# ConveyLinx-IO User's Guide

**Version 1.1**

**October 2017**

**Publication ERSC-1008**

*ConveyLinx module firmware and functionality is protected by U.S. and international patents.*

*For complete patent information visit www.pulseroller.com/patents*

## GLOSSARY OF TERMS

| | |
|---|---|
| **ConveyLinx** | Conveyor controls architecture based upon modular distributed devices connected via Ethernet network. |
| **ConveyLinx-Ai / ConveyLinx-Ai2** | Conveyor control module that is part of the **ConveyLinx** family. Each module can accommodate up to 2 **MDR** conveyor **zones**. The modules allow connection for **Senergy-Ai** platform motor rollers and gear drives. |
| **DHCP** | **D**ynamic **H**ost **C**onfiguration **P**rotocol A protocol for assigning **IP** addresses to devices on a network from a pool of available **IP's**. A dynamic **IP** address changes each time the device connects to the network |
| **M8** | This is the type of a particular connector , which has four connector pins and is used on the **ConveyLinx Ai2** modules for both sensor connectors and MDR connectors |
| **LED** | **L**ight **E**mitting **D**iode – In the context of this document, **LED's** are used on the **ConveyLinx-Ai2** to provide visual indication of module status |
| **Light / Dark Energized** | Term used to describe how the signaling output circuit of a **photo-sensor** is configured when it detects its reflected light. A **photo-sensor** that is **light energized** will activate its output circuit when it detects its reflected light. A **dark energized** photo-sensor will activate its output circuit when it does not detect its reflected light. |
| **Normally Open / Normally Closed** | Control logic terminology to define the state of the output of a Boolean "on" or "off" device. The term specifically describes the state of the output circuit when the device's sensing circuit is un-energized. In the context of **photo-sensors**; a **normally open** wired sensor would have its output circuit energized when it detected its reflected light and its output circuit would be de-energized when it did not detect its reflected light. Conversely a **photo-sensor** wired **normally closed** would energize its output circuit when it did not see its reflected light and it would de-energize its output circuit when it did detect its reflected light. |
| **NPN / PNP** | Electronics term that indicates the type of transistor circuit used for a logical input or output for controllers. **NPN** devices will provide a common or ground connection when activated and a **PNP** device will provide a logic voltage connection when activated. |
| **Photo-sensor** | A device, mounted near the end of the conveyor **zone** to sense the presence of a **load** on the zone |
| **PLC** | **P**rogrammable **L**ogic **C**ontroller – A wide variety of industrial computing devices that control automatic equipment |
| **Retro-reflective / Reflex** | Term used to describe the two basic types of **photo-sensors**. **Retro-reflective photo-sensors** utilize a reflective target that must be aligned with the **photo-sensor** such that the light emitted by the photo-sensor is reflected back to it.<br><br>**Reflex** (or sometimes known as **proximity**) type **photo-sensors** emit light to be reflected back from an object located sufficiently close to the sensor.<br><br>For both types of **photo-sensors**, when they detect their reflected light source, their signaling output circuit changes state. |
| **IP54** | The **IP Code (International Protection Marking)** specifies the device's degree of resistance to intrusions, dust and water. IP54 certified device must be fully protected from splashed water , dust particles and completely protected from contact |

| | |
|---|---|
| **RJ-45** | **R**egistered **J**ack Style **45** – Standard connector / receptacle format utilizing 8 pin connections.  The typical standard for computer network cable connections |
| **TCP/IP** | **T**ransport **C**ontrol **P**rotocol / **I**nternet **P**rotocol - **IP** is the protocol which oversees the transmission of information packets from device to device on an Ethernet network. **TCP** makes sure the packets have arrived and that the message is complete. These two protocols are the basic language of the Internet and are often referred to together as **TCP/IP**. |
| **ZPA** | **Z**ero **P**ressure **A**ccumulation – Term that describes the conveyor controls and mechanical scheme that will cause loads to queue on a conveyor in discrete **zones** such that loads do not touch each other |

## Symbol Conventions

This symbol indicates that special attention should be paid in order to ensure correct use as well as to avoid danger, incorrect application of product, or potential for unexpected results

This symbol indicates important directions, notes, or other useful information for the proper use of the products and software described herein.

## Important User Information

*ConveyLinx-Ai2* modules contain ESD (Electrostatic Discharge) sensitive parts and components. Static control precautions are required when installing, testing, servicing or replacing these modules.  Component damage may result if ESD control procedures are not followed. If you are not familiar with static control procedures, reference any applicable ESD protection handbook.  Basic guidelines are:

- Touch a grounded object to discharge potential static
- Wear an approved grounding wrist strap
- Do not touch connectors or pins on component boards
- Do not touch circuit components inside the equipment
- Use a static-safe workstation, if available
- Store the equipment in appropriate static-safe packaging when not in use

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes, and standards

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Insight Automation Inc. does not assume responsibility or liability (to include intellectual property liability) for actual use based on the examples shown in this publication

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

## Summary of Changes

The following table summarizes the changes and updates made to this document since the last revision

| Revision | Date | Change / Update |
|---|---|---|
| 1.0 | March 2017 | Initial Release |
| 1.1 | October 2017 | Formatting update |
| | | |

## Global Contact Information



WWW.PULSEROLLER.COM
SALES@PULSEROLLER.COM
SUPPORT@PULSEROLLER.COM

## Summary of Hardware Changes

The following table summarizes the changes and updates made to this document since the last hardware revision.

| Hardware Revisions | | |
|---|---|---|
| Revision | Date | Change / Update |
| 1.0 | August 2016 | Initial Release |

# TABLE OF CONTENTS

## PREFACE

### WHO SHOULD USE THIS MANUAL?

This manual is intended for users who need basic product information and simple application procedures to implement *ConveyLinx-Ai2* modules to control simple linear conveyor.

You should have a basic understanding of electrical circuitry and familiarity with relay logic, conveyor equipment, photo-sensors, etc. If you do not, obtain the proper training before using this product.

### PURPOSE OF THIS MANUAL

The purpose of this manual is to:

- Identify the components and ports available on a module

- Provide guidelines for proper installation and wiring

- Provide examples on basic inter-module connections for linear conveyor

- Introduce the *EasyRoll* software tool and provide instructions to configure and modify parameters.

### NOT INCLUDED IN THIS MANUAL

**Because system applications vary; this manual assumes users and application engineers have properly sized their power distribution capacity per expected motor loading and expected operational duty cycle. Please refer to conveyor equipment and/or motor roller manufacturer's documentation for power supply sizing recommendations.**

# INTRODUCTION TO CONVEYLINX®

## CONVEYLINX® CONCEPT

*ConveyLinx* control system as applied to conveyor control or some other industrial application is a series of *ConveyLinx-IO and ConveyLinx-Ai2* modules interconnected via standard Ethernet cabling. A network of *ConveyLinx* modules are primarily used as ZPA conveyor controllers. *ConveyLinx-IO* is available to allow you to place digital IO on the same network along with *ConveyLinx* conveyor control modules.

Each *ConveyLinx-IO* control module is equipped with up to 8 digital inputs / 8 digital outputs. *ConveyLinx-IO* modules can optionally be easily configured with the *EasyRoll* software to be "transparent" to the conveyor flow to ensure that ZPA conveyor is uninterrupted from upstream to downstream



**FIGURE 1 - CONVEYLINX® CONCEPT WITH CONVEYLINX-AI2 MODULES AND CONVEYLINX-IO MODULE**

## CONVEYLINX-AI2 MODULE FEATURES

Each individual *ConveyLinx-Ai2* module has the following features:

- ✓ Built-in Ethernet switch
- ✓ Modular M8 (female) style connection ports for I/O
- ✓ Modular M8 (male) style connection ports for I/O
- ✓ 24VDC power connection with separate power supplies for logic and outputs
- ✓ Context-sensitive multi-color LED indicators
- ✓ Programmable option for push-pull type input devices
- ✓ IP54 compliant

The first sections of this manual will describe in detail the hardware and connectivity requirements for *ConveyLinx-IO* modules and the software configuration procedures for simple linear conveyor installation.

The latter sections of this manual will describe in detail the usage of the *EasyRoll* software package to gain access to the various optional configuration parameters and utilities.

## CONVEYLINX-AI2 MODULE HARDWARE OVERVIEW

*ConveyLinx-IO* modules are designed to be installed and integrated on or near the mechanical equipment requiring I/O interface to the control system.  Each *ConveyLinx-IO* provides connection points for up to 8 configurable digital inputs or outputs, 24V power supply, and Ethernet network cables.
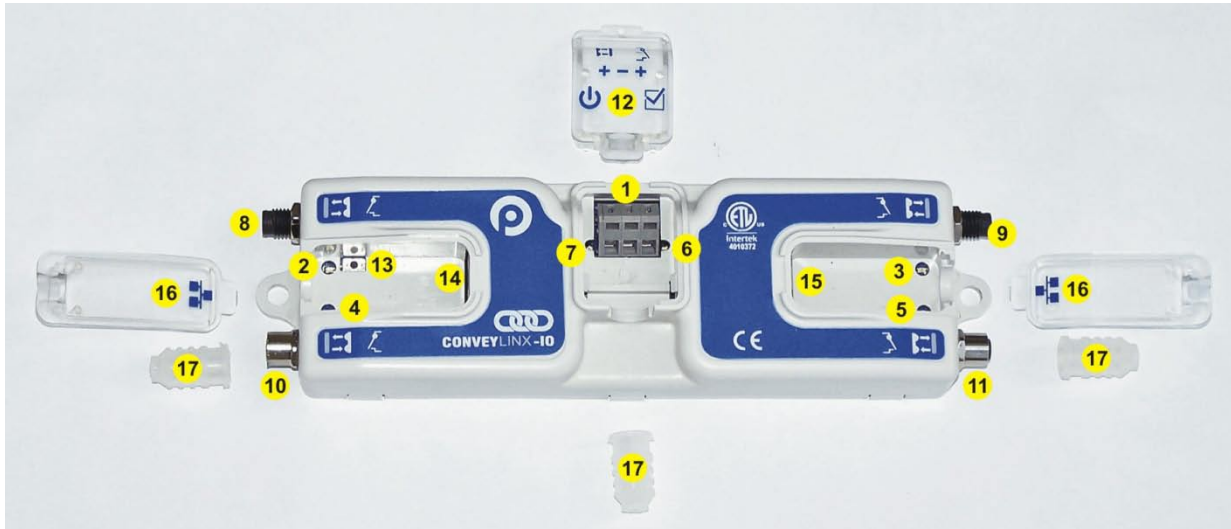


FIGURE 2 – CONVEYLINX-IO MODULE HARDWARE FEATURES IDENTIFICATION

| Item | Description |
|---|---|
| 1 | 24VDC Power Terminals with separate connections for Logic and Motors |
| 2 & 3 | Left Male & Right Male I/O Port Status LED indicators |
| 4 & 5 | Left Female & Right Female I/O Port Status LED Indicators |
| 6 | Module Status LED Indicator |
| 7 | Module Power LED Indicator |
| 8 & 9 | Left & Right Male 4-pin M8 style connector for I/O connections |
| 10 & 11 | Left & Right Female M8 style connector for I/O connections |
| 12 | RemovableIP54 Power Compartment Cover |
| 13 | Module Replacement Button |
| 14 & 15 | Link Left and Link Right – RJ-45 style Ethernet network connection between modules |
| 16 | Removable IP54 Ethernet RJ-45 Port Compartment Cover – Left and Right |
| 17* | IP54 Protection Shrouds for Ethernet cabling and power wiring |

* Indicates items shipped unattached to the module but included in the module's box

**The "left" and "right" naming convention for the module ports is based upon facing the module as shown.**

## HARDWARE CONNECTIONS

### LEFT AND RIGHT MALE I/O PORTS

Both of these ports utilize a 4-pin M8 male receptacle. Each receptacle is mechanically keyed to assure proper orientation upon plugging in. Each of the two signal pins can be configured as a digital input or output.
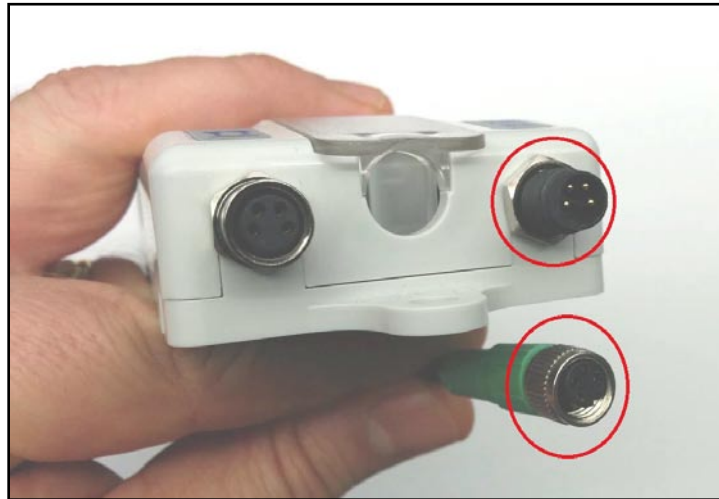


**FIGURE 3 - M8 MALE RECEPTACLE AND FEMALE M8 CONNECTOR**
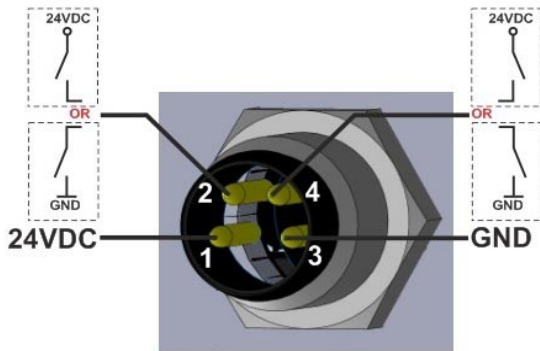
Input Signal Wiring for Male Port             Output Signal Wiring for Male Port
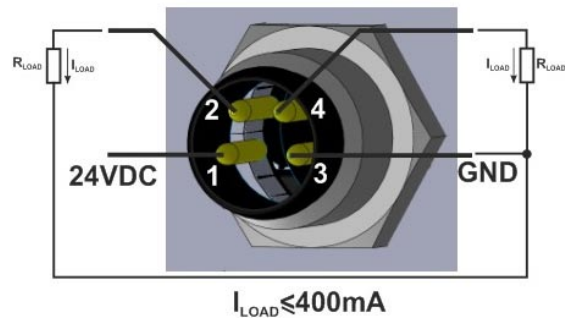
## LEFT AND RIGHT FEMALE I/O PORTS

Each port is a standard M8 Female receptacle. Each receptacle is mechanically keyed to assure proper orientation upon plugging in.
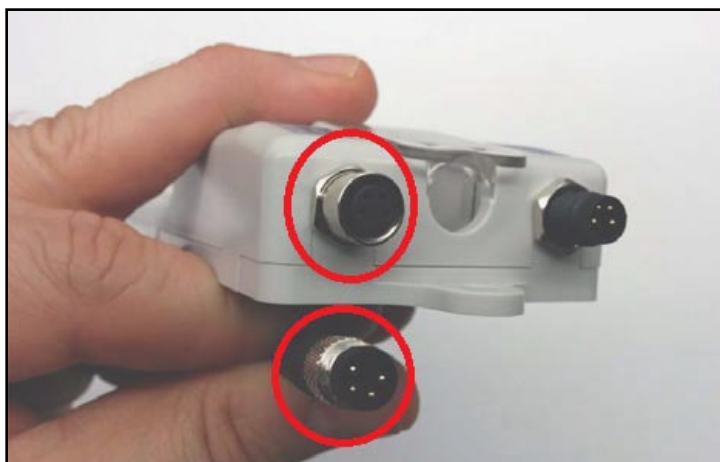


**FIGURE 4 - CONVEYLINX-IO FEMALE M8 PORT AND MATING M8 CONNECTOR**
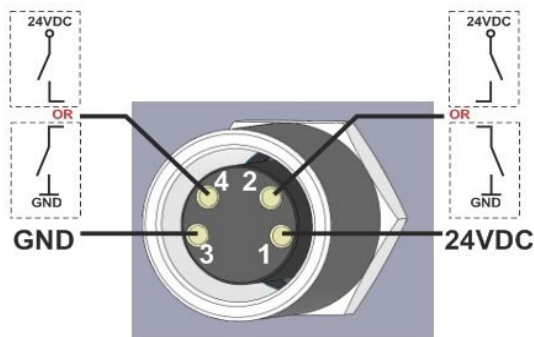
Input Signal Wiring for Female Port          Output Signal Wiring for Female Port
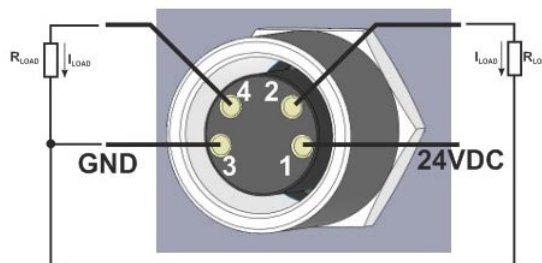


The signals are defined by the following chart:

| Pin | Signal | Description |
| --- | --- | --- |
| 1 | 24V DC | Module 24VDC Supply |
| 2 | I/O | Input or Output signal based upon port configuration |
| 3 | GND | Module DC Common |
| 4 | I/O | Input or Output signal based upon port configuration |

ETHERNET LEFT AND ETHERNET RIGHT PORTS

Both of these ports are standard RJ-45 jacks conforming to standard Ethernet connection pin-out.  In order to maintain IP54 rating; Ethernet cables need to be equipped with protective shrouds.  Figure 5 shows Ethernet cables installed using shrouds to protect the RJ-45 connectors on the Ethernet cables.  Each module is shipped with 3 shrouds.
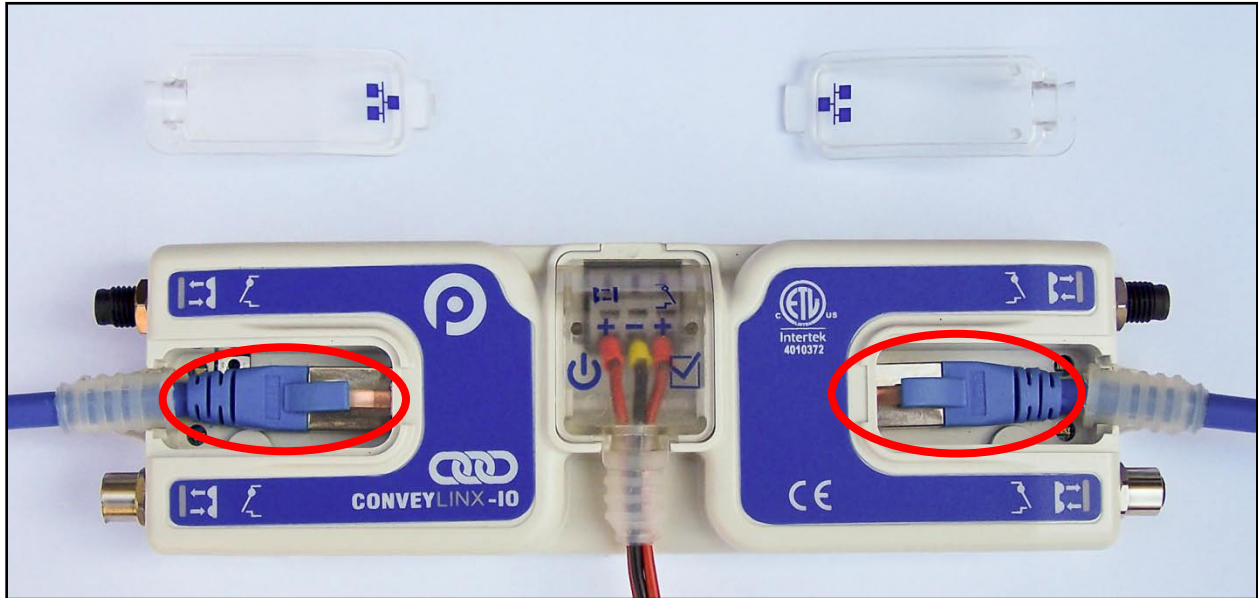


FIGURE 5 – CONVEYLINX-IO WITH LEFT & RIGHT ETHERNET CABLES (COVERS REMOVED)
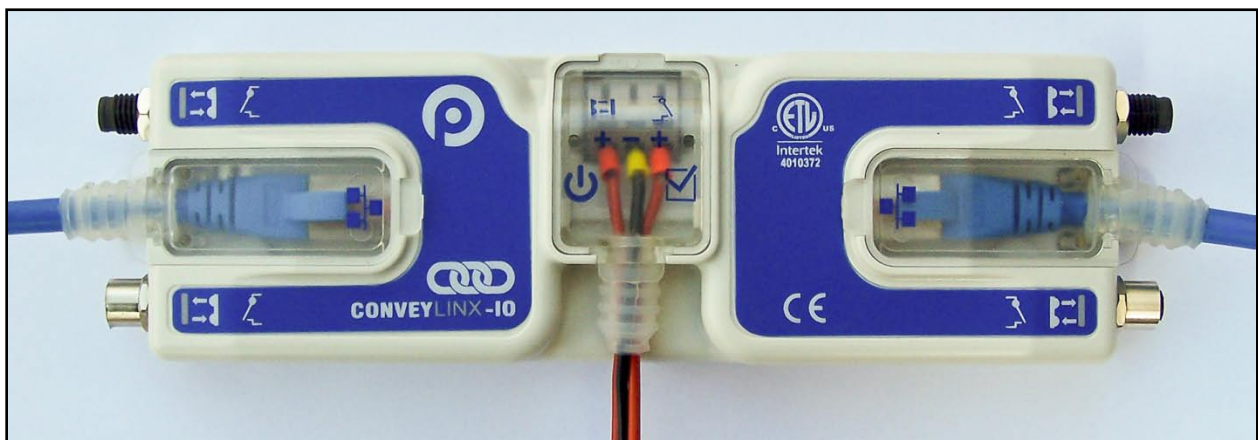


FIGURE 6 - ETHERNET CABLES CONNECTED WITH COVERS ATTACHED

IP54 protective shroud requires a special tool to properly install the shroud onto the RJ-45 connector. Figure 7 shows Phoenix Contact item 2891547 FL IP 54 Assembly Tool.

