PULSEROLLER

ConveyLinx-Ai Family Complete Guide

Version 3.0 — 29 January 2024

For ConveyLinx-Ai2, ConveyLinx-Ai3-24-FC, and ConveyLinx-Ai3-24-RC

Table of Contents

1. About This Manual

Products Covered in this Manual

ConveyLinx-Ai2
All Article Numbers 3022-xxxx

ConveyLinx-Ai2-48
All Article Number 3042-0000
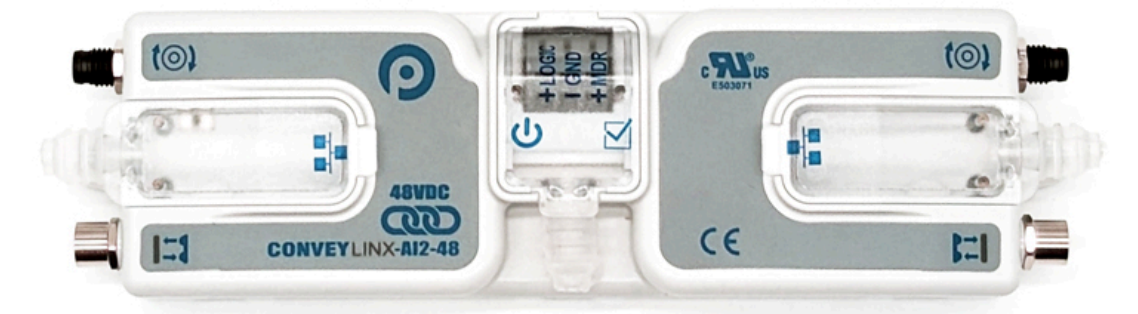
ConveyLinx-Ai3-24-FC Article Number 3023-0001

ConveyLinx-Ai3-24-RC Article Number 3023-0002

ConveyLinx-Ai3-48-FC
Article Number
3043-0000

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

! 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, Pulseroller 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 Pulseroller is prohibited

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

How to Contact Us

Technical Support North & South America: support@pulseroller.com
Sales Support North & South America: sales@pulseroller.com
Technical Support Global: global_support@pulseroller.com
Sales Support Global: global_sales@pulseroller.com
Web Site: www.pulseroller.com

2. Glossary of Terms

Term	Definition
Carton	A separate (usually wrapped or boxed) object to be transported by the conveyor. The terms tray, tote, load, or product may also be used interchangeably in this document.
ConveyLinx	Conveyor controls architecture based upon modular distributed devices connected via Ethernet network.
ConveyLinx-Ai / ConveyLinx-Ai2 / ConveyLInx-Ai3	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. The term Module will be used within this document and will refer to the ConveyLinx-Ai2 device
DHCP	Dynamic Host Configuration Protocol 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
ERSC	Ethernet Roller Speed Control module – Conveyor control module that is part of the ConveyLinx family. Each ERSC can accommodate up to 2 MDR conveyor zones. In this document the term module will be synonymous with ERSC
ERSC-SE4	Designed to "break-out" the RJ11 connection for easy installation. The module has an amplifier to the output giving it up to 100mA output capabilities. Configurable diodes for the inputs to minimize leakage current to and from the ERSC. Module also allows for external power source connection for auxiliary devices.
Hall Effect Sensor	Special sensor embedded within the brushless DC motor of an MDR used to provide motor rotor position feedback to the motor controller
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
JST	This is the name of a particular connector manufacturer that produces a specific plug/socket arrangement for MDR connection to control cards. This name is accepted within the conveyor and MDR industry as a simple description of the particular socket style used on ERSC hardware.
LED	Light Emitting Diode – 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
M8	This is the type of a connector, which has four connector pins and is used on the ConveyLinx Ai2 modules for both sensor connectors and MDR connectors
MDR	Motorized Drive Roller or Motor Driven Roller – Brushless DC motor and gearbox assembly integrated into a single conveyor roller

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 carton on the zone
PLC	Programmable Logic Controller – A wide variety of industrial computing devices that control automatic equipment
PWM	Pulse Width Modulation – a control scheme that utilizes high speed switching transistors to efficiently deliver power in a controlled fashion from the ConveyLinx controller to MDR
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. 'Reflex (or sometimes known as proximity) type photo-sensors emit light to be reflected back from an object located sufficiently close to the sensor. 'For both types of photo-sensors, when they detect their reflected light source, their signaling output circuit changes state.
RJ-11 / RJ-12	Registered Jack Style 11 / 12 – Standard connector / receptacle format utilizing 4 or 6 pin connections. The typical standard connection for telephones. RJ-11 utilizes 4 pins and RJ-12 utilizes 6 pins but both styles use the same physical size.
RJ-45	Registered Jack Style 45 – Standard connector / receptacle format utilizing 8 pin connections. The typical standard for computer network cable connections
Senergy-Ai	PulseRoller brand proprietary motor control platform that provides electronic intelligence inside the motor that can be read by ConveyLinx-Ai Family and MotionLinx-Ai Family control modules. The connection from the motor to the controller is via 4-Pin M8 style connector

Singulation Release	Conveyor control method for zoned controlled conveyor that dictates that when a zone is discharging its carton, the upstream carton waiting to enter must wait until the discharged carton is completely clear before it is allowed to enter
Slave Rollers	A set of non-motorized conveyor rollers mechanically linked to an MDR. The MDR and slave rollers make up a physical zone. All of the slave rollers in a zone rotate at the same speed and direction as the MDR because of their mechanical linkage
TCP/IP	Transport Control Protocol / Internet Protocol – 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.
Train Release	Conveyor control method for zone configured conveyor that dictates that when a zone is discharging, the upstream zone's carton can move in unison with the discharging carton.
Zone	A basic (linear or curved) cell of the conveyor consisting of a set of slave rollers driven by one or more MDR's and a single photo-sensor.
ZPA	Zero Pressure Accumulation – 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

3. Getting Started

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

Who Should Use this Manual?

This manual is intended for users who need basic product information and simple application procedures to implement 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.

What do you want to do?

The Basics

Learn about module hardware ports
Learn about power supply sizing
How to Auto-Configure your network of modules
Find out about what all the LED states mean
Learn about the different release modes and how to change them
Learn about Flex Zone
Learn about jam conditions
How to reset a module back to factory default
How to wire up a Wake Up interlock
How to wire up a Lane Full Interlock
How to Auto-Replace a module

Basic things you can do with EasyRoll and EasyRoll+

Learn about basic navigation through EasyRoll
Learn about ZPA settings and how to change them
How to change motor direction, speed, accel/decel, etc.
Learn about motor status and error indicators
How to change Jam Timers and Auto-Clear Timers

How to change how the module logic uses the block/clear output from your sensors

How to change your Aux I/O Pins to match the signals you are using

Some advanced things you can do with EasyRoll and EasyRoll+

Learn how to discover modules on your network and change IP addresses

How to set up Look Ahead Slow Down feature

How to disable Flex Zone Recognition and why you may want to

How to set up an Extension or slave module

How to connect two separate networks together to operate in ZPA

How to backup and restore module settings

How to upgrade module firmware

4. Module Hardware

ConveyLinx-Ai Family Controllers are designed to operate *Senergy-Ai Technology* based motor units. These motor units are utilized in both *Motorized Drive Rollers (MDR)* for conveyors and *Pulse Gear Drive (PGD)* units that can be used for both conveyor and non-conveyor applications.

Out of the box, **ConveyLinx-Ai Family Controllers** are designed primarily to be installed and integrated into the conveyor's mechanical side frame assembly. The ConveyLinx-Ai Family Controllers is a controller for up to 2 MDR conveyor zones. Each ConveyLinx-Ai Family Controllers provides connection points for 2 MDR units with their corresponding 2 photo-sensors as well as upstream and downstream network and discreet interconnections to form a complete control system for zoned MDR conveyors.

> ✳ The "left" and "right" naming convention for the module ports is based upon facing the front of the ConveyLinx-Ai Family Controllers and is not to be confused with direction of product flow on the conveyor. Product flow will be designated as "upstream" and "downstream"

Learn more:

> [Identifying ConveyLinx-Ai Family Controllers Components](#)
> [Mounting Dimensions](#)
> [Motor Ports](#)
> [Sensor Ports](#)
> [Ethernet Ports](#)
> [Power Connections](#)
> [Power Supply Sizing](#)
> [LED Status Indicators](#)
> [Technical Specifications](#)

4.1. Module Hardware Revisions

ConveyLinx-Ai2

There are currently 3 Generations of Hardware Revisions (HW) for ConveyLinx-Ai2 controllers. These Generations have some restrictions for firmware versions and compatability as neighboring modules for Auto-Replacement firmware upgrade/downgrade capability. The different Generations are identified by the HW Revision for a given module which is shown both on the module's back label and from EasyRoll / EasyRoll+ software

Generation	HW Revisions
Gen 1	HW Revisions 1 through 19
Gen 2	HW Revisions 20 through 39
Gen 3	HW Revisions 40 through 59

- Module functionality (unless otherwise stated in this list) between all 3 Generations will remain identical

- *Gen 2* modules can be seamlessly Auto-Configured and used with *Gen 1* modules

- Module replacement will not update firmware into a *Gen 1* module that is neighboring with *Gen 2* and/or *Gen 3* modules. If a *Gen 2* module is used to replace a *Gen 2* module whose neighbors are also *Gen 2*; then the standard firmware upgrade/downgrade that occurs during the Module Replacement procedure will work as expected

- Module replacement will not update firmware into a *Gen 2* module that is neighboring with *Gen 3* modules. If a *Gen 3* module is used to replace a *Gen 3* module whose neighbors are also *Gen 3*; then the standard firmware upgrade/downgrade that occurs during the Module Replacement procedure will work as expected

- *Gen 2* module's firmware cannot be downgraded below version 6.1.0 and *Gen 3* module's firmware cannot be downgraded below 6.4.0

- Firmware update versions from 6.1.0 through 6.3.x will be compatible for both *Gen 1* and *Gen 2* based modules. EasyRoll and EasyRoll+ will extract and route the appropriate firmware based upon the module's HW Revision

- Firmware update versions 6.4.0 and later will be compatible for all 3 Generations of modules. EasyRoll and EasyRoll+ will extract and route the appropriate firmware based upon the module's HW Revision

- EasyRoll version 4.22 and later will be required to update firmware version 6.1.0 and later

- EasyRoll versions prior to 4.22 will still be able to Discover and connect to *Gen 2* and *Gen 3* modules and perform all functions except firmware upgrade

4.2. Identifying Module Components

The **ConveyLinx-Ai Family Controllers** have very similar module components, but there are small differences between the different models, so each module has its own page for component identification.

[ConveyLinx-Ai2](#)

[ConveyLinx-Ai3-24-FC](#)

[ConveyLinx-Ai3-24-RC](#)

4.2.1. ConveyLinx-Ai2 / ConveyLinx-Ai2-48
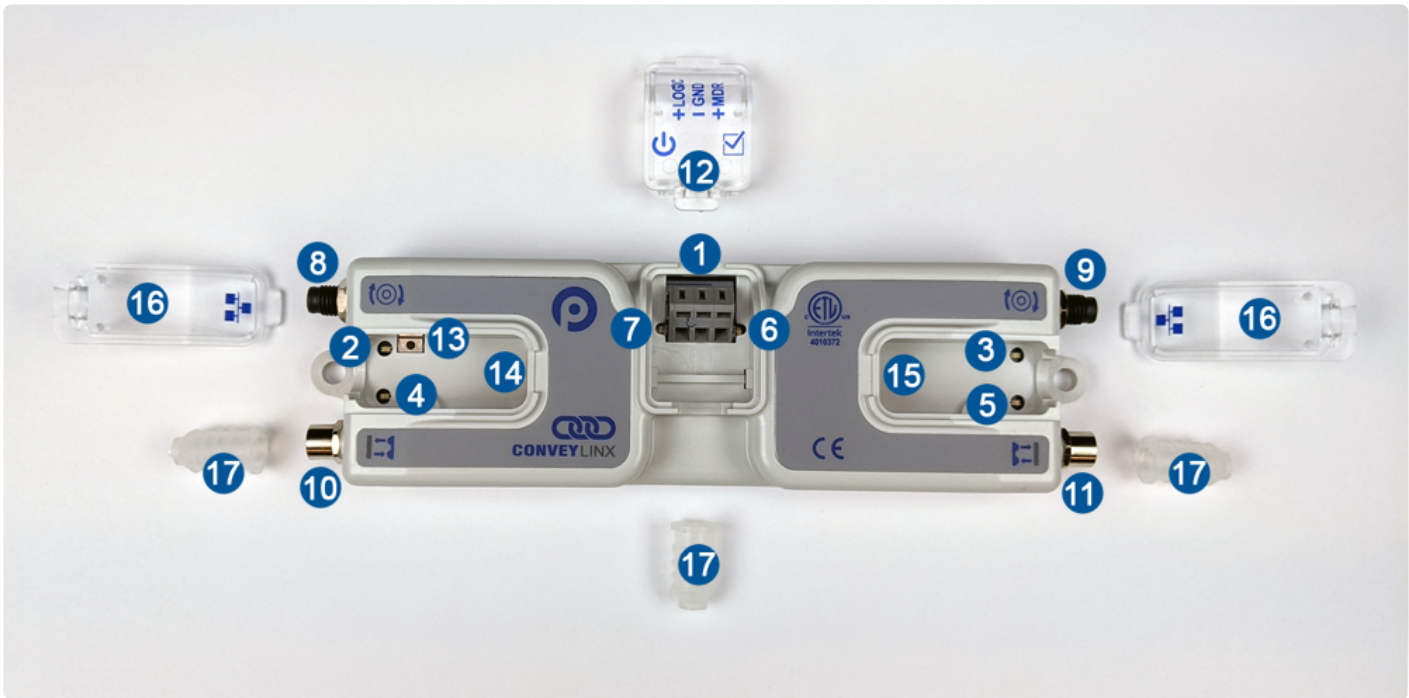
Item	Description
1	24VDC Power Terminals with separate connections for Logic and Motors
2 & **3**	Motor Left LED & Motor Right LED – Motor status indicators
4 & **5**	Left Sensor & Right Sensor Status LED Indicators
6	Module Status LED Indicator
7	Module Power LED Indicator
8 & **9**	Motor Left and Motor Right Port – 4-pin M8 style connector for MDR connection
10 & **11**	Sensor Left and Sensor Right Port – M8 style connector for zone photo-sensor connection
12	Removable IP54 Power Compartment Cover
13	Module Replacement Button

14 & **15**	Link Left and Link Right – RJ-45 style Ethernet network connection between modules including LED Indicators
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 modue but are included in the module's box

4.2.2. ConveyLinx-Ai3-24-FC / ConveyLinx-Ai3-48-FC
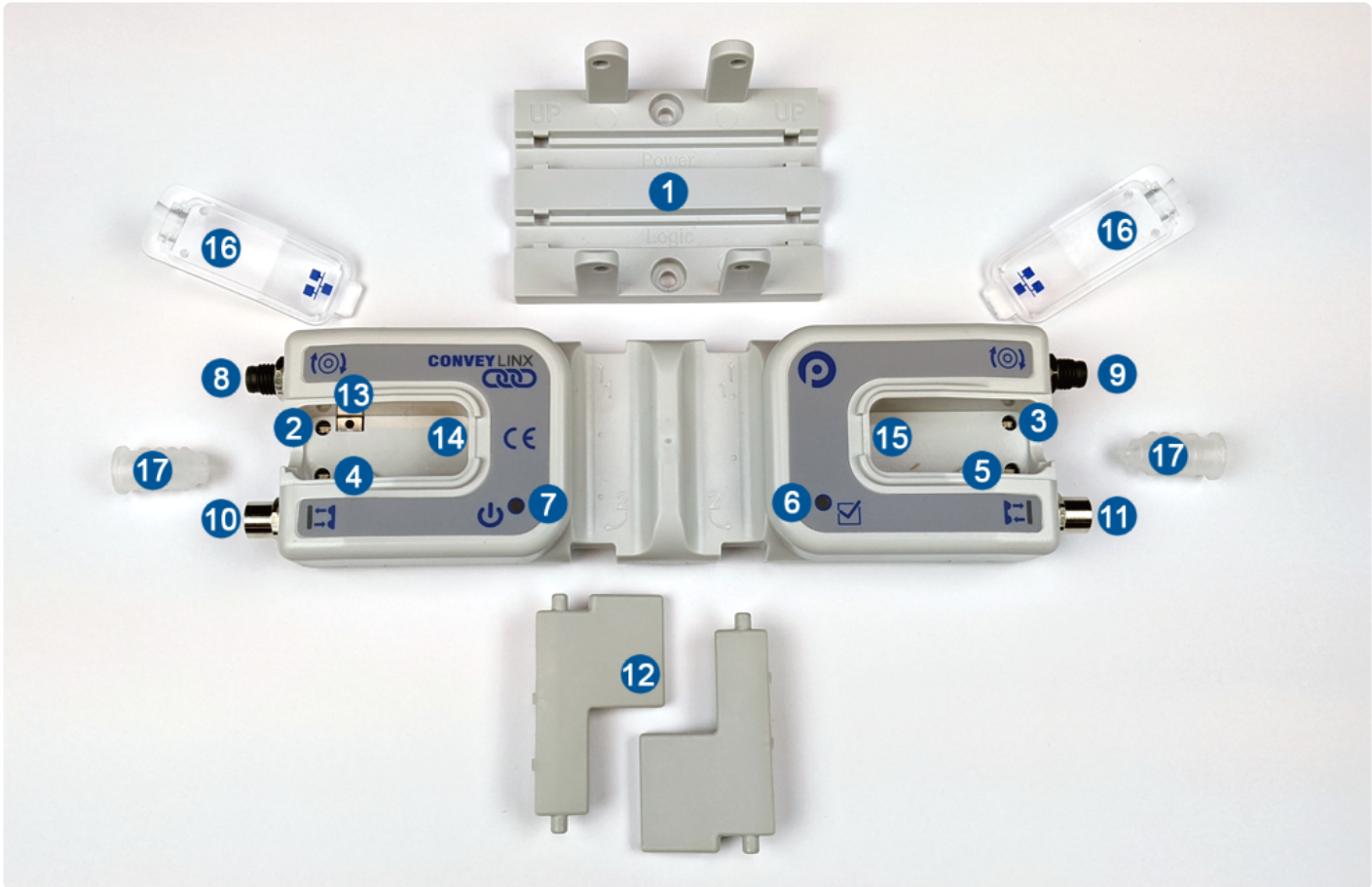
Item	Description
1	Mounting Bracket with FC cable routing channels
2 & **3**	Motor Left LED & Motor Right LED – Motor status indicators
3 & **5**	Left Sensor & Right Sensor Status LED Indicators
6	Module Status LED Indicator
7	Module Power LED Indicator
8 & **9**	Motor Left and Motor Right Port – 4-pin M8 style connector for MDR connection

10 & **11**	Sensor Left and Sensor Right Port – M8 style connector for zone photo-sensor connection
12 *	Module Clamping brackets
13	Module Replacement Button
14 & **15**	Link Left and Link Right – RJ-45 style Ethernet network connection between modules including LED Indicators
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 modue but are included in the module's box

4.2.3. ConveyLinx-Ai3-24-RC

Item	Description
1	Mounting Bracket with RC wire routing channels
2 & **3**	Motor Left LED & Motor Right LED – Motor status indicators
3 & **5**	Left Sensor & Right Sensor Status LED Indicators
6	Module Status LED Indicator
7	Module Power LED Indicator
8 & **9**	Motor Left and Motor Right Port – 4-pin M8 style connector for MDR connection
10 & **11**	Sensor Left and Sensor Right Port – M8 style connector for zone photo-sensor connection

12 *	Module Clamping brackets
13	Module Replacement Button
14 & **15**	Link Left and Link Right – RJ-45 style Ethernet network connection between modules including LED Indicators
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 modue but are included in the module's box

4.3. Mounting Dimensions

The **ConveyLinx-Ai Family Controllers** have very similar mounting dimensions, but there are small differences between the different models, so each module has its own page for mounting details.

[ConveyLinx-Ai2 / ConveyLinx-Ai2-48](#)

[ConveyLinx-Ai3-24-FC / ConveyLinx-Ai3-48-FC](#)

[ConveyLinx-Ai3-24-RC](#)

4.3.1. ConveyLinx-Ai2 / ConveyLinx-Ai2-48
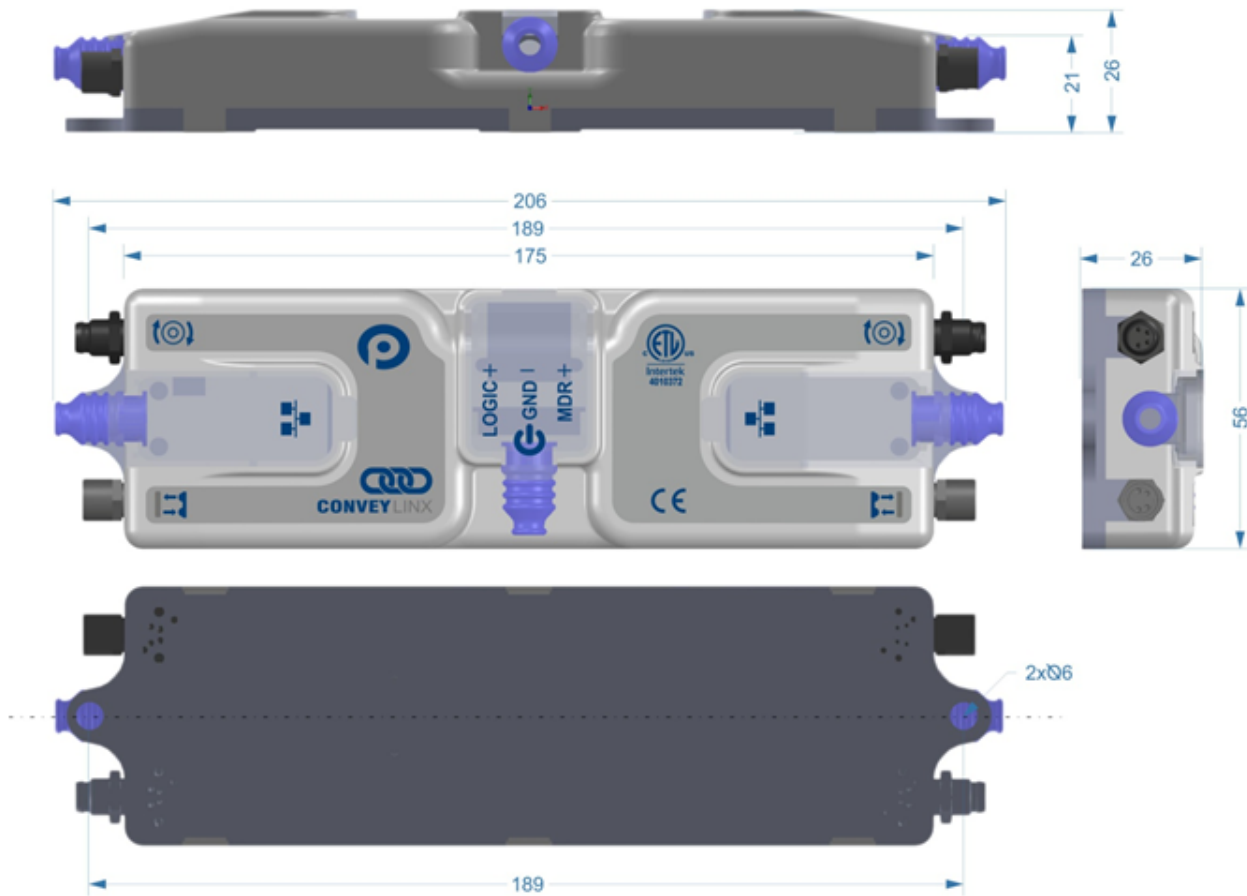
Mounting Hole Dimensions
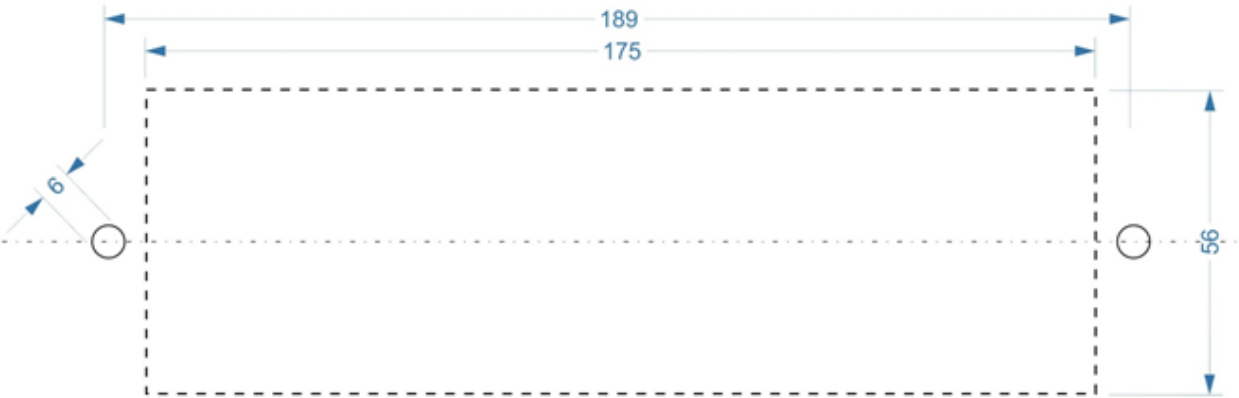
4.3.2. ConveyLinx-Ai3-24-FC / ConveyLinx-Ai3-48-FC

Mounting Hole Dimensions

216

2x⌀4

45

56

80
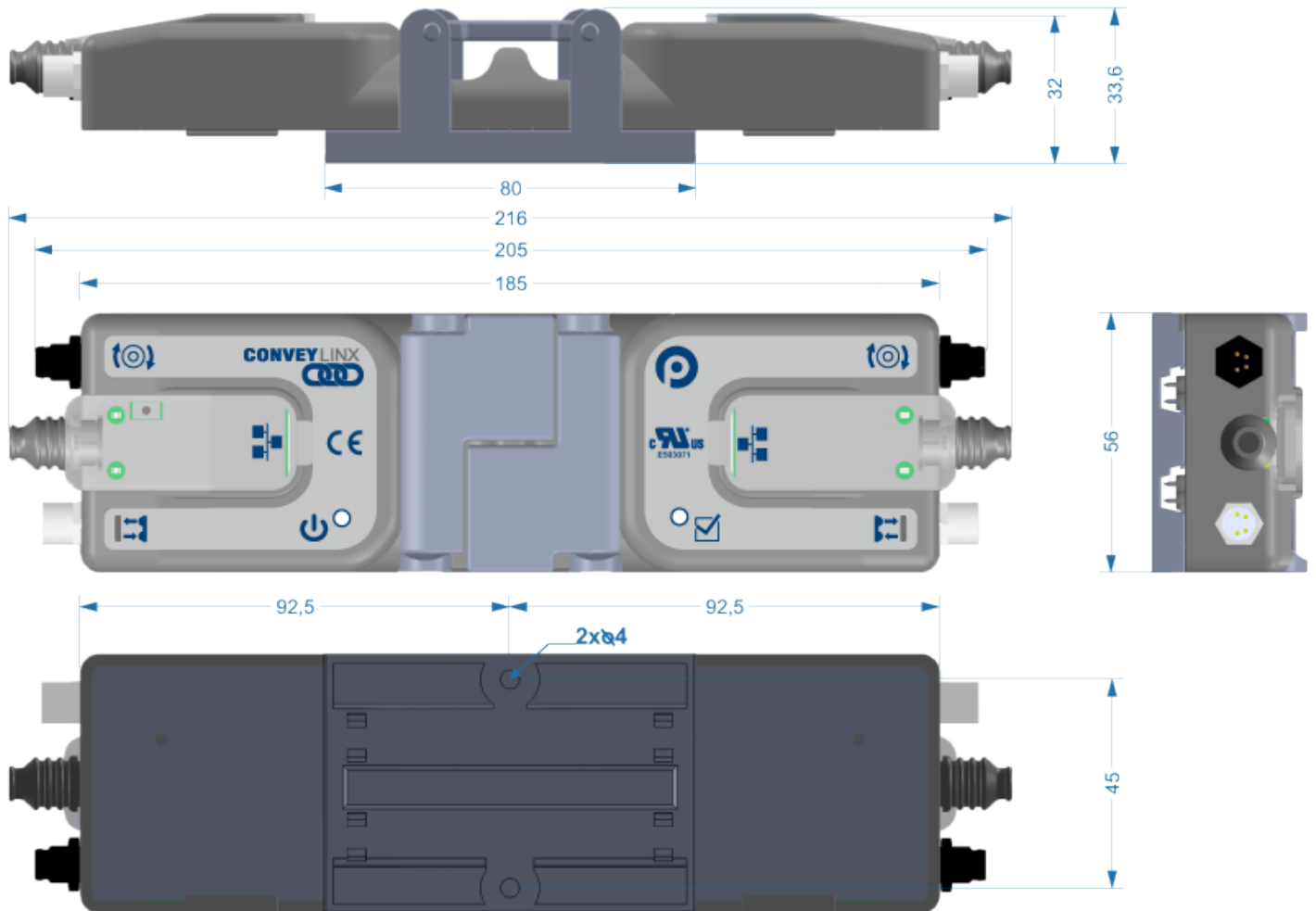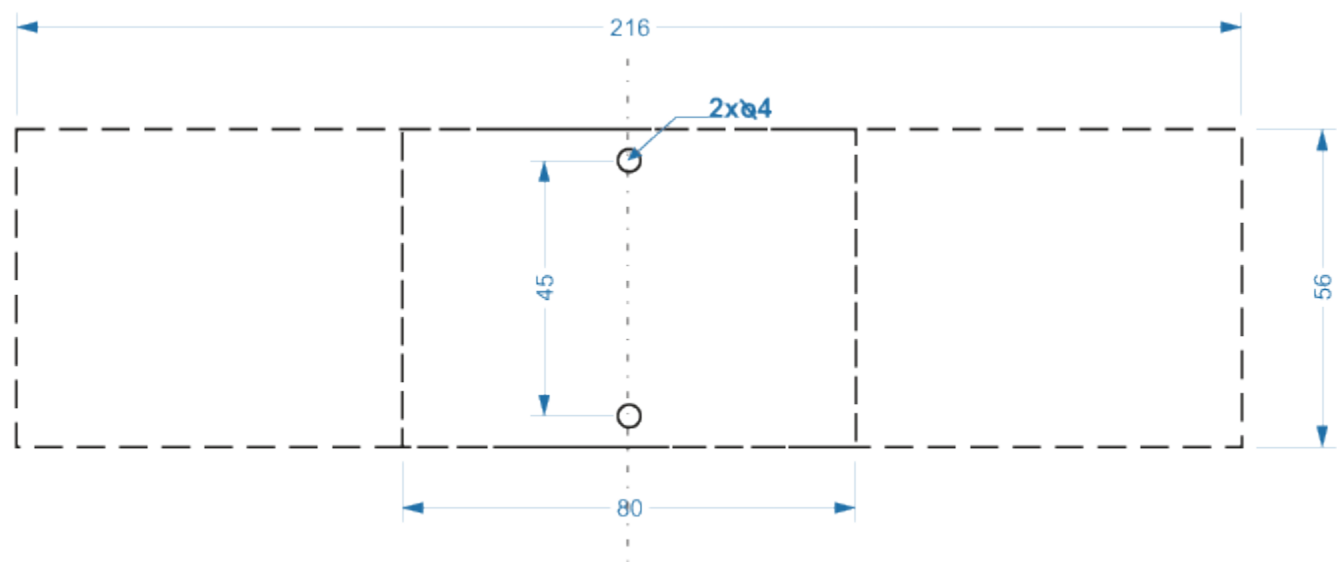
4.3.3. ConveyLinx-Ai3-24-RC
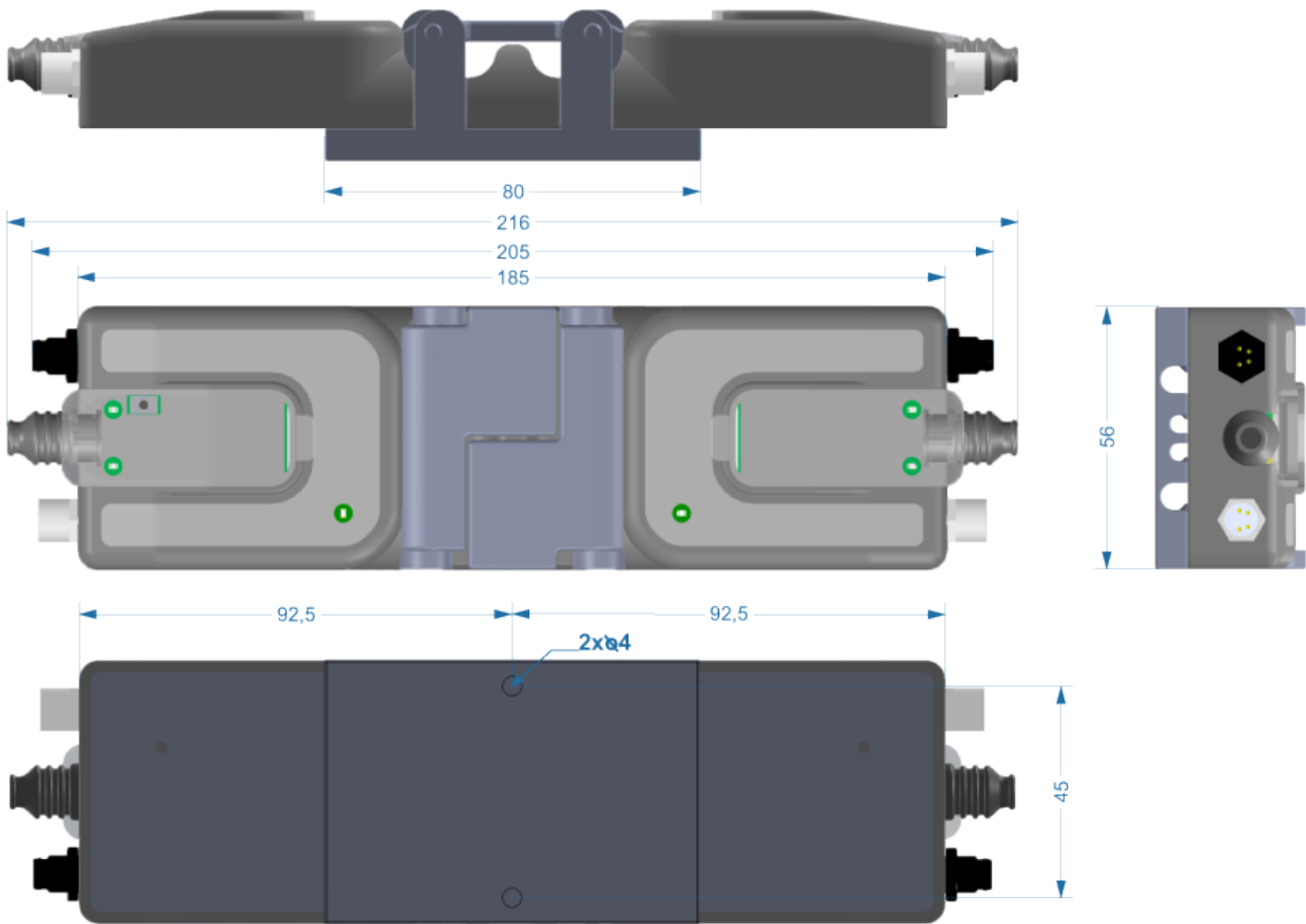
Mounting Hole Dimensions
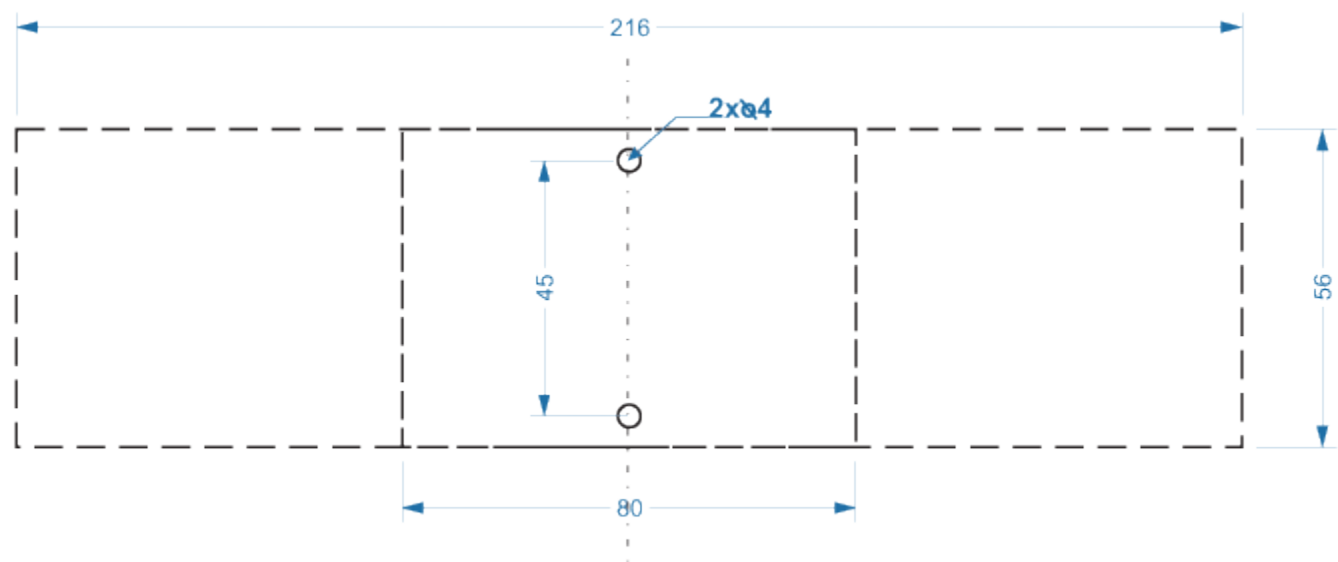
4.4. Module Power Connections

At a minimum, for any of the *ConveyLinx-Ai Family Controllers*, you must supply at least the *Motor Power* connection. The *Logic Power* connection is optional and used when you want to be able to disconnect motor power and yet maintain module communication functionality.

