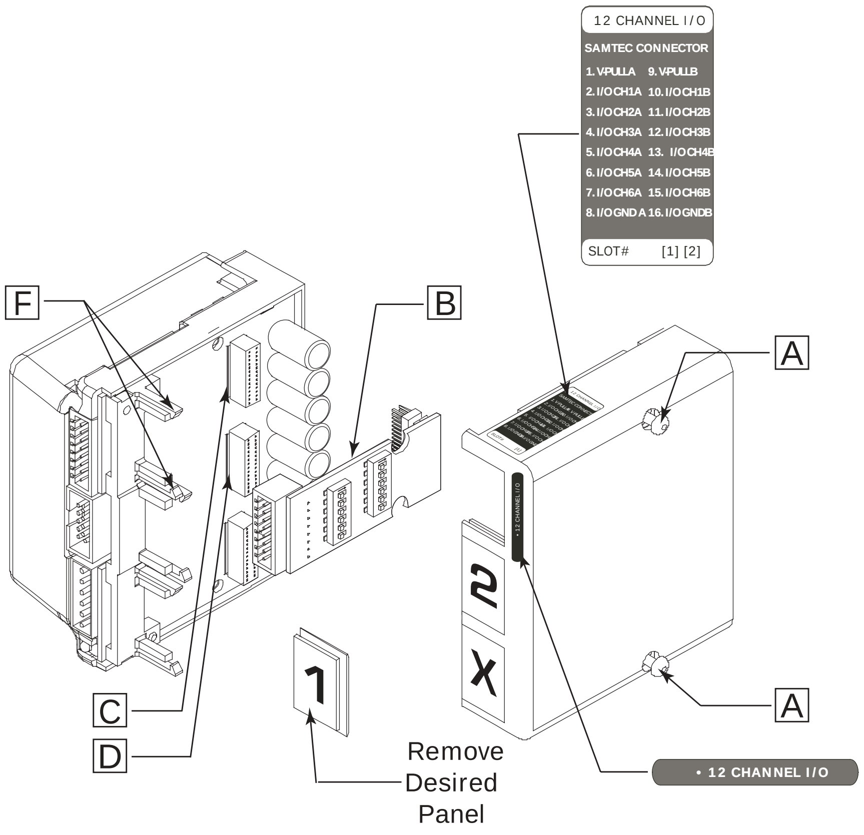


Figure 11.18: Installing the 12 Channel Isolated I/O Module

Pull-up Switches

The Isolated Digital I/O Module is equipped with Pull-up switches which are located on the bottom of the Module. The switches operate in the same manner as the standard Isolated I/O. See Section 10 “Configuring the Isolated Digital I/O” for details.

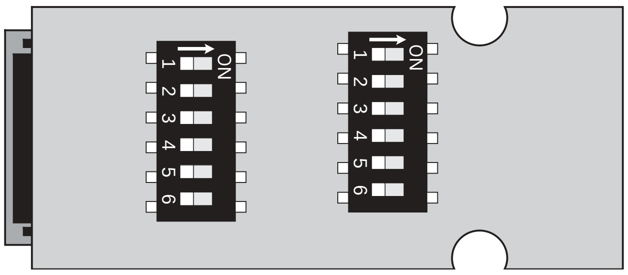


Figure 11.19: 12 Channel I/O Module Pull-up Switches

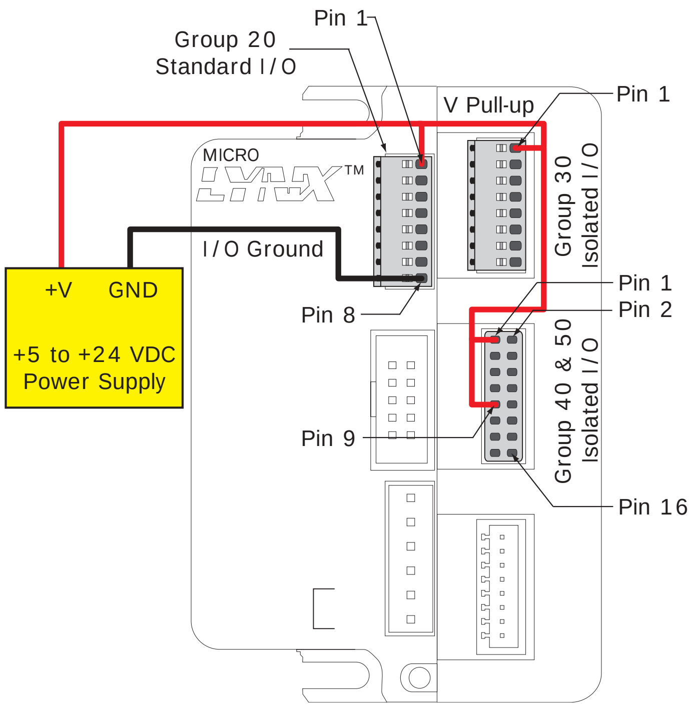


Figure 11.20: Powering Multiple Isolated Digital I/O Modules

NOTE: The Samtec 12 Pin Connector is used in the illustration above. With the Hirose Pin and Receptacle, the physical position of the wires is identical but the Pin numbers are different.

In the illustration above, the Standard Isolated I/O, One Isolated I/O Module, and one 12 Channel I/O Module are shown.

The I/O ground is common internally. Only one ground connection is necessary.

The V Pull-up is NOT common between the modules. This allows the user to power each I/O Group separately if desired.