*This tool must be purchased separately*



**FIGURE 7 - PHOENIX CONTACT ETHERNET SWITCH 2891547 FL IP 54 ASSEMBLY TOOL**



**FIGURE 8 - PROTECTIVE SHROUND PLACED ON TOOL**



**FIGURE 9 - SHROUND STRETCHED WITH TOOL**



**FIGURE 10 - INSERTING RJ-45 END THROUGH SHROUD**
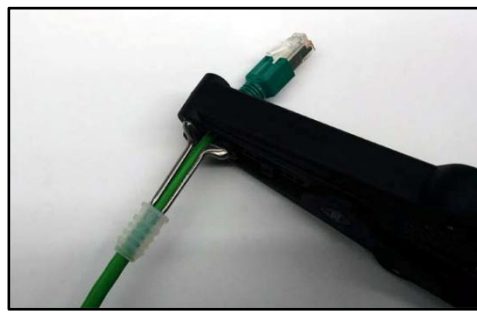


**FIGURE 11 - SHROUD REMOVED FROM TOOL PINS**

**All Ethernet cables for connections between modules are recommended to be shielded. Failure to use shielded cables may result in data loss and unexpected results. Shown above are Ethernet cables with sealing shrouds required for IP54 compliance.**

POWER CONNECTIONS

IP54 INSTALLATION

Item 17 as depicted in Figure 2 may be unattached to the module when shipped and are included in the module's shipping box.  These items are used to maintain an IP54 installation of the power and Ethernet wiring.

Power wires are fed through the protective shroud (Item 17).  The wire terminals are standard cage-clamp style.
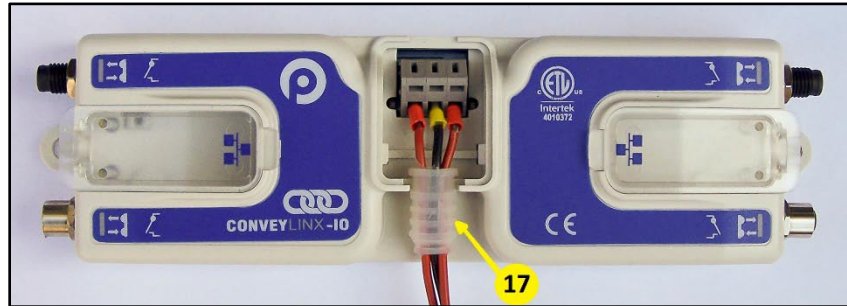
Once wiring has been completed the power wiring compartment is then sealed by snapping into place the Power Compartment Cover (Item 18).
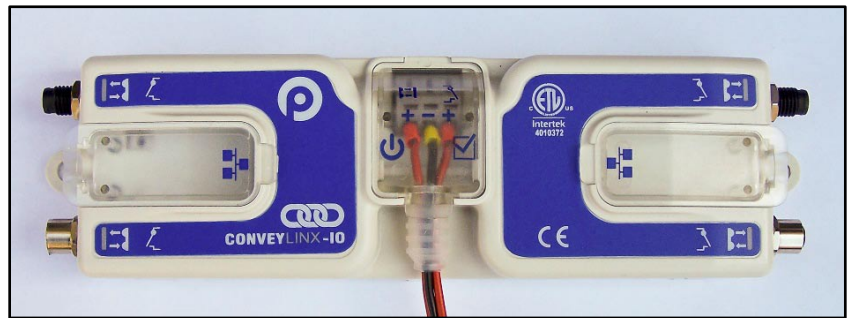
LOGIC AND OUTPUT POWER

The *ConveyLinx IO* module is designed to allow for separate power connections for module logic and digital outputs power so that these can be powered by separate power supplies.  For example, the digital outputs power supply can switched off by an emergency stop control system so that all output devices have power removed. With the outputs power separately switched off; the logic power supply can remain on so that the module's communications can remain active and report status to networked supervisory control system(s).  Figure 13 shows a diagram for separate logic and MDR power supplies and Figure 12 shows a diagram for a single power supply for both logic and MDR power.  Note that powering the Outputs terminal also powers the Logic.
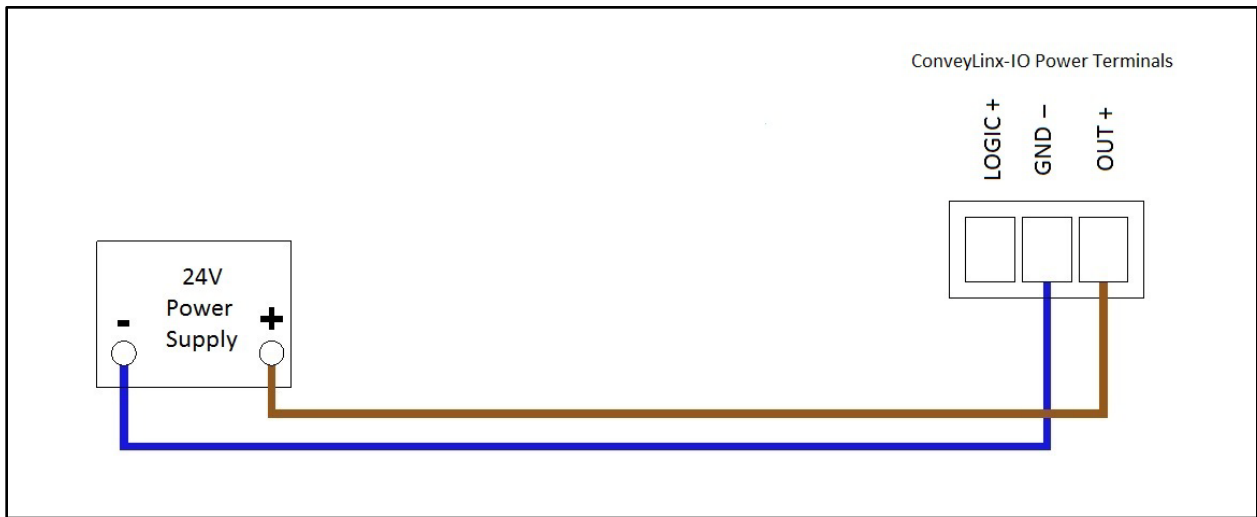
**FIGURE 12 - CONNECTION FOR SINGLE OUTPUTS AND LOGIC POWER SUPPLY**
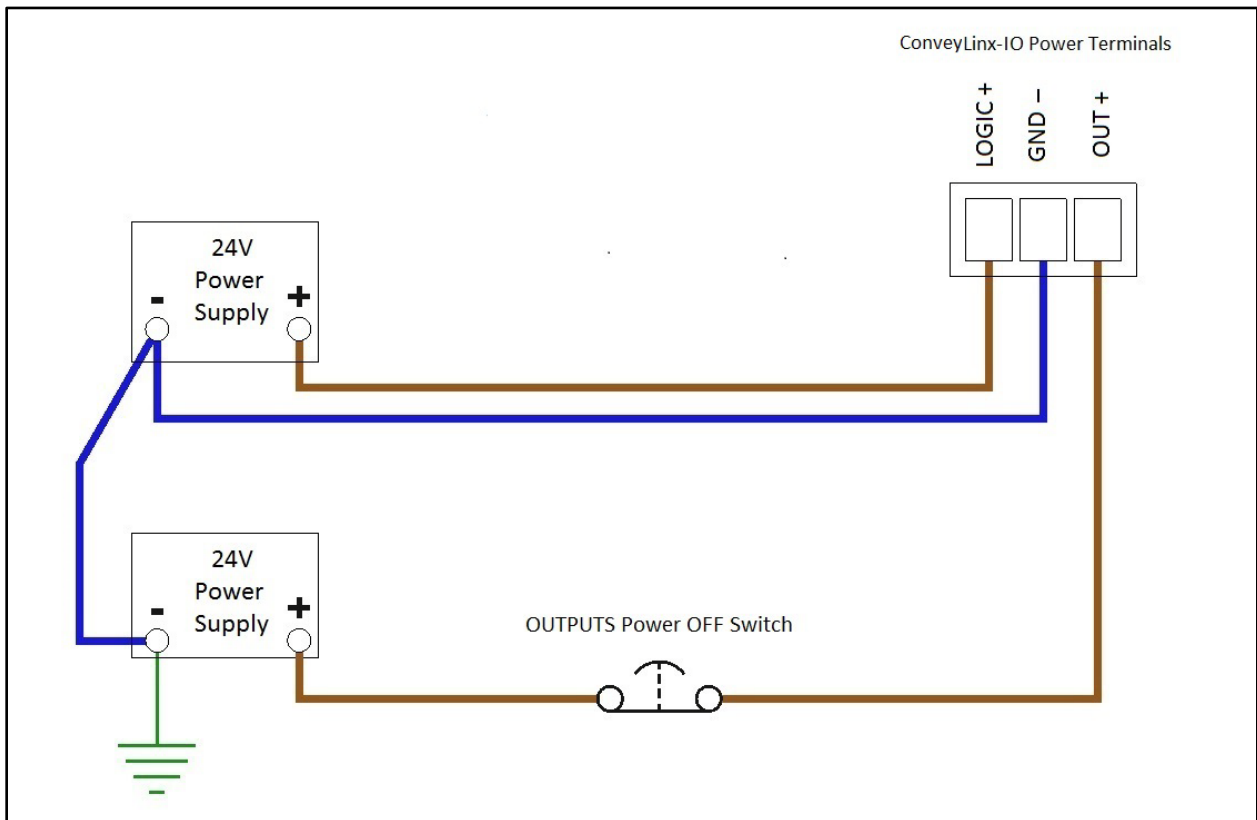


**FIGURE 13 - TYPICAL CONNECTION FOR SEPARATE OUTPUTS AND LOGIC POWER SUPPLIES**

## POWER SUPPLY COMMON GROUNDING

Whether logic and digital outputs are powered together or separately; the DC common ("-") connections on all power supplies should be connected together.  One of the power supplies should have its DC common connected to ground.  Avoid connecting more than one power supply DC common to ground because this can lead to unintended grounding loops.  Figure 14 and Figure 15 show single and separate power supply connections respectively and their DC common connections and grounding connection.
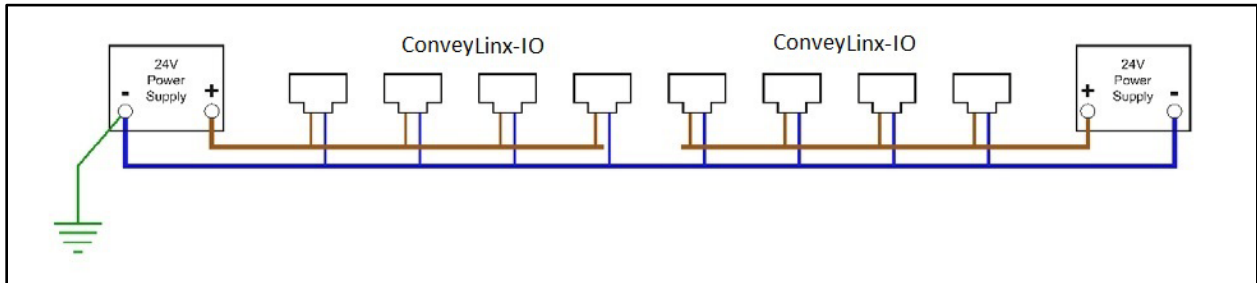

FIGURE 14 – SINGLE POWER SUPPLY CONNECTION WITH DC COMMONS TIED TOGETHER AND TO GROUND
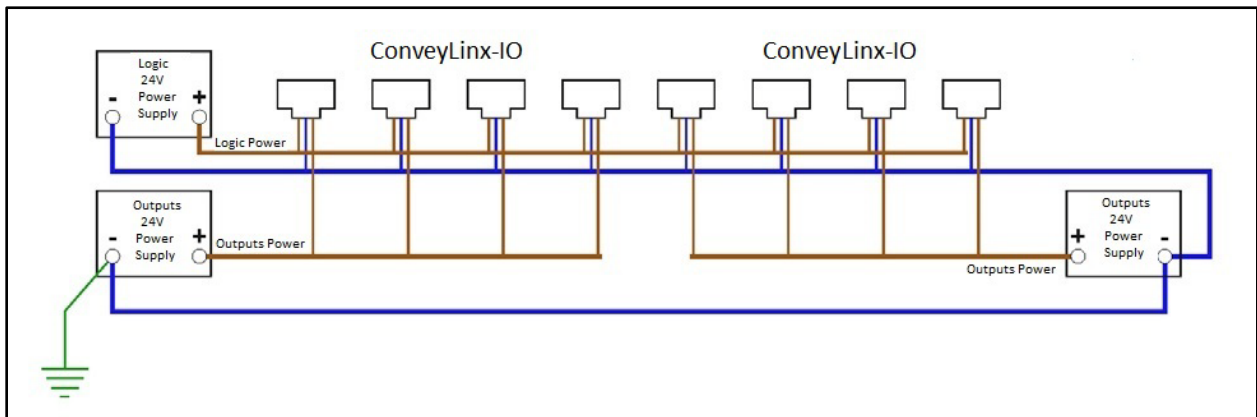

FIGURE 15 - DUAL POWER SUPPLY CONNECTION WITH DC COMMONS TIED TOGETHER AND TO GROUND

This document assumes the user is aware of power requirements for the application and that the user and/or installer have properly sized 24VDC power supplies and wiring based upon all applicable codes and standards.  This document also assumes installation will follow proper equipment grounding practices. "DC common or -"**on all power supplies should always be connected to ground.**  Improper power supply sizing and/or improper grounding practices may produce unexpected results.

## STATUS INDICATORS

*ConveyLinx-IO* module status is indicated by several LED's.  All LED's with the exception of the Ethernet Link and Activity LEDs are multi-coloured and context sensitive.  The following chart indicates the various meanings of all *ConveyLinx-IO* LED indicators.  Please refer to Figure 2 on page 15 for the item number locations on the module. By definition *Blinking* is approximately ½ second on/off cycle and *Flashing* is approximately ¼ second on/off cycle.

### COMMUNICATIONS

| Indicator | Item | LED State | Description |
|---|---|---|---|
| Ethernet Left Link & Ethernet Right Link | On the RJ-45 connectors (13&14) | OFF | No connection established |
| | | Solid Green | Connection is established |
| | | Blinking Green | When data transmission activity is occurring |

### NETWORK & MODULE FUNCTION

| Indicator | Item | LED State | Description |
|---|---|---|---|
| Module Status | 6 | *Blinking* Red | *ConveyLinx-IO* is starting task processes |
| | | *Blinking* Green | *ConveyLinx-IO* is ready |
| | | Flashing Green & Blinking Red | Failsafe Mode |
| | | Flashing Red | Auto Configure Mode is active |
| | | Blinking Amber | Connection to peer lost or performing firmware upgrade check |
| | | Solid Amber | Firmware upgrade in progress |

### INPUTS

| Indicator | Item | LED State | Description |
|---|---|---|---|
| Left & Right IO LEDs | 2, 3, 4 & 5 | OFF | Inputs are OFF |
| | | Solid Green | Input pin4 is ON |
| | | Solid Red | Input pin2 is ON |

### OUTPUTS

| Indicator | Item | LED State | Description |
|---|---|---|---|
| Left & Right IO LEDs | 2 & 3 & 4 & 5 | OFF | Outputs are OFF |
| | | Flash Green | Output pin4 is ON |
| | | Flash Red | Output pin2 is ON |

## POWER

| Indicator | Item | LED State | Description |
|-----------|------|-----------|-------------|
| Power | 7 | Solid Blue | Power supply is connected and voltage is within acceptable range |
| | | Slow Blink Blue | Outputs power is under 18V |

## AUTO-CONFIGURATION

The purpose of Auto-Configuration for networked *ConveyLinx* controls is to provide a simple and easy procedure for linear conveyor system commissioning. *ConveyLinx-IO* also supports Auto-Configuration and can be combined with other *ConveyLinx-IO* modules or *ConveyLinx-Ai2* modules. The Auto-Configuration feature of *ConveyLinx* requires the use of the *EasyRoll* on a PC connected to the most upstream module of the linear network.

### INSTALLING EASYROLL TOOL ON YOUR PC

The files for *EasyRoll* will be typically sent or downloaded in a compressed (i.e. ".zip") format. Once you have extracted the contents of the compressed file; the result will be a folder named with the format "EasyRoll_Vx_nn" where x is the main version number and the nn is the revision level. Inside this folder is a file named "Setup.exe". Double click this file to begin the install procedure. *EasyRoll* installs like any standard Windows application and you will be prompted for typical Windows prompts. By accepting the defaults for the prompts; *EasyRoll* will install on your local *Operating System* drive under the " \Program Files (x86)\Industrial Software\EasyRoll\" or " \Program Files\Industrial Software\EasyRoll\".

### CONVEYLINX ETHERNET DEFINITION

Please refer to **Error! Not a valid result for table.** for pre-requisite information on understanding Ethernet network I.P. addresses and Subnet concepts. Further description in this section assumes you have a general knowledge level of I.P. addressing and subnets.

All *ConveyLinx* modules communicate over Ethernet network and use TCP/IP based protocols for normal function. All TCP/IP protocols require that each device on a network have a unique I.P. address assigned to it in order to function properly.

An I.P. address is in the format of: `AAA.BBB.CCC.DDD` where `AAA,` `BBB,` `CCC`, and DDD are numerical values between 0 and 255.

For the purposes of *ConveyLinx;* the `AAA.BBB.CCC` portion of the I.P. address taken together is defined as the **Subnet**. The DDD value of the address is defined as the **Node**.

For example; if a *ConveyLinx-IO* has an I.P. address of "192.168.25.20" then its *Subnet* address is "192.168.25" and its *Node* is "20"

At the factory, each and every *ConveyLinx-IO* module is assigned a temporary I.P. address that is used by automated testing equipment and fixtures so that every *ConveyLinx-IO* is verified prior to shipment. When a *ConveyLinx-IO* is taken "out of the box" it will still have this I.P. address stored in its memory.