There are differences between each of the ConveyLinx-Ai Family Controllers varieties:

    ConveyLinx-Ai2 / ConveyLinx-Ai2-48

    ConveyLinx-Ai3-24-FC

    ConveyLinx-Ai3-48-FC

    ConveyLinx-Ai3-24-RC

4.4.1. ConveyLinx-Ai2 / ConveyLinx-Ai2-48

The protective shrouds (Item 17 as depicted in *Identifying Module Components*) will be unattached to the module when shipped and are included in the module's shipping box. These shrouds are used to maintain an IP54 installation of the power and Ethernet wiring.

> ✱ 12 AWG (4.0 mm ^{2}) Max wire size, 5-6 mm strip length

> ❗ Images below are shown with individual wires for each power terminal such as would be provided with Pulseroller Accessories PTK and PTK3 series tap kits. Individual wires routed through the protective shroud will NOT PRESERVE IP54 PROTECTION. To achieve IP54 Protection for the power wiring, a multi-conductor cable with a 3 mm minimum outer diameter must be used in order to properly seal the shroud opening.
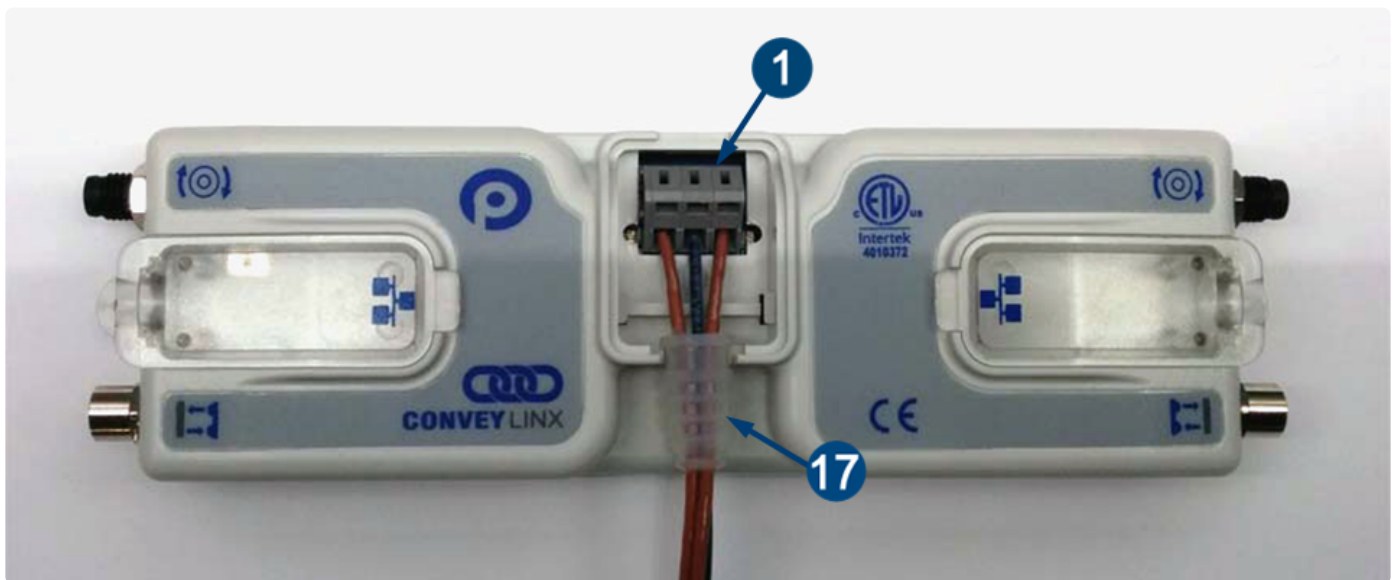
Power wires are fed through the protective shroud (Item 17). The wire terminals (Item 1) are standard cage-clamp style
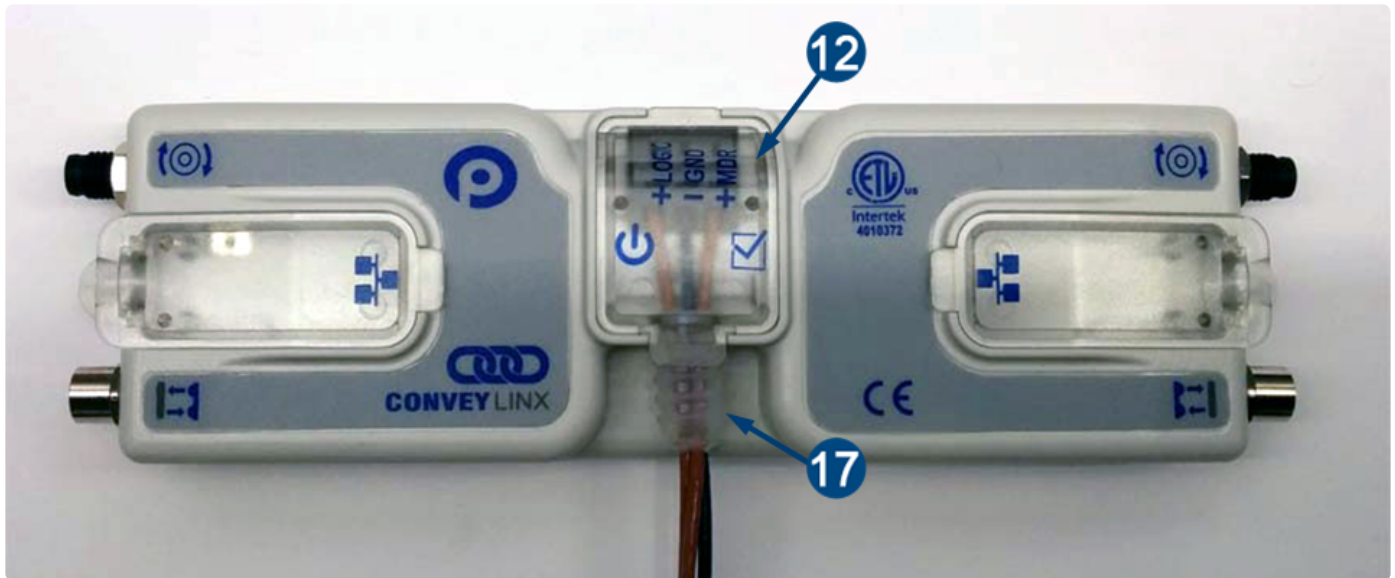
Once wiring has been completed the power wiring compartment is then sealed by snapping into place the Power Compartment Cover (Item 12)

4.4.2. ConveyLinx-Ai3-24-FC

ConveyLinx-Ai3-24-FC
Motor Only Cable Example
Provides Power to BOTH Motors and Logic

+24V

GND

ConveyLinx-Ai3-24-FC
Separate Motor and Logic Power Cable Example
Motor Power Can Be Disconnected and Logic Still Powered

+24V
Motor Power

GND

+24V
Logic Power

GND

4.4.3. ConveyLinx-Ai3-48-FC

ConveyLinx-Ai3-48-FC
Motor Only Cable Example
Provides Power to BOTH Motors and Logic

+48V

GND

BUS AS-INTERFACE LD <G> 2x2,5 RoHS ART. 2

ConveyLinx-Ai3-48-FC
Separate Motor and Logic Power Cable Example
Motor Power Can Be Disconnected and Logic Still Powered

+48V
Motor Power

GND

BUS AS-INTERFACE LD <G> 2x2,5 RoHS ART

+24V or +48V
Logic Power

GND

✱ Please note that for ConveyLinx-Ai3-48-FC, the Logic Power can be +24V or +48V

4.4.4. ConveyLinx-Ai3-24-RC

Motor Power
+24V

Logic Power
+24V

Logic Power
GND

Motor Power
GND

Motor Power
Type MTW
10 AWG
6 mm²

Logic Power
Type MTW
14 AWG
2.5 mm²

4.4.5. Assembly Instructions

The assembly procedure is the same for either the **ConveyLinx-Ai3-24-FC** module or the **ConveyLinx-Ai3-24-RC** module. Once the cabling is properly seated in the mounting mounting bracket, the main module unit snaps into the bracket and the *module clamping brackets* are installed and closed as shown:
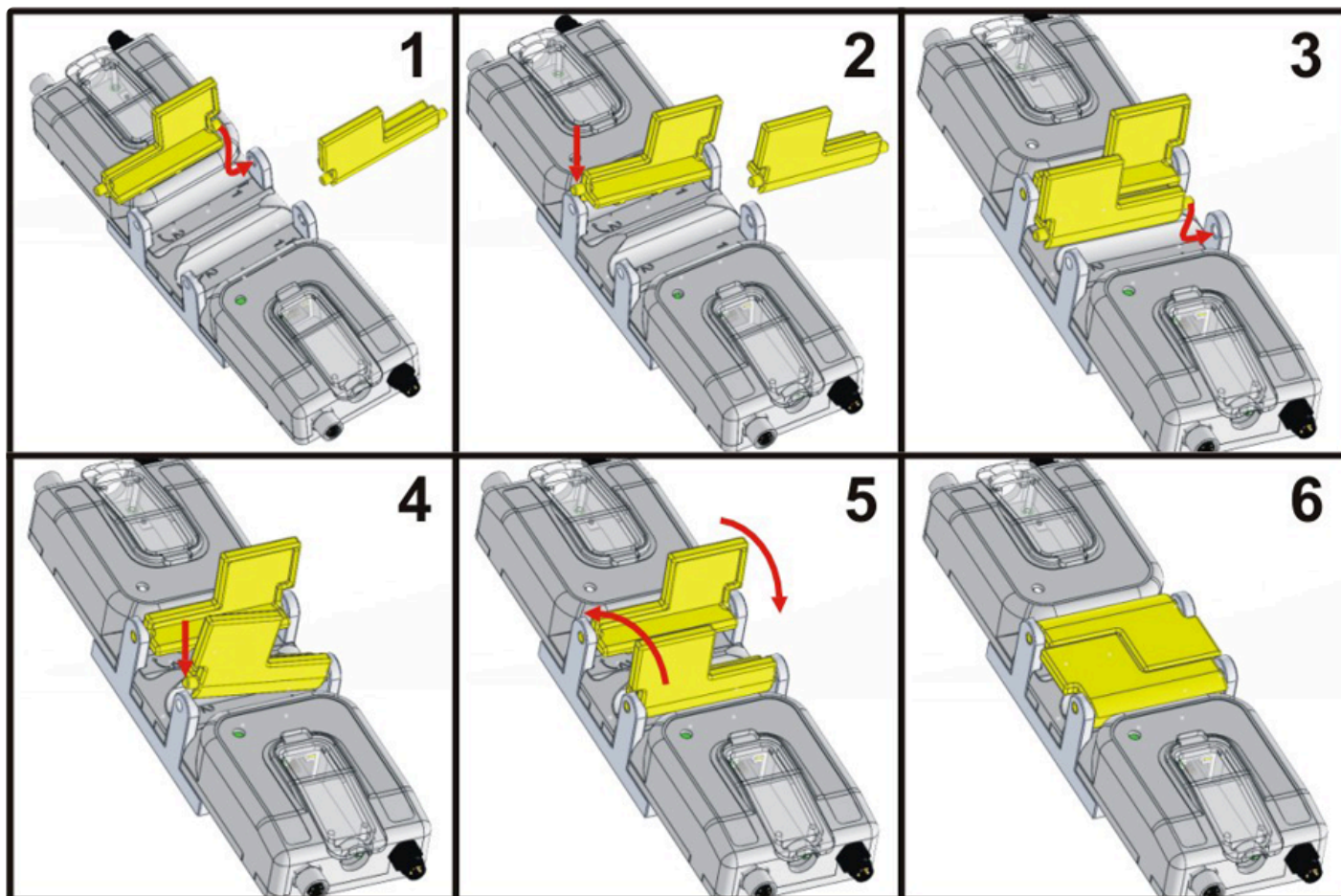
4.5. Motor and Logic Power Distribution

Regardless of the specific *ConveyLinx-Ai Family Controllers*, each is equipped with the ability to separate the *MDR Power* from the *Logic Power*.

> **!** Please note that in all power connection situations it is recommended to tie all DC common terminals together and a single connection to earth ground.

ConveyLinx-Ai2

ConveyLinx-Ai3-24-FC

ConveyLinx-Ai3-24-RC

ConveyLinx-Ai2-48

ConveyLinx-Ai3-48-FC

4.5.1. ConveyLinx-Ai2

Regardless of the specific *ConveyLinx-Ai Family Controllers*, each is equipped with the ability to separate the *MDR Power* from the *Logic Power*.

> **!** Please note that in all power connection situations it is recommended to tie all DC common terminals together and a single connection to earth ground.

Connecting only MDR Power

> **✽** Connecting only to *MDR Power* will provide power to *both* the motor circuits and the logic circuits.

The diagram shows the Power Terminal connections for 8 total modules separated into two groups of 4 modules powered by 2 separate power supplies.

Connecting only Switched MDR Power

These diagrams depict the same examples shown above with the exception of the power supplies being included in an externally controlled switching circuit such as would be used for an E-Stop.

Connecting Separate Logic and Switched MDR Power

> ✱ Providing a separate power connection to the Logic Power, you can keep the logic and communications powered and active and disconnect the MDR Power to assure all motors are disconnected such as in an E-Stop situation

The diagram shows the connection for when Logic Power is coming from a separate power supply than the MDR Power. Please note that the Logic Power supply is connected to the Logic terminals for all 8 modules.

4.5.2. ConveyLinx-Ai3-24-FC

Regardless of the specific *ConveyLinx-Ai Family Controllers*, each is equipped with the ability to separate the *MDR Power* from the *Logic Power*.

> ❗ Please note that in all power connection situations it is recommended to tie all DC common terminals together and a single connection to earth ground.

Connecting only MDR Power

> ✳ Connecting only to *MDR Power* will provide power to *both* the motor circuits and the logic circuits.

The diagram shows the Flat Cable (FC) Power connections for 6 total modules separated into two groups of 3 modules powered by 2 separate power supplies.

Connecting only Switched MDR Power

This diagram depict the same example shown above with the exception of the power supplies being included in an externally controlled switching circuit such as would be used for an E-Stop.

Connecting Separate Logic and Switched MDR Power

> ✱ Providing a separate power connection to the Logic Power, you can keep the logic and communications powered and active and disconnect the MDR Power to assure all motors are disconnected such as in an E-Stop situation

The diagram shows the connection for when Logic Power is coming from a separate power supply than the MDR Power. Please note that the Logic Power supply is connected to the Flat Cable (FC) Logic terminals for all 6 modules.

4.5.3. ConveyLinx-Ai3-24-RC

Regardless of the specific *ConveyLinx-Ai Family Controllers*, each is equipped with the ability to separate the *MDR Power* from the *Logic Power*.

> ❗ Please note that in all power connection situations it is recommended to tie all DC common terminals together and a single connection to earth ground.

Connecting only MDR Power

> ✳ Connecting only to *MDR Power* will provide power to *both* the motor circuits and the logic circuits.

The diagram shows the Flat Cable (FC) Power connections for 6 total modules separated into two groups of 3 modules powered by 2 separate power supplies.

Connecting only Switched MDR Power

This diagram depict the same example shown above with the exception of the power supplies being included in an externally controlled switching circuit such as would be used for an E-Stop.

Connecting Separate Logic and Switched MDR Power

> ✳ Providing a separate power connection to the Logic Power, you can keep the logic and communications powered and active and disconnect the MDR Power to assure all motors are disconnected such as in an E-Stop situation

The diagram shows the connection for when Logic Power is coming from a separate power supply than the MDR Power. Please note that the Logic Power supply is connected to the Flat Cable (FC) Logic terminals for all 6 modules.

4.5.4. ConveyLinx-Ai2-48

Regardless of the specific *ConveyLinx-Ai Family Controllers*, each is equipped with the ability to separate the *MDR Power* from the *Logic Power*.

> **!** Please note that in all power connection situations it is recommended to tie all DC common terminals together and a single connection to earth ground.

> **✱** Please note that ConveyLinx-Ai2-48 **Sensor and I/O ports** accommodate **24V devices** even when powered by **48V power** supplies.
> You can still use your typical 24V sensors and I/O devices.

Connecting only MDR Power

> **✱** Connecting only to *MDR Power* will provide power to *both* the motor circuits and the logic circuits.

The diagram shows the Power Terminal connections for 8 total modules separated into two groups of 4 modules powered by 2 separate power supplies.

Sensor & I/O devices operate at 24V when Motor / Logic power is 48V

Connecting only Switched MDR Power

These diagrams depict the same examples shown above with the exception of the power supplies being included in an externally controlled switching circuit such as would be used for an E-Stop.

Sensor & I/O devices operate at 24V when Motor/Logic power is 48V

Connecting Separate Logic and Switched MDR Power

> ✳ Providing a separate power connection to the Logic Power, you can keep the logic and communications powered and active and disconnect the MDR Power to assure all motors are disconnected such as in an E-Stop situation

The diagram shows the connection for when Logic Power is coming from a separate power supply than the MDR Power. Please note that the Logic Power supply is connected to the Logic terminals for all 8 modules.

4.5.5. ConveyLinx-Ai3-48-FC

Regardless of the specific *ConveyLinx-Ai Family Controllers*, each is equipped with the ability to separate the *MDR Power* from the *Logic Power*.

> ! Please note that in all power connection situations it is recommended to tie all DC common terminals together and a single connection to earth ground.

Connecting only MDR Power

> ✳ Connecting only to *MDR Power* will provide power to *both* the motor circuits and the logic circuits.

> ✳ Please note that ConveyLinx-Ai3-48-FC **Sensor and I/O ports** accommodate **24V devices** even when powered by **48V power** supplies.
> You can still use your typical 24V sensors and I/O devices.

The diagram shows the Flat Cable (FC) Power connections for 6 total modules separated into two groups of 3 modules powered by 2 separate power supplies.

Connecting only Switched MDR Power

This diagram depict the same example shown above with the exception of the power supplies being included in an externally controlled switching circuit such as would be used for an E-Stop.

Connecting Separate Logic and Switched MDR Power

✱ Providing a separate power connection to the Logic Power, you can keep the logic and communications powered and active and disconnect the MDR Power to assure all motors are disconnected such as in an E-Stop situation

The diagram shows the connection for when Logic Power is coming from a separate power supply than the MDR Power. Please note that the Logic Power supply is connected to the Flat Cable (FC) Logic terminals for all 6 modules.

4.6. Proper Motor Grounding Practice

Motor roller drive end shaft and/or fixing bracket must be electrically bonded to the conveyor frame and conveyor frame connected to electrical ground.
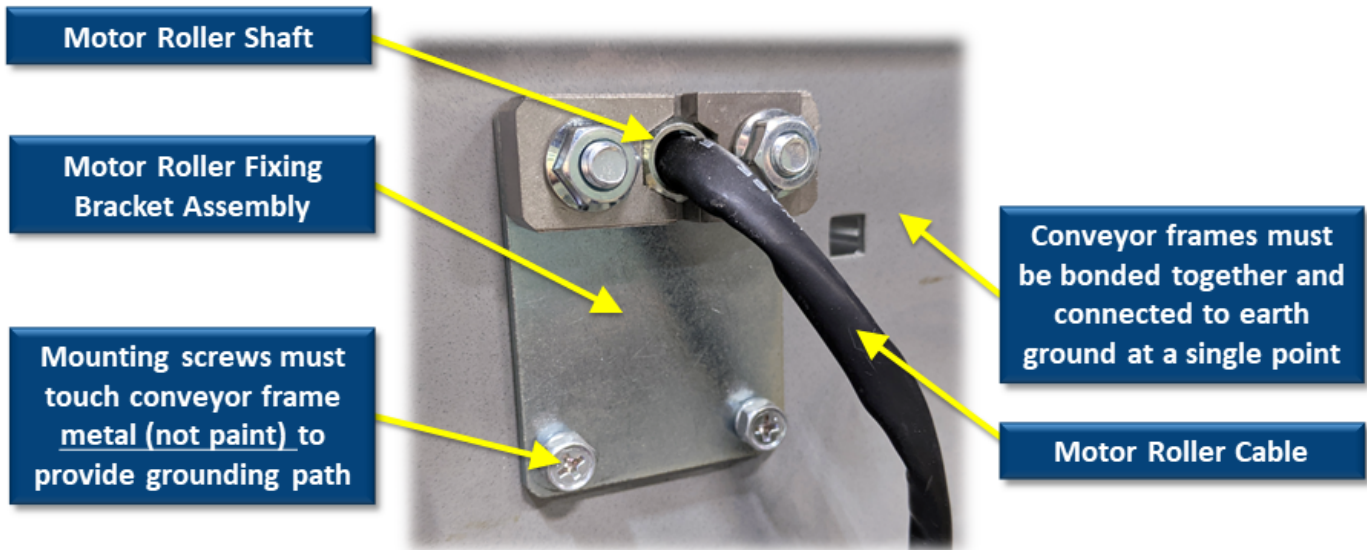
> **!** Improper grounding of MDR and/or Power Supply Common may result in premature MDR and/or ConveyLinx-Ai Family Controllers module failure. Proper grounding techniques MUST be observed for all applications

4.7. Power Supply Sizing

[Motor Power for 24V Controllers](#)

[Motor Power for 48V Controllers](#)

[Logic Only Power](#)

4.7.1. Motor Power for 24V Models

The current loading on the power supply for a group of ConveyLinx-Ai Family Controllerss depends upon the *Motor Type* selected. Each of the motor types available has an associated rated current that the motor will draw at rated torque and maximum speed. Each *Motor Type* also has an associated allowed current draw that is available for a period of time upon the initial starting of the motor. Theses current values and starting times are shown in the following chart:

	ECO	ECO+	BOOST	BOOST 8
Power supply load **per Motor Port** at rated torque at maximum speed	2.5 A	4.1 A	3.5 A	3.5 A
Power supply load **per Motor Port** during motor starting period	3.0 A	4.1 A	5.0 A	8.0 A
Duration of motor starting period	5.0 sec	N/A	1.5 sec	3.0 sec

✱ Please note that the current values shown are **per Motor Port**, so if both Motor Ports are being used on a given ConveyLinx-Ai Family Controllers.
The current load seen by the power supply for that module will be **2 x value** shown.

✱ The current values are at rated speed and at rated torque. The current will be less if rated torque is not required by the motor.

❗ Pulseroller assumes the user is aware of MDR 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. Pulseroller 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.

4.7.2. Motor Power for 48V Models

The current loading on the power supply for a group of ConveyLinx-Ai Family Controllerss depends upon the *Motor Type* selected in **EasyRoll+**. Each of these motor types is based upon the **desired acceleration torque performance** required and is shown in EasyRoll+ as a **current value**.

> ❗ Unlike 24V Senergy-Ai where starting current, starting current duration, and ECO vs. BOOST modes are used; ConveyLinx-Ai2-48 and ConveyLinx-Ai3-48-FC modules provide *Acceleration Torque on Demand* and this torque is limited by the *Motor Type* current limit settings applied in EasyRoll.

	EasyRoll+ Motor Type Setting				
	1.7A	2A	2.5A	3.5A	4A
Power supply load **per Motor Port** at rated torque at maximum speed	1.9 A	2.2 A	2.7 A	3.7 A	4.0 A

> ❗ IMPOTRANT: Even though you can size your power distribution based upon the *Acceleration Torque on Demand Current Limit*, extended operation at any current above the Rated Current of 1.6A will eventually result in overheating of the motor.

> ✱ Please note that the current values shown are **per Motor Port**, so if both Motor Ports are being used on a given ConveyLinx-Ai Family Controllers.
> The current load seen by the power supply for that module will be **2 x value**

shown.

✳ The current values are at rated speed and at rated torque. <u>The current will be less if rated torque is not required by the motor</u>.

! Pulseroller assumes the user is aware of MDR 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. Pulseroller 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.

4.7.3. Logic Only Power

When using a separate power supply for the **Logic Only Power**, the number of modules that can operate from a single power supply depends primarily on the wiring distance and the gauge of the wire. Please use the following guidelines when planning your Logic Only Power Supply design.

Power Supply Size	Wire Gauge	Max. Wire Length	Max. Qty of Modules
10A	16 AWG	50M / 165 ft.	25
10A	14 AWG	50M / 165 ft.	30
20A	14 AWG	50M / 165 ft.	40

4.8. Motor Ports

Both ports utilize a 4-pin M8 male receptacle. Each receptacle is mechanically keyed to assure proper orientation upon plugging in.

M8 4 Pin Male Motor Port with Senergy Ai Motor Female Connector

4.9. Sensor Ports

> ✳ All ConveyLinx-Ai Family controllers **accommodate 24V devices** on their sensor ports regardless if they are powered by **24V or 48V**

Each sensor port is a standard M8 Female receptacle with standard pin-out:
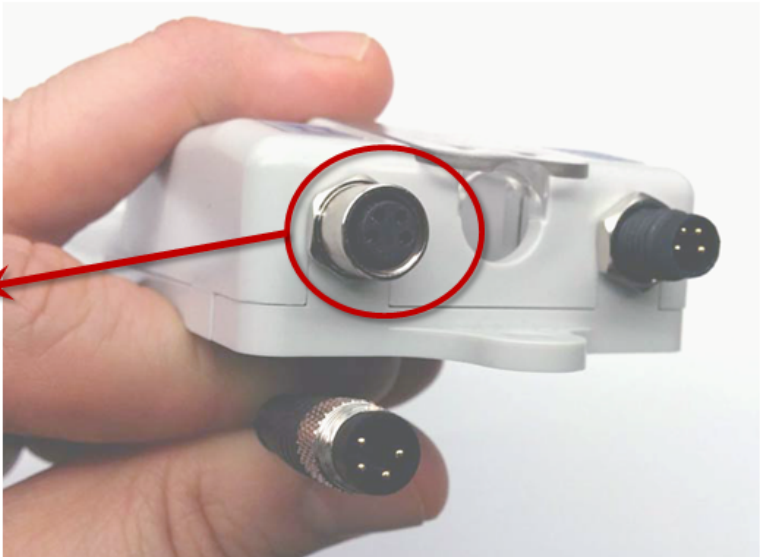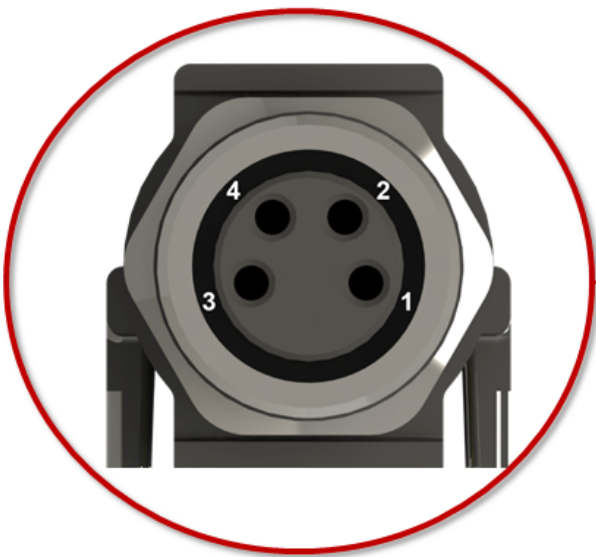
Pin	Signal	Description
1	24VDC	Module's 24VDC Supply
2	Aux I/O	I/O Signal – Input or Output Function configured with EasyRoll software
3	GND	Module's DC Common
4	Sensor Signal	Logical Input for Sensor's state output – Auto detect for NPN or PNP

4.9.1. Electrical Connections for Sensor Port Aux I/O

When connecting to either Sensor port with an M8 connector to access Pin 2 signal; you typically will use a standard parallel type splitter cable assembly that breaks out each of the available pins on the Module to their own M8 connection port. These splitter cables wire the +24V and Common connections in parallel

> ✳ Splitter cables are commercially available from multiple sources. Click here for information for Pulseroller IOX-2 Breakout/Splitter Module
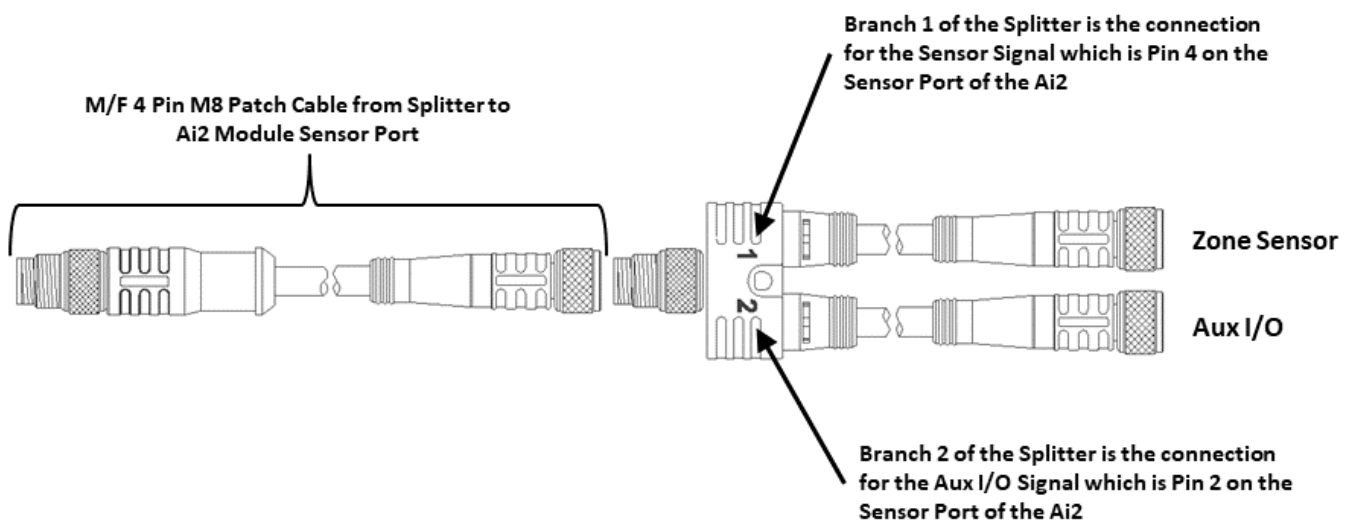
Typical Parallel Splitter Cable Usage

Aux I/O Pin 2 as Input

When the Aux I/O Pin 2 is configured as an input, the circuit is NPN/PNP auto-sensing. The signal can be connected to either +24V or 0V to operate.