When the *Auto-Configuration Procedure* is initiated; one of the many things that occur is that each module is automatically assigned a new I.P. address. This I.P. address for all modules is determined by the **Subnet** of the I.P. address already stored inside whichever *ConveyLinx* module is selected as the *Auto Configuration Master*. Even if all downstream modules from the *Auto Configuration Master* have the same or different *Subnet* or *Node* values; these downstream modules will have their *Subnet* changed to the existing *Subnet* of the *Auto Configuration*

*Master*.   Furthermore, when the *Auto Configuration Procedure* occurs; the *Auto Configuration Master* will also have its **Node** value changed to **20**.   All downstream *ConveyLinx* modules will then have their *Node* values automatically set beginning with 21.

Figure 16 shows 2 *ConveyLinx-IO* and 2 *ConveyLinx-Ai2* installed "out of the box" onto, for example, a conveyor. Once the *Auto Configuration Master* is identified and the *Auto-Configuration Procedure* is performed; all 4 *ConveyLinx* modules will have their I.P. address configured as shown in Figure 17.
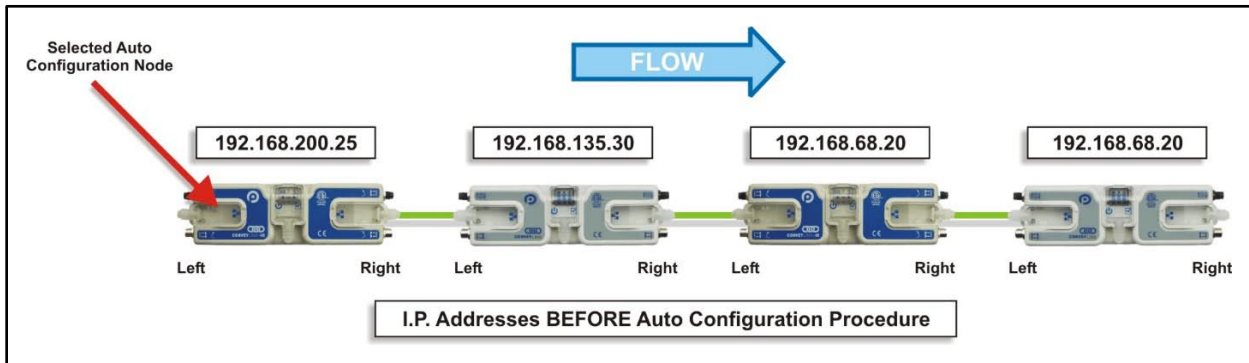


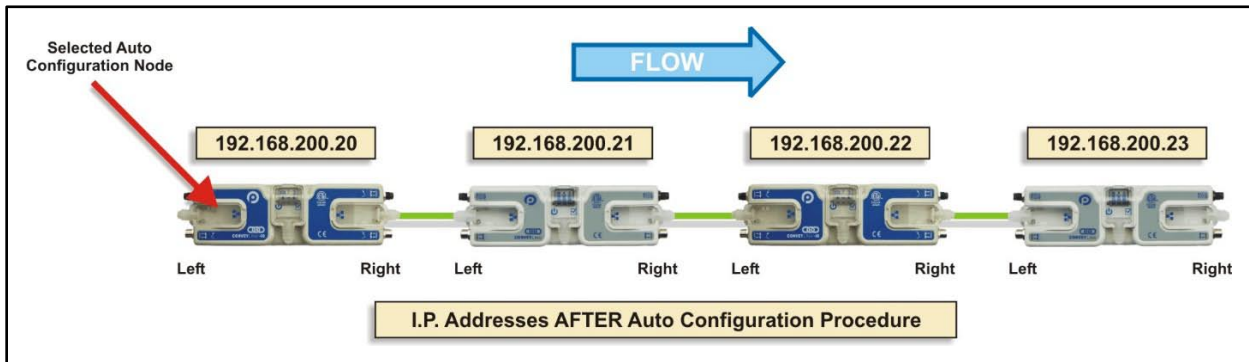**FIGURE 16 - IP ADDRESSES BEFORE AUTO-CONFIGURATION**



**FIGURE 17 - IP ADDRESSES AFTER AUTO-CONFIGURATION**

The *Auto-Configuration Procedure* will assign *Nodes* up to and including *Node 240*.   Therefore each *Subnet* is limited to 220 *ConveyLinx* nodes.

## CONNECTING YOUR PC TO *CONVEYLINX* NETWORK

Using an Ethernet cable, connect your PC's Ethernet port to the *Auto-Configuration Master* as shown below in Figure 18
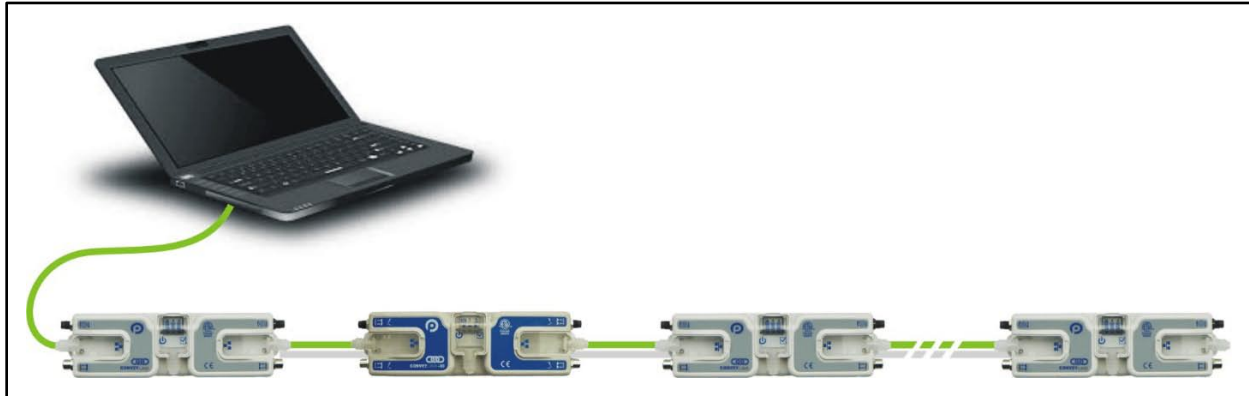


**FIGURE 18 - INITIAL PC CONNECTION TO CONVEYLINX SUBNET**

> ⚠️ **It is highly recommended to connect the PC directly to the ConveyLinx network. Avoid trying to connect via Ethernet switches or wireless router/switches. If a wireless switch is not setup properly then the Discover Feature will not work. Also ensure that network firewall is turned off for proper discovery.**

## AUTO-CONFIGURATION PROCEDURE

The *ConveyLinx* module located at the most upstream or in-feed end of the conveyor is defined as the **Auto-Configuration Node**. The Auto-Configuration procedure is initiated from the *Auto-Configuration Node*. Because of its physical location on the conveyor path and physical location in the Ethernet connection chain; the *Auto-Configuration Node* will automatically connect to all downstream modules and set their I.P. address for communication. Then the routine automatically sets the direction of flow. The following is the procedure to follow:

1. *Start EasyRoll V4.11 or higher* and press F2. In the pop-up panel select "Network Services" and click "Discover".

2. When the *ConveyLinx* modules appear in the module table, select the ones that you want to configure and click "ConveyLinx tree" as shown in Figure 19.

3. When the topology is drawn, select the *ConveyLinx* module from which you want to configure the line and click "Auto-Configure from selected node" as shown in Figure 20.

4. This starts the Auto-Configuration procedure which will spread to all subsequent *ConveyLinx* modules and configure the entire line.

Please note that number of modules on a single Subnet is limited to 220.
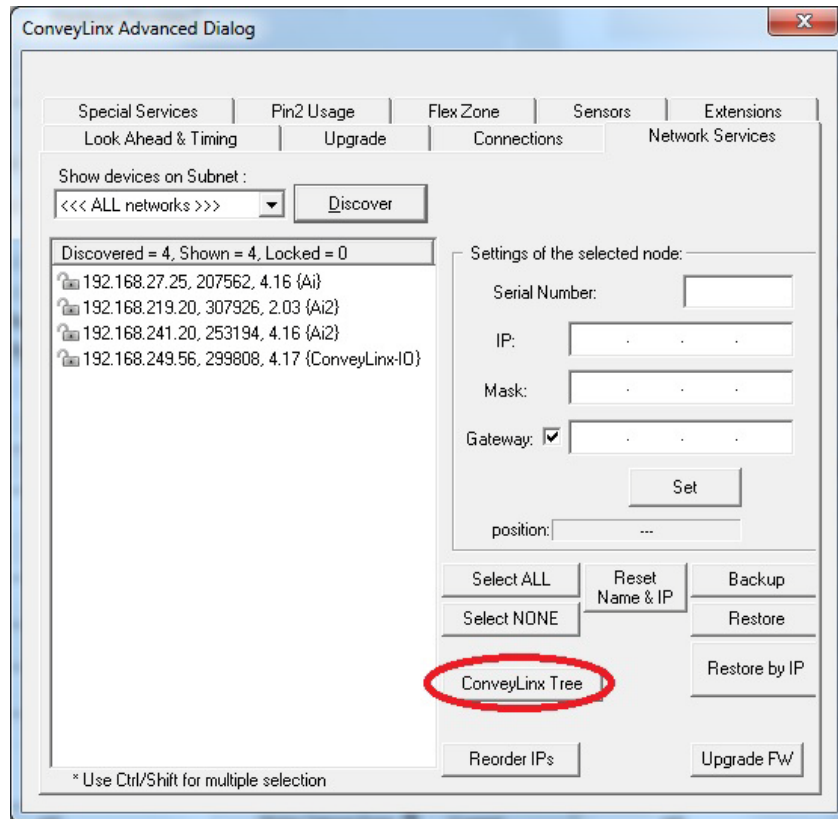


FIGURE 19 - DISCOVER PROCEDURE ON CONVEYLINX-AI2 MODULES
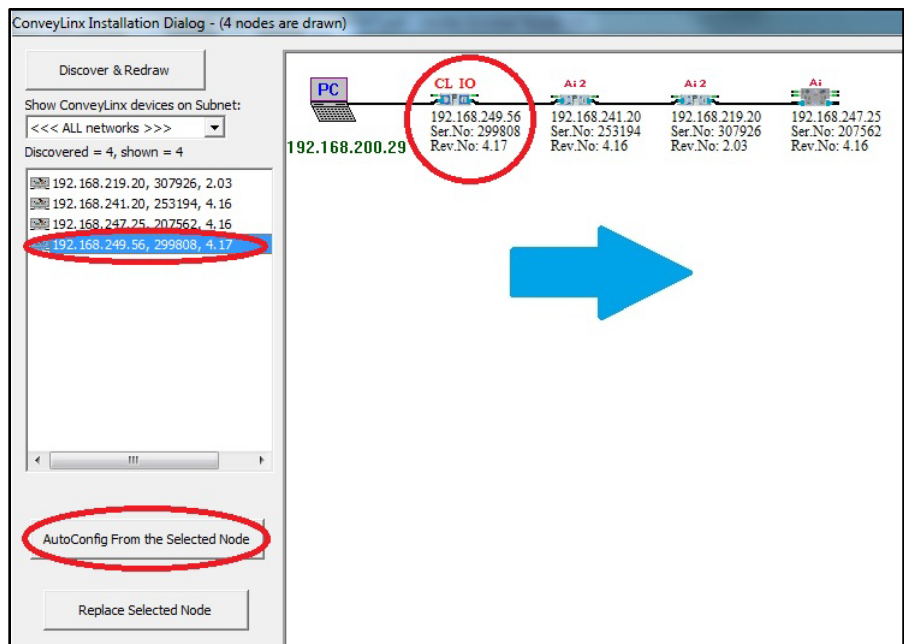
**FIGURE 20 – TOPOLOGY VIEW AND INITIATING AUTO-CONFIGURE FROM THE SELECTED NODE**

You can also select the module graphic from the topology view and right click to show the context menu and initiate the Auto-Configuration from there as shown in Figure 21.
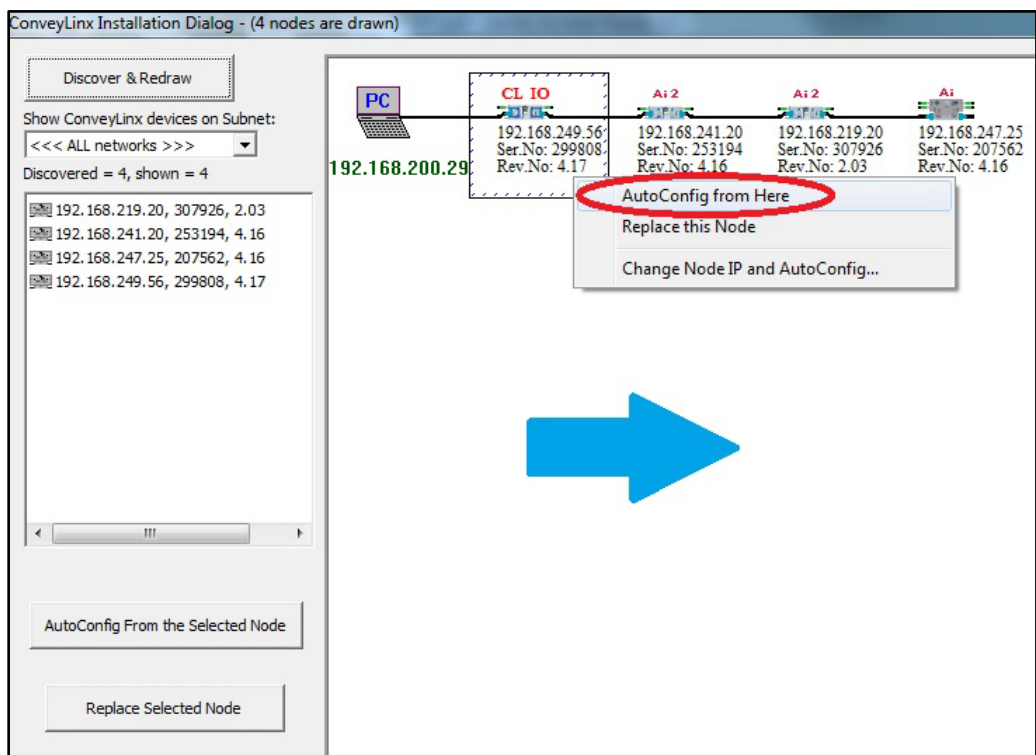


**FIGURE 21 - AUTO-CONFIGURE BY SELECTING NODE IN THE TREE VIEW**

The selected node for Auto-Configuration does not have to be the first module that is connected directly to the PC. The module at the other end of the detected line can also be selected as the Auto-Configuration master as shown in Figure 22.
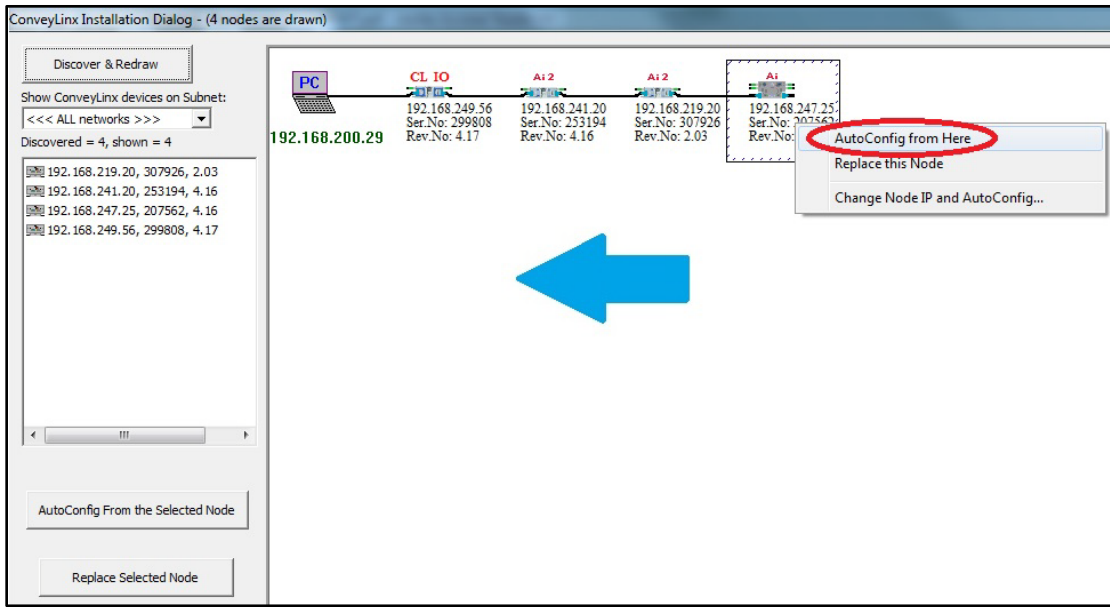


FIGURE 22 - AUTO-CONFIGURATION FROM THE FARTHEST NODE FROM THE PC

Also note that the detail information in the topology view for each module will be shown in RED when the modules are in the same Subnet but with non-consecutive IP addresses or the modules are with duplicate IP addresses.  If some or all of the modules are shown in RED, Auto-Configuration will still function properly.  Figure 23 shows an example where all modules have the same duplicated IP Address.  In this example, you can still perform Auto-Configuration and the addresses will be updated automatically.
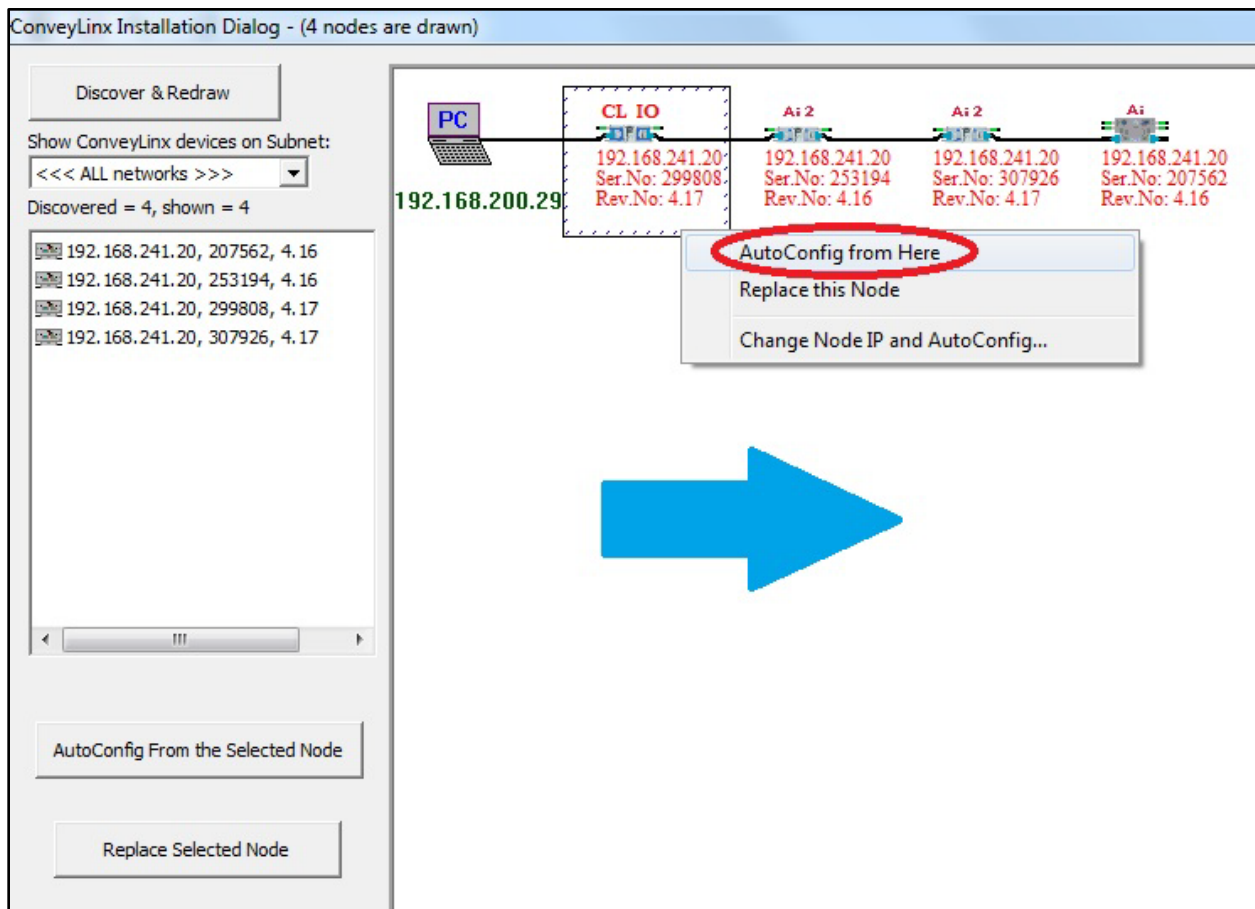
**FIGURE 23 - AUTO CONFIGURATION TOPOLOGY VIEW SHOWING DUPLICATE IP ADDRESSES**

## AUTO CONFIGURATION RESULTS

### NORMAL RESULTS

When the auto-configuration routine is complete, each *ConveyLinx* module will automatically reboot. When a *ConveyLinx-IO* module has been successfully configured and rebooted, its *Module Status* LED will blink on and off green.