Pin 4 of this connector physically connects to Pin 2 on the Ai2 Module because it is connected to Branch 2 of the Splitter

Pin 4

Pin 1 or Pin 3

Aux I/O connects to Branch 2 of the Splitter

Because Aux I/O INPUT is PNP/NPN Auto Sensing, you can use either Pin 1 of Pin 3
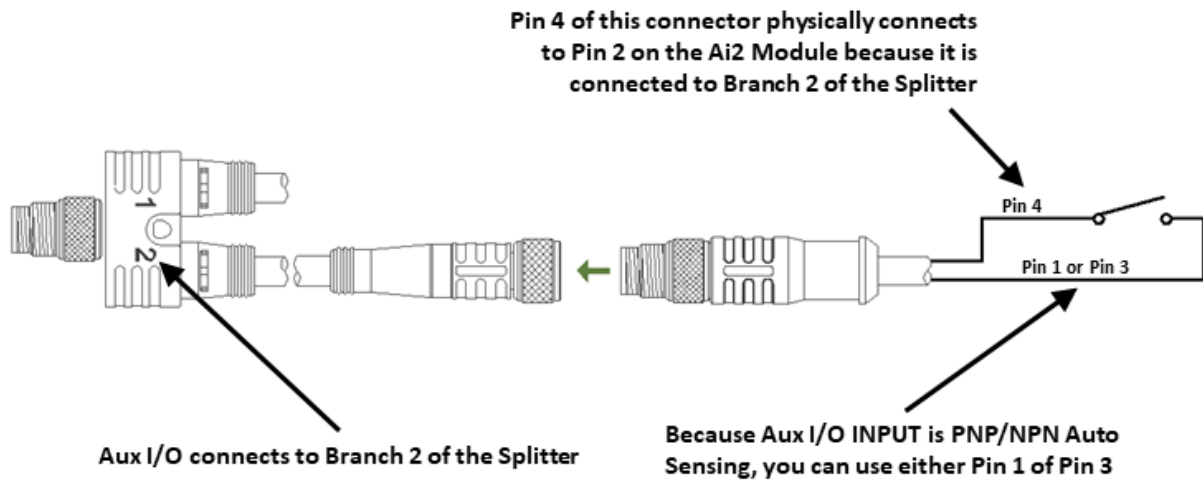
Aux I/O Pin 2 Input Connection Diagram

> ! Because the auto-sensing circuit requires a nominal voltage in order to operate, there will be some small amount of leakage current possible between Module Pin 2 and Pin 3 (GND). Please use caution if you connect a load between Module Pin 2 and Pin 3.

Aux I/O Pin 2 as Output

When the Aux I/O Pin 2 is configured as an output, the circuit is NPN only and requires the load to be connected to Pin 1 (+24V)

24VDC Load 20 mA Max.

Pin 4 of this connector physically connects to Pin 2 on the Ai2 Module because it is connected to Branch 2 of the Splitter

Pin 4   **- LOAD +**

Pin 1

Aux I/O connects to Branch 2 of the Splitter

Note that the Aux I/O OUTPUT is NPN Only
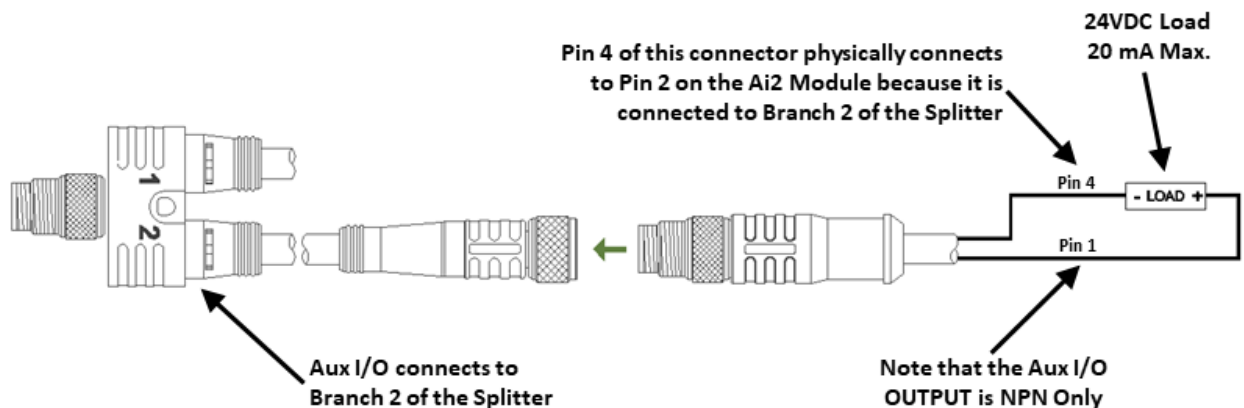
Aux I/O Pin 2 Output Connection Diagram

4.10. Ethernet Ports

Both 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. Shown below are the Ethernet cables installed using shrouds to protect the RJ-45 connectors on the Ethernet cables. Each module is shipped with 3 shrouds – 2 for the Ethernet Cables and 1 for the power wiring.
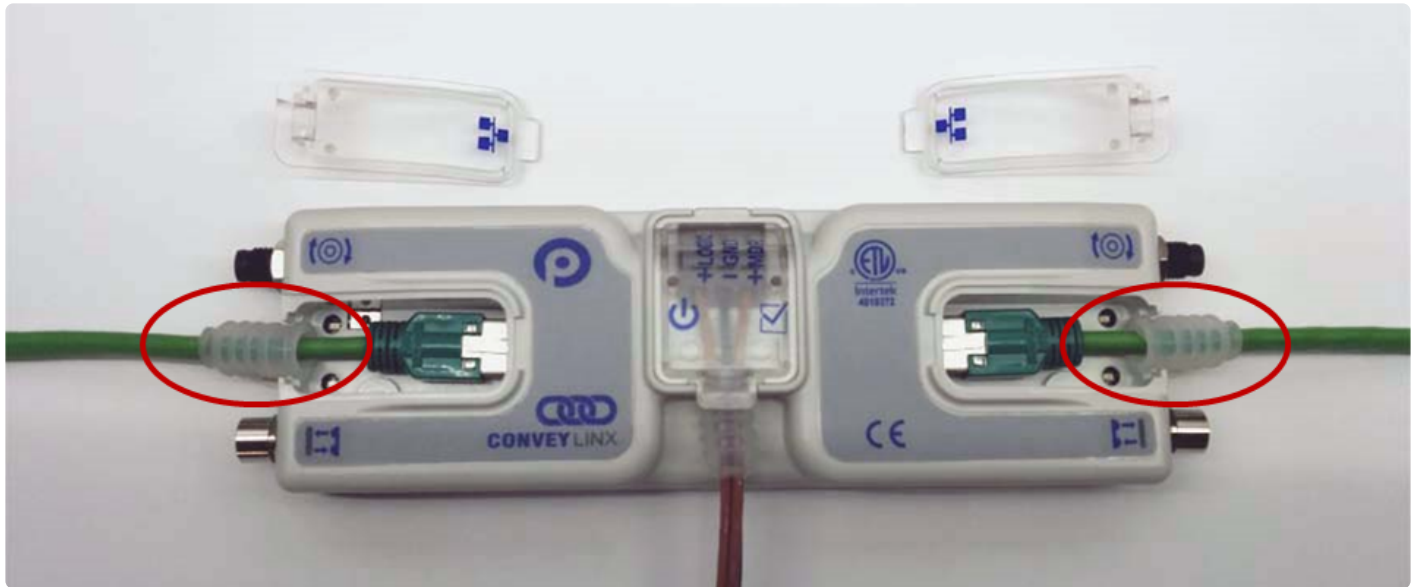
Ai2 Module with Left & Right Ethernet Cable (Covers removed)

Ai2 Module with Left & Right Ethernet Cable with Covers attached

> ✳ Even though the images depict *ConveyLinx-Ai2*, the *ConveyLinx-Ai3-24-FC* and *ConveyLinx-Ai3-24-RC* modules utilize <u>the exact same protective shrouds and covers</u>

How to install shrouds on Ethernet cables

Assembly Tool

Phoenix Contact Part Number 2891547 FL IP 54
or SS TOOL-MET 621-80008

> ✳ Tool must be purchased separately

Putting the shroud on the cable

1: With tool closed, place shroud on the pins

2: Grip the handle to open the pins to stretch open the shroud

3: While continuing to grip the handle, insert the RJ-45 cable through the shroud opening as shown

4: Release the handle and the pins close allowing them to be removed from the shroud
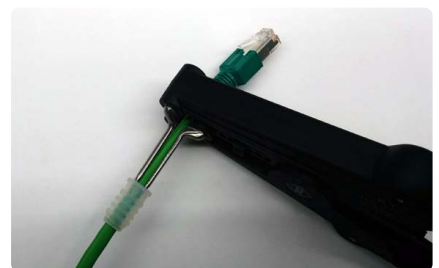
4.11. LED Status Indicators

ConveyLinx-Ai Family Controllers status is indicated by several LED's. All LED's with the exception of the Ethernet Link and Activity LEDs are multi-colored and context sensitive. The following sections indicate the various meanings of all LED indicators.

> ✱ By definition **Blinking** is approximately 1⁄2 second on/off cycle and **Flashing** is approximately 1⁄4 second on/off cycle.

If you need help finding where LED Items are located on the ConveyLinx-Ai Family Controllers

Communications

Indicator	Item	LED State	Description
Ethernet Left Link & Ethernet Right Link	⑭ & ⑮	OFF	No connection established
		Solid Green	Connection is established
		Blinking Green	When data transmission activity is occurring

Motors

Indicator	Item	LED State	Description
Motor Left & Motor Right	② & ③	OFF	Motor is not running and no faults detected
		Solid Green	Motor is running
		Solid Red	If Motor is running – indicates current limit If Motor is stopped – indicates motor is not connected properly or is overheated Power supply is under 18V or above 30V
		Blinking Red	Motor is overloaded and the module is limiting current to reduce temperature
		Flashing Red	Motor short circuit detected between at least two of the phase windings or Motor Power supply is less than 10V
		Solid Amber	Module is booting up

		Slow Blinking Red	Profinet enabled discover function initiated from PLC

Network & Module Status

Indicator	Item	LED State	Description
Module Status	⑥	Blinking Red	Module is starting task processes
		Blinking Green	Module is ready
		Flashing Green & Blinking Red	Module is in Failsafe Mode
		Flashing Red	Auto-Configuration is in progress
		Blinking Amber	Connection to peer lost or performing firmware upgrade check
		Solid Amber	Firmware upgrade in progress
		Solid Red	No Configuration either due to failed Auto-Replace or failed Auto-Configuration

Sensors

Indicator	Item	LED State	Description
Sensors Left & Right	④ & ⑤	Solid Green	Sensor Input is energized
		Solid Amber	Module is booting up
		Flashing Green	When in ZPA Mode and when Sensor is blocked – indicates external device (PLC/PC controller or EasyRoll) has accumulated the zone and inhibiting release
		Solid Red	Aux I/O Pin 2 is energized (when configured as an Input while in PLC I/O Mode)
		Blinking Red	Arrival Jam or missing sensor
		Blinking Green/ Amber	Sensor Jam

		Flashing Red	network Stop condition
		Flashing Amber	Power supply is less than 10V
		Slow Blinking Red	Profinet enabled discover function initiated from PLC

Power

Indicator	Item	LED State	Description
Power	**7**	Solid Blue	Power supply for both Logic and Motors are connected
		Blinking Blue	Motor power supply is less than 18V

Special Cases

Indicator	Item	LED State	Description
All Sensor & Motor	**2** & **3** **4** & **5**	Flashing Red	Module in stopped state
Left Sensor & Left Motor	**2** & **4**	Flashing Green	ZPA zone on Left side of module is in Maintenance Mode*
Right Sensor & Right Motor	**3** & **5**	Flashing Green	ZPA zone on Right side of module is in Maintenance Mode*

✱ *Maintenance mode only accessible via remote PLC.

4.12. Technical Specifications

> ✱ Unless otherwise stated, all specification data is identical for all ConveyLinx-Ai Family modules

Power Connection

ConveyLinx-Ai2 ConveyLinx-Ai2-48	Non-removable cage clamp Degson DG245-5.0, 12 AWG (3.31 mm ^{2}) Max wire size, 5-6 mm strip length
ConveyLinx-Ai3-24-FC ConveyLinx-Ai3-48-FC	Insulation Displacement for standard AS-Interface cable Optional separate cable for logic power
ConveyLinx-Ai3-24-RC	Insulation Displacement for Type MTW (or equivalent) wire Motor Power 10 AWG (6.0 mm ^{2}) 80 strand, insulation hardness 85 or less Optional Logic Power 14 AWG (2.5 mm ^{2}) 48 strand, insulation hardness 85 or less.

Electrical Ratings

Item	ConveyLinx-Ai2 ConveyLinx-Ai3-24-FC ConveyLinx-Ai3-RC	ConveyLinx-Ai2-48 ConveyLinx-Ai3-48-FC
Power supply voltage	24.0V +/- 10%	48.0V +4% /-20%
Input Protection Rating	Class III	

Standby current consumption	< 120mA	
Motor Starting Current	≤ 8A per Motor	Not Applicable
Motor Rated Current	≤ 3A per Motor	≤ 1.7A per Motor

Environmental Ratings

> ! Operating outside these parameters may result in permanent ConveyLinx-Ai Family Controllers failure or unexpected device behavior

Item	ConveyLinx-Ai2 ConveyLinx-Ai3-24-FC ConveyLinx-Ai3-RC	ConveyLinx-Ai2-48 ConveyLinx-Ai3-48-FC
Minimum Operating Voltage	21V	35V
Maximum Operating Voltage	30V	50V
Storage temperature	-25ºC to 70º C (-13ºF to 160ºF)	
Ambient Operating Temperature	Standard Module: 0ºC to 40ºC (32°F to 104°F) Freezer Rated Module: -30ºC to 40ºC (-22°F to 104°F)	
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 motor start current	8A	5A

Certifications & Standards

BDS EN 61131-2:2008	Programmable controllers — Part 2: Equipment requirements and tests
BDS EN 61000-6-2:2006	Electromagnetic compatibility (EMC) — Part 6-2: Generic standards – Immunity for industrial environments
BDS EN 61000-6-4:2007+A1:2011	Electromagnetic compatibility (EMC) — Part 6-4: Generic standards – Emission standard for industrial environments
BDS EN 55016-2-1:2009+A1:2011	Specification for radio disturbance and immunity measuring apparatus and methods Part 2-1 Methods of measurement of disturbances and immunity. Conducted

	disturbance measurements
BDS EN 55014-1:2007+A1:2009 +A2:2011	Electromagnetic compatibility – Requirements for household appliances, electric tools and similar apparatus — Part 1: Emission
BDS EN 61000-4-2:2009	Electromagnetic compatibility (EMC) Part 4-2: Electromagnetic discharge Immunity test
BDS EN 61000-4-4:2012	Electromagnetic compatibility (EMC) Part 4-4 Electrical fast transient/burst immunity test.
BDS EN 61000-4-5:2007	Electromagnetic compatibility (EMC) Part 4-5 Surge immunity test.
BDS EN 61000-4-6:2009	Electromagnetic compatibility (EMC) Part 4-6 Immunity to conducted disturbances, induced by radio-frequency field
BDS EN 61000-4-11:2009	Electromagnetic compatibility (EMC) Part 4-11 Voltage dips, short interruptions and voltage variations immunity tests

Sensor Port I/O

> ✳ <u>Please Note</u>: All ConveyLinx-Ai Family Sensor Ports accommodate **24V devices** regardless of whether they are powered by **24V or 48V**

Each Module is equipped with two 4-pin female M8 style Sensor I/O ports primarily used to connect a photo-electric sensor to the module. Each of these ports has one pin dedicated as an input for the sensor (pin 4) and one Aux I/O pin (pin 2) that is configurable to be either an input or an output.

Pin 1 – 24VDC

Pin 2 – Configurable Aux I/O Pin

Pin 3 – GND

Pin 4 – Input

Sensor Input Signal (M8 Pin 4)

The sensor input (pin 4) is auto-sensing for PNP or NPN circuit type such that both sourcing and sinking current will activate the input based upon the following conditions:

Minimum ON Current	1.5 mA
Maximum OFF Current	0.4 mA

Aux I/O Signal (Pin 2) when configured as as INPUT

When configured as an INPUT, the Aux I/O (pin 2) is auto-sensing for PNP or NPN circuit type such that both sourcing and sinking current will activate the input based upon the following conditions:

Minimum ON Current	1.5 mA
Maximum OFF Current	0.4 mA

Aux I/O Signal (Pin 2) when configured as an OUTPUT

When configured as an OUTPUT, the Aux I/O (Pin 2) provides an NPN only circuit as illustrated:
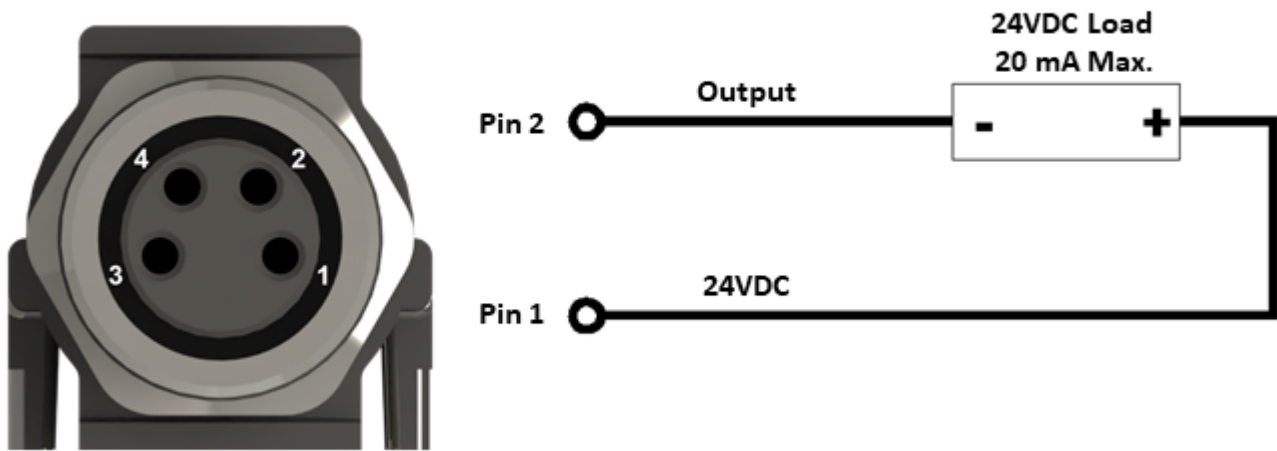
Sensor Port 24VDC (Pin 1) and GND (Pin 3)

Pin 1 of each Sensor Port provides 24V for powering up a sensor device and/or for supplying the load for the Aux I/O (Pin 2) when configured as an output. The available current for the two control ports on the module is limited internally by a solid-state fuse. The maximum combined current consumption for the two sensor ports is 100 mA.

> **!** Current in excess of 100mA drawn from the sensor port's 24V pin may cause permanent damage to the sensor detection circuit. Care should be taken to avoid excess loads, short circuits and miss-wiring of the sensor port.

Motor Port

Item	ConveyLinx-Ai2 ConveyLinx-Ai3-24-FC ConveyLinx-	ConveyLinx-Ai2-48 ConveyLinx-Ai3-48-FC

	Ai3-RC	
Supported motor types	Senergy Ai	Senergy-Ai-48
PWM frequency*	25 kHz +/- 0.1%	40 kHz +/- 0.1%
Maximum starting current	8A	N/A
Maximum rated current	3A	1.7A
Motor Protection**	Coil-to-coil short, coil-to-Vcc short, overheating, over-voltage, under-voltage, stall sensing and protection	

*The PWM frequency is firmware version dependent

> **!** **During normal operation as an MDR port, the internal protection circuitry is not capable of detecting a short-circuit between a BLDC coil output and ground. Such a short-circuit will cause damage to the high-side bridge transistors. When operating these outputs as general purpose outputs, the high-side transistors are disabled, so a pin-to-ground short-circuit is not an issue

Motor Ports in Digital Output Mode (not applicable for ConveyLinx-Ai3-48-FC & ConveyLinx-Ai2-48)

Either the Left or Right (or both) Motor Ports can be configured to operate their respective motor coil output transistors as 24V DC digital output signals. These digital output signals are sinking type only and will accommodate up to 1A on a given output pin, but restricted to a total of 1.5A for both pins on a given port.

For each of the Motor Ports, only 2 out of the 3 total motor coil output pins are available as digital outputs for a total of 4 Motor Port digital outputs available per module. Please note that these 2 available motor coil pins are different between the Left and Right Motor Ports as illustrated:

RIGHT Side View

Pin 3
Pin 4
Pin 2
Pin 1

Digital Output for RIGHT Motor Port Only

+24V
Power Supply

4 Pin
M8 Style
Female Plug

Plug into male M8
RIGHT Motor Port

Pin 3 (V)

- LOAD +

Pin 2 (U)

- LOAD +

No Connection to Pin 1
No Connection to Pin 4

24VDC Load
1A Max. per Pin
1.5A Max. Total between both Pins

LEFT Side View

Pin 1
Pin 2
Pin 4
Pin 3

Digital Output for LEFT Motor Port Only

+24V
Power Supply

4 Pin
M8 Style
Female Plug

Plug into male M8
LEFT Motor Port

Pin 4 (W)

- LOAD +

Pin 3 (V)

- LOAD +

No Connection to Pin 1
No Connection to Pin 2

24VDC Load
1A Max. per Pin
1.5A Max. Total between both Pins

Ethernet

- 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 Protocols

- Modbus/TCP
- EtherNet/IP
- Profinet IO
- CCLink IEF Basic

5. Auto-Configuration

> ! Either EasyRoll or EasyRoll+ Software is required to Auto-Configure ConveyLinx-Ai Family Controllers

> ＊ Use **EasyRoll** or **EasyRoll+** for all <u>**24V based**</u> ConveyLinx controllers

> ! You MUST use **EasyRoll+** for all <u>**48V based**</u> ConveyLinx controllers

See more about EasyRoll Software

See more about EasyRoll+ Software

The purpose of *Auto-Configuration* for networked ConveyLinx controls is to provide a simple and easy procedure for linear conveyor system commissioning. The *Auto-Configuration Procedure* requires the use of *EasyRoll* on a PC.

Connecting Devices to ConveyLinx-Ai Family Controllers
Connecting Nodes Together
Installing EasyRoll
ConveyLinx Ethernet Definition
Connecting Your PC to Network
Auto-Configuration Procedure

Linear Conveyor Definition

Auto-Configuration is only applicable to a Linear Conveyor arrangement. A Linear Conveyor arrangement is defined as a single uninterrupted path of conveyor with no merge or diverts mechanisms. A Linear Conveyor can include curved sections, but the flow of cartons on the conveyor is continuous from in-feed zone to discharge zone. A networked ConveyLinx solution can control more complex conveyor paths that include diverting and merging equipment by utilizing external PLC/PC control and/or on-board ConveyLogix programming.
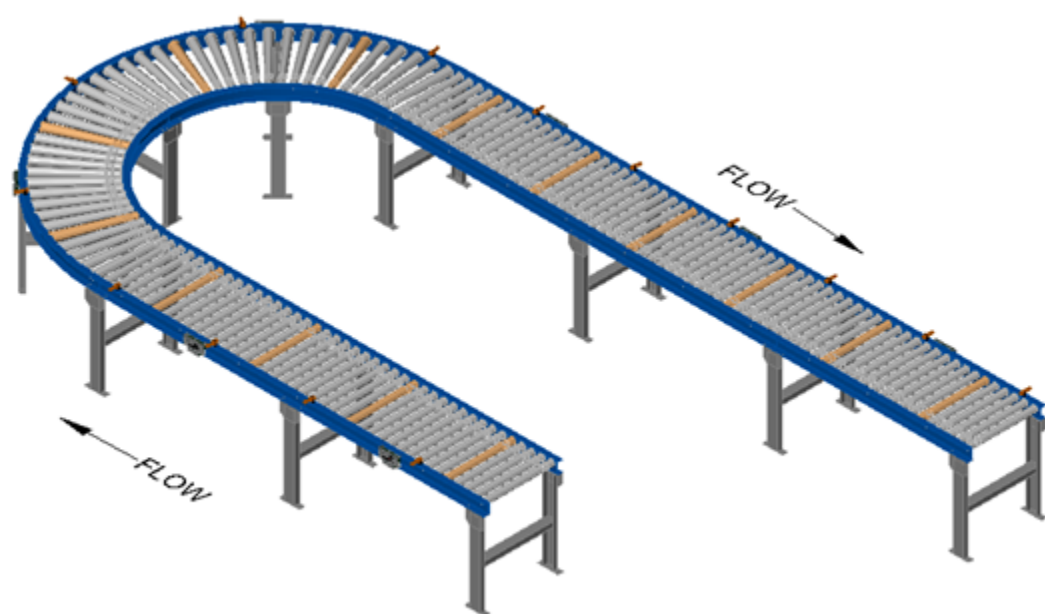
Example of a Linear Conveyor

5.1. Device Connections to Modules

Before the *Auto-Configuration Procedure* can be performed; each individual ConveyLinx-Ai Family Controllers needs to have its associated MDR's and photo-sensors connected in the proper way for expected operational results.

In general, each ConveyLinx-Ai Family Controllers detects which *Sensor* ports have a device connected and will use this to determine its specific configuration once it has been instructed to self-configure by the *Auto-Configuration Procedure*.

Before starting to configure your system to operate, each MDR and photo-sensor needs to be properly connected to the ConveyLinx-Ai Family Controllerss mounted on the conveyor. Modules will determine how to operate based upon how the photo-sensors and MDR's are connected.

> ✳ The number of *Sensors* connected will determine the total number of *Zones*. Once *Auto-Configuration* is complete, this number of *Zones* cannot be modified or over-ridden without performing another *Auto-Configuration Procedure*.

A single ConveyLinx-Ai Family Controllers can operate as a:

2 zone controller with 2 MDR's and 2 photo-sensors

1 zone controller with 1 MDR and 1 photo-sensor on Left or Right Side

OR

1 zone controller
with 2 MDR's and
1 photo-sensor
with Sensor on
Left or Right Side

OR
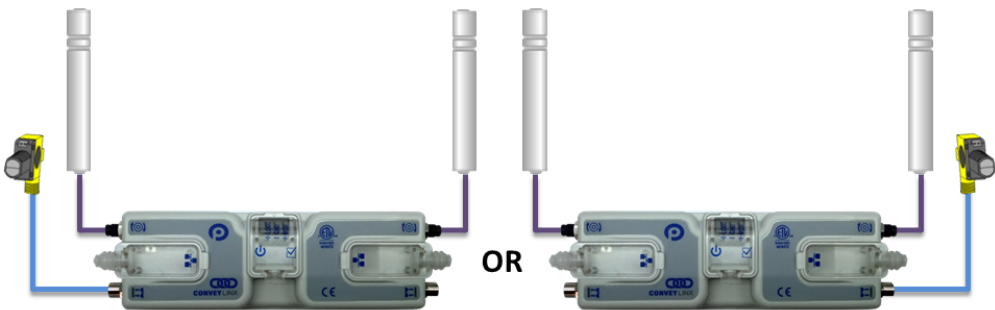
5.1.1. Examples that will generate errors

> ✱ These examples are not necessarily invalid and will not cause the Auto Configuration procedure to fail or abort, but they will result in module errors in ZPA Mode.

This module will configure as a two zone module but will generate a motor connection error for the missing motor on the Right side

This module will configure as a two zone module but will generate a motor connection error for the missing motor on the Left side

This module will configure as a single zone module on the Right side but will generate a motor connection error for the missing motor on the Right Side

This module will configure as a single zone module on the Left side but will generate a motor connection error for the missing motor on the Left Side

5.2. Motor Rotation Definition

The ConveyLinx-Ai Family Controllers uses a *Clock-Wise (CW)* and *Counter Clock-Wise (CCW)* motor rotation definition. The reference for this distinction is based upon viewing the MDR from the cable exit end of the roller.

Motor Rotation Definition

✱ *Auto Configuration Procedure* will automatically adjust the motor rotation direction based upon the flow established by the procedure as long as the cable end of the roller is on the same side of the conveyor as the ConveyLinx-Ai Family Controllers. If the rotation direction is not correct after the *Auto-Configuration Procedure*, you can change it in *EasyRoll*. How to change the motor rotation direction

5.3. Node Connections for a Subnet

With your motors and sensor connected to your individual ConveyLinx-Ai Family Controllerss, you will need to connect the Ethernet network cabling from module to module in a daisy chain fashion. The most upstream module in the direction of flow is node 1.

> ✳ Please note that the *Flow* is based upon the Ethernet cable routing during the *Auto-Configure Procedure*. The *Flow* direction cannot be changed or over-ridden. The only way to change the *Flow* direction is to perform another *Auto-Configuration Procedure*
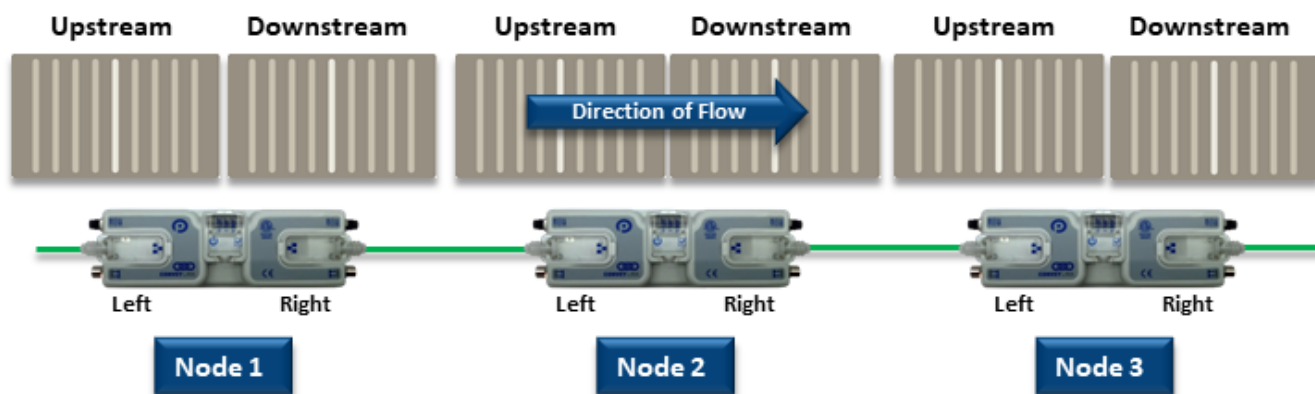
Subnet with Left to Right Flow
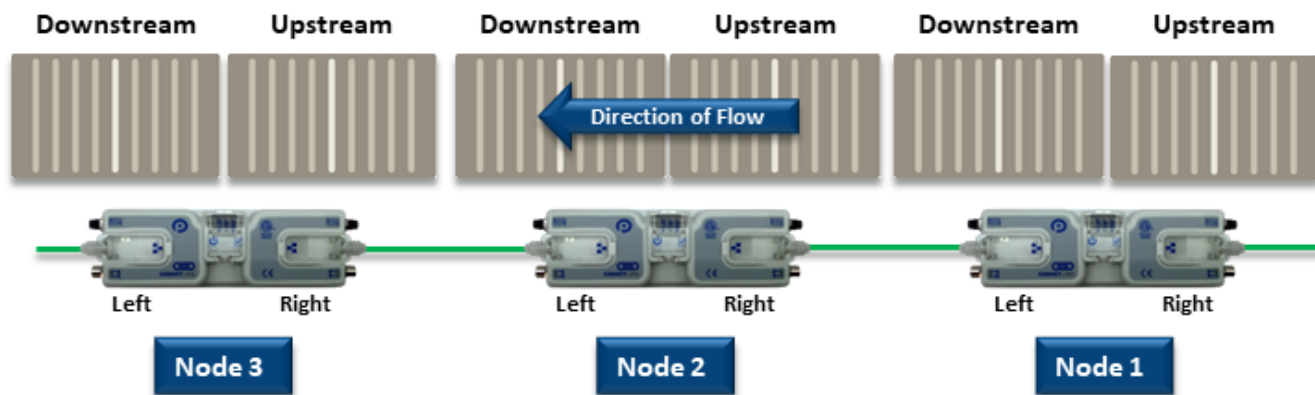
Subnet with Right to Left Flow
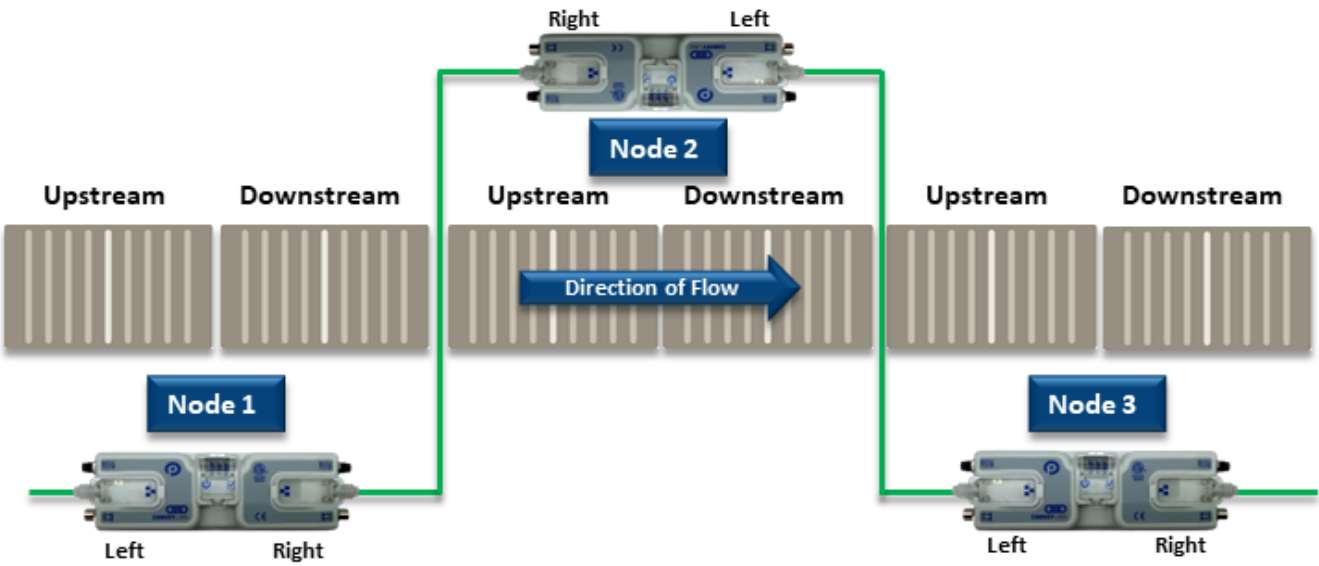
Subnet showing one module on the opposite side of conveyor

5.4. ConveyLinx Ethernet Definition

All ConveyLinx-Ai Family Controllerss 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 minus 19 is defined as the **Node**.