> Please note that the time to complete the Auto-Configuration procedure is dependent on the number of modules being configured. Larger networks will take more time than smaller networks.

The topology view will show the module detail information in black text (no longer red) when the auto configuration is complete and successful as shown in Figure 24.
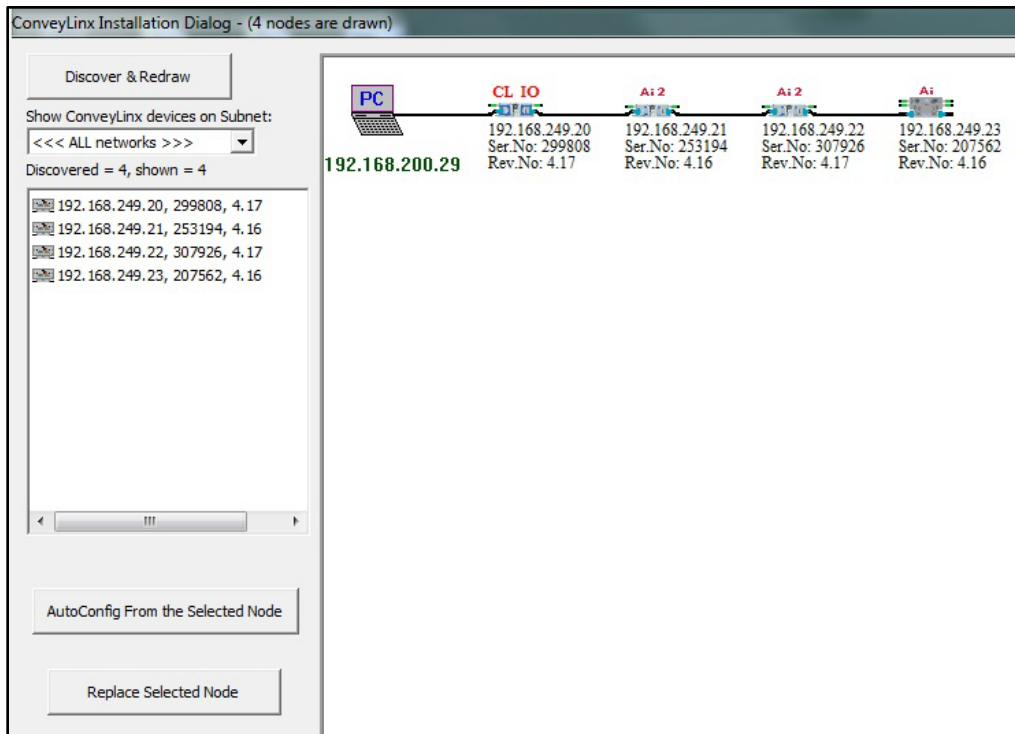
**FIGURE 24 - AUTO-CONFIGURATION SUCCESS**

Please note that once a network has been configured; attempting to again Auto-Configure any *ConveyLinx* module that is not the *Auto Configuration Master* will not initiate a new Auto Configuration procedure.  The *ConveyLinx* module will detect that it is not the most upstream unit and abort the procedure.  However the *ConveyLinx* module will perform its local re-booting procedure.  This procedure will take a few seconds to complete

## TROUBLE-SHOOTING FAILED AUTO-CONFIGURATION

The following chart lists some typical failed condition indicators and actions to take for resolution.

| Failed Condition | Action |
| --- | --- |
| Status LED's OK with Unexpected Result | • Check all network and power connections and perform procedure again.<br>• For ConveyLinx-Ai2 modules controlling conveyors and/or motors, please refer to the ConveyLinx-Ai2 User's Guide for troubleshooting details |
| Status LED blinking or solid red on one or more modules | • Verify that there are no Ethernet Switches or PC's connected between *ConveyLinx* modules. The Auto Configuration procedure will abort if a non-*ConveyLinx* device is detected along the path before reaching the last node. Modules up to that point will be configured, but the remaining modules will not.<br>• When removing a *ConveyLinx* module from an existing network that is already operational; be sure to wait at least 2 minutes to allow the Ethernet switches on the remaining modules to reset before attempting a new Auto Configuration procedure. |

## LOW VOLTAGE FAULT

In instances when the *ConveyLinx-IO* module detects that its supply voltage has dipped below 16VDC; the *ConveyLinx-IO* will turn off its digital outputs.  The *ConveyLinx-IO* will keep this state until it has detected that its power supply voltage has risen to at least 21VDC.

## AUTOMATIC MODULE REPLACEMENT

Once a *ConveyLinx-IO* module has been configured with its connections and settings (including transparency options and optional ConveyLogix program); this data is automatically stored in its upstream and downstream neighboring modules.  This configuration data is automatically updated even if settings and options have been modified since the initial Auto-Configuration by the *EasyRoll* software for example.  The *ConveyLinx-IO* firmware uses this feature to allow for easy module replacement so that the entire linear conveyor does not have to be re-configured in order to replace a single module.

*Automatic Module Replacement procedure will work to replace a ConveyLinx module on properly configured systems with multiple subnets.  You do not have to temporarily disconnect any network connections or otherwise isolate the particular subnet where the replacement procedure needs to occur.*

### CONVEYLINX MODULE REPLACEMENT PROCEDURE USING EASYROLL

1. Disconnect existing module's network, digital I/O, and power connections.  The order of disconnection does not matter.
2. Connect new module's network, digital I/O, and power connections.
3. Start *EasyRoll V4.12* or higher. Press F2 and go to tab **Network Service**, click on **Discover** button. All modules in the network should be display. Click on **ConveyLinx Tree** button as shown in Figure 25 . Topology of the ConveyLinx modules will be display in a few seconds. Select the *ConveyLinx* module from the list on the left side and click on **Replace Selected Node** as shown in Figure 26. Note that this module's text detail in the topology view will be in RED.
4. Wait until *ConveyLinx* module performs its internal boot-up procedures which will be indicated when the Module Status LED blinks on and off green.
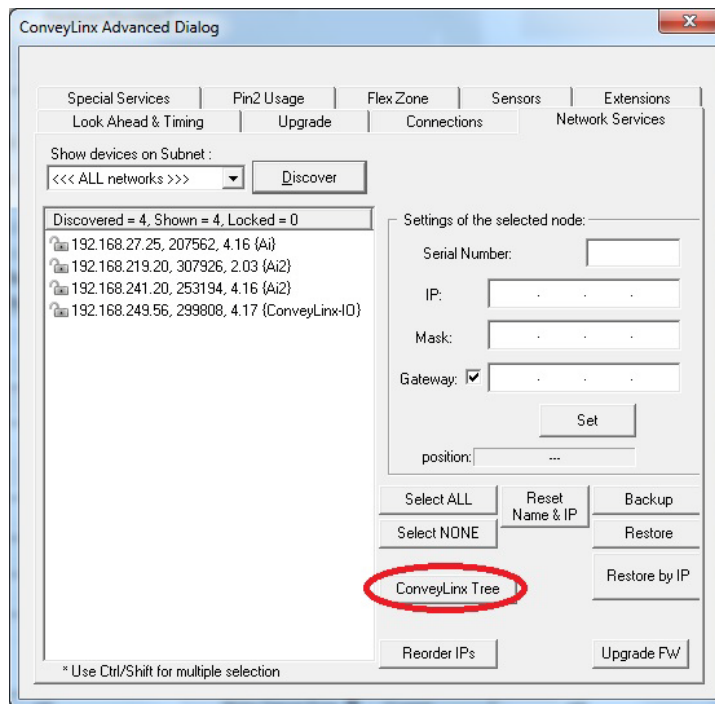
FIGURE 25 - EASYROLL NETWORK SERVICES TAB AFTER DISCOVERY
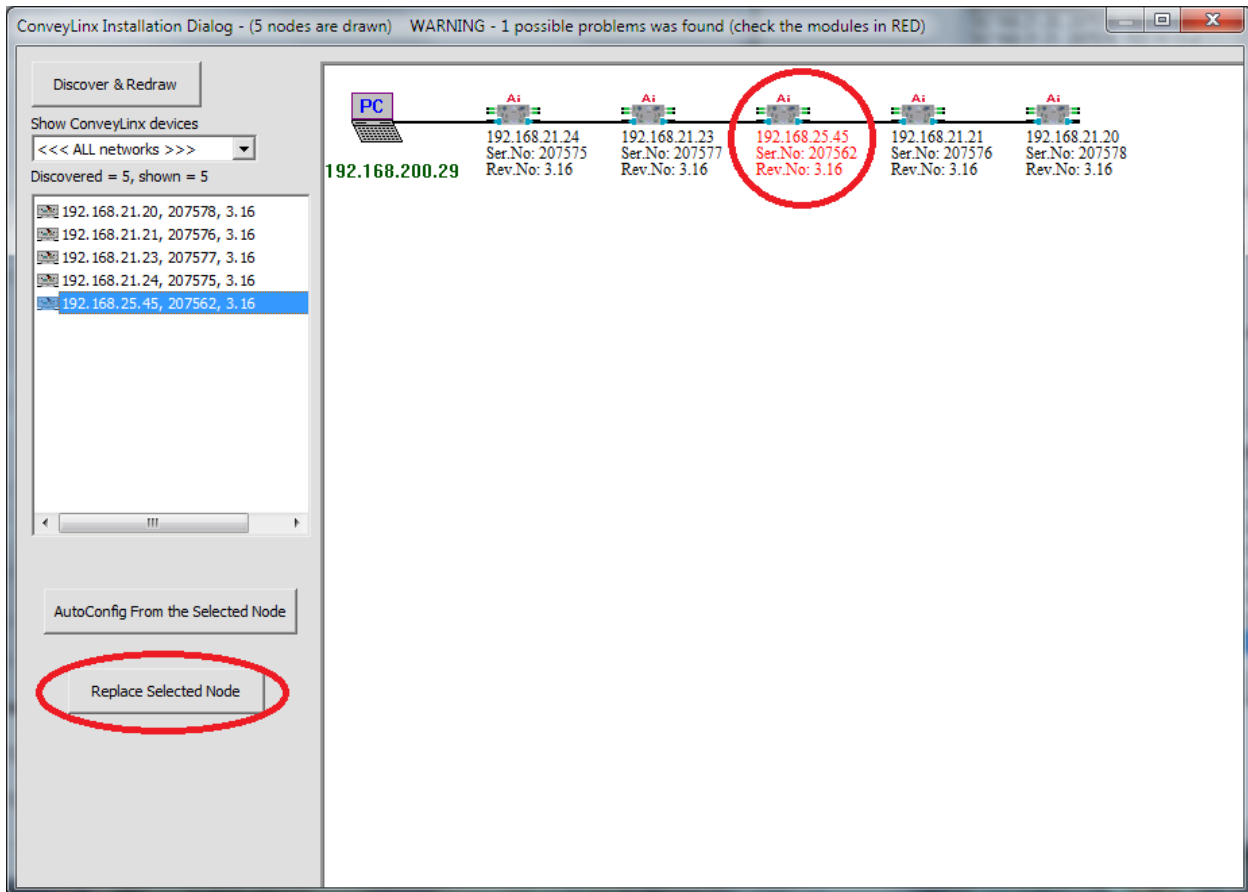
**FIGURE 26 - TOPOLOGY VIEW AND REPLACING THE SELECTED NODE**

You can also replace a module by selecting the module graphic directly in the topology view, right click to display the context menu, and select "Replace this Node" as shown in Figure 27.
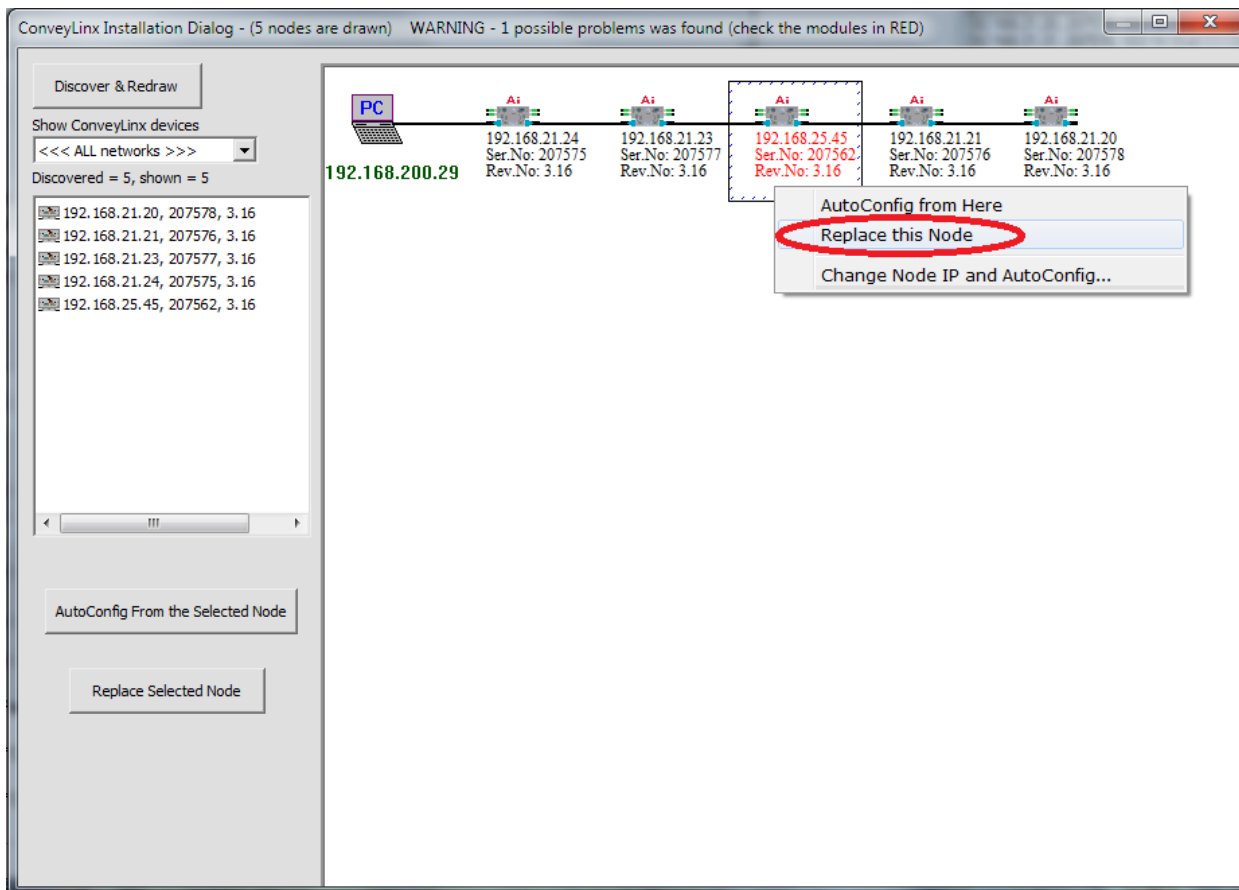
**FIGURE 27 - MODULE REPLACEMENT FROM TOPOLOGY VIEW**

## MODULE REPLACEMENT USING REPLACEMENT BUTTON

You can replace a single *ConveyLinx-IO* module without using *EasyRoll* by utilizing the Replacement Button located inside the Left Ethernet cable connection compartment. Simply install the replacement module and connect all sensors, motors, network cables, and power connections. The procedure is illustrated below.

The Module Replacement Button is located in the Left Ethernet cable compartment. Remove the compartment cover to gain access to the button
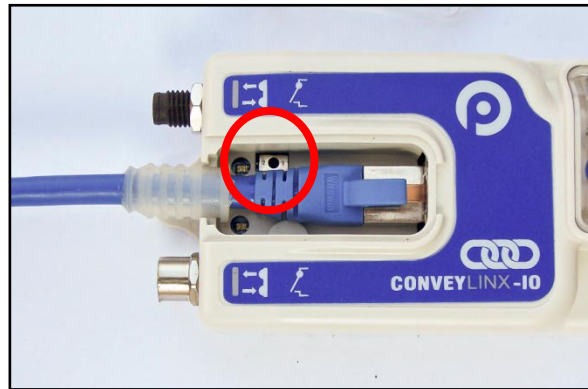


With a pointed instrument (pencil or small screwdriver) press and hold the button for 2 seconds until the Status LED flashes red. Then release the button.
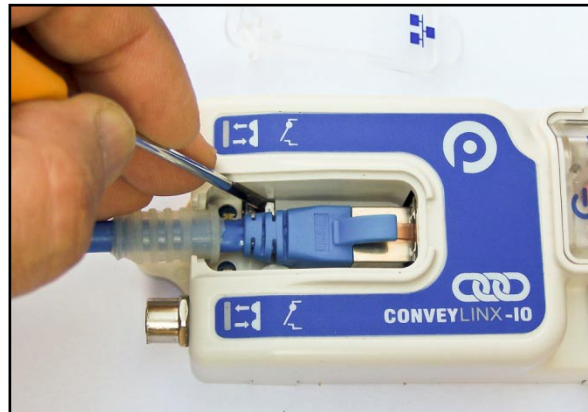


**Please note that the Module Replacement Procedure (either with EasyRoll or Replacement Button) can take a few minutes to complete. The replaced module will restart itself a minimum of 4 times during the procedure. It will restart 5 times if firmware is being upgraded or downgraded.**

# EASYROLL SOFTWARE CONFIGURATION TOOL

## INTRODUCTION

The *EasyRoll* Software Configuration Tool is a PC based application that provides the means to configure a *ConveyLinx* controlled conveyor system. *EasyRoll* also provides the ability to change *ConveyLinx* module default parameters.

> **Installing EasyRoll and performing the initial Auto Configuration is described in the previous section**

## BASIC FEATURES

Some of the basic module parameters that can be modified by *EasyRoll* are:

- Graphical view of the current state of each digital input and output
- Configure the module to be *Transparent* to the flow of ZPA *ConveyLinx-Ai2* in a network
- *Blink &Wink* function used to visually locate an *ConveyLinx-Ai2* on the conveyor

*EasyRoll* has the ability to display the status information for any *ConveyLinx* module on the network's subnet.

## ADVANCED FEATURES

Some of the advanced features available with *EasyRoll* are:

- Firmware Upgrade utility for one or a group of *ConveyLinx* modules
- *Network Services Discover* utility used to find all *ConveyLinx* modules on a network and manually set their I.P. addresses.
- Module *Connection* mapping to logically link two or more separate *ConveyLinx* networks.
- The ability to Back-Up and Restore the Network Configuration.
- The ability to restore a backup by IP or by Nodes. Restoring by IP's is useful when you need to duplicate part or all of a system.
- Access to the Module Auto-Configuration function
- Access to the Module Replacement function

## OPTIONS FOR CONFIGURING YOUR PC'S IP ADDRESS

> *Note that your PC's I.P. address does not have to be in the same subnet as your ConveyLinx modules in order to perform the Auto Configuration procedure with EasyRoll as described in the previous section. However, your PC's I.P. address and/or subnet mask has to be properly set in order to read and change ConveyLinx module's parameters and settings.*

Once a *ConveyLinx* network or *Subnet* has been configured by the *Auto-Configuration Procedure* with the *Subnet* value taken from the factory "out of the box" IP address of the *Auto Configuration Master* (similar to example shown above); you will need to do **one** of the **three** following procedure in order to have your PC be able to connect to the *Subnet* and use *EasyRoll* software:

| Option | Description |
|---|---|
| Method 1 | Allow *ConveyLinx's* built-in *DHCP* service automatically assign an I.P. address to your PC |
| Method 2 | Manually change the I.P. address and/or subnet mask of your PC to match the *ConveyLinx Subnet* |
| Method 3 | Manually change the I.P. address of the *Auto-Configuration Master* to a new *Subnet* that is accessible from the I.P. address already configured in your PC |

Any of these choices is equally valid and totally dependent on user preference.


### METHOD 1 - USING DHCP SERVICE FOR PC I.P. ADDRESS

For installations where you are connecting to a single simple *Subnet* and your PC is already configured to have its I.P. address assigned; it is recommended to allow the *ConveyLinx* network automatically assign an I.P. address to your PC utilizing *ConveyLinx's* built-in *DHCP* service. This is the easiest method particularly if your PC is already set-up to have its I.P. addressed assigned.

If your PC is already configured to obtain an IP address automatically; then by simply connecting you PC as shown in *Figure 18 - Initial PC Connection to ConveyLinx Subnet* is all you have to do to have the PC's I.P. address configured so you can use *EasyRoll*
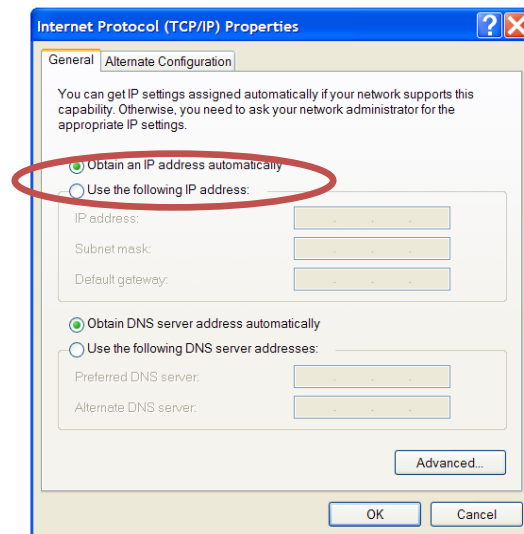
## MANUAL I.P. ADDRESS CONFIGURATION METHODS

Manual configuration of your PC's I.P. address may be your preference for larger system configurations with multiple *Subnets* and/or installations where you want to keep a dedicated PC connected all the time.

For installations where there are multiple *ConveyLinx Subnets* that share the same physical Ethernet cabling (either directly or through Ethernet switches); it is recommended that the *Subnets* be pre-determined and that each *Auto-Configuration Master* have its *Subnet* set in advance of performing each of their respective Auto Configuration Procedures.  By pre-determining all *Subnets* required; your PC can have its I.P. address and subnet mask set to appropriate values so that you can access all of your *ConveyLinx Subnets* from a single PC with *EasyRoll*.

Regardless of which manual procedure you choose, with your PC and *EasyRoll* you can easily accomplish either.

## USING EASYROLL TO LOCATE AUTO-CONFIGURATION MASTER

For either manual method of I.P. address configuration, you must access the I.P. address information of the *Auto-Configuration Master ConveyLinx-Ai2* on your network by using *EasyRoll*.

## STARTING EASYROLL APPLICATION

If you followed the default installation setting when you installed *EasyRoll*; the program should be selected from "Start – All Programs – Industrial Software – EasyRoll".  If you selected a different location when you installed; go to that location and run "EasyRoll.exe".

When you first run *EasyRoll*; you should see a window similar to this with greyed out status values and blank parameter boxes. Pressing F2 key or clicking the "Advanced Dialog" button will open the Advanced Dialog window.
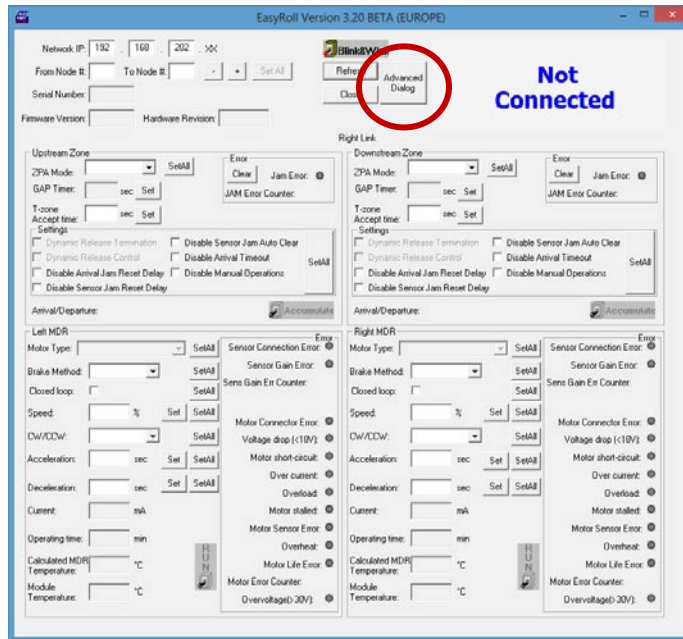
Regardless of whether you need to change your PC's I.P. address to match your already configured *ConveyLinx Subnet* or change the *ConveyLinx Subnet's Auto-Configuration Master* to match a subnet address you want to use; you have to connect to the *Auto-Configuration Master*.

### USING THE NETWORK SERVICES UTILITY

One of the features of *EasyRoll* is that it has a utility called *Network Services Discover* that allows your PC to go and find any *ConveyLinx* modules that may be physically connected to you network regardless of the I.P. address settings of your PC or the I.P. address settings of the *ConveyLinx* modules.

To access the UDP Discover Utility; you need to invoke the *ConveyLinx Advanced Dialog* window. To do this, press **F2.**



After pressing F2 or clicking the "Advanced Dialog" button on the main screen, *EasyRoll* will display the *ConveyLinx Advanced Dialog* screen. Click on the tab *Network Services.*
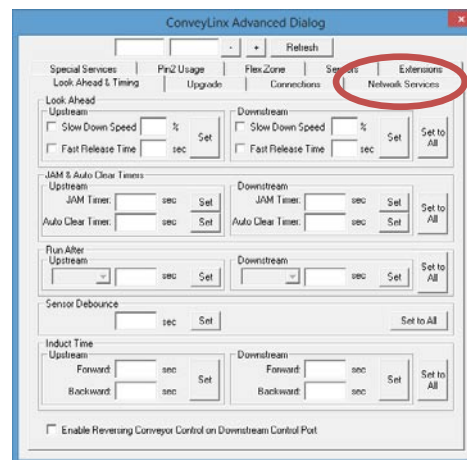
After clicking the *Network Services* tab, you will see the screen you will use to both "discover" the *ConveyLinx* modules that can be found as well as select a specific *ConveyLinx* module in which to modify its I.P. address settings.

On this screen, click the "Discover" button
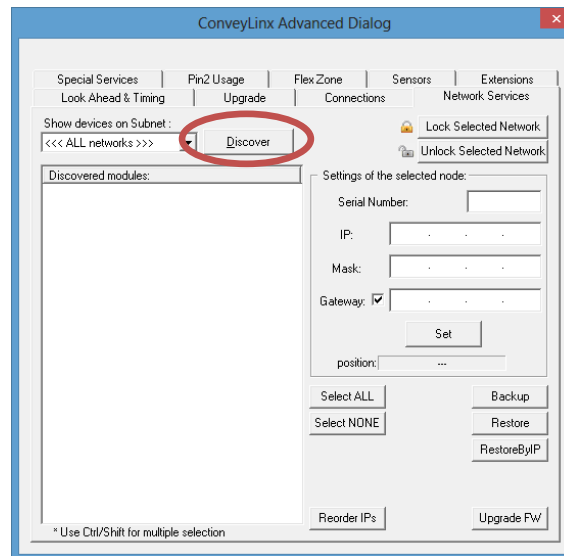
After clicking the "Discover" button, *EasyRoll* will query the network and return a list of all *ConveyLinx* modules it finds and shows each module's I.P. address, serial number and current Firmware. We already know that the *Auto-Configuration Master* is the *ConveyLinx* module with the last I.P. address octet of ".20".

In this example, 2 *ConveyLinx-Ai2s* and 2 *ConveyLinx-IO* modules were found and the *Auto-Configuration Master* is at 192.168.241.20, its serial number is 275463 and has Firmware 4.17
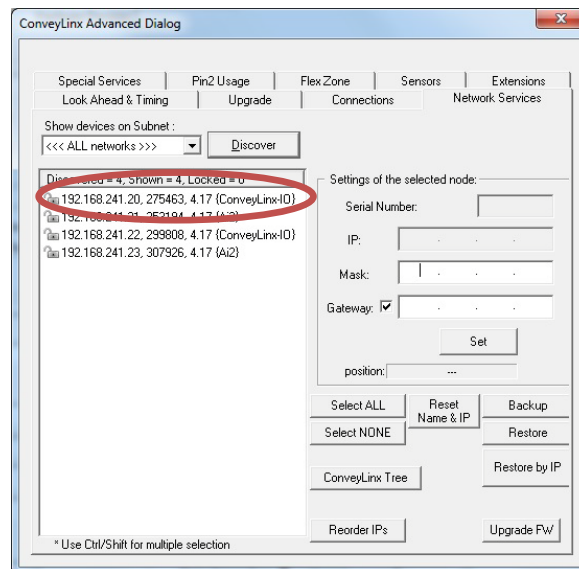
Please refer to section *ConveyLinx Advanced Dialog* on page 51 for further descriptions for the remaining *ConveyLinx Advanced Dialog* screen selection tabs.

## METHOD 2 - CHANGE PC TO MATCH AUTO-CONFIG MASTER

At this point, because you now know the *Auto-Configuration Master*'s I.P. address you can simply change your PC's IP address configuration so that is can have access to the *Auto-Configuration Master's Subnet*. In the example above, the *Auto-Configuration Master's* I.P. address is 192.168.241.20 therefore the *ConveyLinx Subnet* is 192.168.241.

METHOD 3 - CHANGE AUTO-CONFIG MASTER I.P. ADDRESS

In cases where you want to set the *Auto-Configuration Master's* I.P. address to something other than the default it used when the *Auto Configuration Procedure* was performed, you can do this from the same *Network Services* screen.



Double click the *Auto-Configuration Master* in the list.  When you do this, its I.P. address information is filled in as shown.  Simply enter in the new I.P. address information you want to use and then click the "Set" button.
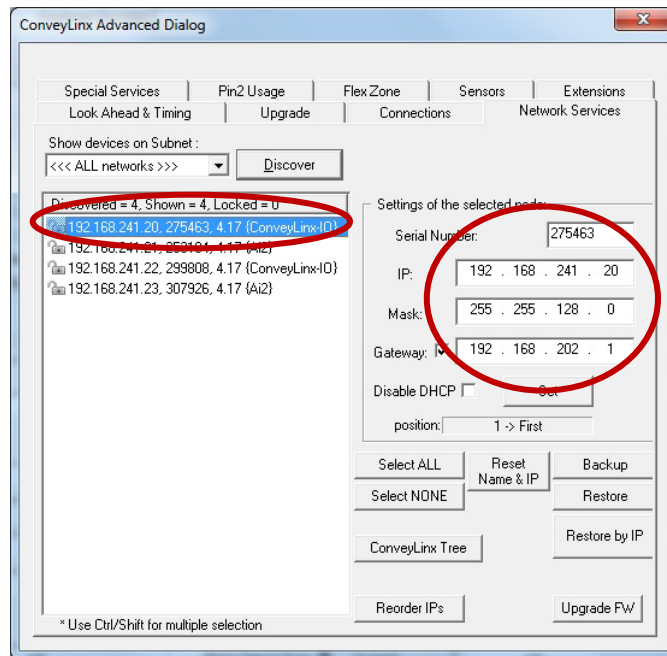
After clicking the "Set" button, you can click the "Discovering" button again and *EasyRoll* will refresh the list of modules at the left.  You can verify that the module has the new I.P. address settings.

At this point, this particular *ConveyLinx Subnet* will no longer operate because its *Auto Configuration Master's* I.P. address has been changed.  You must perform the *Auto Configuration Procedure* again so that all downstream *ConveyLinx* modules will have their I.P. address updated to match the *Auto Configuration Master's* new *Subnet.*

## EASYROLL MAIN SCREEN

Assuming you have either changed your PC's configuration or changed the *Auto Configuration Master's* configuration as described above; you should now be able to use the *EasyRoll* main screen to view your system's status and change operational parameters. If you have followed the above example, simply closing the *ConveyLinx* Advanced Dialog will show the main screen. The main screen is also shown when you first run *EasyRoll*. *Figure 28 - EasyRoll Main Screen* shows a typical main screen.



FIGURE 28 - EASYROLL MAIN SCREEN FOR CONVEYLINX-IO

The numbered items in *Figure 28 - EasyRoll Main Screen* show the basic functional areas and detailed descriptions will follow for each.

| Item | Description |
|------|-------------|
| 1 | Network IP – This is where you enter the *Subnet* of the particular *ConveyLinx* network you wish to connect |
| 2 | Node No. – This is where you enter a range of *Nodes* in which to connect.  Entering values here will cause the "Refresh" button to enable.  Clicking this button will cause the rest of the items (3, 4, and 5) to be populated. |
| 3 | ConveyLinx-IO Connection Status –. This indicates whether a remote PLC is currently connected to the module. |
| 4 | ConveyLinx-IO Transparency – See section *ConveyLinx-IO Transparency Mode* below. |
| 5 | Indicates the current on/off state of each digital input and output |

Please note that some of the detailed information shown in this figure may be different for your particular system and that most of these fields will be blank until you actually initiate communications.

## CONNECTING TO CONVEYLINX

Once the Network IP boxes (1) have been entered with the correct *Subnet*, you then type in a range of *Nodes* (2) you wish to connect; the "Refresh" button will become enabled.  Click the "Refresh" button and data for the rest of the main screen should fill in.

### NODE NAVIGATION

Whatever *Node* is entered in the first box will be the particular *ConveyLinx* data shown in the remainder of the main screen.
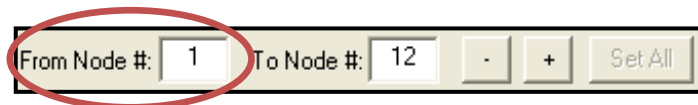


The *Node* value entered in the second box does not have to be the actual "last" node of the network.  If you enter a *Node* value higher that exists; an error message will display after you click the "Refresh" button. For the example shown; if there were only 10 *Nodes* installed and you entered 12, you would receive 2 error messages in succession after clicking the "Refresh" button.



Clicking the "+" and "-" buttons will increment / decrement the *Node* value in the first box and display the *ConveyLinx* data for the new *Node* selected.  Please note that if you increment past the value of the last physical *Node* installed, you will receive an error message.
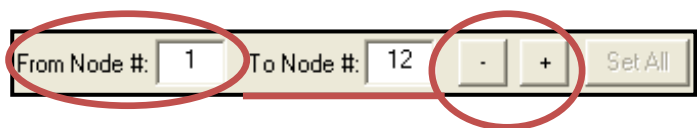
NODE IDENTIFICATION

*EasyRoll* main screen has a feature identified as "Blink & Wink" that allows you to visually verify the *Node* you have selected.

If a valid *Node* is selected in the first text box in the "Node No." area and its information is displayed on the main screen; clicking the "Blink&Wink" switch will signal the selected *ConveyLinx* module to blink on and off all of its LED indicators.  Click the "Blink&Wink" switch again to turn this off.
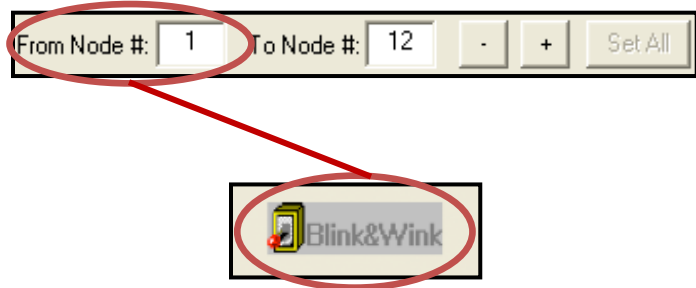
## ConveyLinx-IO Transparency Mode

*ConveyLinx* modules are most often used on MDR conveyor applications where a network of modules is controlling zones of conveyor in a ZPA fashion. Part of the Auto-Configuration procedure establishes a "connection" between each module where each module receives status from its neighbours as to the status of each other's zones. This is mechanism that provides the zone to zone flow of items on the conveyor.
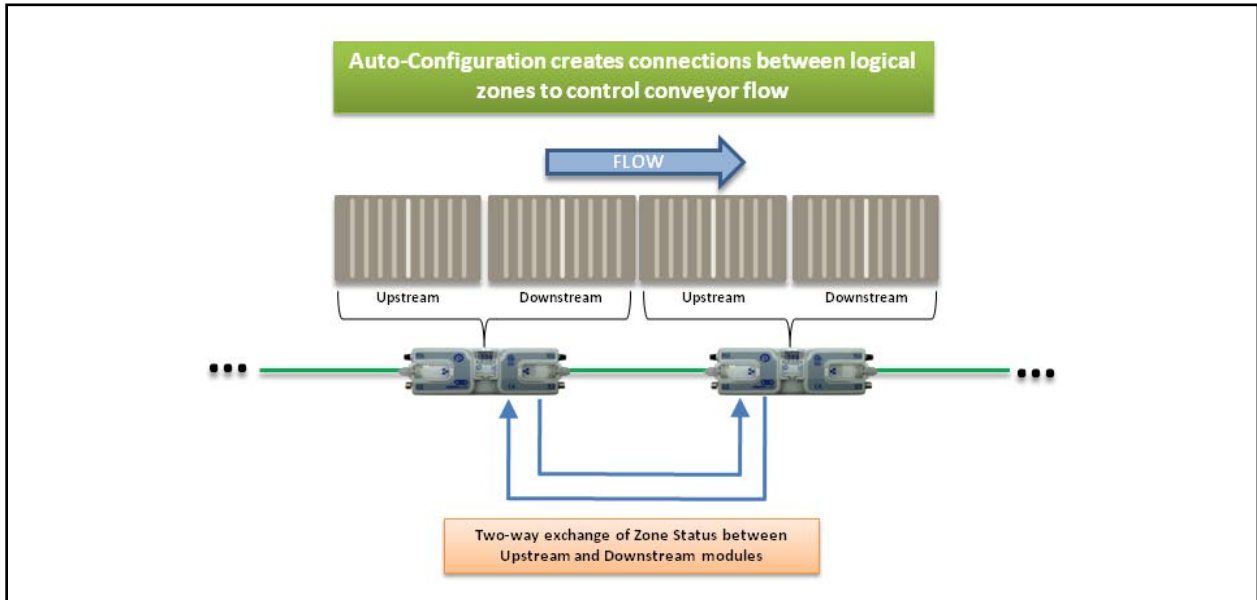