For example; if a module has an I.P. address of "192.168.25.20" then its *Subnet* address is "192.168.25" and its *Node* is 1 (i.e. 20-19 = 1)

At the factory, each and every module is assigned an I.P. address that is used by automated testing equipment and fixtures so that every module is verified prior to shipment. When a module 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 the most upstream module. Even if all downstream modules from the most upstream have the same or different *Subnet* or *Node* values; these downstream modules will have their *Subnet* changed to the existing *Subnet* of the most upstream module. Furthermore, when the *Auto Configuration Procedure* occurs; the most upstream module will also have its *DDD* octet value changed to 20. All downstream Modules will then have their respective *DDD* values automatically set beginning with 21.

An Example

Here is a 4-module network with possible I.P. addresses that could have been on the module from the factory. Note that their *Subnets* could be different as well as there could be duplicate addresses.
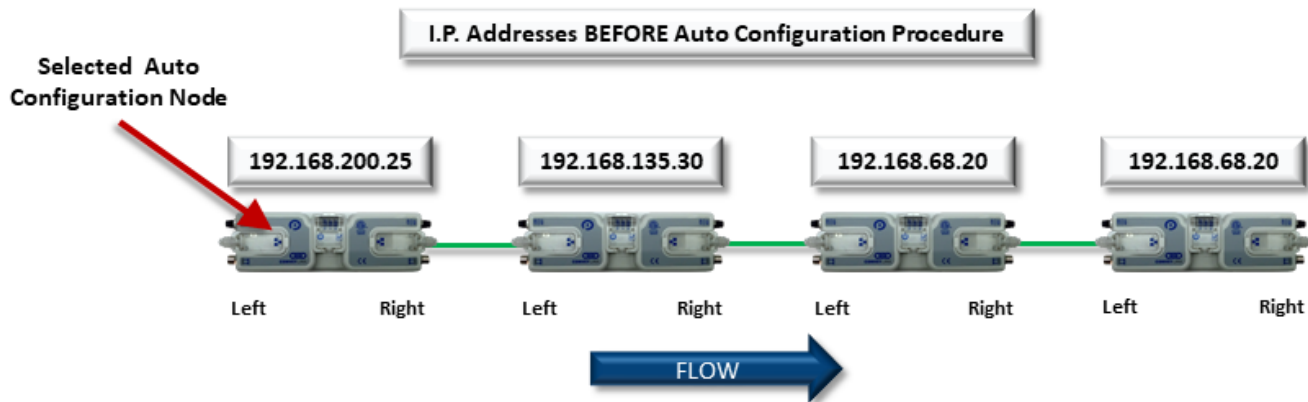
Once the *Most Upstream Module* is identified as the *Selected Auto-Configuration Node* and when the *Auto-Configuration Procedure* is performed; all 4 modules will have their I.P. address configured as shown.
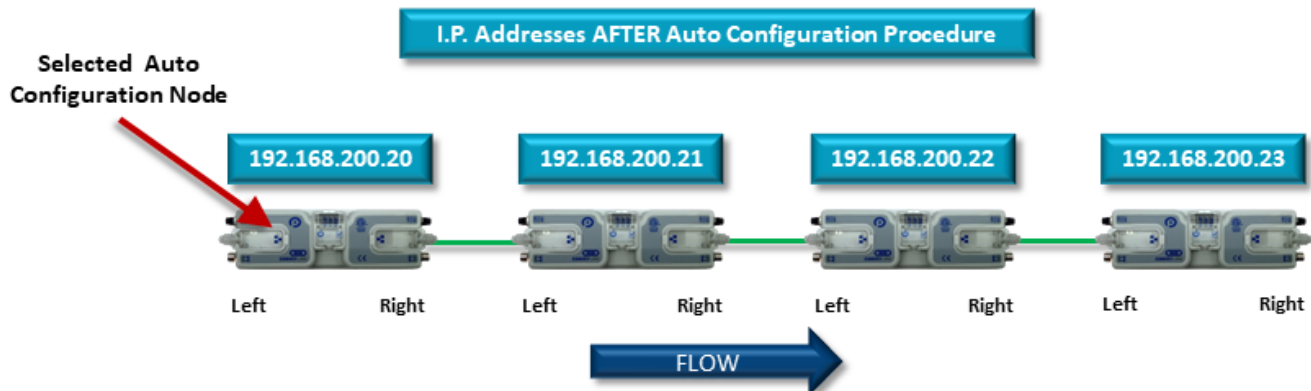
> ✳ The Auto-Configuration Procedure will assign Nodes up to and including Node 240. Therefore, each Subnet is limited to **221 Nodes**

5.5. Connecting Your PC to the Network

You can connect your PC to a ConveyLinx network with a standard RJ-45 Ethernet cable at either end of the string of modules.

Windows PC with EasyRoll

REGARDLESS OF FLOW

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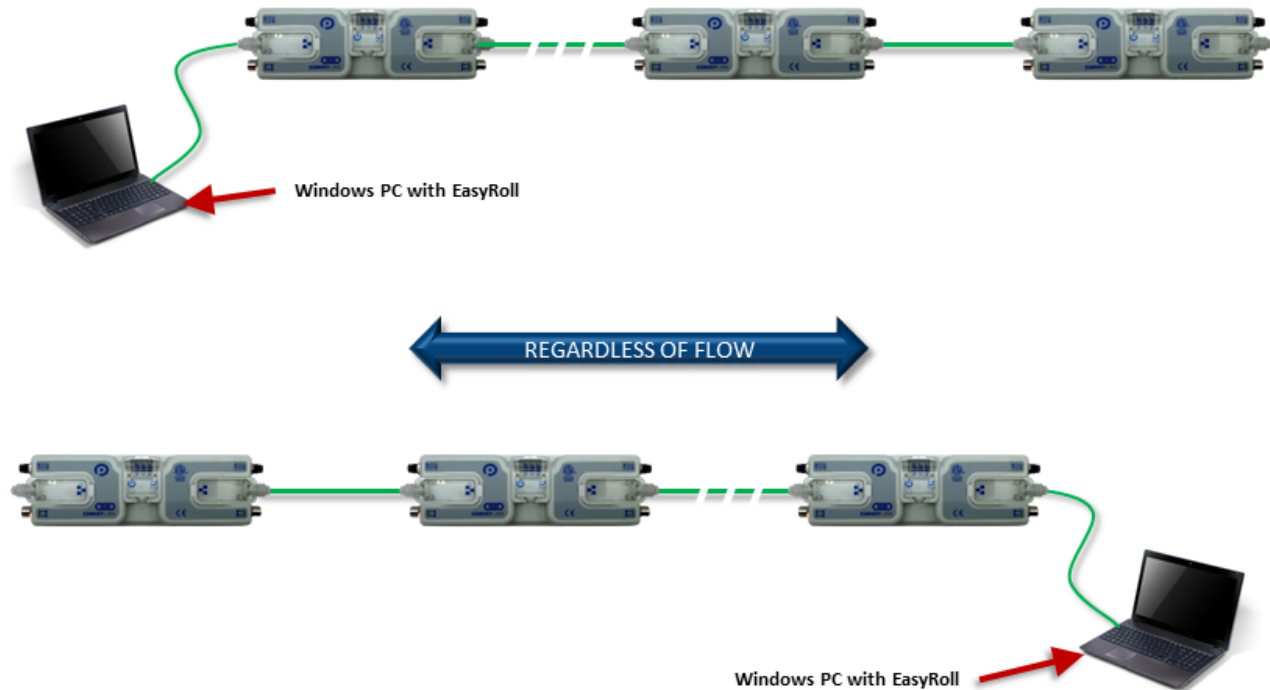
> **!** It is highly recommended to connect the PC directly to the ConveyLinx network. Avoid trying to connect via Ethernet switches or wireless router/switches. If a wireless switch is not setup properly then the Discover Feature will not work. Also ensure that network firewall is turned off for proper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

5.6. Auto-Configuration Procedure

The direction of flow of the conveyor dictates how to begin the Auto-Configuration procedure. The 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.

Before You Begin

- All sensors should be clear. *Auto-Configuration* assumes that state of the sensor is the "zone empty" state
- Make sure you have no more than 221 ConveyLinx-Ai Family Controllerss connected. *Auto-Configuration* will not configure past 221 ConveyLinx-Ai Family Controllerss in one network
- Make sure that there are no other Ethernet devices or switches connected in your string of ConveyLinx-Ai Family Controllerss. After *Auto-Configuration* is complete and successful, you can interrupt the Ethernet string with other devices and/or switches

For Auto-Configuration – make sure all modules are connected without any other devices

Once Auto-Configuration is complete – you can insert Switches and connect devices as needed
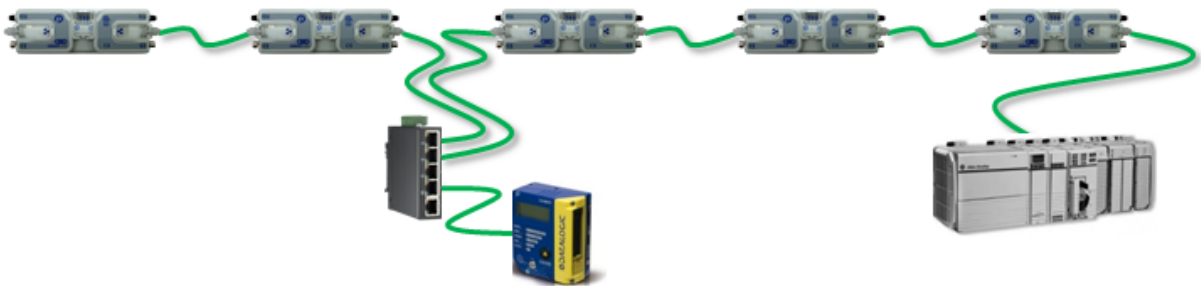
The direction of flow of the conveyor dictates where to begin the *Auto-Configuration Procedure*. The ConveyLinx-Ai Family Controllers located at the most upstream or in-feed end of the conveyor is the module where the procedure must be initiated.. Because of its physical location on the conveyor path and physical location in the Ethernet connection chain; the most upstream ConveyLinx-Ai Family Controllers will automatically connect to all downstream

ConveyLinx-Ai Family Controllerss and set their I.P. address for communication. Then the routine automatically sets the direction of flow.

> **!** For the Auto-Configuration to work properly, all loads, totes, product, containers, cartons, etc. must be removed from the entire conveyor path and all photo-sensors must be aligned and adjusted so that none are detecting that their respective zone is occupied. Failure to meet these conditions will produce unexpected results

Auto-Configuration Procedure

> **✱** Examples shown are taken from EasyRoll. EasyRoll+ provides identical functionality

Start EasyRoll V4.04 or higher and invoke the *Advanced Dialog*. In the pop-up panel select *Network Services* and click *Discover*

After a few seconds, the discovered Modules appear in the module table, click the *AutoConfig* button
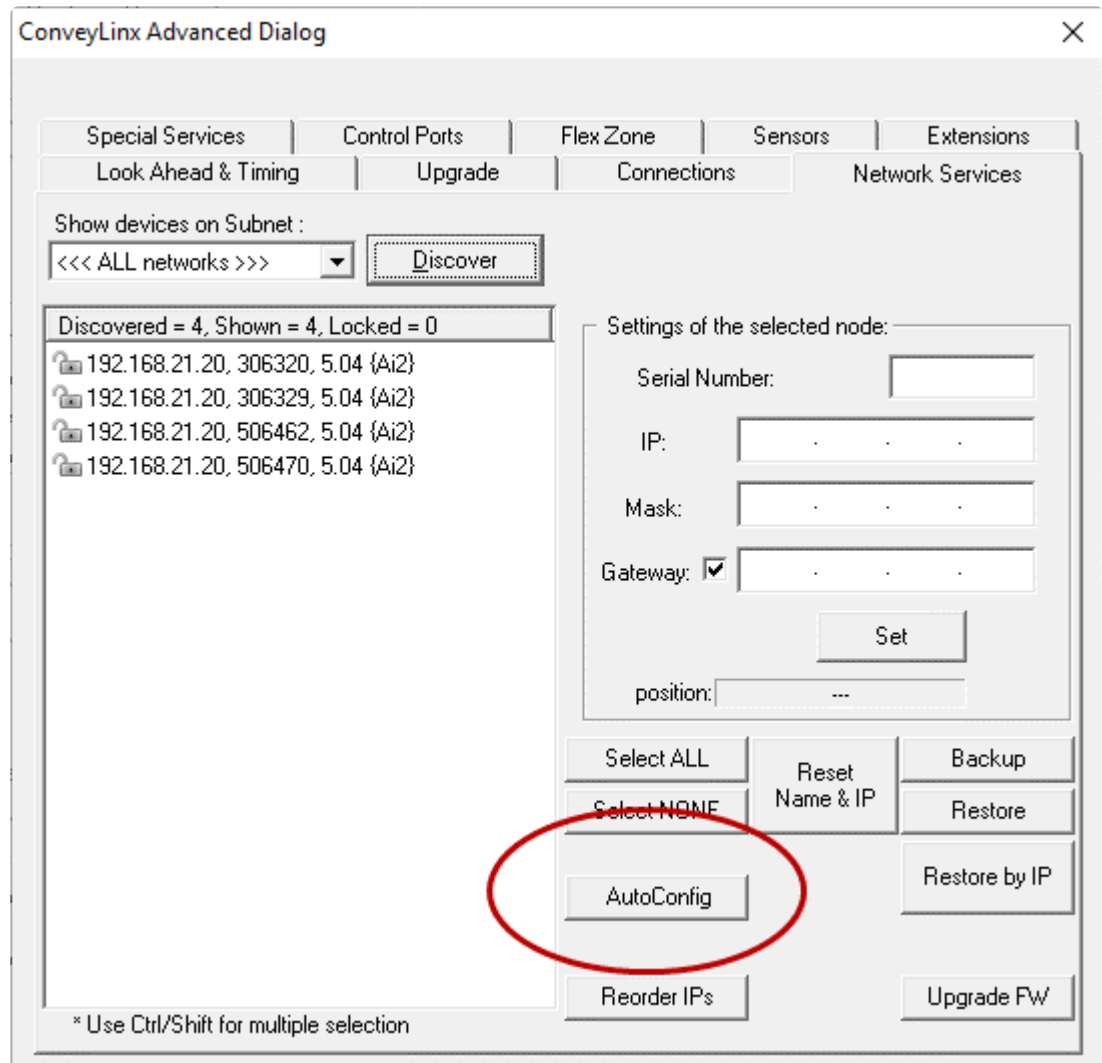
After a few seconds each Module is shown in physical topology order from the PC
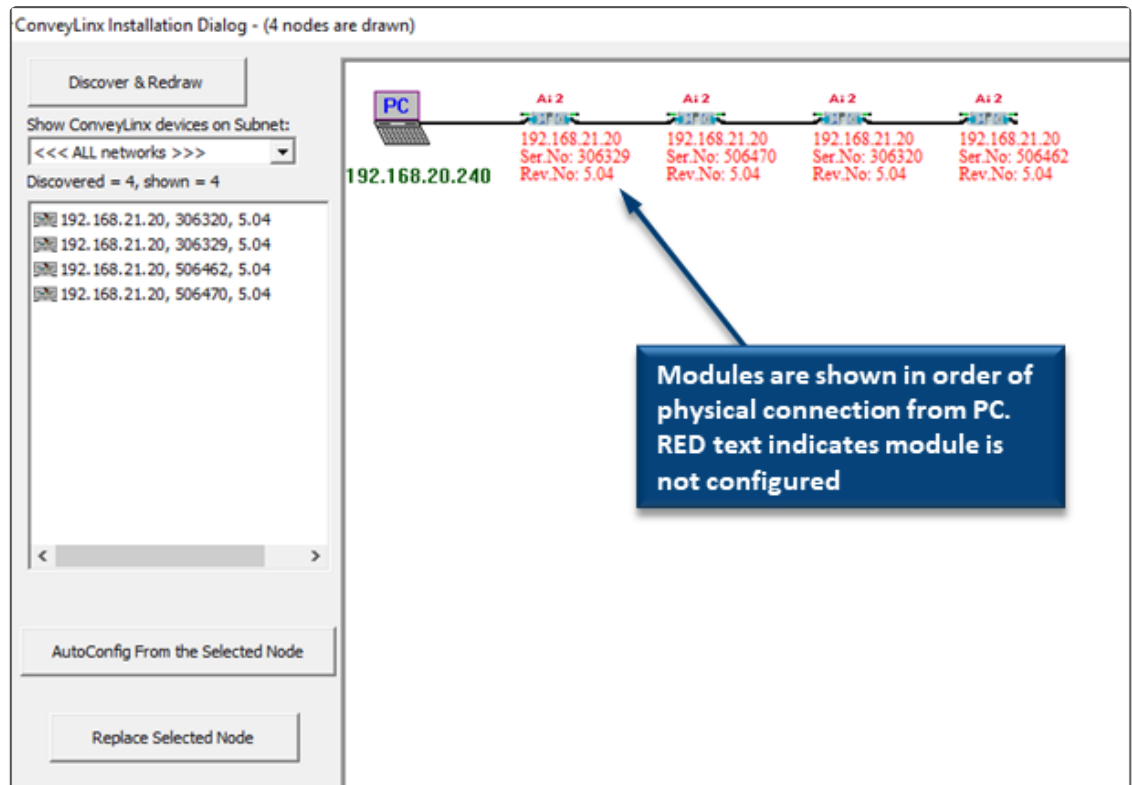
If the most upstream module is the one closest to the PC, select it, then right click and select *AutoConfig from Here*. This will start the procedure.
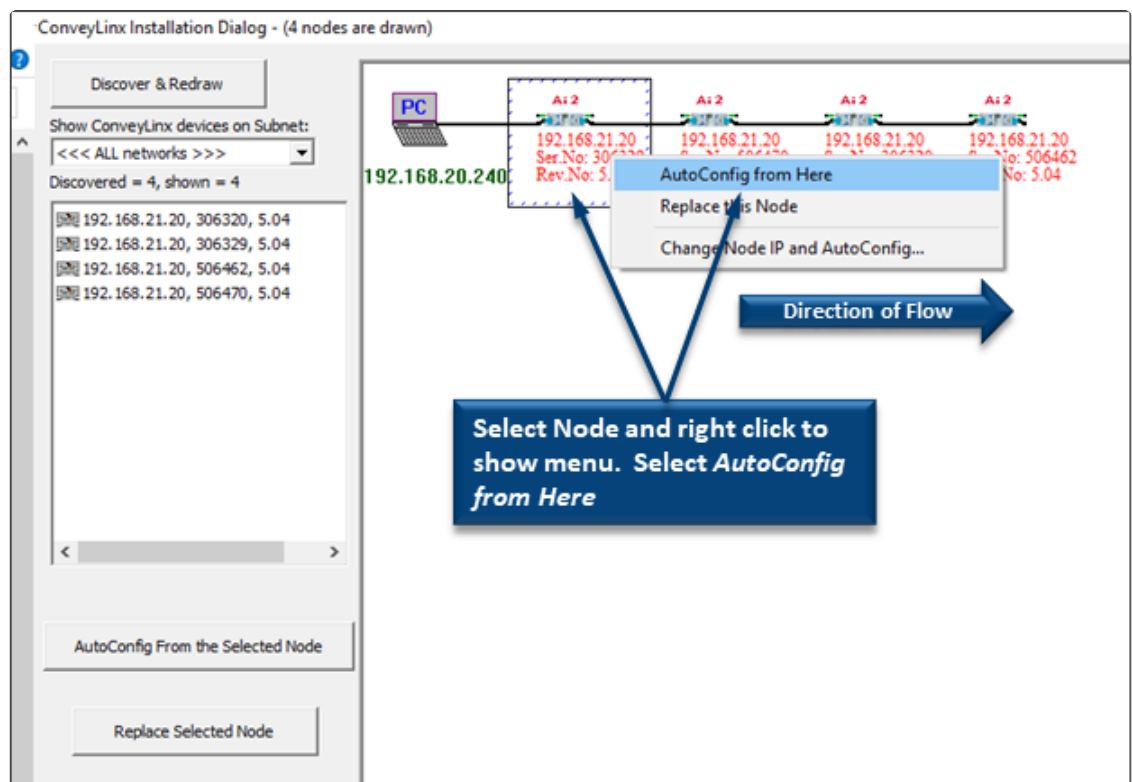
When the procedure is done, you will see the text go from red to black and you will notice that the I.P. addresses have updated per the direction of flow
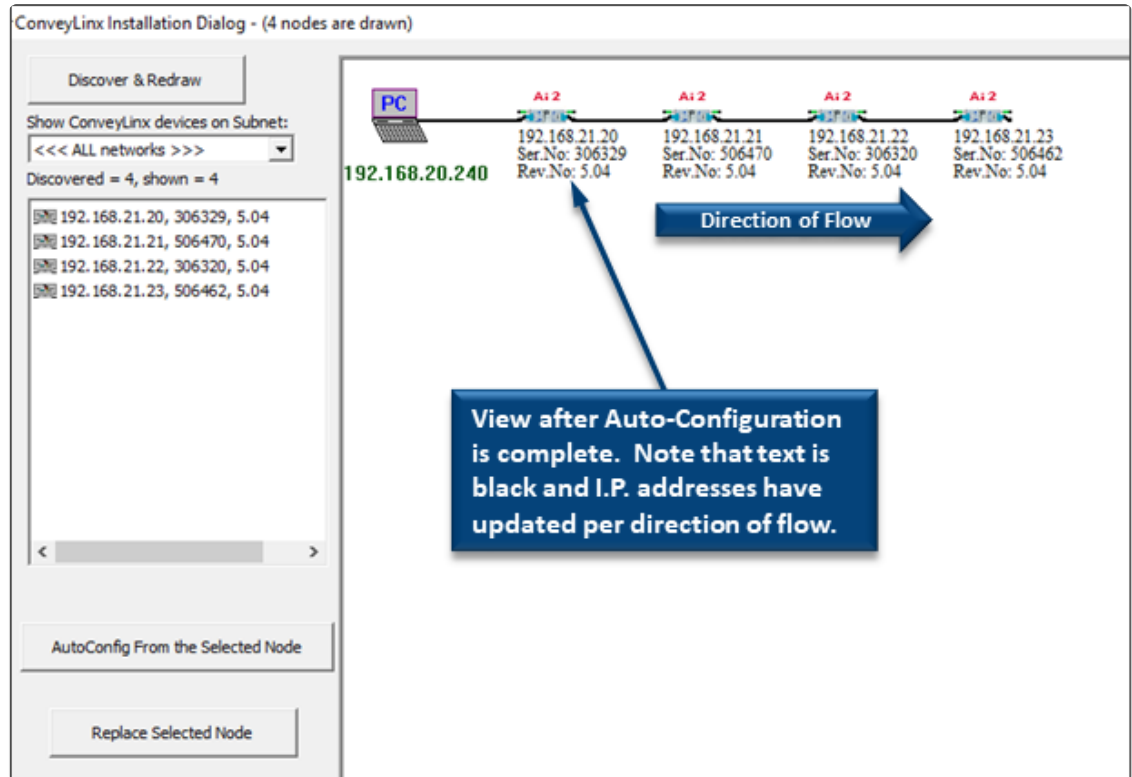
To Change the Direction of Flow – Select the Module at the Opposite End of the Network

If the most upstream module is the one farthest from the PC, select it, then right click and select *AutoConfig from Here*. This will start the procedure.
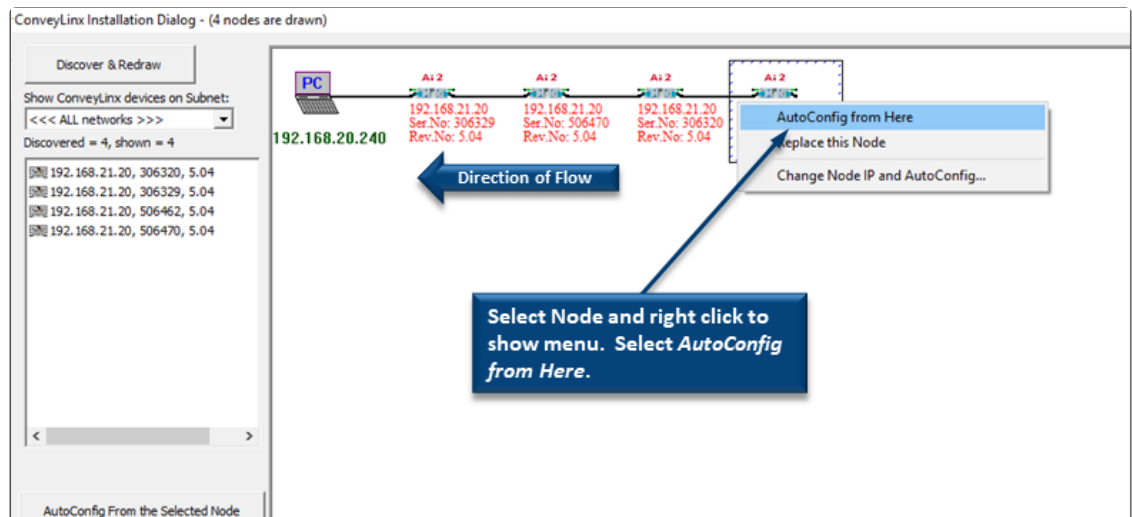
When the procedure is done, you will see the text go from red to black and you will notice that the I.P. addresses have updated per the direction of flow
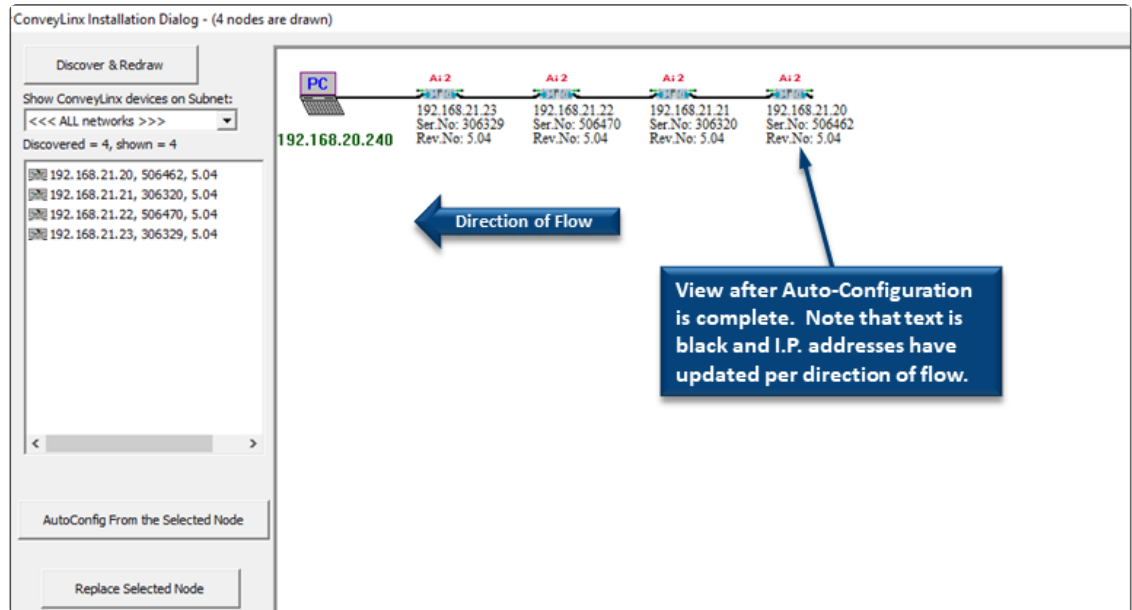
✱ Please note that the time to complete the *Auto-Configuration Procedure* is dependent on the number of modules being configured. Larger networks will take more time than smaller networks

5.7. Expected Results

When the *Auto-Configuration Procedure* routine is complete, each ConveyLinx-Ai Family Controllers will automatically reboot. When a ConveyLinx-Ai Family Controllers has been successfully configured and rebooted, its *Module Status* LED will blink on and off green.

When the module has been properly configured and is operational, its *Module Status* LED will blink on and off *Green*

Please note that the location of the *Module Status* indicator is different between Ai2 based and Ai3 based controllers

> ✳ Please note that the time to complete the *Auto-Configuration Procedure* is dependent on the number of ConveyLinx-Ai Family Controllerss being configured. Larger networks will take more time than smaller networks

How to verify success

Conveyor Operation

Place a carton on your empty conveyor so that it blocks the most upstream zone's sensor. It should convey all the way to the discharge end and the last most downstream zone should try to convey it off the end of its zone. If this does not happen, then at the point where the carton stops, check that module's motor connection is sound and that the zone's sensor is properly aligned. If the sensor was not properly plugged in and powered when the *Auto-Configuration Procedure* was performed, you will have to perform the procedure again. If the sensor was powered but is misaligned when the *Auto-Configuration Procedure* was performed, you can either perform the procedure again or use *EasyRoll* to change the logical polarity of that particular sensor.

Examine the Network

While the *Auto-Configuration Procedure* is in process, you should see all of your modules' LEDs flashing on and off red. If this is not the case, then check your Ethernet cables and connections and power connections.

To verify that all the modules you expected to be configured are in fact configured, with *EasyRoll* you can connect to your newly configured ConveyLinx-Ai Family Controllers network and perform a *Discover* function. The *Discover* function will display all modules that it finds and from the list your most upstream ConveyLinx-Ai Family Controllers should have the 4th octet of its IP Address as .20 and you should see each module you configured in the list.

5.8. What to do if things go wrong

Module Status LEDs Blinking Green with unexpected results

- Check that all sensors are operational and that all zones are clear then perform procedure again. How to change block/clear sensor logic without having to perform another Auto-Configuration
- Check all network and power connections and make sure there is 24V on each ConveyLinx-Ai Family Controllers then try the Auto-Configuration procedure again
- Verify that all of the module sensor/motor combinations are valid. Check here to find out what module set-ups will generate errors

Module Status LED is Blinking or Flashing Red

- Make sure there are no other Ethernet devices (PC's, Switches, Scanners, etc.) connected between any of the module on your network. The Auto-Configuration procedure aborts when it encounters a non ConveyLinx device along the network path before reaching the last module. ConveyLinx-Ai Family Controllerss up to that point should be configured properly but the remaining modules will not.
- When removing a ConveyLinx-Ai Family Controllers from an existing configured network that is already operational; be sure to wait 1 or 2 minutes to allow the Ethernet Switches on the modules to reset their MAC Address ARP table before attempting a new Auto-Configuration procedure

6. Default Settings and Operation

After Auto-Configuration, the modules on your newly configured network are in ZPA mode and set to Singluation Release.

If all module and/or Network Status LED's are blinking green; then to fully verify configuration is to place a single carton onto the most upstream zone and see that it conveys to the discharge zone. If it does, then the Auto-Configuration Procedure is successful.

Learn About:

Singulation Release Mode
Flex Zone Recognition
Jam Conditions
Automatic Module Replacement
How to Reset to Factory Defaults

6.1. Singulation Release Mode

Singulation Release is the mode where the ConveyLinx-Ai Family Controllers requires that its neighboring downstream zone must be clear before it releases an item.