**FIGURE 29 - TYPICAL CONVEYLINX CONVEYOR FLOW**

When an application calls for additional digital I/O to be localized on such a conveyor network, it is convenient to install *ConveyLinx-IO* along the conveyor path. However, by default of the Auto-Configuration procedure, the *ConveyLinx-IO* gets configured along with its neighboring ZPA control *ConveyLinx* modules and is set up with these zone to zone flow connections. However, the zone to zone status data associated with these conveyor connections do not automatically pass through the *ConveyLinx-IO* module. The result is that conveyor flow will be interrupted. This is illustrated in Figure 30.
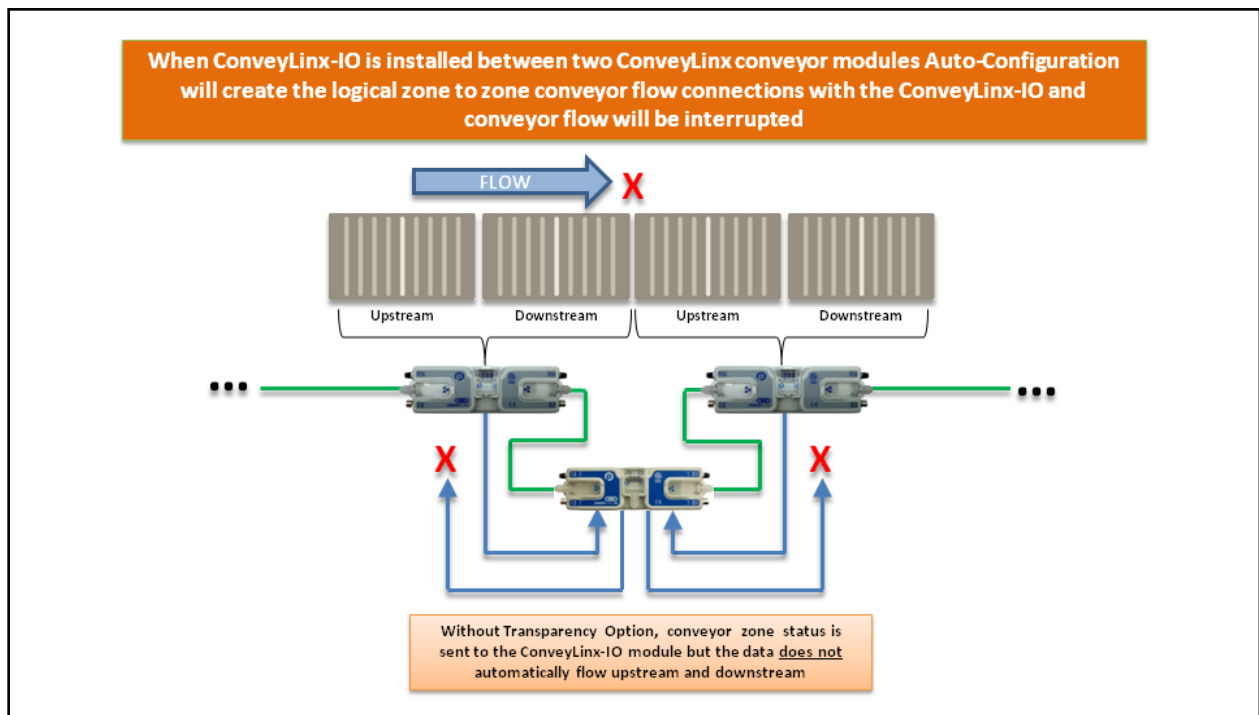
**FIGURE 30 - CONVEYLINX CONVEYOR WITH CONVEYLINX-IO CONFIGURED IN THE NETWORK**

In applications where a *ConveyLinx-IO* module is installed between two *ConveyLinx-Ai2* modules that need to maintain their logical upstream to downstream flow; the Transparency Option instructs the *ConveyLinx-IO* module to automatically pass through the conveyor flow status data between its *ConveyLinx-Ai2* neighbours. This in essence makes the *ConveyLinx-IO* "transparent" to the conveyor flow. The conveyor will operate as if the two *ConveyLinx* conveyor controllers are connected to each other as illustrated in Figure 31 .
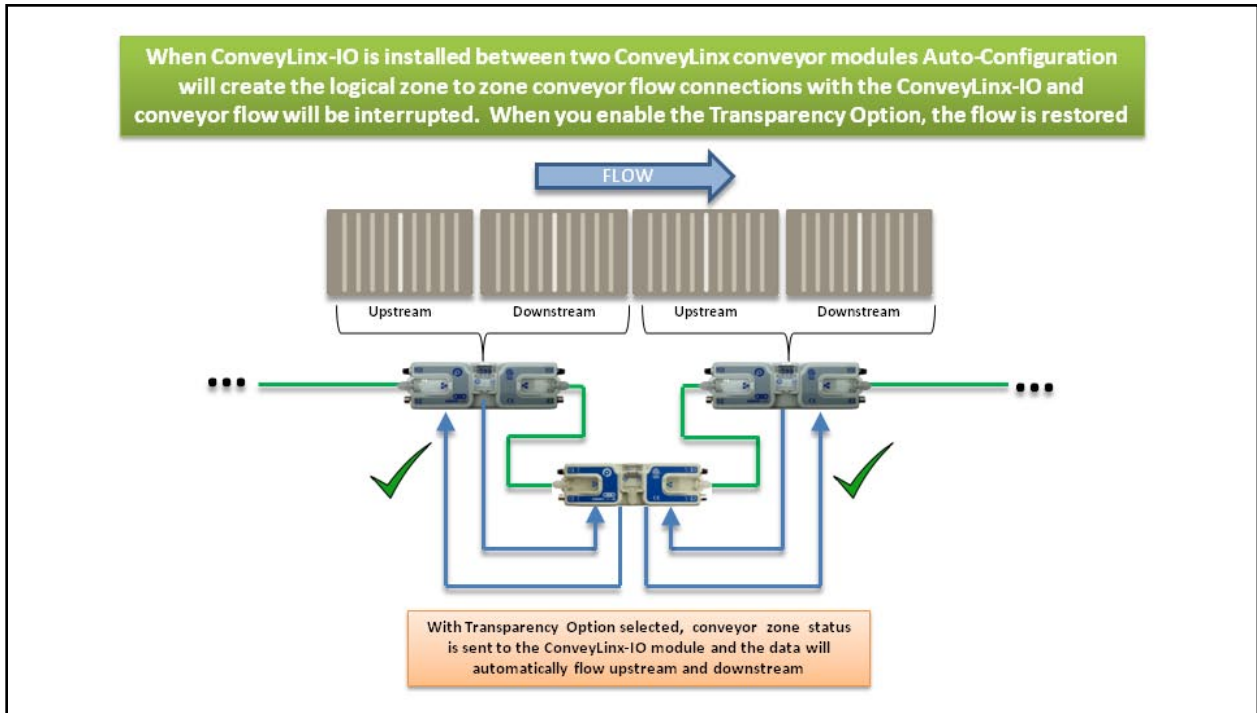
**FIGURE 31 - CONVEYLINX-IO WITH TRANSPARENCY OPTION ENABLED**

DISABLING TRANSPARENCY MODE

| | |
|---|---|
| Transparency is enabled by default during Auto-Configuration.  Disabling Transparency mode is done on the EasyRoll main screen by clicking the check box as shown |  |

## CONVEYLINX ADVANCED DIALOG

We introduced the *ConveyLinx Advanced Dialog* briefly in section *Using the Network Services Utility* on page 42. This section will define the remaining tab selections from this screen.

### INVOKING THE CONVEYLINX ADVANCED DIALOG

To invoke the *ConveyLinx Advanced Dialog*, do any one of the following:

- Press F2 keyboard key
- Press [Ctrl][Shift] U keys
- Click the "Advanced Dialog" button at the top of the Main Screen

Whatever *Node* is entered in the first box when you open the Advanced Dialog will be the particular *ConveyLinx Node* data in context for the *ConveyLinx Advanced Dialog* tab selections when the dialog screen pops up.

Because *EasyRoll* is used for the entire *ConveyLinx* family of modules, there will be several tabs shown in the *Advanced Dialog* screen that are not applicable to the *ConveyLinx-IO* modules.  These will be blank or otherwise disabled when you are connected to a *ConveyLinx-IO* module.

### UPGRADE TAB

The *Upgrade* tab screen gives access to the *ConveyLinx* update firmware utility.  Over time, enhancements and features may be added to the *ConveyLinx* family of products.  These features and enhancements are typically made available to customers in the form of firmware upgrade files that need to be uploaded to your *ConveyLinx* modules.

A firmware upgrade will be in the form of a data file sent to you or made available for download.  The *Upgrade* utility allows you to browse for this data file and then select a single *Node* or group of *Nodes* to upload.

Upon selecting the *Upgrade* screen tab, *EasyRoll* fills in the I.P. address of the range of *Nodes* entered on the main screen. Click the "Browse" button to open a file selection dialog window.
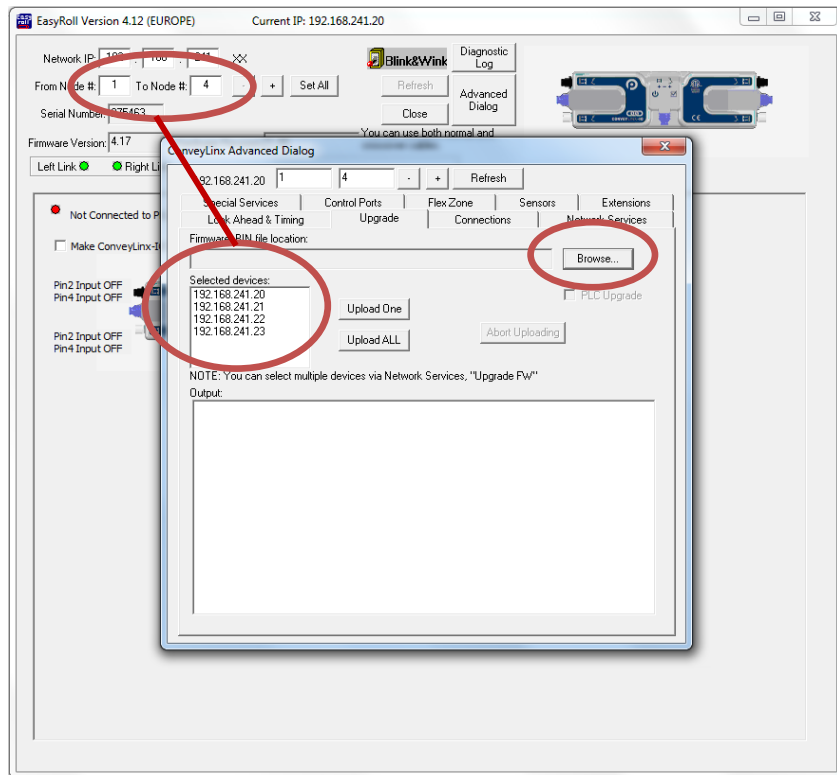


With the "Open" dialog displayed, navigate to the location on your PC where you placed the firmware upgrade file you received. Select the file and click "Open".
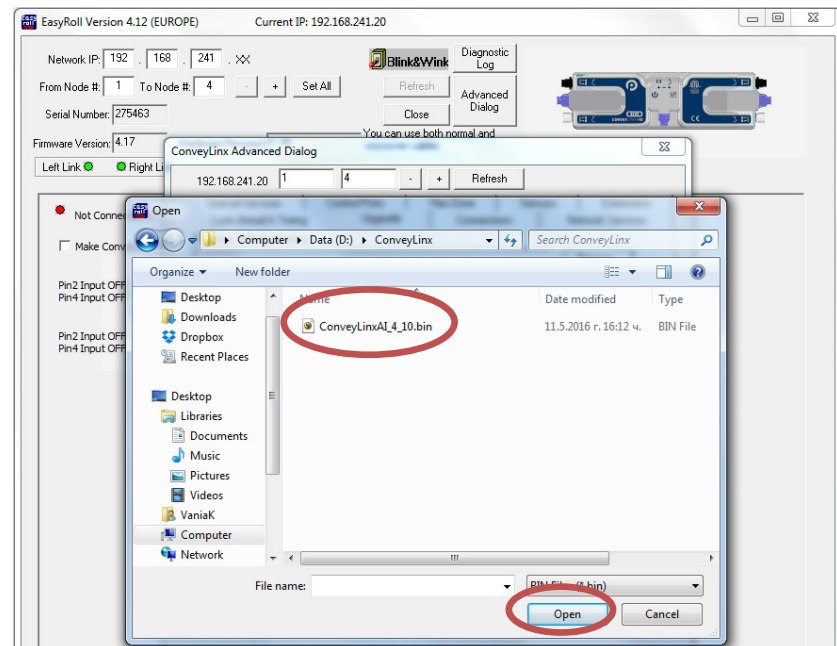
In this example, we clicked "Upload ALL" so the selected firmware upgrade file will be sent to all 4 *Nodes*.  The "Output" window will update the progress of the file uploading process. The time it takes for this process will vary depending upon how many *Nodes* are being uploaded.
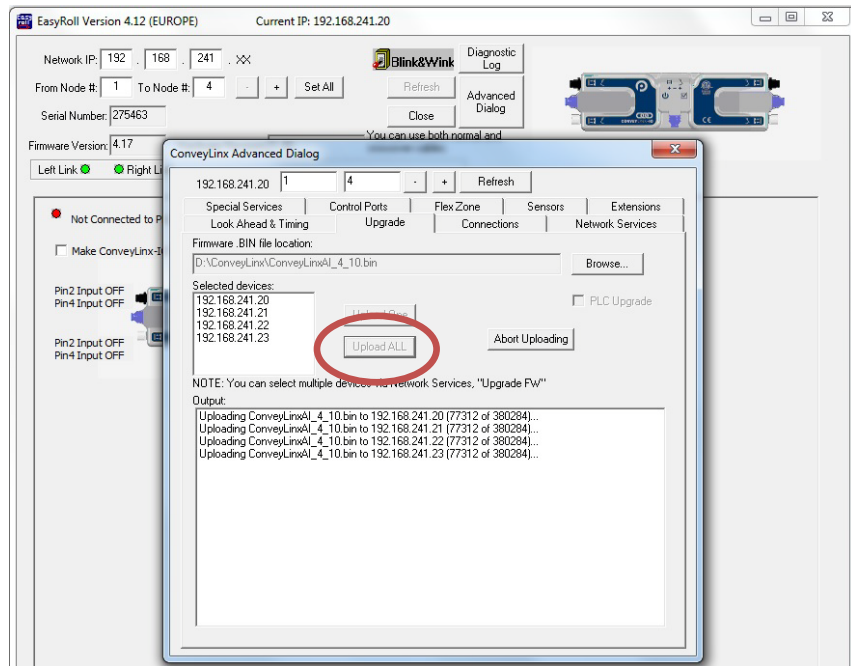


When all *Nodes* report back to the "Output" window with a status of *Done*; then the upload is complete and you can close the *ConveyLinx Advanced Dialog* window.
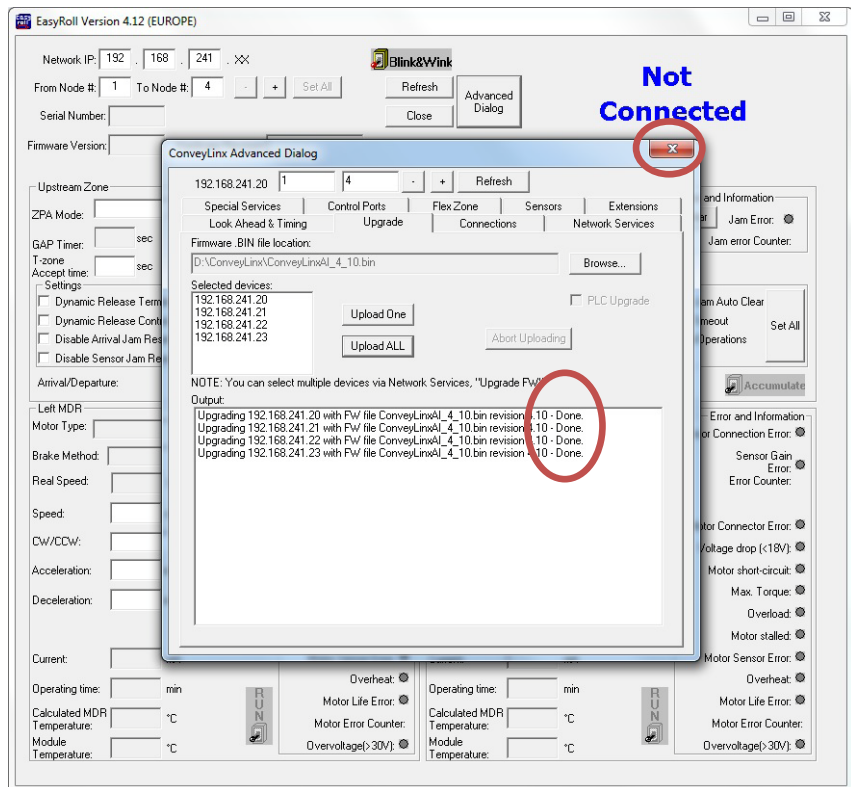
CONNECTIONS TAB

The *Connections* utility uses *EasyRoll* to instruct a given *ConveyLinx* module to make a logical connection to another *ConveyLinx* module that it otherwise would not have made during the *Auto-Configuration Procedure*. For applications where you have more than one *ConveyLinx Subnet*, this would be the way to logically connect the most downstream *Node* of one *Subnet* to the most upstream *Node* of another *Subnet*.

NETWORK SERVICES

The *Network Services* screen is used to both explore any networks reachable by the PC for any *ConveyLinx-Ai2* modules and it is used to set the I.P. address of a selected *ConveyLinx-Ai2*. Refer to section *Using the Network Services Utility* on page 42 for details.

## CONNECTING TO A REMOTE PLC CONTROLLER

*ConveyLinx-IO* is a device that requires a remote PLC controller to connect and transfer data to control its digital I/O points.  This section describes the network architecture utilized for the module and provides the data mapping details to access I/O for the 3 main protocols that the module recognizes.

### CONVEYLINX NETWORK ARCHITECTURE

Each module communicates to its adjacent modules and to any connected PC or PLC via Ethernet physical media. *ConveyLinx* modules recognize (3) TCP/IP based protocols: Modbus TCP, Profinet I/O, and EtherNet/IP.  Modbus TCP is the "native" protocol for communications between *ConveyLinx* modules and the *EasyRoll* PC software. When *ConveyLinx* modules create connections between them, they utilize Modbus TCP for inter-module communication.  EtherNet/IP is recognized by *ConveyLinx* modules and any given module can be attached to an EtherNet/IP capable PLC (Allen-Bradley ControlLogix or CompactLogix platforms for example) and be recognized as a "Generic Ethernet Module" or explicitly as a module by using Insight Automation provided EDS file.  Profinet I/O is recognized by the modules and can be attached to any Profinet I/O capable PLC (applicable Siemens platforms). All protocols access the same internal data locations on a given module.

Each *ConveyLinx-IO* module's internal data structure is arranged and addressed as Modbus Holding registers.  The on-board communication and control processes attach logical meanings to each holding register and read and write data to specific registers to initiate and/or react to events.  Certain registers contain information as to how the module is configured (Inputs, Outputs, polarity etc.) for its local controls.