All 3 Cartons are accumulated

Carton 1 releases – Cartons 2 and 3 remain accumulated

When Carton 1 clears PE1 then Carton 2 releases
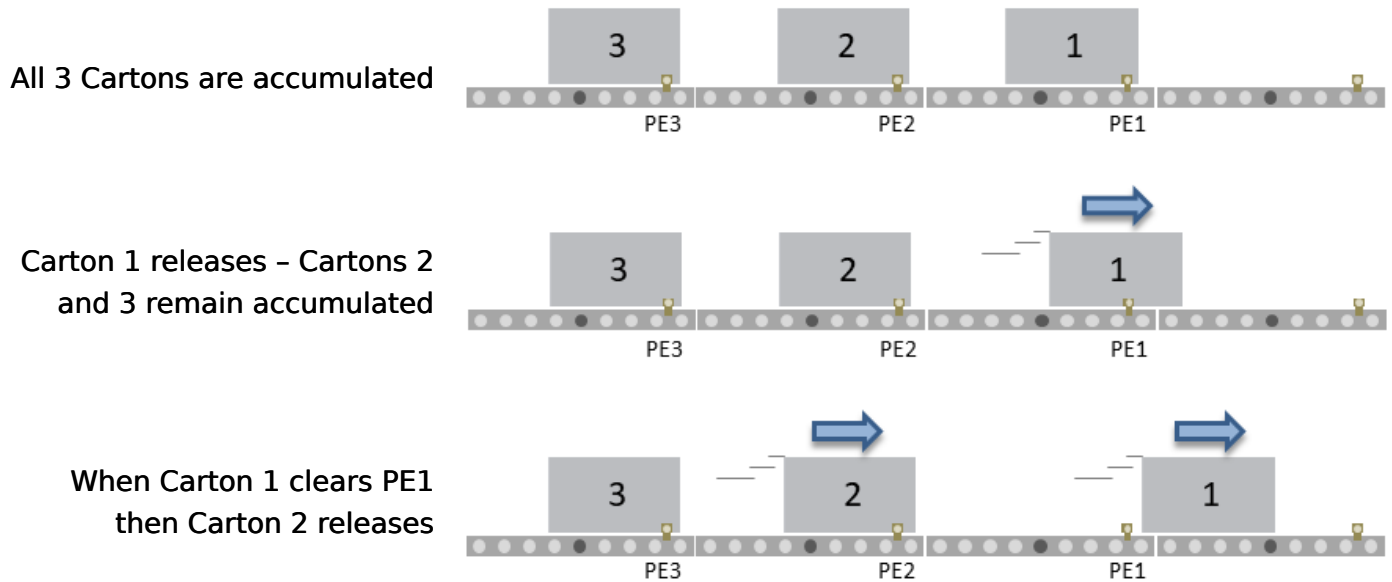
✱ View other ZPA Release Modes and how to change them

6.2. Flex Zone Recognition

ConveyLinx-Ai Family Controllerss will automatically detect that a given carton is longer than one zone length and automatically adjust accumulation control so that the longer carton occupies two logical zones and will keep the next upstream carton from conveying into the longer carton.

> ❗ Flex Zone mode only functions in singulation release mode

All 3 Cartons are accumulated and the Zone at PE3 has Flex Zone enabled
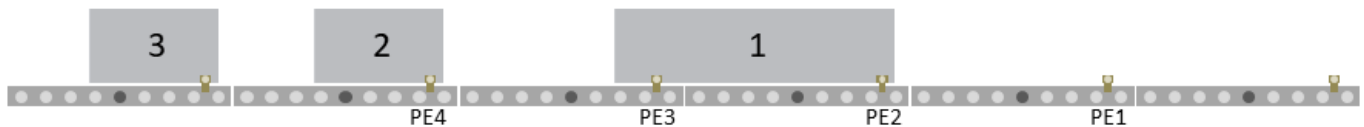
Carton 1 Releases – Even though PE3 is clear, because it has Flex Zone enabled, Carton 2 remains accumulated
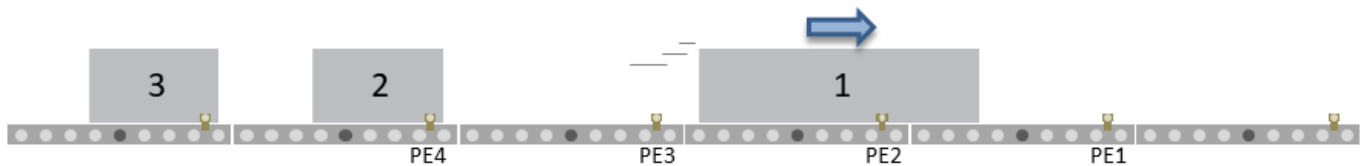
When Carton 1's leading edge reaches PE1, Flex Zone is cleared from the Zone at PE3 and Carton 2 releases
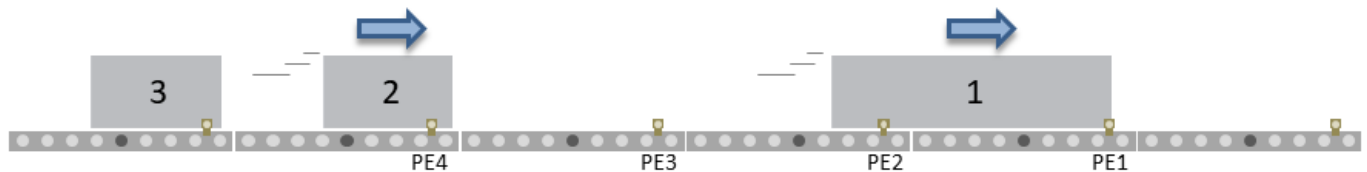
! Please note that Flex Zone mode operates for carton lengths up to 2 zone lengths only. Operating conveyor system with cartons whose lengths are in excess of 2 zone lengths may produce undesirable results such as excessive detected jam conditions and faults.

How to disable Flex Zone operation with EasyRoll

6.3. Jam Conditions

There are two (2) types of Jam conditions detected by the ConveyLinx-Ai Family Controllers:

Sensor Jam

Arrival Jam

✱ Both of these Jam conditions utilize a *Jam Timer* that has to expire in order for the condition to be active. Once either of these jam conditions becomes active; they will automatically clear in the logic after the *Auto Clear Time* has expired. Both the *Jam Time* and *Auto Clear Time* values are 5 seconds each by default. Please see How to Change Jam and Auto Clear Timers

6.3.1. Arrival Jam

When a carton leaves an upstream zone and is conveyed to its next downstream zone, this upstream zone expects positive confirmation of carton arrival from the downstream zone. This communication occurs automatically along the ConveyLinx network. If a new carton arrives at this upstream zone while this upstream zone is waiting for a downstream arrival confirmation, the new carton will accumulate on this upstream zone. If the upstream zone does not receive this confirmation within the *Jam Timer* interval, the ConveyLinx-Ai Family Controllers will produce an *Arrival Jam* fault. Once an *Arrival Jam* occurs, the ConveyLinx-Ai Family Controllers will automatically hold any new carton at the upstream zone for a pre-determined *Auto Clear Time* Timer value and then resume normal ZPA function. By default, the *Jam Timer* and the *Auto Clear Timer* values are set to 5 seconds for each. This condition will be indicated on the corresponding Sensor LED on the module

Carton 1 has left PE1 and is on its way to PE2 and because PE1 is clear, Carton 2 is entering Zone 1
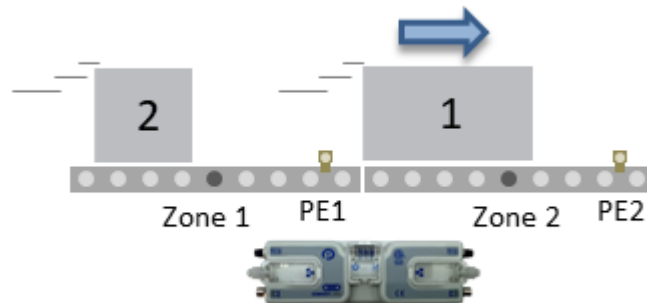
Carton 1 disappears but the Zone 2 motor continues to run for the Jam Time period (5 sec. by default). During this time Carton 2 arrives at PE1 but Zone 1 stops because Jam Timer is still timing
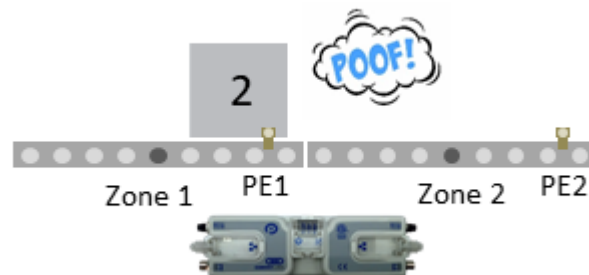
When the Jam Timer expires, the Sensor LED blinks to indicate an Arrival Jam and keeps blinking for the Auto Clear Time (5 sec. by default). When Auto Clear Time expires, the Arrival Jam is reset and Carton 2 will release if Zone 2 is empty
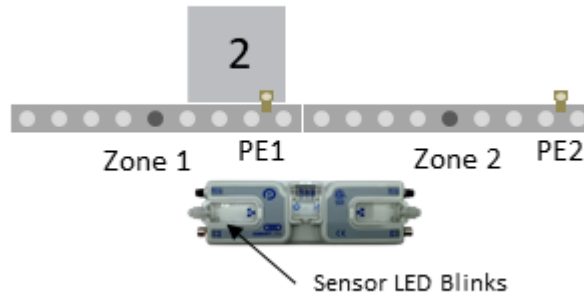
✳ How to Disable Arrival Jam

✳ How to change the Jam and Auto Clear Timers

6.3.2. Sensor Jam

While a zone is releasing a carton; if this carton remains blocking the photo-sensor for the Jam Timer period (default of 5 seconds), the ConveyLinx-Ai Family Controllers will detect a Sensor Jam. This will be indicated on the corresponding Sensor LED on the module as shown here

Carton 1 is leaving Zone 1 but has got stuck. Zone 1 and Zone 2 motors are running but PE1 is remaining blocked
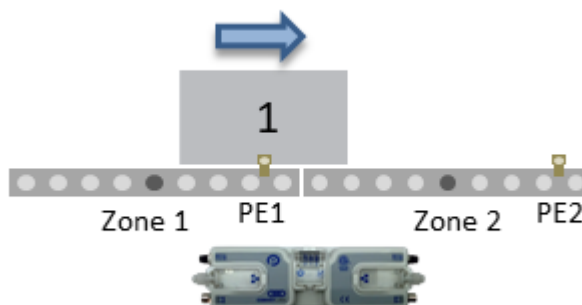
When the Jam Timer has expired (5 sec. by default) Zone 1 and Zone 2 motors stop and Zone 1 Sensor LED flashes green/amber for the Auto Clear Time (5 sec. by default)
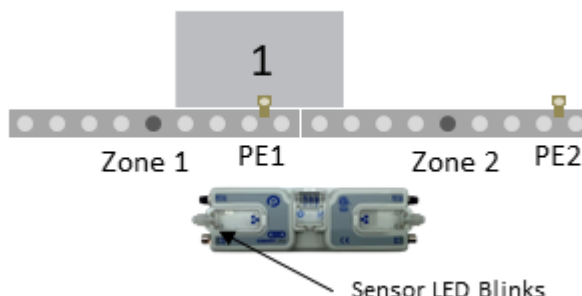
When the Auto Clear Timer has expired, Zone 1 will start performing the Sensor Jam Auto Clear procedure
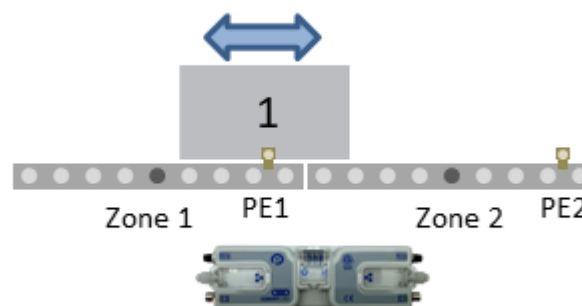
Here is what happens when Sensor Jam Auto Clear procedure occurs

1. Run the zone motor in reverse until the sensor is clear or 1 second has elapsed, whichever happens first
2. Wait for the Auto Clear Timer to expire

3. Run the zone motor forward to attempt discharge to the downstream zone (Attempt #1 complete)
4. If sensor is still blocked after discharge attempt, the zone motor runs in reverse until the sensor is clear or 1 second has elapsed, whichever happens first
5. Wait for the Auto Clear Timer to expire
6. Run the zone motor forward to attempt discharge to the downstream zone (Attempt #2 complete)
7. If sensor is still blocked after discharge attempt, the zone motor runs in reverse until the sensor is clear or 1 second has elapsed, whichever happens first
8. Wait for the Auto Clear Timer to expire
9. Run the zone motor forward to attempt discharge to the downstream zone (Attempt #3 complete)
10. If sensor is still blocked after Attempt #3, the sensor must be cleared manually to reset the Sensor Jam condition. Please note that if any of the discharge to downstream attempts (steps 3, 6, or 9) are successful, the Sensor Jam condition is automatically reset.

✱ If item is still blocking the sensor after Sensor Jam Auto Clear procedure is complete you have to physically clear the jam by removing the item or you can attempt another Auto Clear procedure

✱ The 5 second Jam Timer and Auto Clear Timer values are default settings. How to change Jam and Auto Clear Timers in EasyRoll

✱ How to Disable Sensor Auto Clear Function

6.4. Automatic Module Replacement

Once a linear conveyor has been commissioned by Auto-Configuration Procedure, the ConveyLinx-Ai Family Controllers store configuration data about its upstream and downstream neighboring modules. This configuration data is automatically updated even if the linear conveyor has had its parameters modified by the EasyRoll software. The ConveyLinx-Ai Family Controllers firmware uses this feature to allow for easy module replacement so that the entire linear conveyor does not have to be Auto-Configured again in order to replace a single ConveyLinx-Ai Family Controllers.

> **!** <u>For Automatic Module Replacement to work:</u>
>
> ● The modules initially must have been properly Auto-Configured
>
> ● None of the modules have had their I.P. Addresses changed manually
>
> ● None of the Ethernet cables have been re-routed since initial Auto-Configuration

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

Module Replacement Using EasyRoll

1. Disconnect existing module's motor(s), network, photo-sensor(s), and power connections. The order of disconnection does not matter.

2. Connect new module's motor(s), sensor(s), network connections and power connections.

3. Start EasyRoll V4.04 or higher. Invoke the Advanced Dialog and go to the Network Services tab, click on the Discover button. All modules in the network should be display. Click on AutoConfig button and after a few seconds the topology of the ConveyLinx modules will be displayed.

4. Click on the module in the Tree View you want to add as the replacement module.

5. Right-Click and select Replace this Node from the menu to begin the replacement

process as shown. Note that you can also select the node from the list of modules on the left-hand side and then click the Replace Selected Node button.

6. Wait until Module performs its internal boot-up procedures which will be indicated when the Module Status LED blinks on and off green.
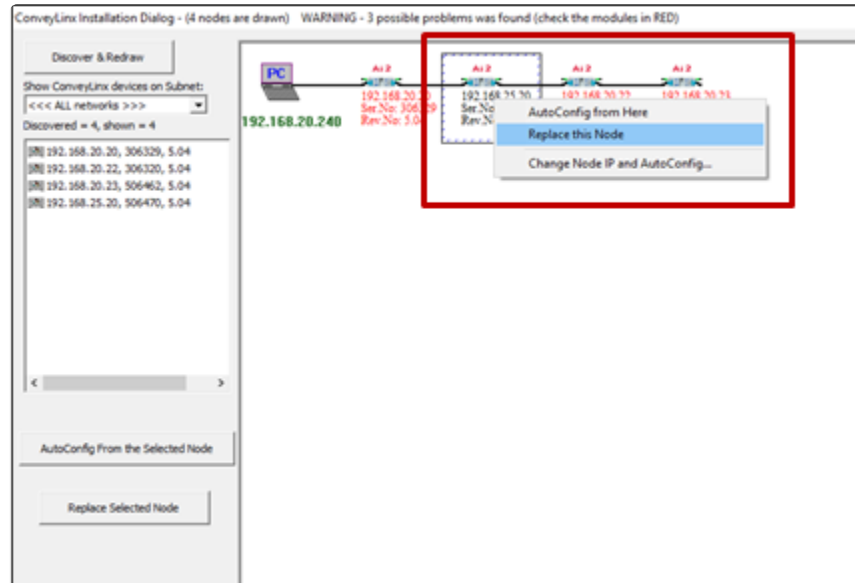
Module Replacement Using Replacement Button

Disconnect existing module's motor(s), network, photo-sensor(s), and power connections. The order of disconnection does not matter. Connect new module's motor(s), sensor(s), network connections and power connections. The *Module Replacement Button* is located in the Left Ethernet cable compartment. Remove the compartment cover to gain access to the button
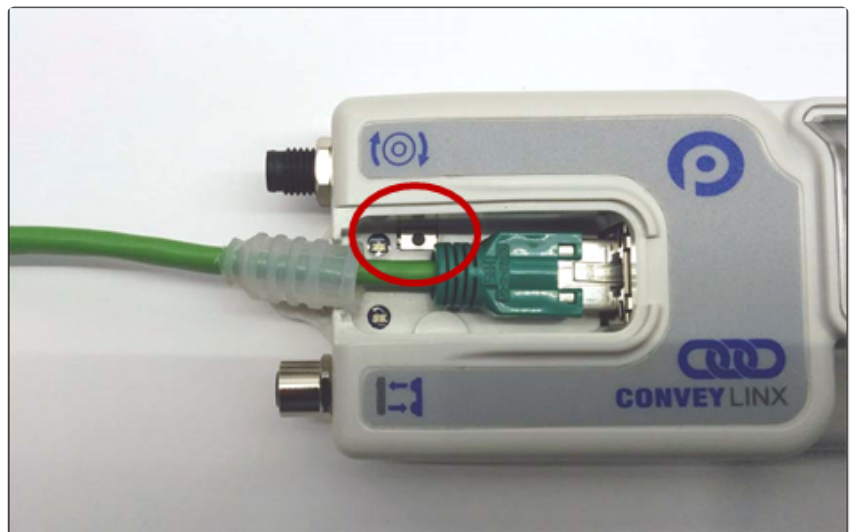
With a pointed instrument (pencil or small screwdriver) press and hold the button for 2 seconds until the *Status LED* flashes red and then release the button
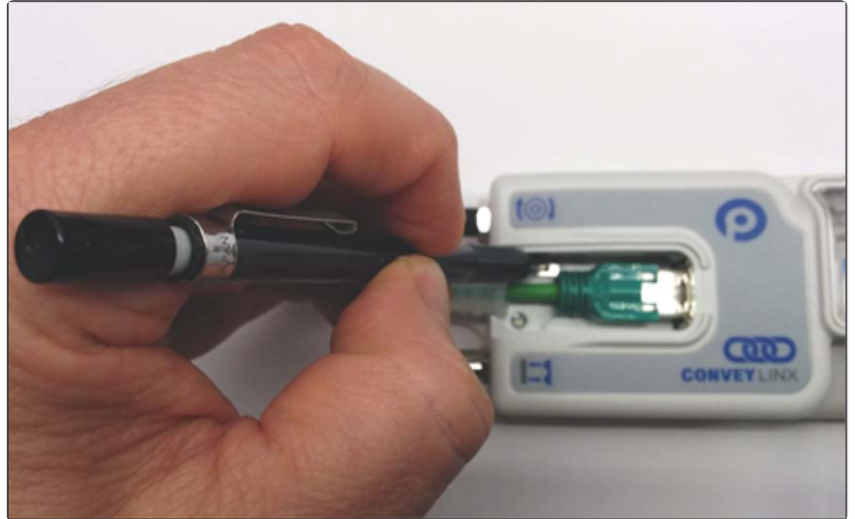
> ✱ Please note that the Module Replacement Procedure (either with EasyRoll or Replacement Button) can take a few minutes to complete. The replaced module will restart itself a minimum of 4 times during the procedure. It will restart 5 times if firmware is being upgraded or downgraded

6.5. Reset to Factory Default Settings

There may be instances when you want to return a ConveyLinx-Ai Family Controllers to its "factory default" state. The procedure to do this is the following:

1. *Unplug all* sensors, devices, motors, and network cables such that the only thing connected is power
2. Press and hold the *Module Replacement Button* until the Module LED begins to flash and then release
3. When the Module Status LED blinks green, the procedure is complete

Factory Default Settings

Item	Value or Setting
IP Address Settings	IP: 192.168.21.20
	Subnet Mask: 255.255.128.0
	Default Gateway: 192.168.202.1
Left & Right Motor Settings	Senergy ECO
	Normal Braking
	1 m/s Speed
	CW Direction
	30 mm Acceleration
	30 mm Deceleration
Module Mode	PLC I/O
Look Ahead & Timing Settings	JAM Timer = 5.00
	Auto Clear Timer = 5.00
	Run After Timer = 5.00

	Sensor Debounce = 0.10
	All Induct Timers = 0.00
Sensors	Sensors are "ON is Blocked"
Connections	All are Cleared

7. EasyRoll Software

✻ Use **EasyRoll** for all <u>**24V based**</u> ConveyLinx controllers

❗ You MUST use **EasyRoll+** for all <u>**48V based**</u> ConveyLinx controllers

[Installing EasyRoll](#)

[Main Screen](#)

[Advanced Dialog](#)

7.1. Installing EasyRoll

The files for *EasyRoll* can be downloaded for free at **www.pulseroller.com** and will be typically 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\

> **!** Please note that some anti-virus and/or security updates block the usage of WinPcap utility which is used by EasyRoll. It is recommended that when you run "Setup.exe" that you "Run as Administrator"

7.2. Main Screen

Indicator	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	Upstream Zone / Downstream Zone – These selections allow you to change the ZPA mode of the particular zone as well as several check boxes to change

	the default operation of certain jam conditions, etc.
4	Upstream Zone / Downstream Zone – Selector to cause the local zone to Accumulate if a carton arrives and to cause the local zone to be in Accumulate mode upon power up of the module
5	Left MDR / Right MDR – Selections for MDR type, braking options, speed, acceleration, deceleration, etc.
6	Left MDR / Right MDR – Selector to click to jog the MDR, click again to stop
7	Left MDR / Right MDR – Visual indicators for various MDR status and diagnostics information
8	Diagnostic Window – Click the graphic image to open a details Diagnostics window

✱ 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
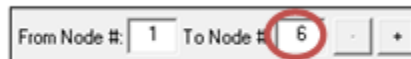
7.2.1. Node Navigation

Whatever value is entered in the *From Node #* box will be the particular module Node data shown in the remainder of the main screen.

The value entered in the *To Node #* box does not have to be the actual "last" Node of the network. If you enter a value higher than what exists; an error message will display after you click the *Refresh* button. For the example shown; if there were only 4 Nodes installed and you entered 6, 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 *From Node #* box and display the module 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

✱ Anywhere on the Main Screen where you see a *Set All* button next to a parameter or data entry selection, all the nodes in the range entered in the *From Node #* and *To Node #* will be updated with the same parameter or data entry selection when you click the associated *Set All* button. A dialog box will appear to confirm your selection.

7.2.2. 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 *From Node #* text box and its information is displayed on the main screen; clicking the *Blink & Wink* switch icon will signal the selected module to blink on and off all of its LED indicators. Click the *Blink & Wink* switch again to turn this off
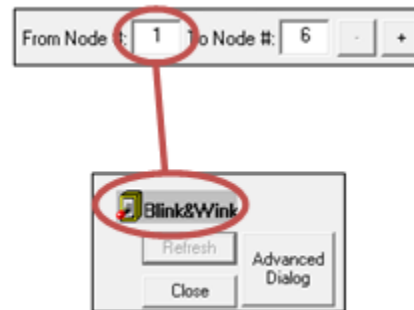
❗ Please note that when Blink & Wink is active, the module stops all motors

7.2.3. ZPA Upstream/Downstream Zone Settings

On the main screen there are areas to configure the *Upstream* and *Downstream* Zone settings. The *Upstream* and *Downstream* settings are set independent of one another so you can customize your ZPA functionality based upon your specific needs.
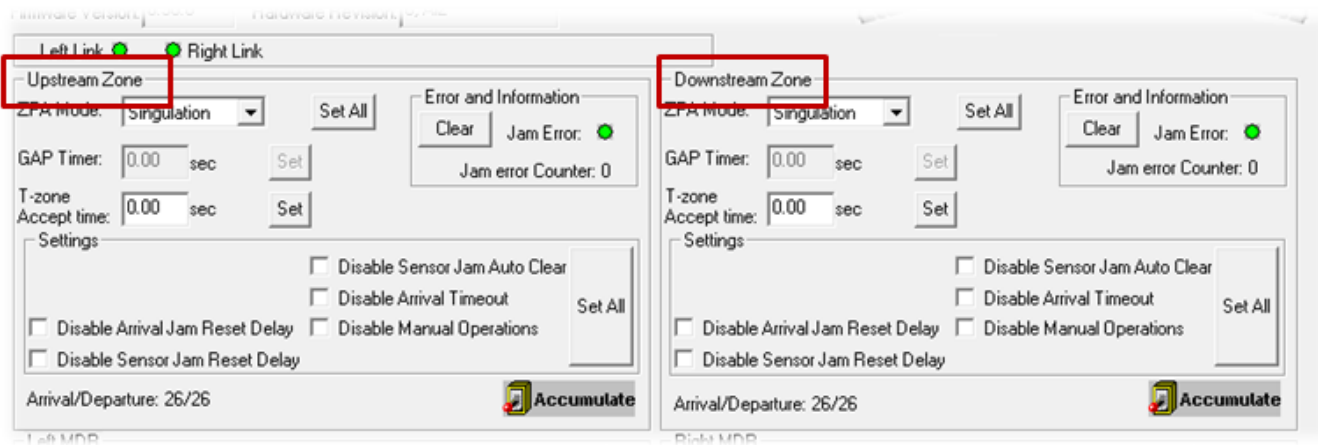
7.2.3.1. Release Mode

Selecting the pull-down box for ZPA Mode will show the available selections. Singulation is the default configuration
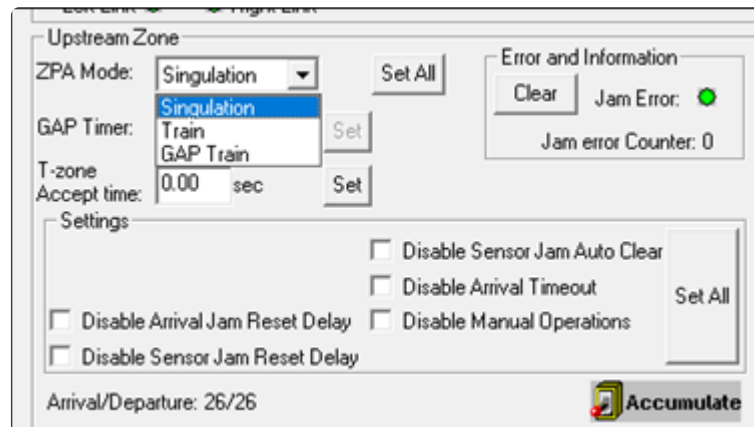
> * Selecting a new setting from the ZPA Mode drop down box immediately changes the zone's mode. If you want to set all Upstream zones for the range of nodes entered in the Node No. text boxes, then click the Set All button. Similarly, you can do the same operation in the "Downstream Zone" portion of the main screen.

> * Singulation Release Mode is the default setting and it is described in the Default Settings and Operation section

> * Please note that Singulation and Train Release Modes are configurable per Zone and can be mixed on the same network

7.2.3.1.1. Singulation Release

✱ Singulation Mode is the default setting and it is <u>described in the Default Settings and Operation section</u>

7.2.3.1.2. Train Release

All 3 Cartons are Accumulated
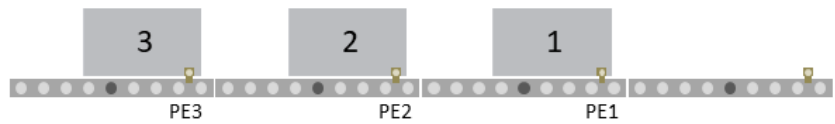
When Carton 1 releases – Cartons 2 and 3 also release at the same time
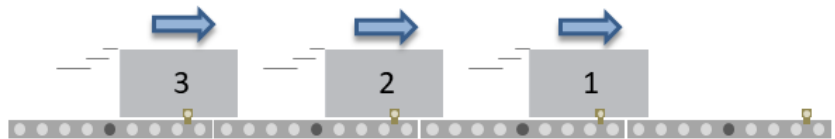
Select *Train* from the pull-down box for *Train Mode*

> **!** Please note that starting many zones in Train Release mode particularly with heavy loads can cause voltage drops on your power supplies. Be sure to take care in sizing your power supply needs. You should consider GAP Train Release Mode if power supply sizing is a concern

7.2.3.1.3. GAP Train Release

All 3 Cartons are Accumulated
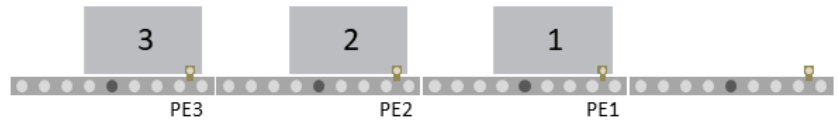
When Carton 1 releases and Zone 2's GAP time has expired, Carton 2 will begin to release. Carton 3 remains accumulated
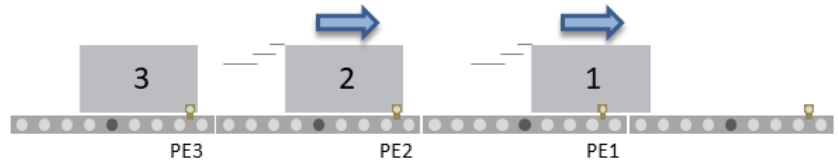
When Carton 2 releases and Zone 3's GAP timer has expired, Carton 3 will begin to release
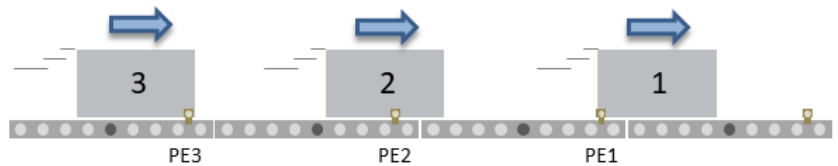
When you select *Gap Train* from the ZPA Mode drop-down box; the *Gap Timer* data entry box and *Set* button are enabled. Simply enter the desired time value and click the *Set* button to update the value in the selected Node. In this example we entered 0.35 seconds
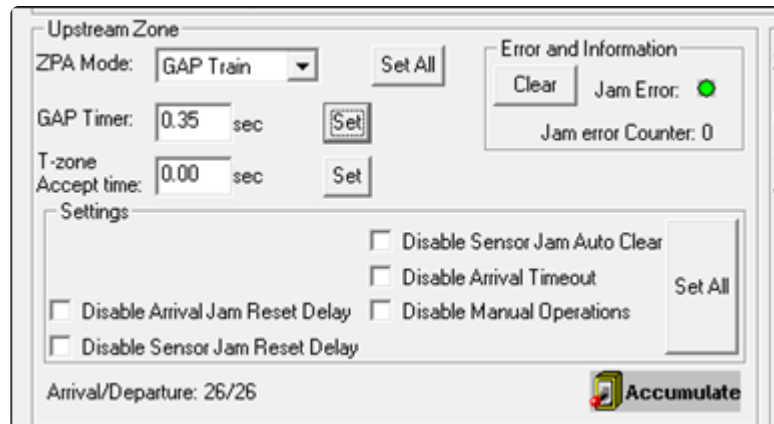
Please note the GAP time <u>does not create a gap</u> or cause any delay when items are in transportation. The Gap timer is only activated when a given zone accumulates an item and then it is allowed to release

✱ If more than one consecutive zone is configured as *Gap Train*; then each of these zones will in turn require that their respective gap timers expire. If the time value is long enough, the result may appear to be *Singulation Release* mode or even Singluation Release with extra time delay. 0.2 to 0.3 seconds are typical values used when you want to help balance the loading on your power supplies but still have the operation be very close to *Train Release Mode*

7.2.3.2. T-Zone Settings

In conveyor applications, transferring a carton at a right angle from one conveyor to another often requires special lifting and lowering mechanisms. In certain applications, one conveyor can simply drive its carton off of its downstream zone directly onto the upstream zone of another conveyor that is perpendicularly oriented. This type configuration is commonly defined as a *T-Zone* arrangement. ConveyLinx contains the logic to control a T-Zone arrangement without requiring any external control interface or programming.
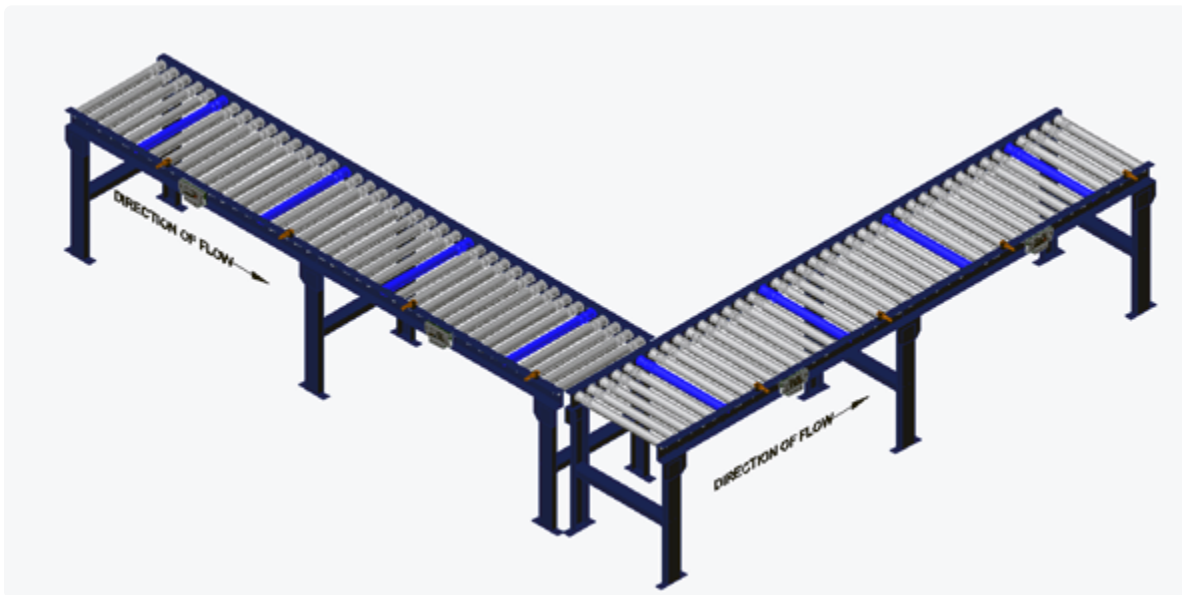
> **!** Material handling considerations such as discharge conveyor speed and load weight have to be analysed prior to implementing a T-Zone configuration. Be sure to verify your mechanical design and carton characteristics before utilizing a T-Zone arrangement.