### UNDERSTANDING ASSEMBLIES

The holding registers described above we will define as **Module Register Addresses** and are used for the function of the module regardless of if any remote PLC or PC is connected.  There are many *Module Register Addresses* that are used for the operation of the module that are not applicable, required, or should even be made available to a remote PLC.  When connecting to a PLC, the module needs to gather certain specific *Module Register Addresses* located scattered throughout the entire listing of *Module Register Addresses* into a concise contiguous group or **Assembly** that the PLC can efficiently read from and write to.  Figure 32 illustrates this concept.
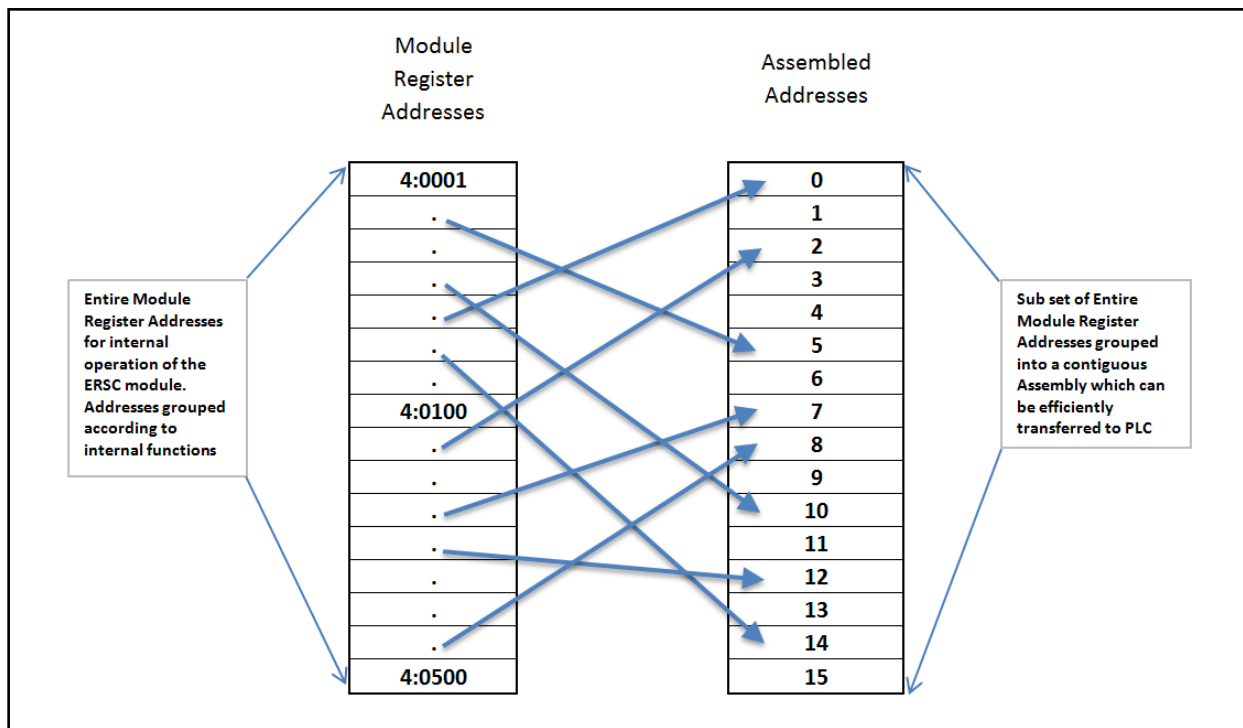
**FIGURE 32 - PLC ASSEMBLY CONCEPT**

So, no matter which of the three connection protocols you use (Modbus TCP, EIP, or Profinet I/O), for any given assembly, the internal data you are accessing with your PLC is the same.

## MODBUS ASSEMBLY INSTANCE STRUCTURE

Each *ConveyLinx-IO* utilizes Modbus register architecture for remote data access over Ethernet.  Modbus TCP is a simple protocol for data exchange based upon a query/response mechanism.  Each *module*'s memory structure contains a fixed array of internal data locations that are constructed as Modbus *Holding Registers*.  Each *module* has a fixed reserve of *Holding Registers* with each capable of holding a 16-bit numerical integer value.  Modbus TCP protocol provides for read/write access to any available *Holding Register*.  The structure of these registers allows for individual *module*'s to read from and write to specific register address locations to achieve inter-module communications.  Certain registers are read from and written to by the *EasyRoll* software in order to monitor and/or change default configuration values.

Modbus TCP addressing convention utilizes a "4:xxxx" notation.  The "4:" in Modbus protocol designates that the address is a *Holding Register* and the xxxx is a numerical value representing the offset or index for a specific location.  The "xxxx" values used in this document are to be interpreted as if they are for a Modbus PLC which means that the first register address is "4:0001" and that there is no "4:0000" register.  Some PLC data structures and PC development environments utilize the "4:0000" designation and their indexes will be offset by 1.  Please refer to your PLC or PC application documentation for the Modbus convention used on their platforms.

Any Modbus TCP capable PC or PLC can connect to any *ConveyLinx-IO* visible on its network and access Input and Output *Holding Register Assemblies*.  *ConveyLinx* supports the following Modbus TCP Service Codes:

- Service Code 3 - Read Holding Register (up to 45 registers per instruction)
- Service Code 6 - Write Single Register
- Service Code 16 - Write Multiple Registers (up to 45 registers per instruction)
- Service Code 23 - Read/Write Multiple Registers (up to 45 registers per instruction)

**When using Input and Output Assemblies with Modbus TCP PLC, it is important to always use the first address shown in the assembly group as the beginning register to read or write regardless of which register in the assembly is needed by the PLC.  Trying to access Input or Output Assemblies starting with any register in the assembly other than the first register will cause the module to return an error.**

**For more information and open protocol specification, please visit www.modbus.org**

## ETHERNET I/P ASSEMBLY INSTANCE STRUCTURE

When a *ConveyLinx-IO* module is attached to an external Ethernet I/P controller (Logix 5000 based PLC, for example), it is done so as a Generic Ethernet I/O device.

*For details on connecting to Rockwell PLC – see Appendix B – Connecting to Rockwell Ethernet I/P PLC.*

Part of this procedure in the PLC is to instruct the Generic device as to which data configuration or instance of Ethernet I/P the Generic device is to use to report and respond to data to and from the PLC.

**From this point forward, it is assumed the reader is familiar with Allen-Bradley Logix platform addressing notation:**

**[*ModuleName*]:O.Data[*Index*].*Bit***
**[*ModuleName*]:I.Data[*Index*].*Bit***
**[*ModuleName*]:C.Data[*Index*].*Bit***

**Where:**
- ***ModuleName* is the user-defined name of the device**
- **"O.Data" indicates data written by the PLC to the device**
- **"I.Data" indicates data read by the PLC from the device**
- **"C.Data" indicates configuration data sent to the device at initial connection**
- **"[*Index*].*Bit*" indicates the word and bit within the image.  If the bit notation is absent the notation refers to the entire word data type**

These instances essentially group the appropriate Modbus registers into contiguous Input and Output array image that fit into the Allen-Bradley Logix 5000 controller tags.

> *Please refer to* <u>Connecting ConveyLinx to Rockwell PLC's</u> *–*
> *Insight Automation publication ERSC-1520 for details on how to*
> *connect ConveyLinx to Rockwell Ethernet I/P capable PLCs.*

## PROFINET IO ASSEMBLY INSTANCE STRUCTURE

Prior to connecting any *modules* to a Profinet IO capable (Siemens, for example) PLC, a *GSDXML* configuration file needs to be installed in your PLC programming software (i.e. STEP 7 or TIA Portal). The *GSDXML* file contains, among other things, the definition of memory size required when connecting a *module* to a Siemens PLC with Profinet IO. When assigning an individual *module* to a PLC, you decide what memory address to use. The *GSDXML* file stipulates that each module requires 64 bytes of input data from the *module* to the PLC and 64 bytes of output data from the PLC to the *module*. When assigning an individual *module* to a PLC, you decide what physical PLC memory address to use as the beginning of these 64 byte blocks.

In all of the charts showing *module* data registers and their assigned function, for Profinet IO these are indicated as "Byte 0, Byte 1,...Byte 32", etc. These are offsets from the starting address in the Siemens PLC memory assigned by the Siemens programming software when the *module* was installed into the network. For example, if a *module's* inputs are installed beginning at %IB256, then from our charts, the data for "Byte 6 and Byte 7" would be at addresses %IB262 and %IB263 respectively.

> *Please refer to* <u>Connecting ConveyLinx to Profinet PLCs</u> *– Insight*
> *Automation publication ERSC-1525 for details on how to connect*
> *ConveyLinx to Profinet I/O capable PLCs.*

## ConveyLinx IO Configuration Data

The configuration data is sent from the controller to the module upon initial connection in order to set optional module operational parameters.

### Register Chart Legend

M: 4:1502 → Indicates Modbus TCP Addressing Notation

E: I.Data[2] → Indicates Ethernet I/P Addressing Notation

### Configuration Assembly

This chart shows the data sent to the PLC from the module when connected. All registers in the module are 16-bit integer data type.

| Register Name / Module Address | Assembled Address for PLC | Description |
|---|---|---|
| Input/Output 4:0497 | M: 4:13800 E: C.Data[0] | **Input/Output Configuration**<br>1 = set Pin as Output<br>0 = set Pin as Input<br><br>bit 0 : Left Male Pin2<br>bit 1 : Left Female Pin2<br>bit 2 : Right Male Pin2<br>bit 3 : Right Female Pin2<br>bit 4 : Left Male Pin4<br>bit 5 : Left Female Pin4<br>bit 6 : Right Male Pin4<br>bit 7 : Right Female Pin4<br>bit 8 to bit 15 - reserved |
| Input Polarity 4:0034 | M: 4:13801 E: I.Data[1] | **Inputs Polarity Configuration**<br>0 = Input signal and logic states equal<br>1 = Invert logic from signal state<br><br>bit 0 : Left Male Pin2<br>bit 1 : Left Female Pin2<br>bit 2 : Right Male Pin2<br>bit 3 : Right Female Pin2<br>bit 4 : Left Male Pin4<br>bit 5 : Left Female Pin4<br>bit 6 : Right Male Pin4<br>bit 7 : Right Female Pin4<br>bit 8 to bit 15 - reserved<br><br>See Note ① |

| Register Name / Module Address | Assembled Address for PLC | Description |
|---|---|---|
| Output Polarity<br>4:0038 | M: 4:13802<br>E: I.Data[2] | **Outputs Polarity Configuration**<br>0 = Output signal and logic states equal<br>1 = Invert logic from signal state<br><br>bit 0 : Left Male Pin2<br>bit 1 : Left Female Pin2<br>bit 2 : Right Male Pin2<br>bit 3 : Right Female Pin2<br>bit 4 : Left Male Pin4<br>bit 5 : Left Female Pin4<br>bit 6 : Right Male Pin4<br>bit 7 : Right Female Pin4<br>bit 8 to bit 15 - reserved<br><br>See Note ② **Error! Reference source not found.** |
| PushPull<br>4:0496 | M: 4:13803<br>E: I.Data[3] | **Push/Pull Inputs Configuration**<br>0 = Input circuit Not Push-Pull type<br>1 = Input circuit is Push-Pull type<br><br>bit 0 : Left Male Pin2<br>bit 1 : Left Female Pin2<br>bit 2 : Right Male Pin2<br>bit 3 : Right Female Pin2<br>bit 4 : Left Male Pin4<br>bit 5 : Left Female Pin4<br>bit 6 : Right Male Pin4<br>bit 7 : Right Female Pin4<br>bit 8 to bit 15 - reserved<br><br>See Note ③ |
| State On Disconnect<br>4:0255 | M: 4:13804<br>E: I.Data[4] | **Output State on Disconnect Configuration**<br>0 = Output Pin de-energizes upon communication loss with remote controller<br>1 = Output Pin remains in last state upon communication loss with remote controller<br><br>bit 0 : Left Male Pin2<br>bit 1 : Left Female Pin2<br>bit 2 : Right Male Pin2<br>bit 3 : Right Female Pin2<br>bit 4 : Left Male Pin4<br>bit 5 : Left Female Pin4<br>bit 6 : Right Male Pin4<br>bit 7 : Right Female Pin4<br>bit 8 to bit 15 - reserved<br><br>See Note ④ |

NOTE ①

This register is only used for Pins configured as Inputs in the *Input/Output* register.  If a given Pin's corresponding bit is 0 in this register, the logical state of the Pin's data seen in the Inputs register of the Inputs Assembly will match the logical state of the electrical signal on the Pin.  If a given Pin's corresponding bit is 1 in this register, the

logical state of the Pin's data seen in the *Inputs* register of the Inputs Assembly will be inverted from the logical state of the electrical signal on the Pin.

This is provided as a convenience for the remote controller programmer.  For example, the remote controller could have re-useable code or routines that expect a photo sensor to have its electrical signal ON when it is "blocked".  Let's say on a given module, you need to connect a photo sensor whose electrical signal is OFF when "blocked".  Instead of modifying the program logic in each and every place this input is used; you can simply change its logical polarity with this register

Please note that this register also controls the state of the LED indicators for the given Pin.  The LED state will match the logical state of the pin.

For example, by default when the Left Sensor input is electrically energized, its corresponding LED illuminates green and bit 4 of the *Inputs* register in the Inputs Assembly is a 1.  If you set bit 4 in the *Input Polarity* register; when the Left Sensor input is electrically energized, bit 4 of the *Inputs* register is 0 and its corresponding LED is off.

NOTE ②

This register is only used for Pins configured as Outputs in the *Input/Output* register.  If a given Pin's corresponding bit is 0 in this register when the remote controller sets the logical state of the Pin in the Outputs register in the Output Assembly, the electrical signal on the Pin will match this logical state.  If a given Pin's corresponding bit is 1 in this register, the electrical signal on the Pin will be inverted from the logical state.

NOTE ③

This register is only used for Pins configured as Inputs.

NOTE ④

This register is only valid for Pins configured as Outputs

**Upon power-up, all I/O Pins default to being inputs.  Because these input pins are NPN/PNP auto-sensing, there is a voltage (approx. 12V) on the circuit that will produce up to a 2 mA leakage current.  When using a given Pin as an output, there may be a time delay between module boot-up and the remote controller configuring the Pin as an output.  During this time delay, if you have a device connected to the Pin, it may receive the auto-detect circuit's voltage and leakage current.**

## INPUT ASSEMBLY (CONVEYLINX-IO ➔ PLC)

| Register Name / Module Address | Assembled Address for PLC | Description |
|---|---|---|
| Inputs 4:0035 | M: 4:13500 E: I.Data[0] | Bitwise Value - Read only<br><br>bit 00 = Left Male Pin 2<br>bit 01 = Left Female Pin 2<br>bit 02 = Right Male Pin 2<br>bit 03 = Right Female Pin 2<br>bit 04 = Left Male Pin 4<br>bit 05 = Left Female Pin 4<br>bit 06 = Right Male Pin 4<br>bit 07 = Right Female Pin 4<br>bit 08 = Error Left Male Pin 2<br>bit 09 = Error Left Female Pin 2<br>bit 10 = Error Right Male Pin 2<br>bit 11 = Error Right Female Pin 2<br>bit 12 = Error Left Male Pin 4<br>bit 13 = Error Left Female Pin 4<br>bit 14 = Error Right Male Pin 4<br>bit 15 = Error Right Female Pin 4<br><br>See Note ⑤ |

### NOTE ⑤

Bits 8 through 15 indicate error conditions for each port pin when they are configured as outputs. If any of these bits are set, it indicates that there is an over-current or short circuit detected on its respective port's power supply.

## OUTPUT ASSEMBLY (PLC ➔ CONVEYLINX-IO)

| Register Name / Module Address | Assembled Address for PLC | Description |
|---|---|---|
| Outputs 4:0037 | M: 4:13600 E: O.Data[0] | Set bit to energize Output<br><br>bit 0 : Left Male Pin2<br>bit 1 : Left Female Pin2<br>bit 2 : Right Male Pin2<br>bit 3 : Right Female Pin2<br>bit 4 : Left Male Pin4<br>bit 5 : Left Female Pin4<br>bit 6 : Right Male Pin4<br>bit 7 : Right Female Pin4<br>bit 8 to bit 15 - reserved<br><br>These bits are only recognized for a given port pin if it has been configured as an Output |

### NOTE ⑥

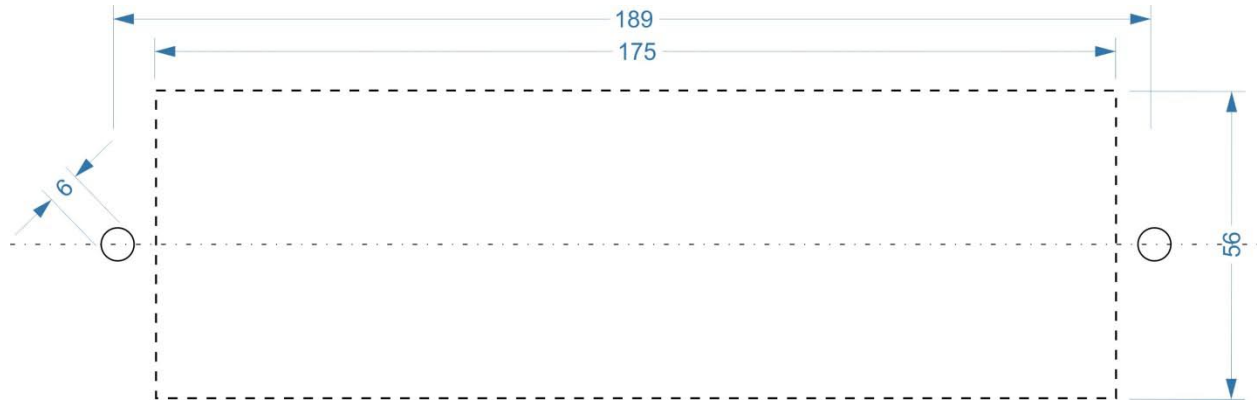These bits are only recognized for a given port Pin if it has been configured as an Output

## APPENDIX A – MODULE SPECIFICATIONS

### MODULE DIMENSIONS

Dimensions in mm
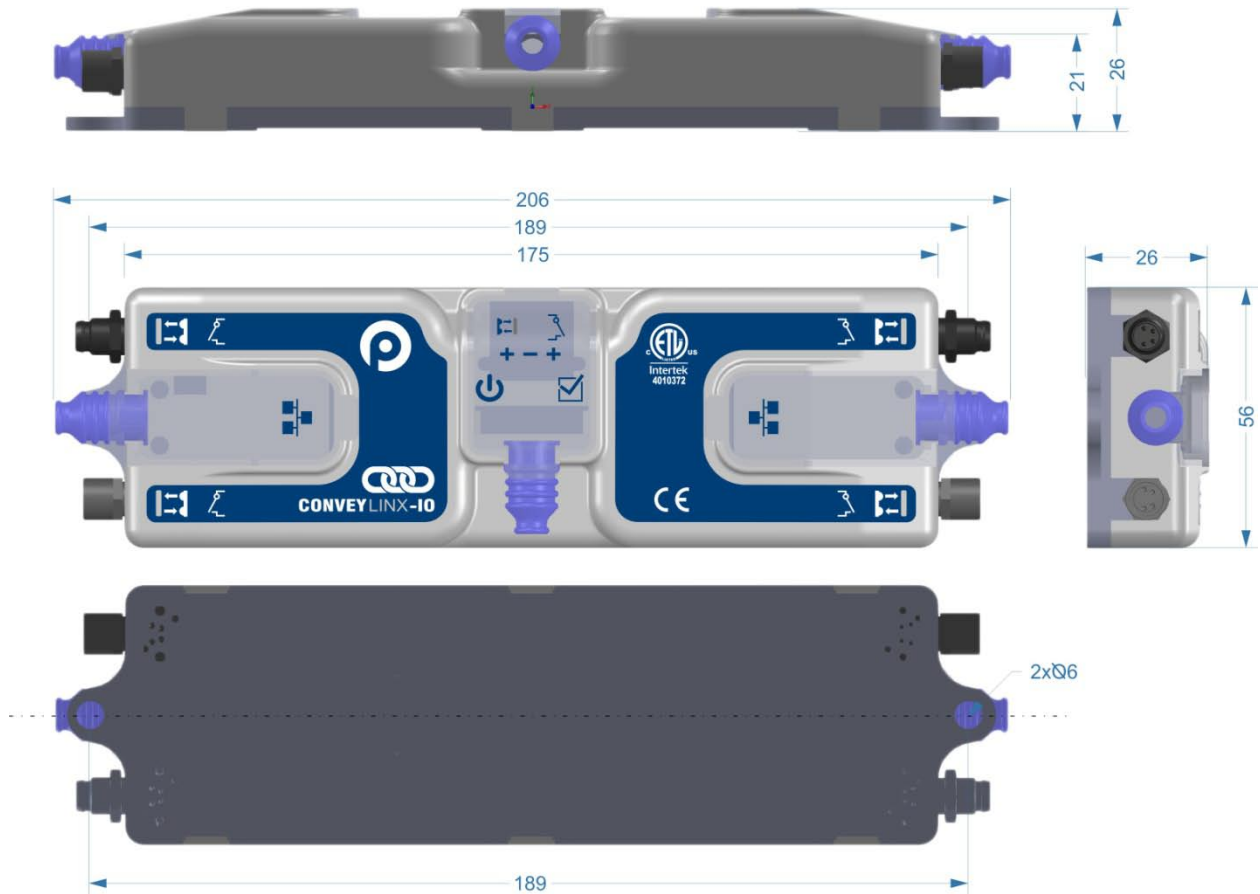
## MOUNTING DIMENSIONS



## ENVIRONMENTAL AND ELECTRICAL

| | |
|---|---|
| **Power supply Voltage** | 24.0V +/- 10 % |
| **Standby current consumption** | less than 70mA |
| **Minimum Operating Voltage** | 21V |
| **Maximum Operating Voltage** | 30V |
| **Storage temperature** | -40C to 120C(-40F to 248F) |
| **Ambient Operating temperature** | 0C to 50C(32F to 122F) |
| **Humidity** | 5% to 95% non-condensing |
| **Vibration** | 0.152 mm (0.006 in.) displacement, 1G peak |
| **Mechanical Shock** | 20G peak for 10ms duration (1.0 ms) |
| **Enclosure IP Rating** | IP54 |
| **Maximum peak current** | 2.4A |

## INPUTS

| Inputs | • PNP/NPN Auto-sensing (Default)<br>• Up to 8 per module - 2 inputs per port<br>• Programmable Push-Pull selectable |
|---|---|
| **Minimum ON current** | 1.5mA |
| **Maximum OFF current** | 0.4mA |
| **Short Circuit protection( per side)** | 500mA max |

Sensor port inputs are auto-sensing for the connected circuit type of either PNP or NPN. Please note that both sourcing and sinking current will activate the input.