The T-Zone Accept Time is the amount of time between the leading edge of the Accepting Zone's Sensor becoming blocked and the starting of the Accepting Zone's motor. This time delay allows the Sending Zone enought time to fully transfer the item into the Accepting Zone.

T-Zone Between 2 Separate Modules
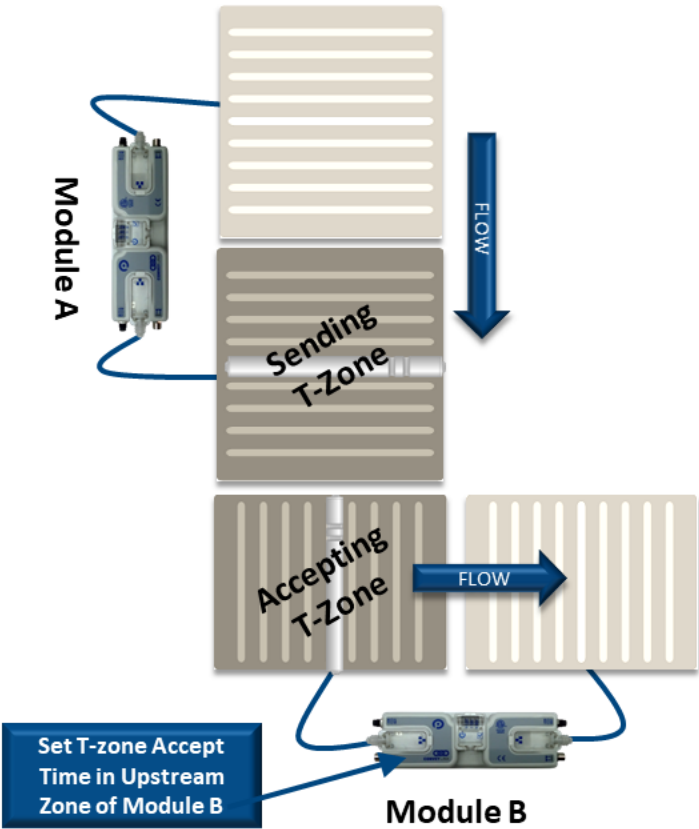
T-Zone on a Single Module
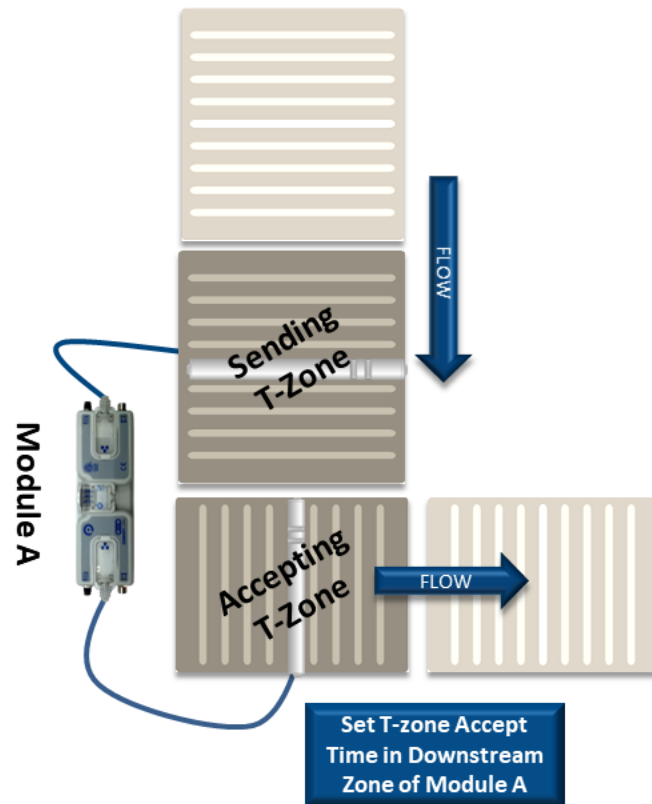
Enter the value and click the Set button. In this example we entered 1.5 seconds. Whether to change the Upstream Zone or Downstream Zone value on the main screen is dependent upon which zone is the **accepting zone**

❗ The T-zone Accept time is always applied to the accepting zone

7.2.3.3. ZPA Error and Information

There is an *Error and Information* area in the upper right corner in both the Upstream and Downstream Zone areas of the *Main Screen*. There is an indicator that shows if a Jam condition is active (either *Arrival Jam* or *Sensor Jam*). There is a numerical indicator of all occurrences of any Jam conditions since the last module power cycle. There is also a Clear button to allow you to clear a *Sensor Jam* if it is active.
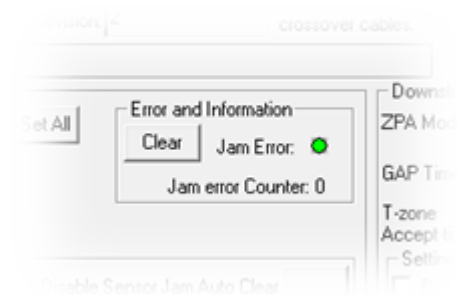
✳ The *Clear* button is a convenience so you do not have to go the zone in question and remove or otherwise clear the item from the zone sensor. If Sensor Jam Auto Clear is enabled, clicking the Clear button will re-initiate this procedure.

7.2.3.4. Accumulate Control from Main Screen

Clicking the *Accumulate* switch icon will place the zone in accumulation mode and the next carton that arrives at that zone will stop and remain until you click the switch again to turn off the accumulation mode

Clicking the icon will initiate the *Accumulate* function and the icon will highlight and change state

Clicking the icon again will turn off the *Accumulate* function

> ✱ When you set *Accumulate* from *EasyRoll*, this is saved to the module's flash memory so that when you cycle power on the module, it will power up in the accumulated state.

7.2.3.5. Settings Checkboxes

Both the *Upstream* and *Downstream* ZPA Zones have Settings check boxes to allow you to customize some of the ZPA behavior. These settings can apply to only the zone you are connected to and viewing or you can apply the same settings to multiple modules in a range of module nodes.

Disable Reset Delays

Disable Sensor Jam Auto Clear

Disable Arrival Timeout

Disable Manual Operation

7.2.3.5.1. Disable Reset Delays

Any individual zone or group of zones can be configured to ignore the Auto Clear Time delay for either or both of the Arrival Jam and Sensor Jam.

Clicking either or both check-boxes will cause the zone's logic to ignore the *Auto Clear Time* delay for the particular jam condition

> **!** Selecting either of these options will not eliminate the detection of the particular jam condition; it simply eliminates the default *Auto Clear Time* delay the logic utilizes before automatically clearing the condition.

7.2.3.5.2. Disable Sensor Jam Auto Clear

Sensor Jam Auto Clear Procedure describes how the logic will make 3 attempts to clear a Sensor Jam if one occurs. There may be instances at specific zones or range of zones where you do not want this functionality to happen. There is a check box that allows you to disable this functionality

Clicking the *Disable Sensor Jam Auto Clear* check box will disable the 3 attempts to clear the *Sensor Jam* and will cause the zone to remain in the jam state until the sensor is cleared manually

⁂ You can also attempt to clear a *Sensor Jam* using the *Clear* button for the Zone

7.2.3.5.3. Disable Arrival Timeout

[Arrival Jam](#) describes how default zone to zone logic utilizes the Jam Time value for the discharging zone to wait for the accepting zone to indicate successful arrival of the carton into the accepting zone. In certain applications for a given zone, you may want to disable this functionality
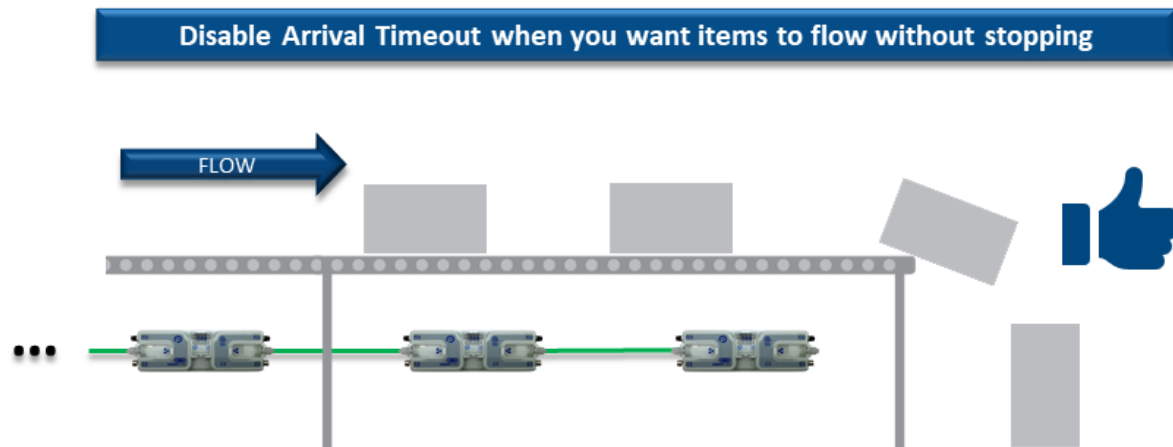
Clicking the *Disable Arrival Timeout* check box will prevent the discharging zone from registering an *Arrival Jam* condition. Upstream cartons will not wait to enter the discharging zone once a carton has left the discharging zone

7.2.3.5.4. Disable Manual Operation

In normal ZPA operation, if a zone is in Accumulation (either by external device or if Aux I/O Pin 2 is set to Accumulate) and the carton is subsequently manually removed from the conveyor; the zone downstream of the accumulated zone will run in an attempt to "find the lost carton". In certain applications or situations where manually removing cartons from accumulated zones is expected; you can disable the running of the downstream zone to "find the lost carton"

Without Disable Manual Operations box checked

Carton accumulated in Upstream Zone by Aux I/O Pin 2 Accumulate signal or external network signal
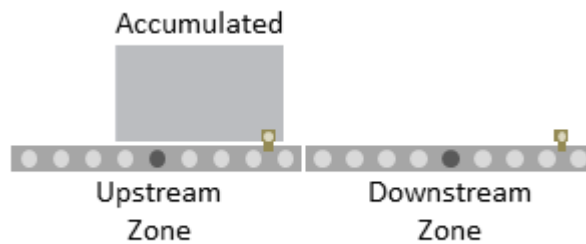
If carton is removed from Upstream Zone, the Downstream Zone will run to "find the missing carton". Run time is the duration of the Jam Timer setting (5 sec. by default)
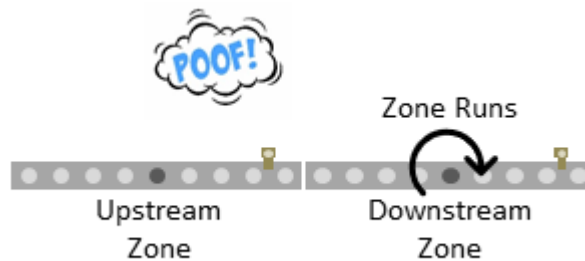
With Disable Manual Operations box checked

You first navigate the main screen to the *zone you want to prevent from running* when it's upstream neighboring zone has its carton removed. Clicking the *Disable Manual Operations* check box will prevent this zone from running when its upstream neighbor has an accumulated carton manually removed
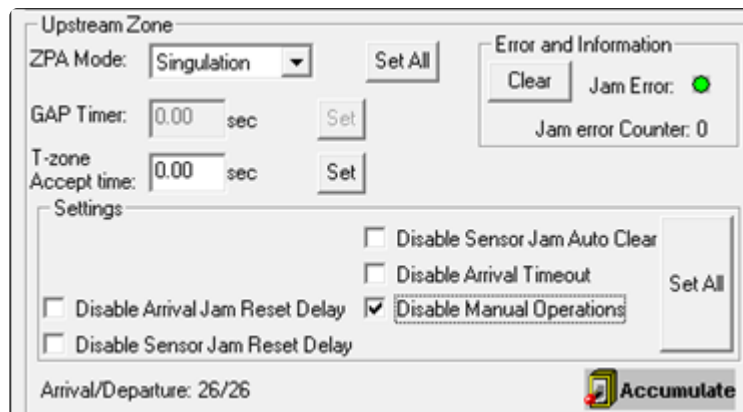
When you check the *Disable Manual Operations* check-box for the Downstream Zone, then it will not run when upstream carton is removed
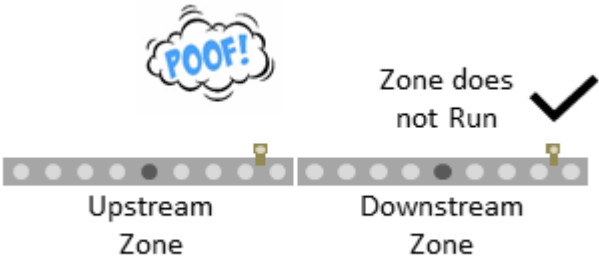
7.2.4. Motor Settings

The motor settings for the current selected module can be modified on the Main Screen

> ✳ Please note that the Motor Settings grouped by motor plugged into the physical Left and Right sides of the module and are not based upon Upstream or Downstream product flow.

7.2.4.1. Motor Type

The Motor Type pull-down box lists all motor brand and types whose profiles are available for the module. Senergy-Ai ECO is the default setting upon completion of the Auto-Configuration Procedure. The new settings are downloaded to the selected Node upon selecting a new item from the list.

In this example, clicking *Set All* will download the selected setting to the *Left MDR* of all modules entered in the range of Nodes at the top of the main screen in the *From Node # / To Node #* boxes. If for example these boxes had the values of 1 and 6; changing the selection in the *Motor Type* pull down will change Node 1 and clicking Set All will change Nodes 2 thru 6 to the same setting as Node 1
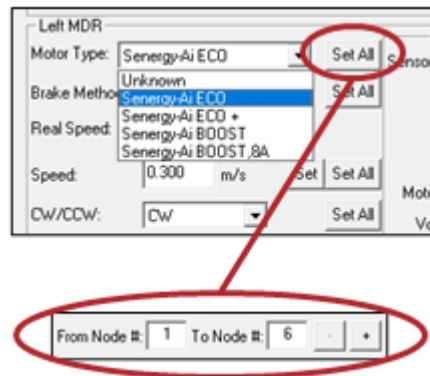
> **!** Please consult the Pulseroller Catalog Senergy Ai motor documentation and review your application if you are unsure as to which motor-type setting to use

7.2.4.2. Brake Method

The *Brake Method* pull-down box lists all the holding brake methods available for the module. *Normal* is the default setting upon completion of the *Auto-Configuration Procedure*. The new settings are downloaded to the selected Node upon selecting a new item from the list.

In this example, clicking *Set All* will download the selected setting to the *Left MDR* of all modules entered in the range of Nodes at the top of the main screen in the *From Node # / To Node #* boxes. If for example these boxes had the values of 1 and 6; changing the selection in the *Brake Method* pull down will change Node 1 and clicking *Set All* will change Nodes 2 thru 6 to the same setting as Node 1

Method	Description
Normal	Once the controls have decelerated the rotor to a stop, the motor coil are internally connected. The permanent magnet forces in the rotor and the mechanical inertia of the gearbox holds the rotor in place. This is the MDR industry standard holding brake method and is often termed short circuit or shunt. *Normal* is the default factory setting for all module zones from the *Auto-Configuration Procedure*
Free	Once the controls have decelerated the rotor to a stop, the motor coils are internally disconnected and only the mechanical gearbox inertia holds the rotor in place
Servo Brake	When the controls have decelerated the rotor to a stop, the processor notes the *Hall Effect* sensor status. If the *Hall Effect* sensor status changes indicating a change in position of the rotor, the controls will inject current into the motor coils in the proper sequence to move the rotor back to its original stop position
Continuous Torque	When the controls have decelerated the rotor to a stop, the processor monitors the *Hall Effect* sensor status. If the *Hall Effect* sensor status changes indicating a change in position of the rotor, the controls will inject current into the motor coils to generate torque in order to try and stop the rotor. Unlike the *Servo Brake* function, the controls do not try and return the rotor to its original stop position

! Servo Brake function utilizes motor power and depending on the torque demanded by the motor to hold the load the potential for heat build-up exists

✳ *Continuous Torque* Brake method required *EasyRoll* version 4.21 or later

7.2.4.3. Speed

The *Speed* setting value is in meters per second (m/s). The Senergy-Ai motor roller contains data as to its gear reduction ratio and roller diameter that is readable by the ConveyLinx-Ai Family Controllers. *EasyRoll* then uses this information to indicate whether the speed you enter is valid for the connected roller.

Setting Speed Value

For our example roller, entering a value of "0.300" for the speed and clicking Set will set the speed to 0.3 m/s for the roller. If the background of the speed entry box remains white, then the speed is valid for the roller part number detected by the module
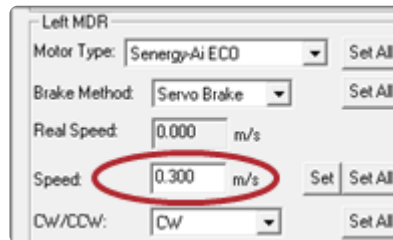
Speed Setting Too High

For our example roller, entering a speed of 2 m/s is over its allowable maximum speed. The background of the speed entry box will change to red indicating the speed setting is too high
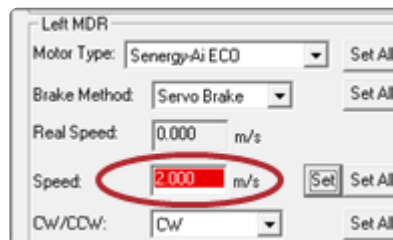
> ✽ Setting the speed too high will result in the motor running at is maximum defined speed

Speed Setting Too Low

For our example roller, entering a speed of 0.1 m/s is under its allowable minimum speed. The background of the speed entry box will change to blue indicating the speed setting is too low
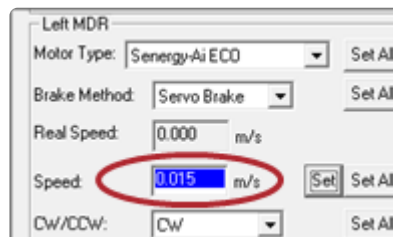
> ✽ Setting the speed too low will result in the motor running at is minimum defined speed

Setting Same Speed for Multiple Modules

Type in the desired speed in m/s. You can also use the Set All button to set the entered speed to each motor in the *From Node # / To Node #* boxes. In this example the Left MDRs from nodes 1 thru 6 will each get the same speed setting

7.2.4.4. Rotation Direction

This setting is either Clock-Wise (**CW**) or Counter-Clock Wise (**CCW**)and is determined for each module based upon the *Auto-Configuration Procedure* results. Please refer to section [Motor Rotation Definition](#) for details on how to determine rotation direction with respect to MDR installed orientation.

Selecting a direction from the drop-down box immediately sets the rotation direction. You can also use the *Set All* button to set the selected rotation direction to each motor in the *From Node # / To Node #* boxes. In this example the Left MDRs from nodes 1 thru 6 will each get the same *CW/CCW* rotation direction setting
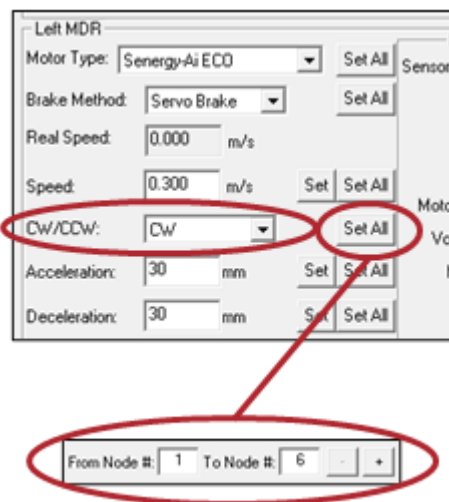
✱ Please note that if you change the *Rotation Direction* and this module goes through another *Auto Configuration* procedure, the *Rotation Direction* setting **will not** reset back to default. It will remain unchanged at the last selected setting.

7.2.4.5. Acceleration/Deceleration

Acceleration and Deceleration settings are fields you enter values similar to Speed. The units of the value you enter are in mm which means that the control will accelerate and decelerate for the distance specified within the controller's current limits and the mechanical limits of the connected drive train.

In this example, clicking *Set All* will download the selected setting to the *Left MDR* of all modules entered in the range of Nodes at the top of the main screen in the *From Node # / To Node #* boxes. If for example these boxes had the values of 1 and 6; changing the selection in the *Motor Type* pull down will change Node 1 and clicking Set All will change Nodes 2 thru 6 to the same setting as Node 1
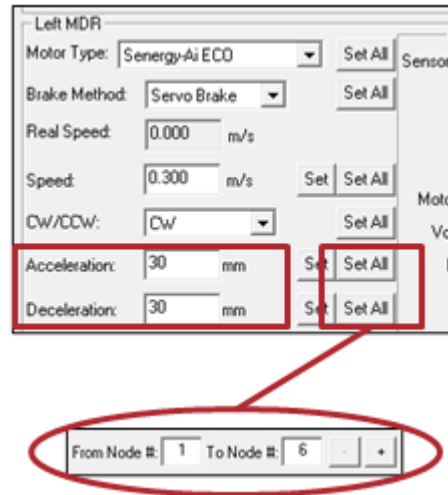
7.2.4.6. Motor Jog and Error Indicators

Motor Jog (Run) Function

In *EasyRoll*, the Left and Right motors have their own *Run* buttons on the *Main Screen*. These are used to temporarily run the motors for verification of function and diagnostics.

Click the *Run* icon to turn on the motor*
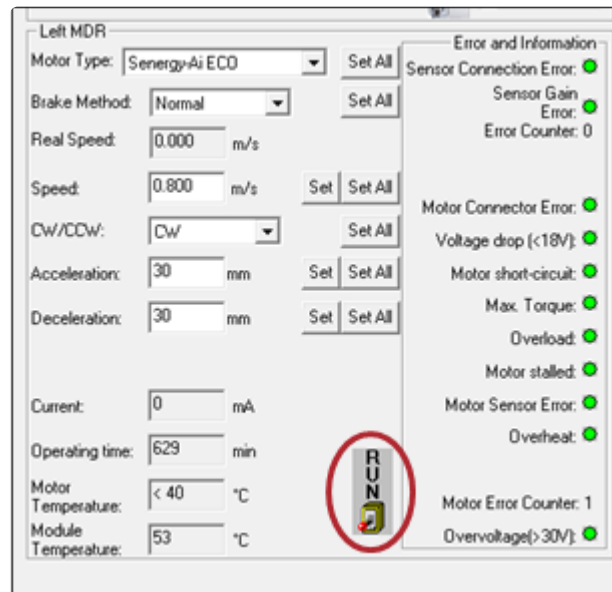
When *Run* is activated the icon highlights. Click the icon again to turn the motor off
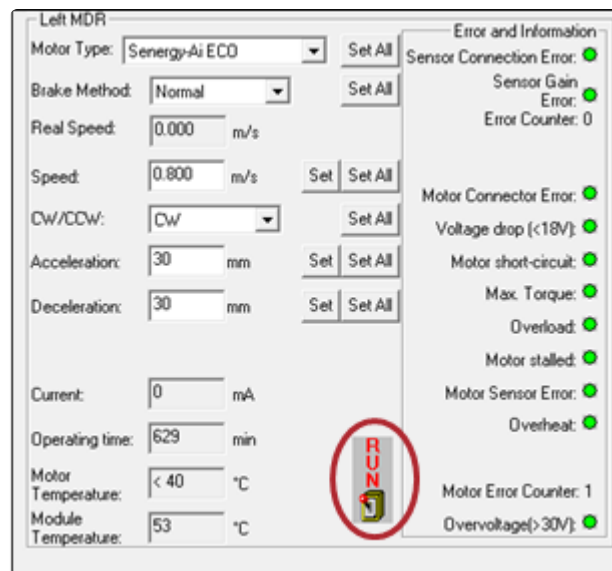
✱ Explanation of unexpected rotation direction when you run a motor from this jog function

Error Indicators

In *EasyRoll*, the Left and Right motors have their own status indicators on the *Main Screen*. These are used to provide visual indication of certain conditions. There is also status indicators for the Sensors connected to the Left and Right sides of the modules.
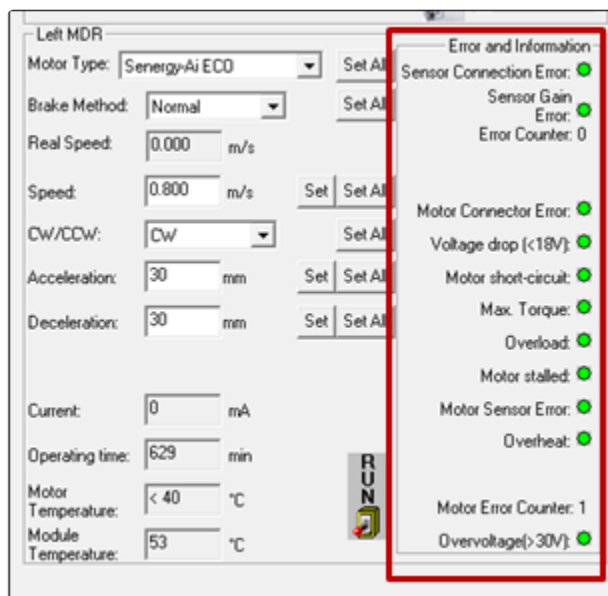
Indicator	Description
Sensor Connection Error	Occurs when the particular side of the module was Auto Configured as a zone and the sensor is unplugged or otherwise not drawing power
Sensor Gain Error	Occurs when the sensor health/gain signal on pin 2 of the Sensor port changes state from its Auto-Configuration state. This is only applicable if the Aux I/O Pin 2 Usage has been set to "Sensor Error (DI)"
Error Counter	A running numerical counter that increments for each occurrence of connection or gain errors. Counter resets to 0 upon power cycle
Motor Connection Error	Occurs when a motor that is expected to be connected becomes disconnected by either unplugging or internal damage
Voltage drop [<18V]	Occurs if the incoming power to the module drops below 18 volts
Motor short circuit	Occurs if the module detects a short circuit condition
Max. Torque	Indicator turns Yellow if the motor is delivering the maximum torque according to the selected motor profile
Overload	Occurs if motor is commanded to run and is in a stalled condition for more than 20 seconds
Motor Stalled	Occurs when the motor is commanded to run and the rotor is not

	turning at least at 10% of its set speed
Motor Sensor Error	Occurs if one or more of the 3 Hall Effect Sensors does not produce a signal or does not change state when motor is running
Overheat*	Occurs when the motor temperature has exceeded 105°C or the module temperature has exceeded 90°C. When the motor temperature reaches 105°C the motor is stopped and is not allowed to run again until the temperature falls to 80°C
Motor Error Counter	A running numerical counter that increments for each occurrence of any of the motor error conditions. Counter resets to 0 upon power cycle
Overvoltage [>30V]	Occurs when the incoming power to the module exceeds 30V

*When the motor temperature reaches 101°C, the current is limited to 1.3A and the motor is still allowed to run.

7.2.5. Diagnostic Window
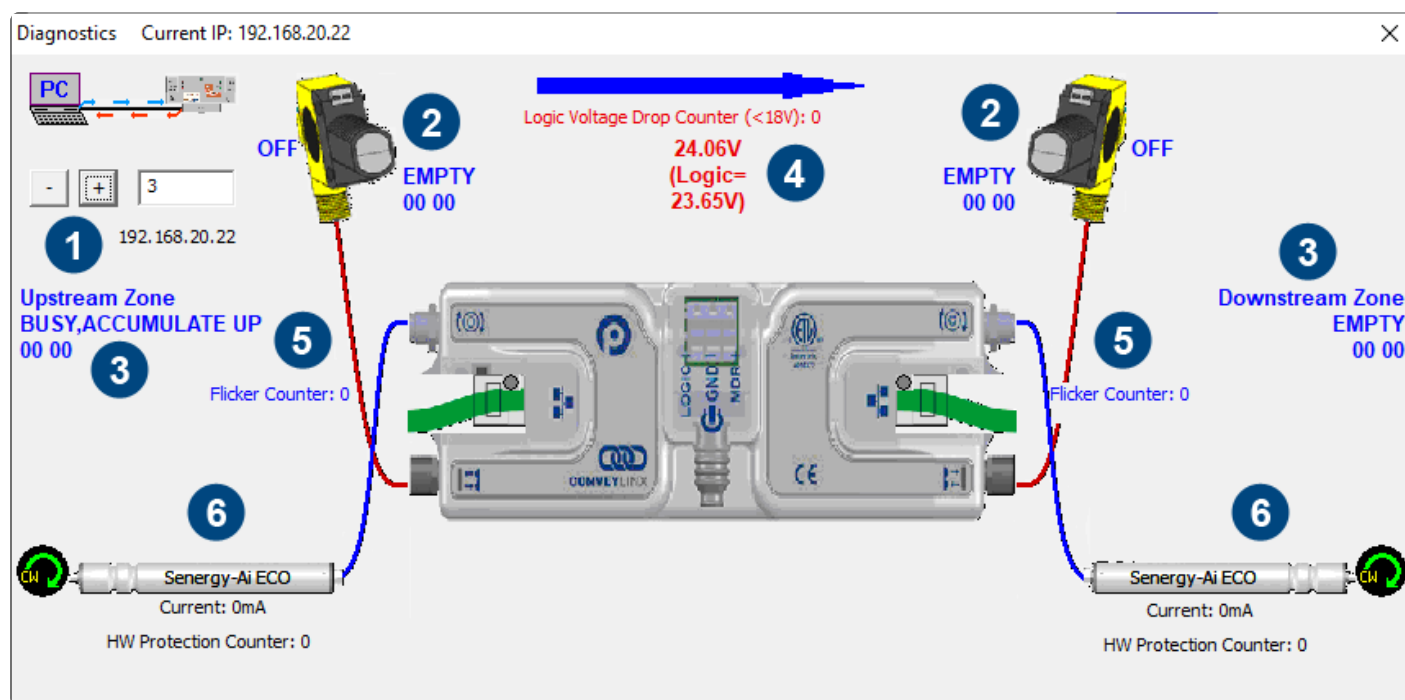
Diagnostic Window Example

Indicator	Description
1	You can navigate to the next or previous module's Diagnostic Window by clicking the + and – buttons
2	Displays the current ZPA status of the zone and the state of the sensor
3	Displays the status of the upstream and downstream zones to the selected module. Note that the values depicted below the text ("00 00" in the example) indicate the contents of the Tracking Registers.
4	Displays the current input voltage to the module as well as the count of the number of times the power supply went below 18 volts but did not completely shut off. This is useful for diagnosing possible power supply issues
5	Displays current value of the sensor input's Flicker Counter
6	Displays current status of each motor. Please note that this data is also shown on the main screen as well

Accumulation Status

If a carton is accumulated on a particular zone, the Diagnostic Window will indicate a reason for the accumulated state. Also note that in situations where an external device (PLC or PC controller) or the *Accumulate* icon on the Main Screen has been activated; the Sensor LED on the module for the stopped carton's zone will be fast blinking green. You can then consult the Diagnostic Window for more detailed information on the exact reason

As shown in this example, for the upstream zone on the selected Module, it shows "PLC" as the reason a carton is stopped at this zone even though its immediate downstream zone is empty
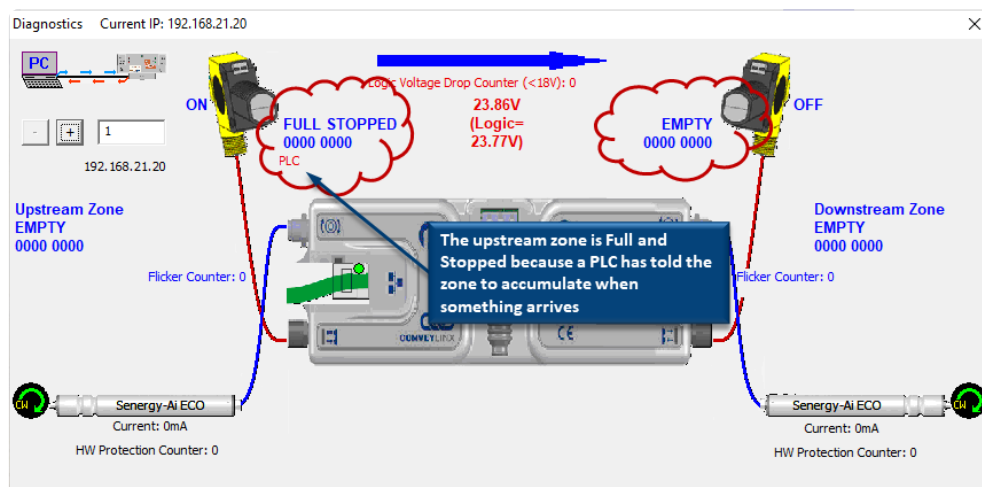
Motor Roller Information

One very useful feature of the Diagnostics window is that you can mouse-over the image of each roller and a pop-up will display the pertinent motor parameters as well as the part number and serial number of the connected roller as shown
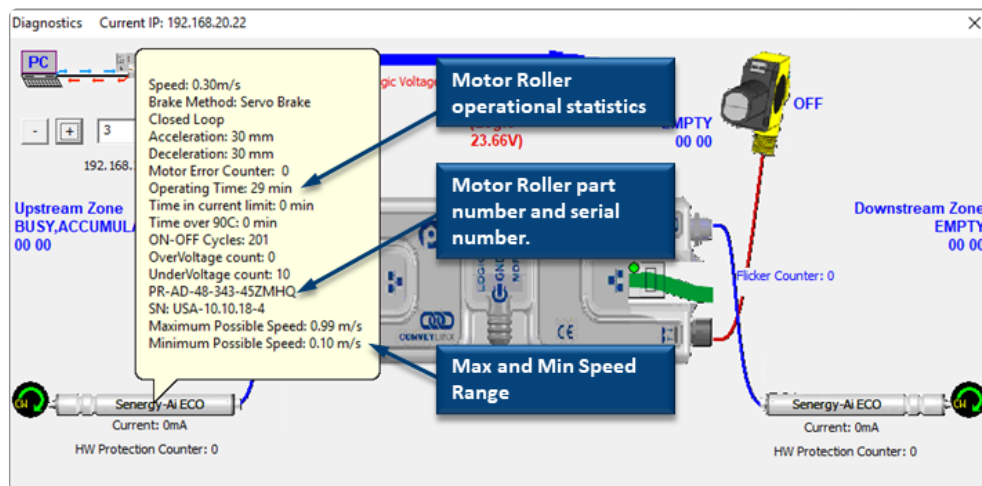
Flicker Counter

The *Flicker Counter* contains a running total of the number of state changes that have occurred during the *Sensor Debounce* time window. This can be used as diagnostic information to help you adjust the proper Sensor Debounce time for a given application

7.3. Advanced Dialog

The Advanced Dialog is where you can make further changes to module operation and access utilities to help maintain your system of modules.

To invoke the ConveyLinx Advanced Dialog you can do any of the following:

- Click *Advanced Dialog* button
- Press F2
- Simultaneously press [Ctrl] [Shift] U

The *ConveyLinx Advanced Dialog* pops up over the *Main* screen display and defaults to show the *Look Ahead & Timing* tab

7.3.1. Look Ahead and Timing Tab

The Look Ahead and Timing Tab contains 4 sections of settings pertaining to ZPA functionality:

Look Ahead
Jam & Auto Clear Timers
Run After & Induct
Sensor Debounce

7.3.1.1. Look Ahead Slowdown Feature

The *Look Ahead* feature configures the logic to "look ahead" to its next downstream zone and if it is occupied when a carton is entering its zone, the module will dynamically adjust the MDR to the selected speed. This feature in intended to be used in higher speed applications were increased stopping distance is required to keep cartons from over-travelling their stop positions. This function can be applied per zone or for multiple zones

A carton is accumulated and stopped at Zone 1 and another carton is conveying at normal set speed
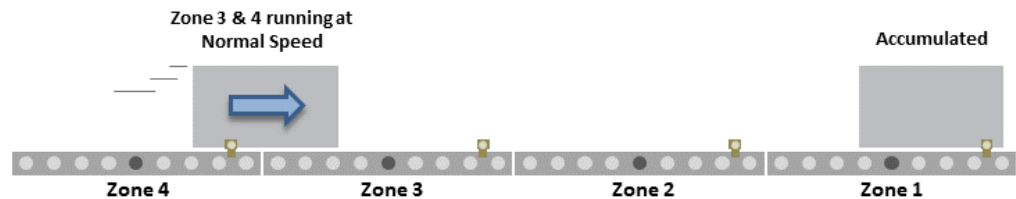
When carton reaches end of Zone 3, Zone 2 will wake up and run at the Look Ahead Speed
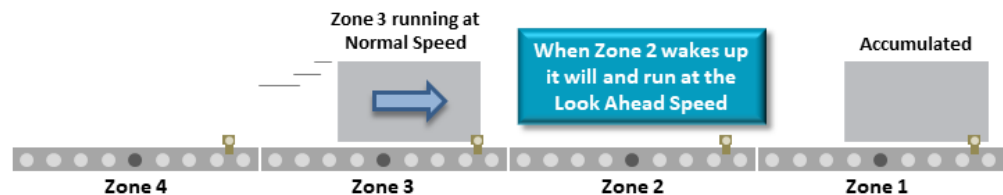
Click the checkbox to enable the *Look Ahead* feature for the selected Node. Clicking the *Set* buttons will download the setting to the respective zone on the selected Node. The value entered for the *Slow Down Speed* is in percent of the Zone's normal speed as set on the main screen. In this example, the *Slow Down Speed* speed will be proportional to 80% of 0.300 m/s. Also note there is a *Set All* button that will apply the settings to the range of Nodes entered
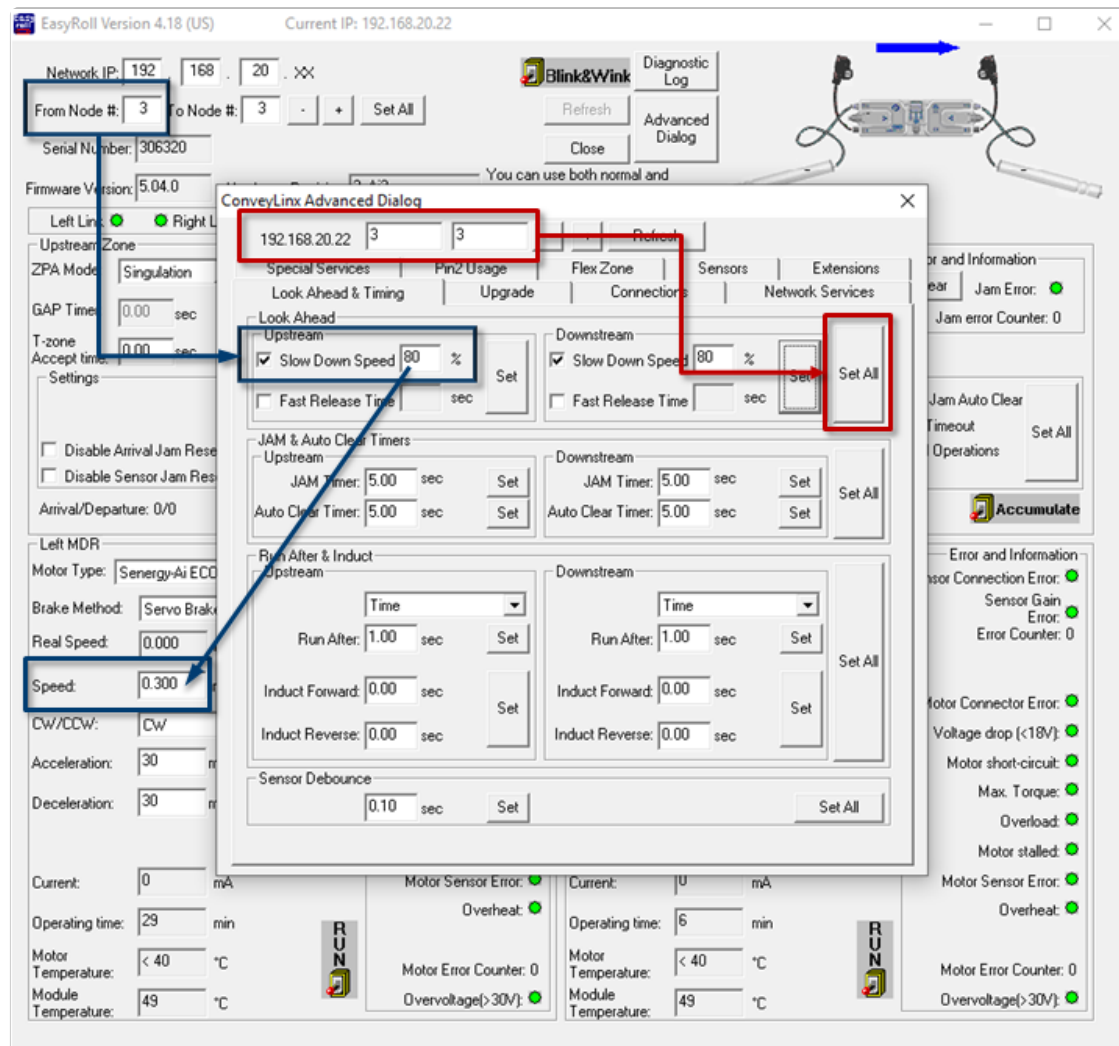
Fast Release Time

The *Fast Release Time* option allows you to set a delay before the *Slow Down Speed* is engaged for the slow-down zone

When Zone 2 wakes up it will run at normal set speed until the *Fast Release Time* has expired, then it will switch to run at the *Slow Down Speed*
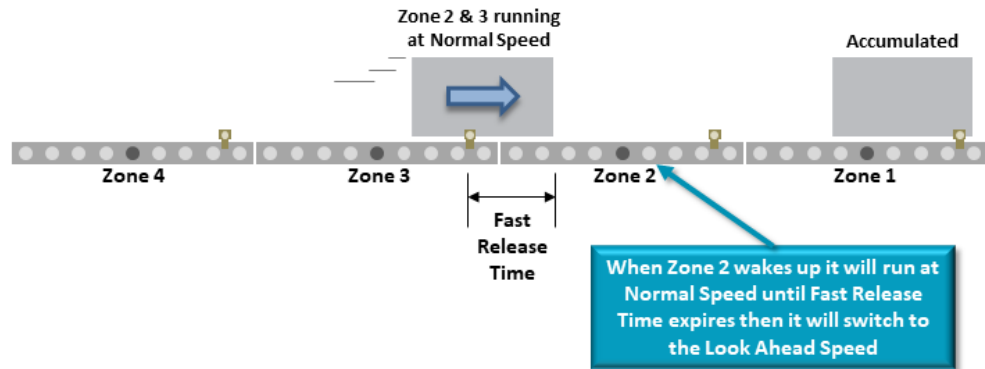
To use the *Fast Release Time* option, click the *Fast Release Time* checkbox and enter a time value (in seconds) into the entry box. In this example we entered 0.5 seconds. Click *Set* to write the changes to the module and use the *Set All* feature as desired
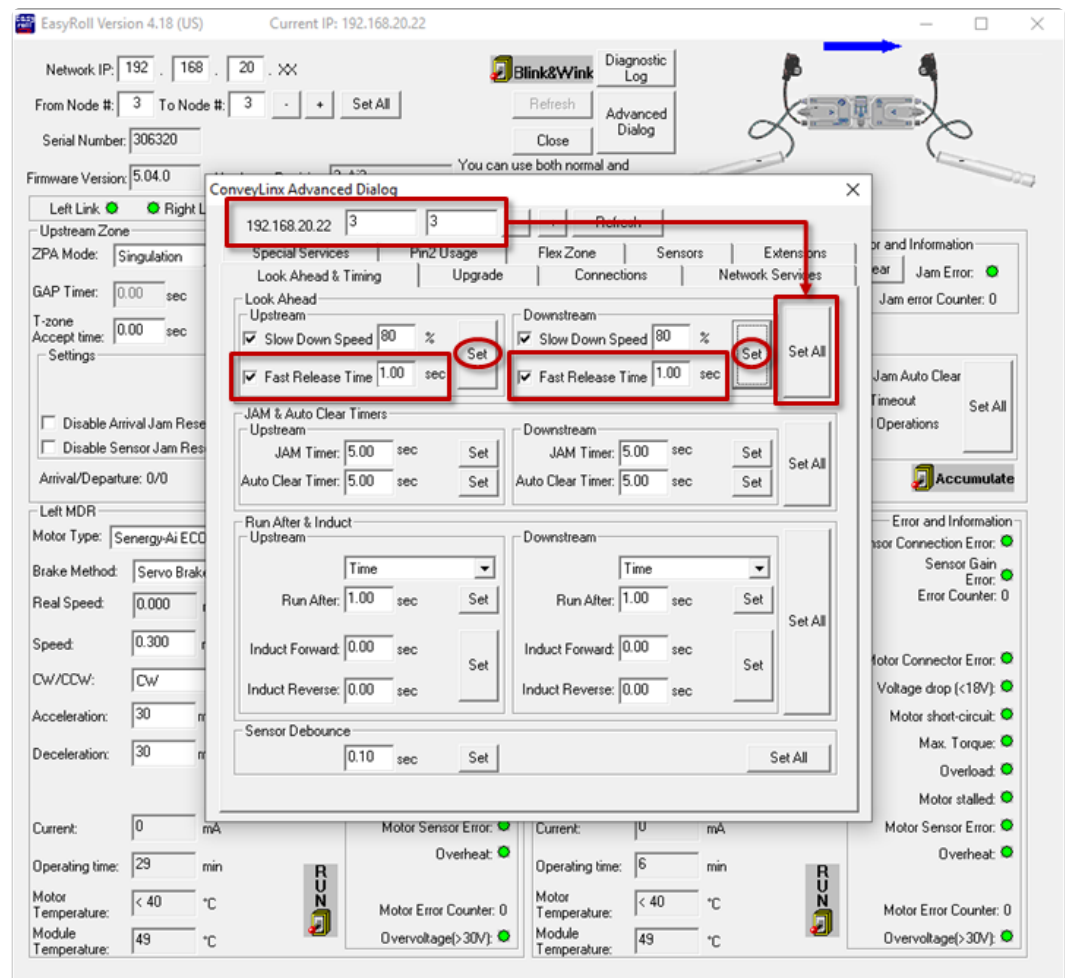
7.3.1.2. Jam & Auto Clear Timers

The *Jam Timer* for a given Upstream or Downstream zone is used for both detecting an Arrival Jam and a Sensor Jam. The *Jam Timer* should be set as the maximum expected time it takes for a carton to travel from one zone to the next plus a small addition to prevent nuisance jam occurrences. The default value for the Jam Timer is 5 seconds and the valid range of values is from 1 seconds to 65 seconds.

The *Auto Clear Timer* is the amount of time that the logic maintains the jam condition before auto resetting the jam. The default value for the Auto Clear Timer is 5 seconds and the valid range of values is from 0 seconds to 65 seconds

Enter new values for either or both the *Jam Timer* and/or *Auto Clear Timer* and click the corresponding *Set* button for each. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog

7.3.1.3. Run After Time/Distance

The *Run After* time value is used by the logic for normal zone discharge. This is the amount of time the zone's MDR will continue run after its photo-sensor has been clear when discharging to the next downstream zone.

This extra run time allows the zone to run so that the trailing edge of the carton can completely pass the photo-sensor and fully enter the next zone. This value is adjustable to compensate for special conditions where a zone photo-sensor is required to be placed farther upstream or downstream
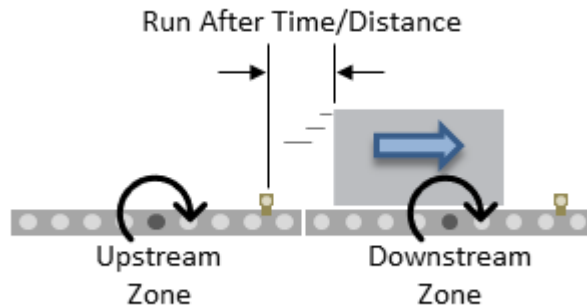
> ✳ Please note that Run After <u>does not</u> affect throughput rate. A longer Run After time will not prevent an upstream item from entering the zone.

Run After Time

Enter new values for upstream and/or downstream *Run After* time and click the corresponding *Set* button. The default is 5 seconds and the valid range is 0 to 65 seconds. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog. In this example we changed the *Run After* time to 1.00 seconds and using the *Set All* button to update this setting in Nodes 3 thru 4

Run After Distance

EasyRoll provides the option to change the *Run After* metric to be distance based instead of time based. When the metric is distance, the value entered is millimetres instead of seconds

Select *Distance* from the drop-down box and enter new values for upstream and/or downstream *Run After* distance and click the corresponding *Set* button. The default is 1000 mm and the valid range is 0 to 65,535. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog. In this example we changed the *Run After* distance to 100 mm and using the *Set All* button to update this setting in Nodes 3 thru 4.
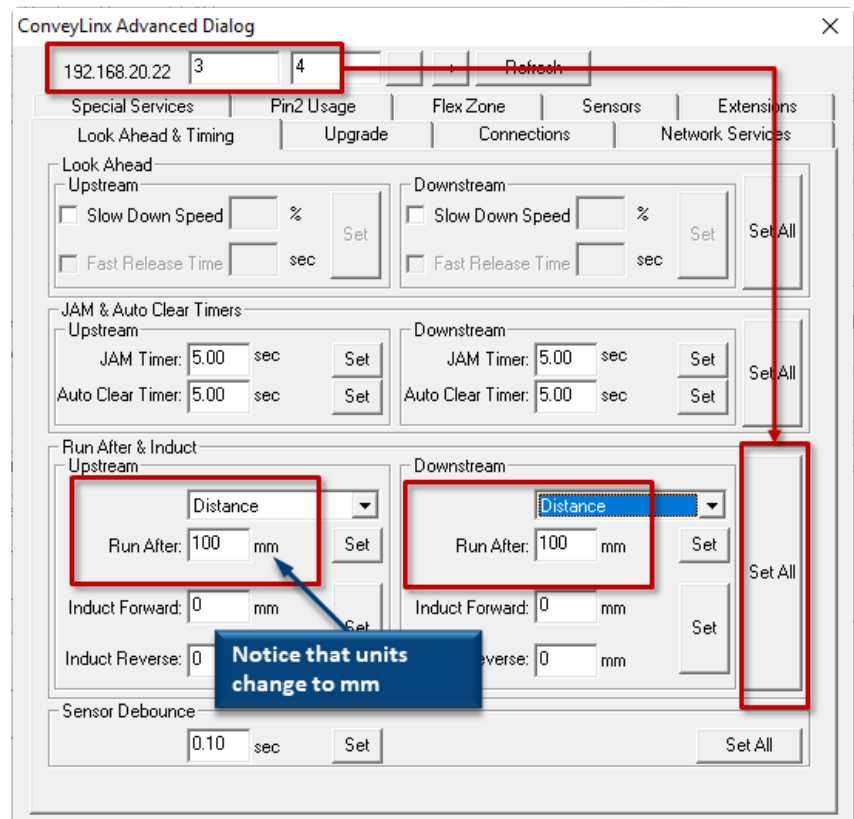
7.3.1.4. Induct Forward Time/Distance

Induct Forward value is used to cause the MDR to continue to run after the zone's photo-sensor has been blocked when receiving a carton from upstream.

This value is adjustable per zone to compensate for special conditions when for example a zone's photo-sensor needs to be placed farther upstream from the discharge end of the zone
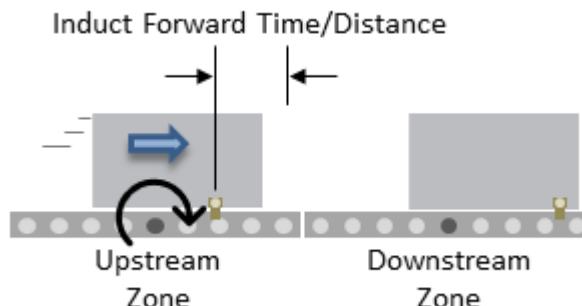
Induct Forward Time

Enter new values for upstream and/or downstream *Induct Forward* time and click the corresponding *Set* button. The default is 0 seconds and the valid range is 0 to 65 seconds. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog. In this example we are changing the *Induct Forward* time to 1.00 seconds for both the upstream and downstream zones and updating this setting to Nodes 3 thru 4
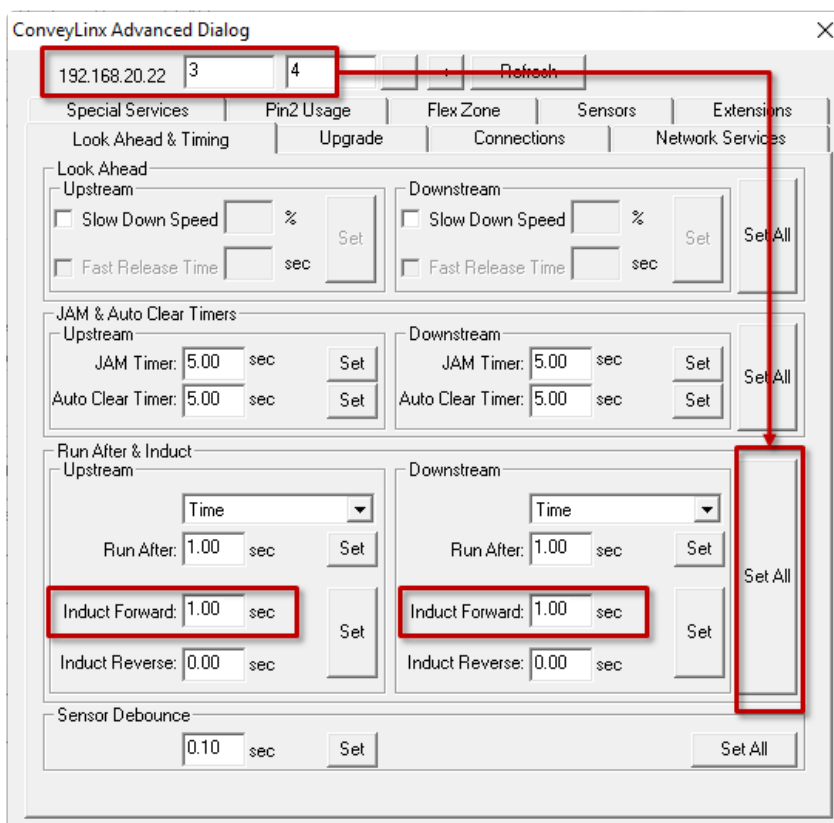
Induct Forward Distance

Select Distance from the drop down box and enter new values for upstream and/or downstream *Induct Forward* pulses and click the corresponding *Set* button. The default is 0 mm and the valid range is 0 to 65,535 mm. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog. In this example we are changing the *Induct Forward* distance to 100 mm for both the upstream and downstream zones and updating this setting to Nodes 3 thru 4
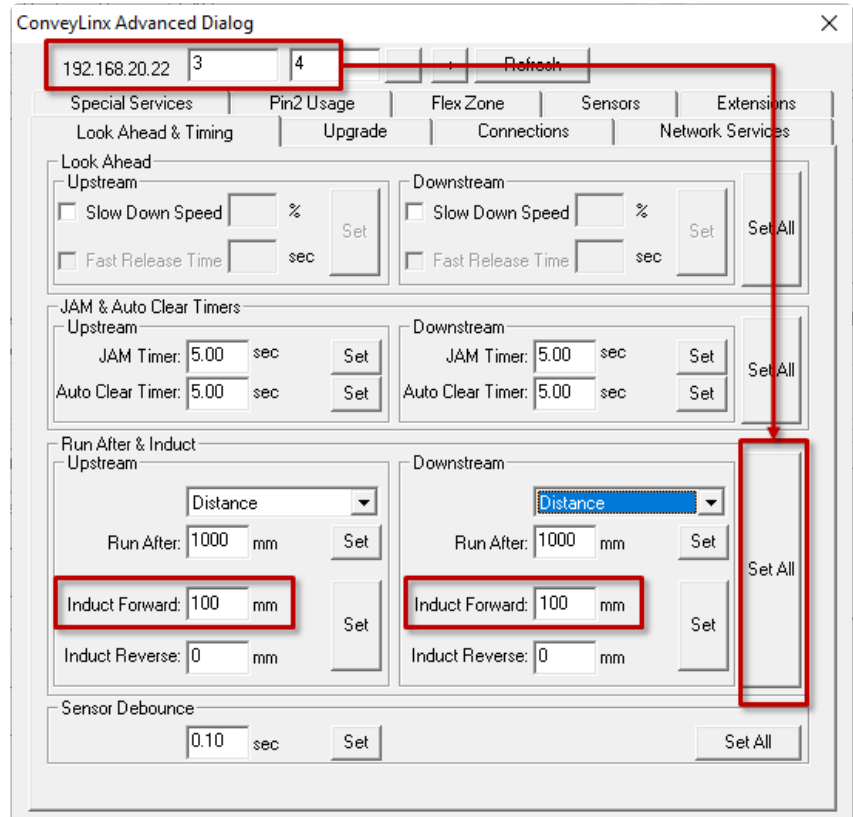
7.3.1.5. Sensor Debounce

Sensor Debounce setting is the time the logic holds the state of its Sensor inputs after a change of state. Keep in mind this is <u>not a delay prior to detecting a carton</u> when it first blocks the sensor. The module will detect the leading edge of a carton and hold this state for the *Sensor Debounce* time period. Similarly, when the trailing edge of the carton clears the sensor, the logic holds this state for the *Sensor Debounce* time period.
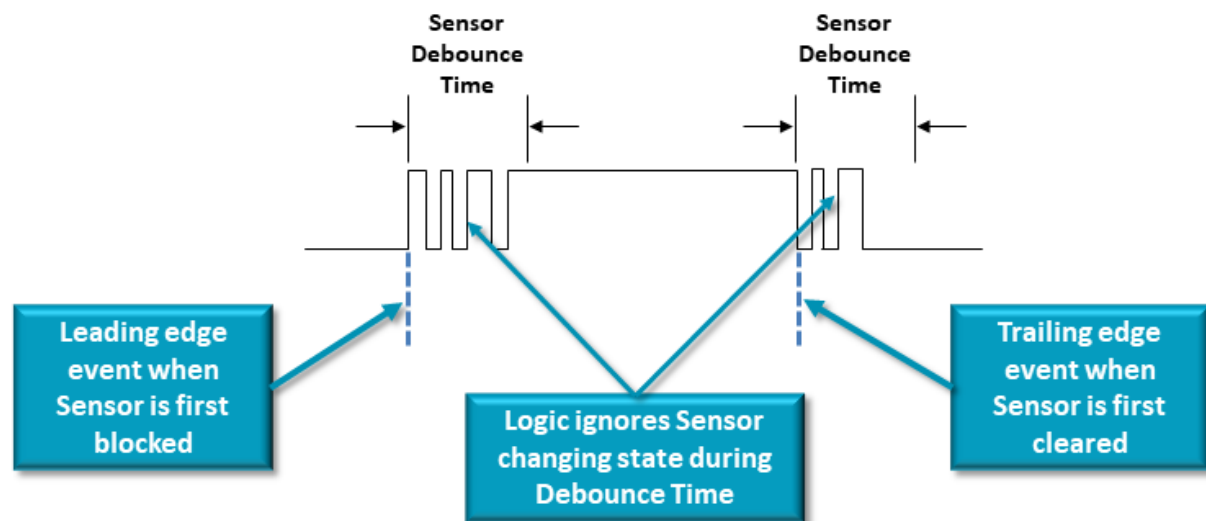
> ✻ The ConveyLinx-Ai Family Controllers keeps a running count of the number of changes of state that have occurred during the *Sensor Debounce Time* window even though the logic is ignoring these state changes. This running value is called the *Flicker Counter* and it can be viewed on the *Diagnostic Window*

Enter the desired value in seconds and click the *Set* button. The default is 0.1 seconds and the valid range is 0 to 2 seconds. Please note this setting applies to **both** the **Left and Right Sensor** ports. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog. In this example we are changing the *Sensor Debounce* time to 0.30 seconds and updating this setting using the *Set All* button for Nodes 3 thru 4.

7.3.2. Upgrade Tab

The *Upgrade* tab screen gives access to the module 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 modules. These files are available for download from our website. Once you have downloaded the desired file, the Upgrade utility allows you to browse for it and then select a single Node or group of Nodes to upload

✱ *VERY COOL:* Please note that if you upgrade firmware, all of each module's settings (motors, speeds, ZPA options, etc.) are preserved

❗ Please note that if module power is lost during a firmware upgrade, the module may become inoperable.

If this occurs, the module must be returned to the factory in order to be recovered.

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
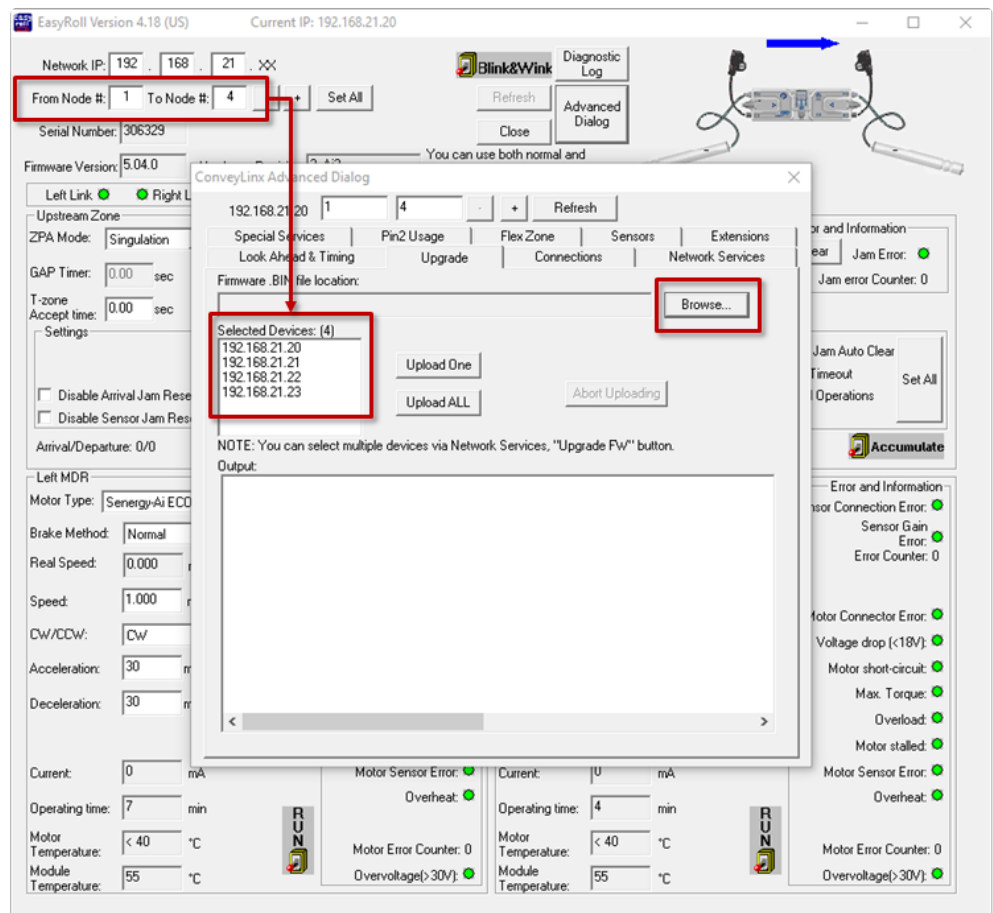
With the *Open* dialog displayed, navigate to the location on your PC where you placed the firmware upgrade file you received. Select the file and click *Open*
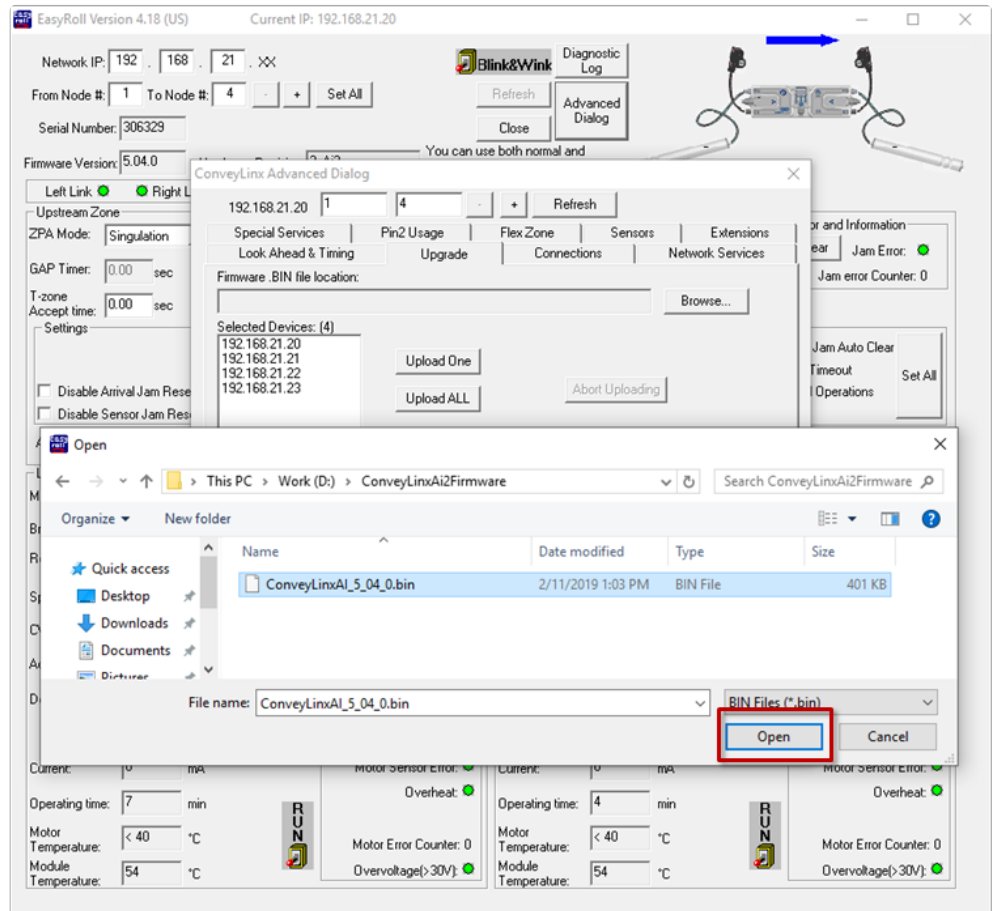
In this example, we clicked *Upload ALL* so the selected firmware upgrade file will be sent to all 6 Nodes. The *Output* window will update the progress of the file uploading process. The time it takes for this process will vary depending upon how many Nodes are being uploaded
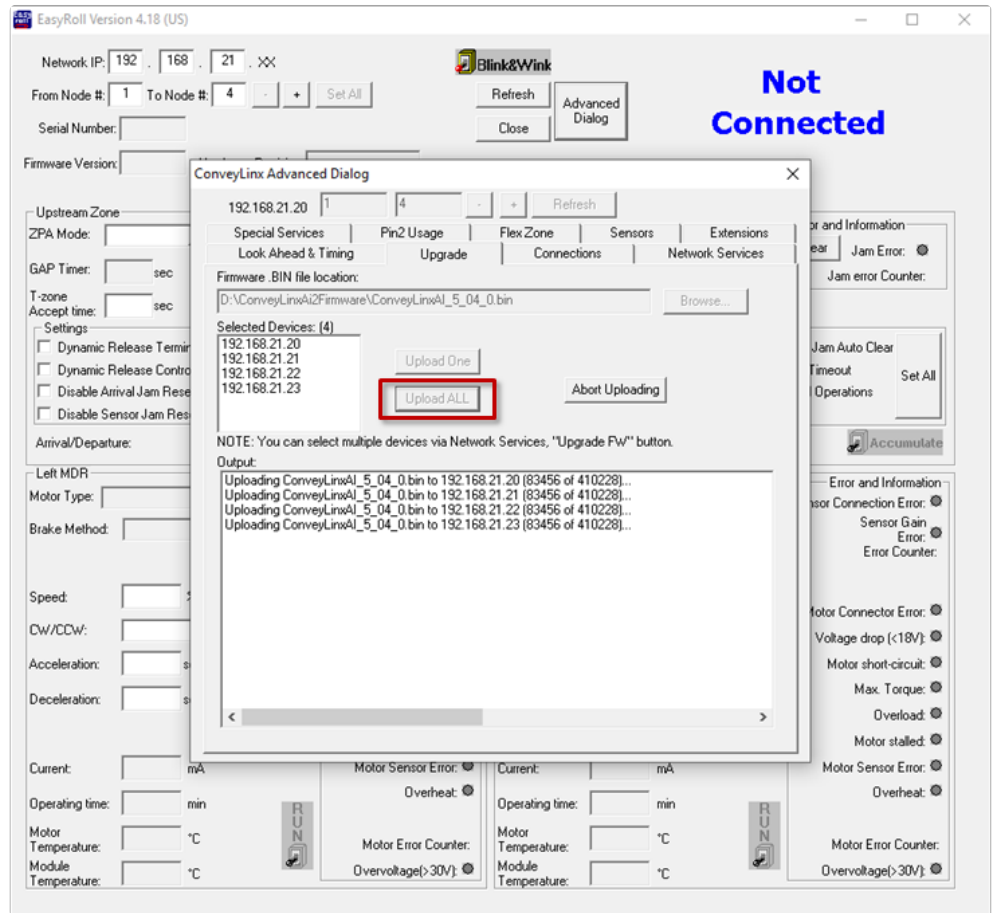
When all Nodes report back to the *Output* window with a status of *Done*; then the upload is complete and you can close the ConveyLinx Advanced Dialog window