Each port utilizes load detection circuits between the +24V (Pin 1) and Gnd (Pin 3). This circuit provides input to the processor to detect if a power consuming device is connected. Short circuit between Pin 1 and Pin 3 can cause damage to the load detection circuit and render this function inoperable.

## OUTPUTS

| **Outputs** | PNP only |
|---|---|
| **Output count** | Up to 8 per module |
| **Maximum current ( per Output)** | 300mA |
| **Short-circuit protection threshold( per Output** | 500mA |

## ETHERNET SPECIFICATION

- 3 port integrated switch ( 2 external ports and 1 port for the on-board processor)
- Automatic speed setup (10Base-T / 100Base-TX)
- Automatic duplex configuration (Full / Half)
- Automatic straight/crossover cable detection ( Auto MDI/MDI-X)
- PAUSE frame support
- Back pressure flow control support
- Maximum segment length: 100m / 328ft

## SUPPORTED NETWORK PROTOCOLS

- Modbus/TCP
- EtherNet/IP
- Profinet I/O

## APPENDIX B – CONNECTING TO ROCKWELL ETHERNET I/P PLC

In RSLogix-5000, when you add a new module in your Ethernet tree, you select "Generic Ethernet Module" from the "Communication" category as shown below in Figure 33:
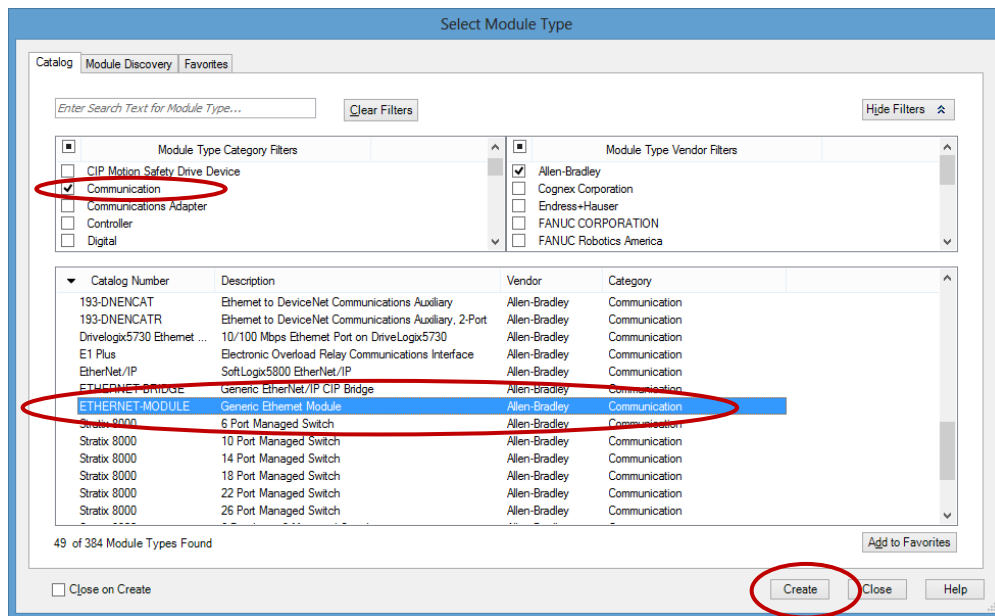


**FIGURE 33 - RSLOGIX 5000 SELECT MODULE TYPE SCREEN**

When you click "Create", you get the new module configuration window as shown below in Figure 34 . Please note that the Name and IP address are for example only. The Comm Format and Connection Parameters must be entered as shown.
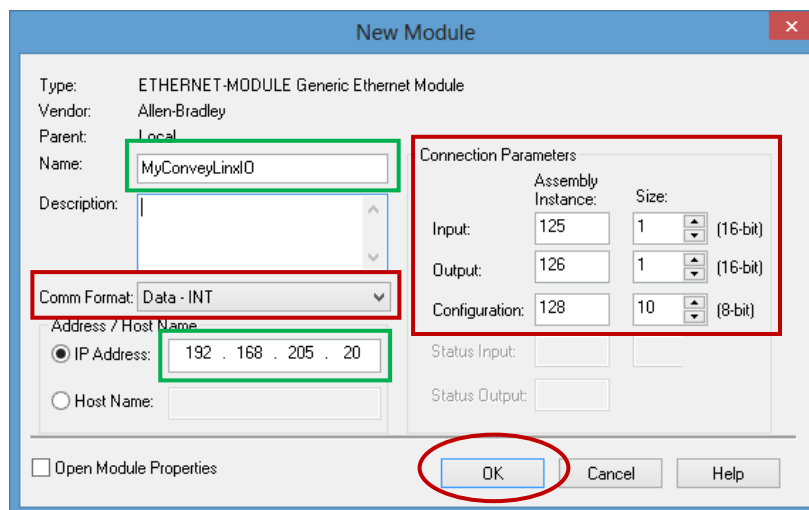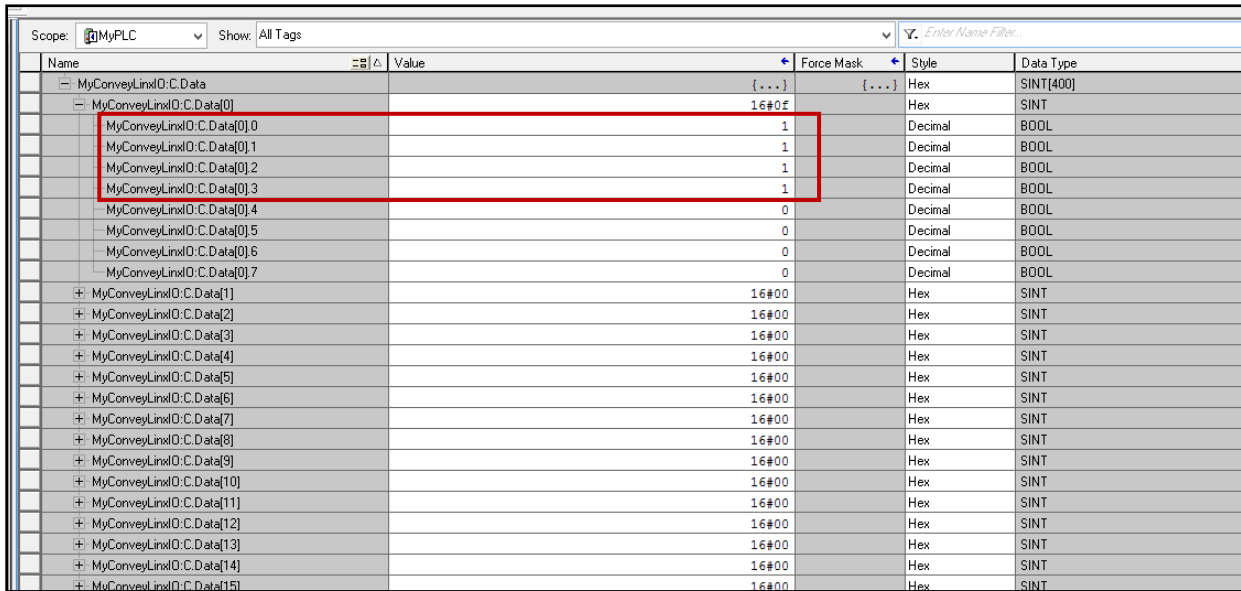


**FIGURE 34 - RSLOGIX 5000 NEW MODULE CONFIGURATION SCREEN**

Once you have created and configured the new module, you can see the data tags created your Controller Tags view as shown below in Figure 35.  For example, if you populate bits 0 thru 3 in C.Data[0] with a "1"; by referencing the chart in section *Configuration Assembly* on page 59, this data configures Pin 2 on each Port to be a digital output.



**FIGURE 35 - CONTROLLER TAG VIEW SHOWING CONFIGURATION DATA**

Please note that RSLogix 5000, by default, creates an array tag of "SINT[400]" for the Configuration Assembly for any Generic Ethernet Module.  When the program is actually downloaded to the PLC processor and begins communications with *ConveyLinx-IO*; the data exchanged will be the quantity of bytes entered when the module was created.  For *ConveyLinx-IO* this is Configuration Assembly Instance 128 with a size of 10 bytes as shown in Figure 34.

## APPENDIX C – CONNECTING TO SIEMENS PROFINET-IO PLC

ConveyLinx-IO supports PROFINET I/O and can be used as a remote I/O module to any Siemens Profinet I/O capable PLC. ConveyLinx-IO only exchanges 2 input bytes and 2 output bytes with the PLC so it uses very little of the PLC's available I/O memory.

> *Connecting ConveyLinx-IO to Siemens PLC requires a GSDXML file to be installed into your Siemens TIA Portal programming environment. Please refer to www.pulseroller.com for downloading information.*
>
> *Please refer to* Connecting ConveyLinx to Profinet PLCs *– Insight Automation publication ERSC-1525 for details on how to connect ConveyLinx to Profinet I/O capable PLCs.*

Once you have the appropriate GDSXML file installed into your TIA Portal programming environment, *ConveyLinx-IO* will support Topological discovery and Media redundancy. The configuration parameters (i.e. the equivalent parameters shown in the chart in section *Configuration Assembly* on page 59) can be found in the "Properties" of the device under "Module parameters" tab. The configuration data bits have been segmented into simple Boolean check box selections as shown below in Figure 36 .
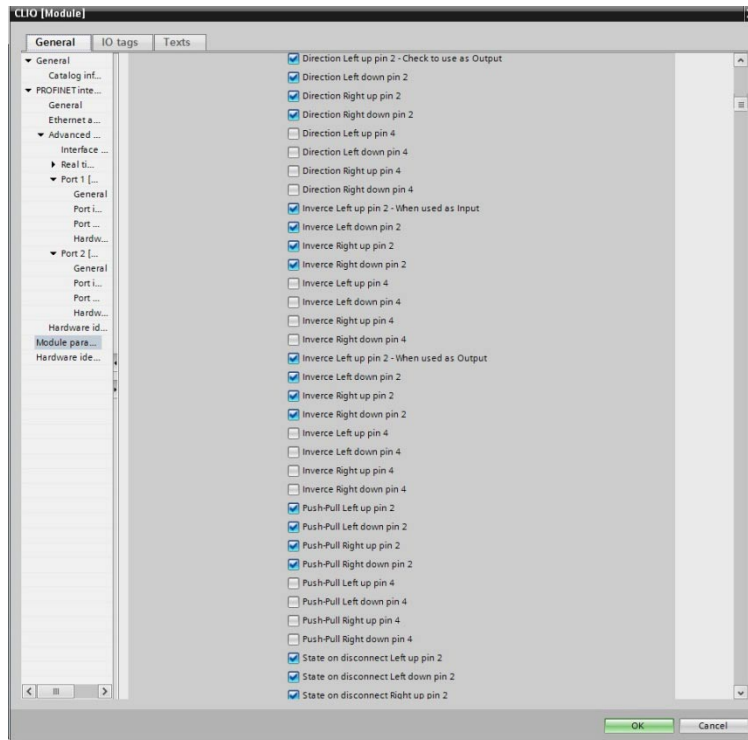


**FIGURE 36 - SIEMENS TIA PORTAL MODULE CONFIGURATION PARAMETER EXAMPLE**

# APPENDIX D – UNDERSTANDING IP ADDRESSES AND SUBNETS

## CONVEYLINX, IP ADDRESSES, AND SUBNETS

In order to connect to a *ConveyLinx* network and/or utilize and manage a multiple subnet *ConveyLinx* conveyor installation; a certain level of Ethernet I.P. addressing knowledge is required.  This reference provides some background information and a quick guide for setting up your PC to be able to take full advantage of *ConveyLinx* and *EasyRoll*.

Your PC's I.P. address is used by an Ethernet network to identify the PC on a network.  An I.P. address is constructed of 4 numbers or *octets*.  Each of the numbers can be a value from 0 to 255.  The format of an I.P. address is:

<p align="center">AAA.BBB.CCC.DDD</p>

Where AAA, BBB, CCC, and DDD can theoretically be any values from 0 to 255 each.  For any given network, this I.P. address is unique for each PC on the network.  The AAA value identifies the *Class* of the network and is most relevant to I.T. professionals and other entities such as internet providers, etc.  For our purposes, we will use a *Class C* type network which uses the value *192* for AAA.  For the BBB value we will use *168*.  The *192.168* value for the first 2 octets of our I.P. address is the most common for user configurable networks.  The values AAA.BBB.CCC together identify the *Subnet* that the PC will be connected.  The *Subnet* can be thought of as a group of PC's or *ConveyLinx* modules that can all communicate directly with each other.  For example, if a PC's I.P. *Subnet* (AAA.BBB.CCC) address is 192.168.0; then any other PC or device on the same network who's *Subnet* is equal to 192.168.0 can communicate with each other.  In this case, our network can have up to 256 devices because the DDD octet has to be in the range of 0 to 255 and each complete I.P. address has to be unique.  Any other PC's or *ConveyLinx* modules on our network who's *Subnet* does not equal 192.168.0 will not be able to communicate with each other.

In order to allow your PC to communicate with more than 256 possible address on its network; your PC's I.P. address configuration also uses another 4 octet value known as the *Subnet Mask*.  This value allows your PC to see other *Subnets* on the same network.

The following figure shows some typical values for *Subnet Mask* and the resulting number of *Subnets* that can be addressed:
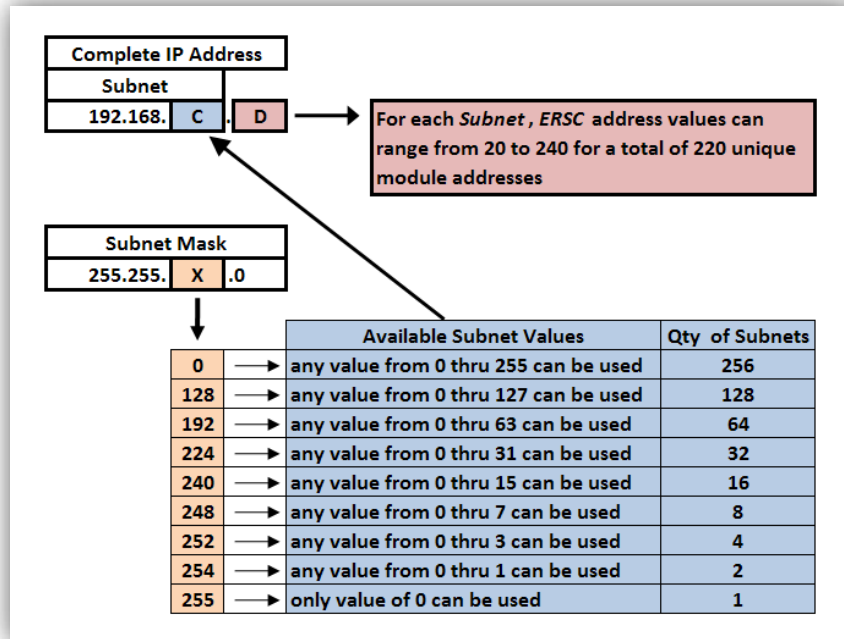
**FIGURE 37 - AVAILABLE SUBNETS PER TYPICAL SUBNET MASK VALUES**

As you can see, by simply manipulating the *Subnet Mask* values, you can configure your PC to see multiple *ConveyLinx* networks.

## CONFIGURATION EXAMPLE

Your PC's I.P. address is used by an Ethernet network to identify the PC on a network. For most office networks, the I.P. address is automatically assigned by your office network or in smaller networks (like a home network) the IP address is assigned by a router device. In some cases, your I.T. department may assign your PC or laptop a fixed I.P. address.

For our example we wish to be able to communicate with up to 4 separate *ConveyLinx Subnets*. With a properly configured PC, we can use *EasyRoll* to view and set parameters for all modules on all 4 networks.

The following figure illustrates how we want our PC's I.P. address settings to be configured:
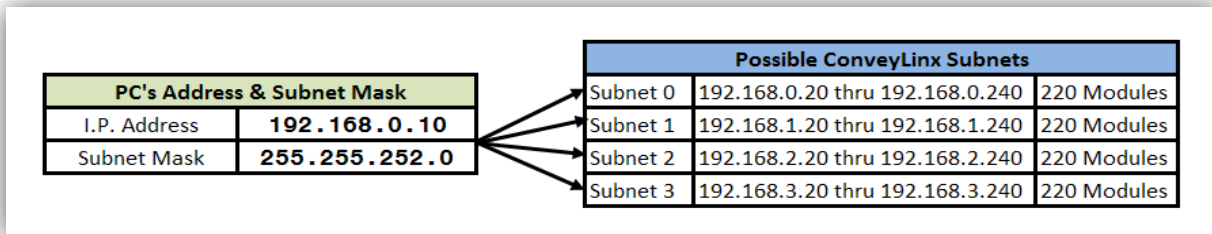
| Possible ConveyLinx Subnets | | |
|---|---|---|
| Subnet 0 | 192.168.0.20 thru 192.168.0.240 | 220 Modules |
| Subnet 1 | 192.168.1.20 thru 192.168.1.240 | 220 Modules |
| Subnet 2 | 192.168.2.20 thru 192.168.2.240 | 220 Modules |
| Subnet 3 | 192.168.3.20 thru 192.168.3.240 | 220 Modules |

**FIGURE 38 - IP ADDRESS CONFIGURATION EXAMPLE**

**Please Note:** The *ConveyLinx* IP address structure is designed such that all *ConveyLinx* last *octet* (DDD) of their address is greater than or equal to 20 and less than or equal to 240. This leaves 36 spare valid addresses (256 – 220 = 36) on the same *Subnet* for other devices such as PC's and PLC's. In our example, the last *Octet* for the PC's I.P. address is arbitrarily set to 10. This value could be any value from 0 to 19 or 241 to 255. Network conventions are such that on a given *Subnet* the last octet (DDD) values of 0 and 1 are usually reserved for the *Default Gateway* which is often the address of an Ethernet router.

Also note that our example is utilizing all the possible *Subnets* for the Subnet Mask (255.255.252.0) shown. From Figure 37 above; we could have selected any of the values for X on the chart that was listed above the 252 value. In these cases there would simply be more *Subnets* available to address.

N<span>OTES</span>:

# PULSEROLLER

**WWW.PULSEROLLER.COM**
**SALES@PULSEROLLER.COM**
**SUPPORT@PULSEROLLER.COM**