---
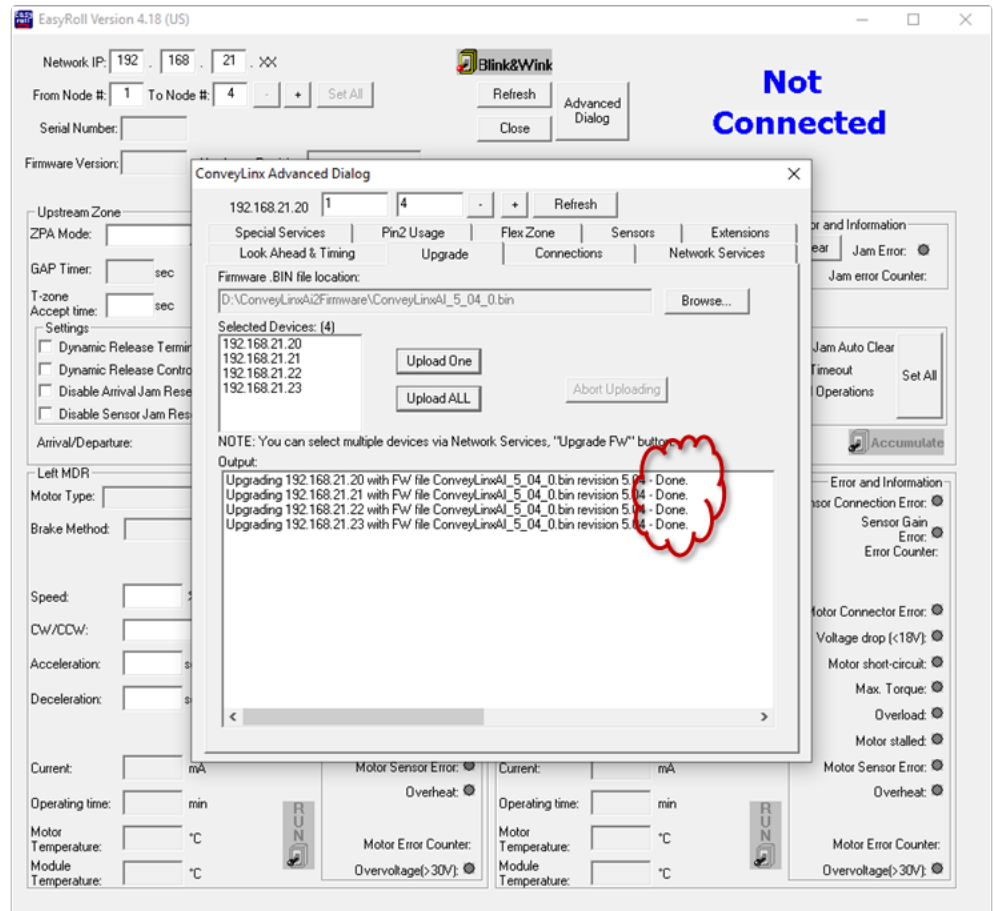
✱ Please note that you typically arrive at the *Upgrade* tab from the *Network Services* tab where you select the modules first and click the *Upgrade* button there. That will automatically take you to the *Upgrade* tab and populate the *Selected Devices* list box on the *Upgrade* tab with your selection from the *Network Services* tab

7.3.3. Connections Tab

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

Connecting Two Subnets Together

Here is a typical boundary between two *Subnets*. The most downstream Node of the first *Subnet* has an I.P. address of 192.168.21.23 and the most upstream Node of the second *Subnet* has an I.P. address of 192.168.22.20.
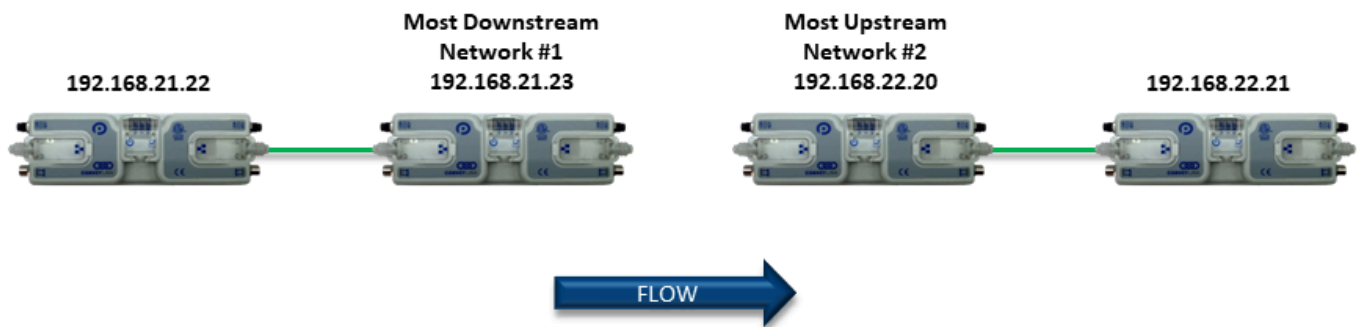
By simply connecting an Ethernet cable between these two boundary Nodes and then using *EasyRoll* to establish the "logical" connection between the two *Subnets*; you can achieve seamless flow between the two networks. The procedure requires that you have to instruct Node at 192.168.21.23 to convey cartons to Node at 192.168.22.20, and likewise you have to instruct Node at 192.168.22.20 to accept cartons from Node at 192.168.21.23
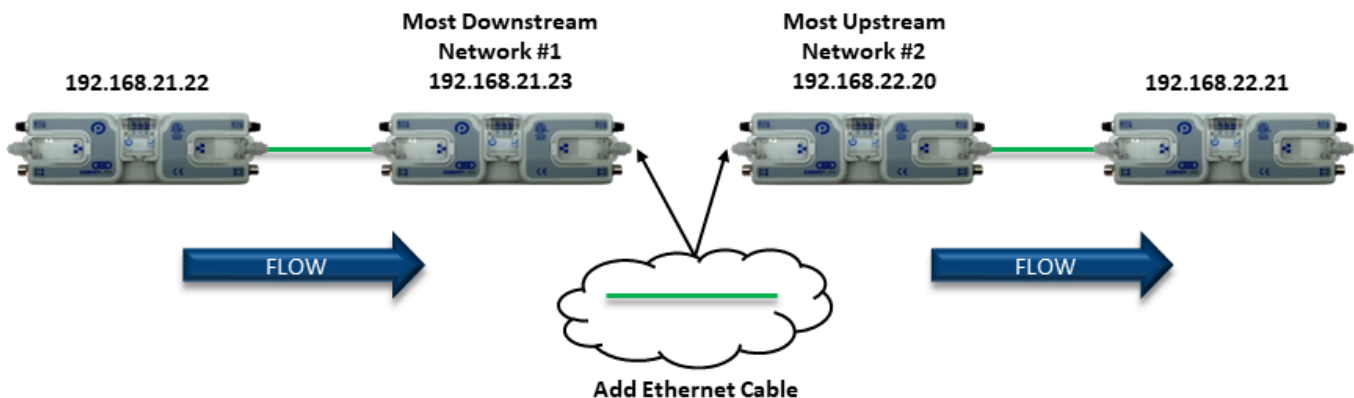
Navigate to Node 4 of the 192.168.21 subnet (i.e. 192.168.20.25). Notice that the module's network data appears in the center. Also notice that the Downstream connection for this Node is *None*

Enter *192.168.22.20* as the I.P. address for Node 4's new Downstream connection. Click *Apply* to make the change. Please note that it will take a few seconds for this to complete

Navigate to Node 1 of the 192.168.22 subnet (i.e. 192.168.22.20). Notice that the module's network data appears in the center. Also notice that the Upstream connection for this Node is *None*
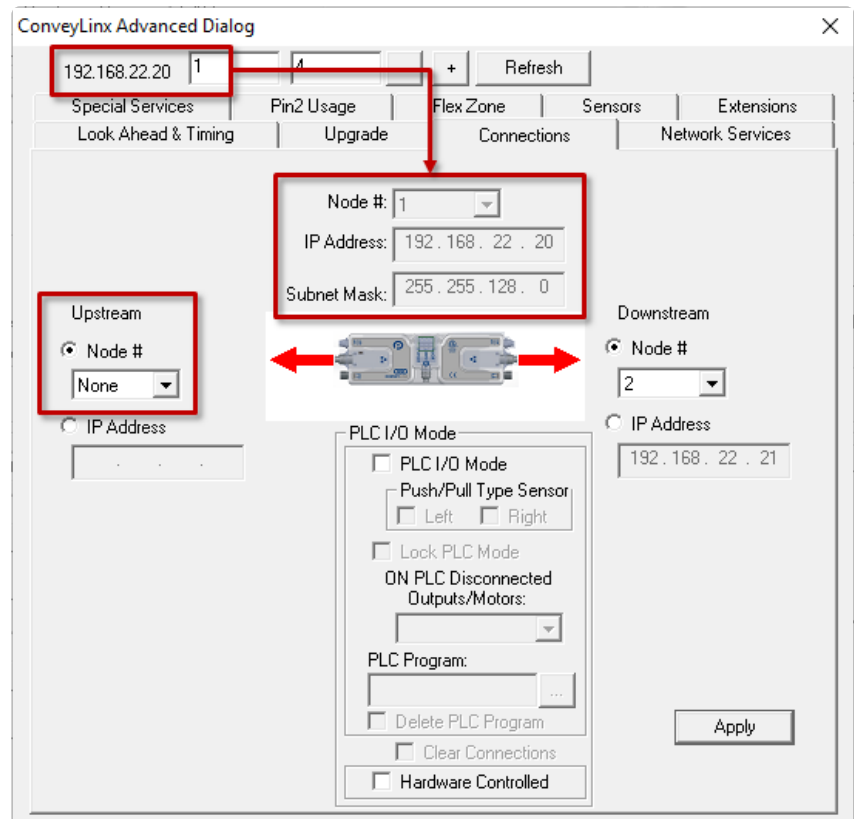
Enter *192.168.21.23* as the I.P. address for Node 1's new Upstream connection. Click *Apply* to make the change. Please note that it will take a few seconds for this to complete
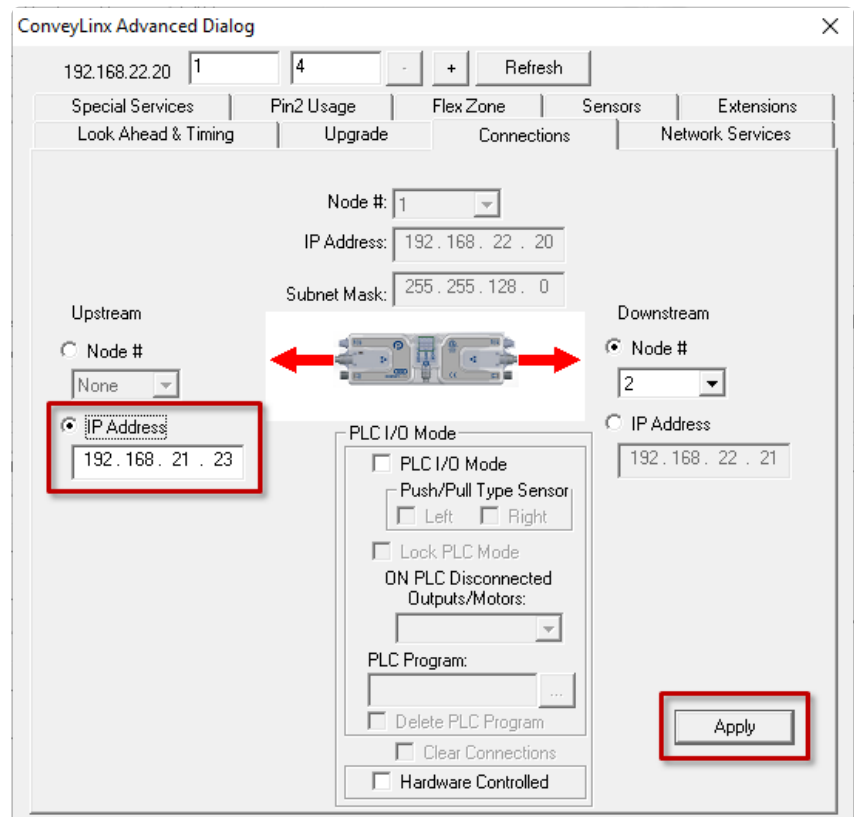
7.3.4. Network Services Tab

The Network Services screen provides multiple functions related to module and network management. These functions are:

- Discover and IP Address Set
- Backup and Restore
- Firmware Upgrade

7.3.4.1. Discover & IP Address Set

One of the features of *EasyRoll* is that it has a utility called *Discover* that allows your PC to go and find any 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 modules

Clicking the *Discover* button will cause the list box to populate with all the modules in all *Subnets* that your PC can see out on the network
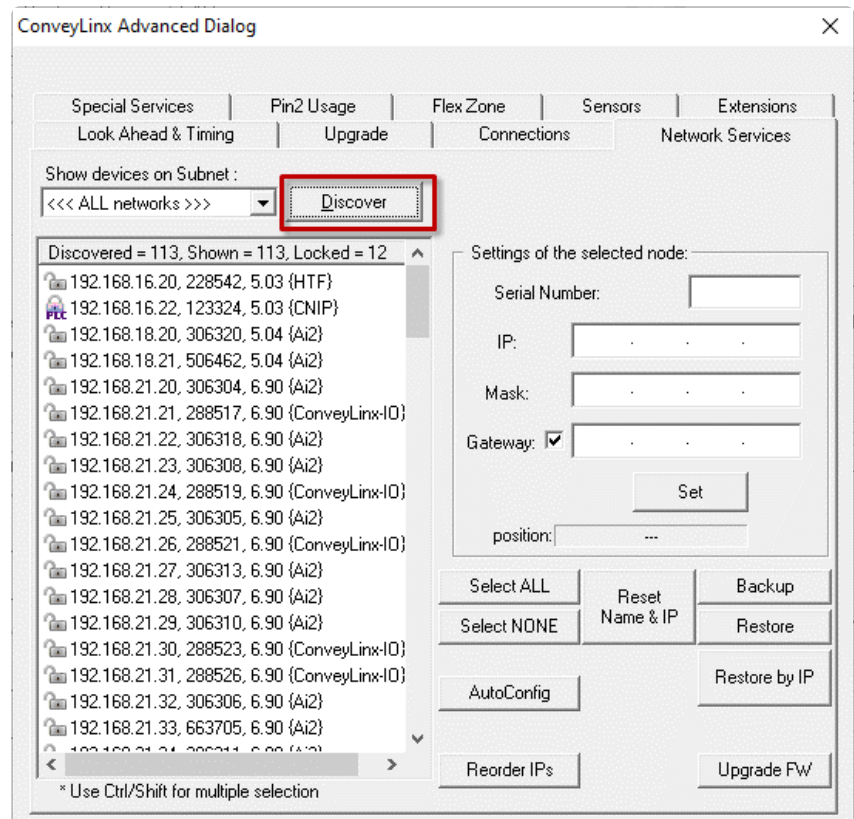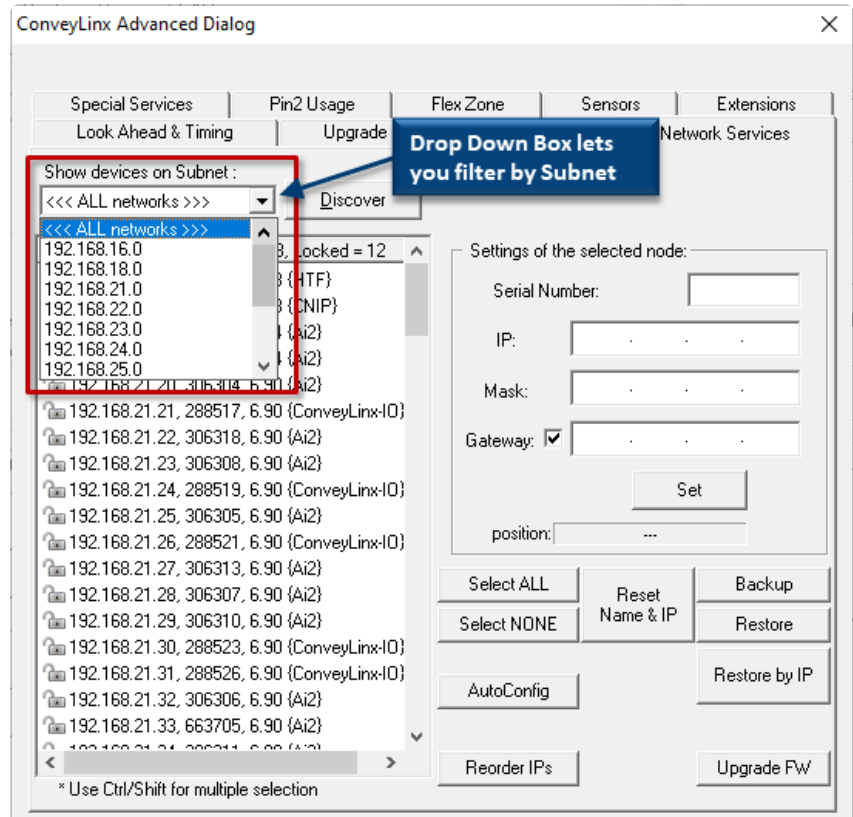
The drop-down box at the top of the list box lets you filter the list by *Subnet*
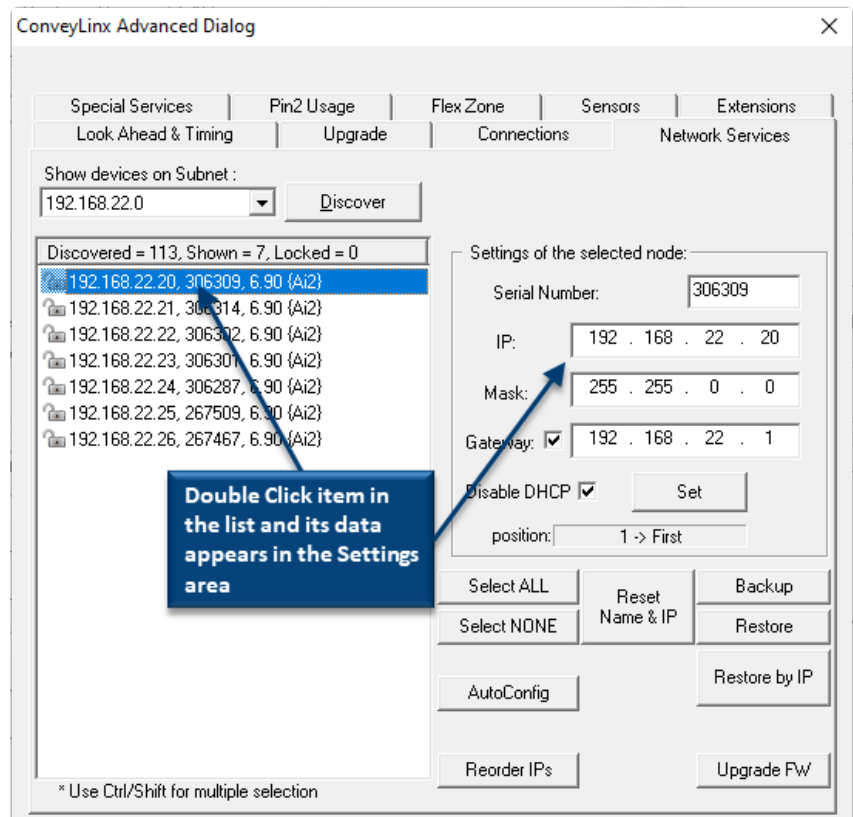
To change the IP Address, Subnet Mask, etc. of a module, double click it from the list and its data can be modified

Once you have edited the data, click the *Set* button and the changes are sent to the module

7.3.4.2. Backup & Restore

You have the ability to select a **Subnet** or all **Subnets** discovered and generate a *Backup* file that will contain all the parameters and settings for each module included in the selection. This means all motor settings (speed, acceleration, deceleration, braking, etc.), ZPA settings, Advanced Dialog settings, etc. are captured in this file that can be saved on your PC. Conversely you can use this *Backup* file to *Restore* settings to a **Subnet** or all **Subnets** in the event settings are inadvertently modified or you simply want to return your modules to a previously known state

To perform a backup, click the *Select ALL* button to select all the items in the list. In this example we are going to backup **Subnet** 192.168.29. Once all modules are highlighted, click the *Backup* button.
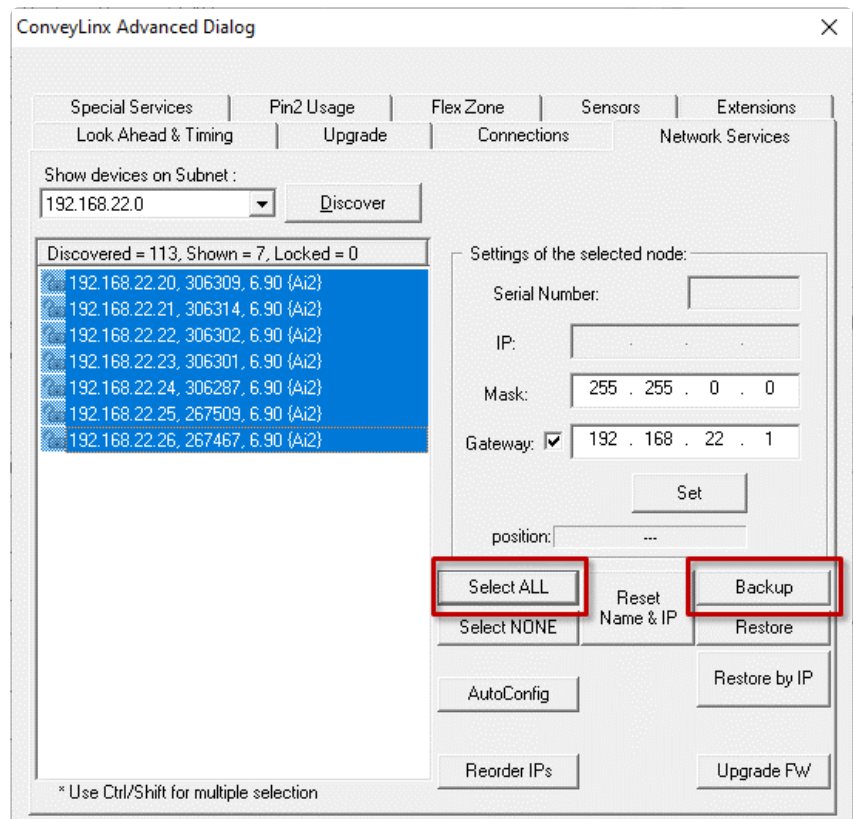
A *Save As* dialog will appear and you provide a filename and location to save the file
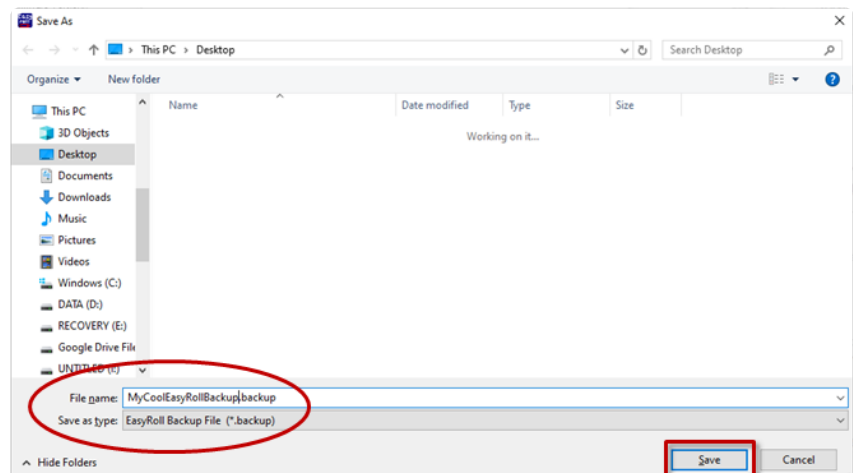
Conversely, if you have a *Backup* file that you want to use to get your **Subnet's** settings restored; click the *Select ALL* button and then click the *Restore button*.
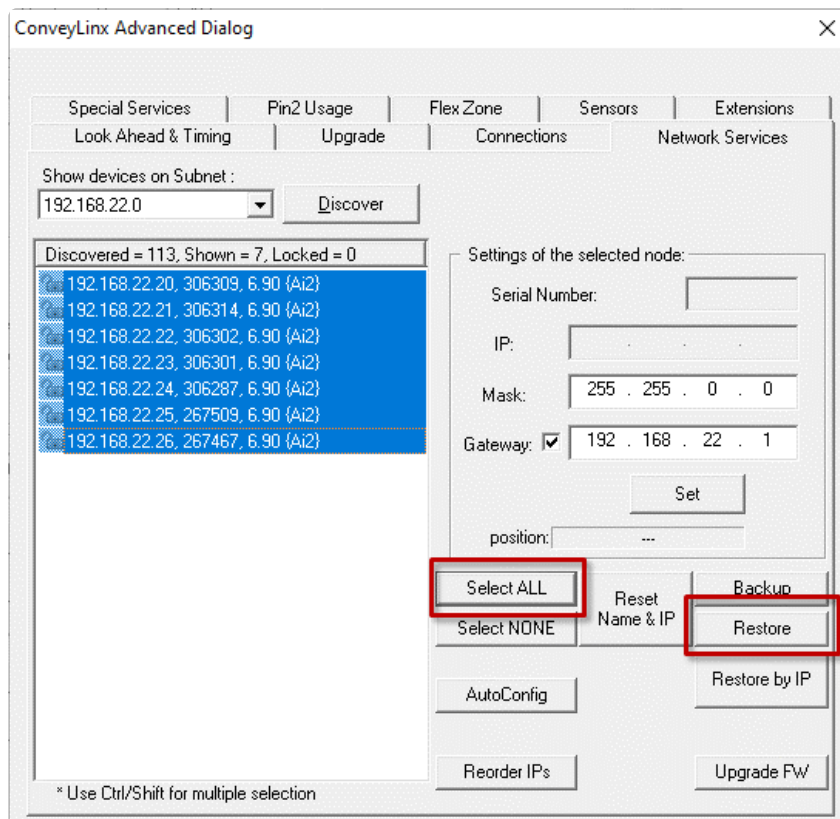
An "Open File" dialog will appear for you to navigate to the location of your backup file, select it and click "Open".
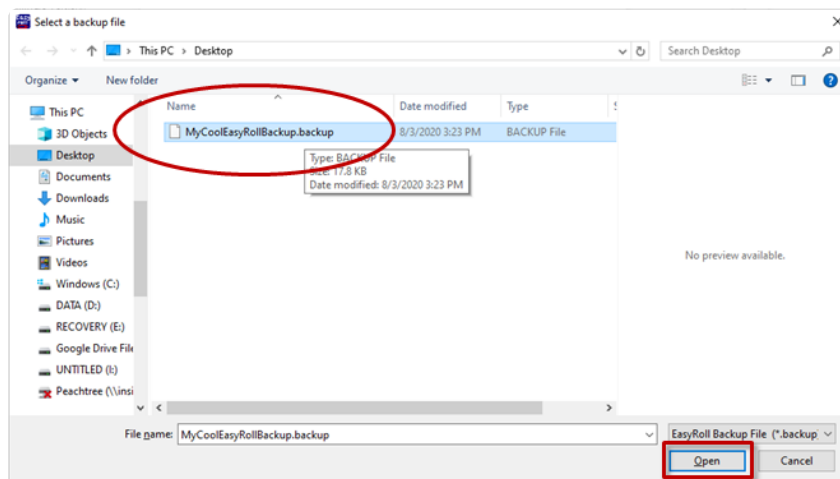
✳ Please note that the *Restore* function will only restore settings to modules that have a matching Serial Number in the **Backup** file. For situations where module serial number are different but functionality still needs to be restored, you will need to use the *Restore by IP* function

---

Restore by IP Function

Because the default *Restore* function restores settings by module Serial Number only, in situations such as duplicating a **Subnet** or restoring functionality from an older backup where some of the modules were replaced since the **Backup** file was generated, you need to restore

the settings by IP address instead of by module Serial Number. To do this you follow the same restore procedure as previously shown except you click the *Restore by IP* button instead of the Restore button

Backup and Restore Recommendations and Tips

Sometimes there are situations where just because a software application will allow you to do something does not always mean that you should. The *Backup* and *Restore* functions in *EasyRoll* can fall into this scenario. Even though you can technically create a backup file for a few or even a single module from within a **Subnet** of many modules, it is not recommended that you do so. ConveyLinx support engineer's experience to date has indicated that when customers perform *Backup* and *Restore* functions for, at a minimum, all modules in a **Subnet**, you will have fewer or no problems with your ConveyLinx networks. Systems that were once working and then start producing unexpected results often stem from performing a *Restore* function to a small portion of modules instead of the entire Subnet and/or starting with a backup file that did not include all modules in the **Subnet**. So, here is a list of tips:

- When you perform a *Backup*; at a minimum, select all modules in a given **Subnet**
- When you perform a *Restore*; at a minimum, select all modules in a given **Subnet**
- Perform a new *Backup* after you make any settings changes to one or more modules in a **Subnet**
- Perform a new *Backup* after every time you perform an Auto Configure Procedure

7.3.5. Special Services Tab

Clear Motor Short Circuit Error

Another function on the *Special Services* tab is a button used to clear an MDR short circuit error. This particular error is not logically cleared based upon an elapsed period timeout or other such reset. An MDR short circuit error requires that either the module be powered down and then powered back up or by clicking the Reset button on this tab. This function is made available in *EasyRoll* as a convenience so you don't have to cycle the power on the module.
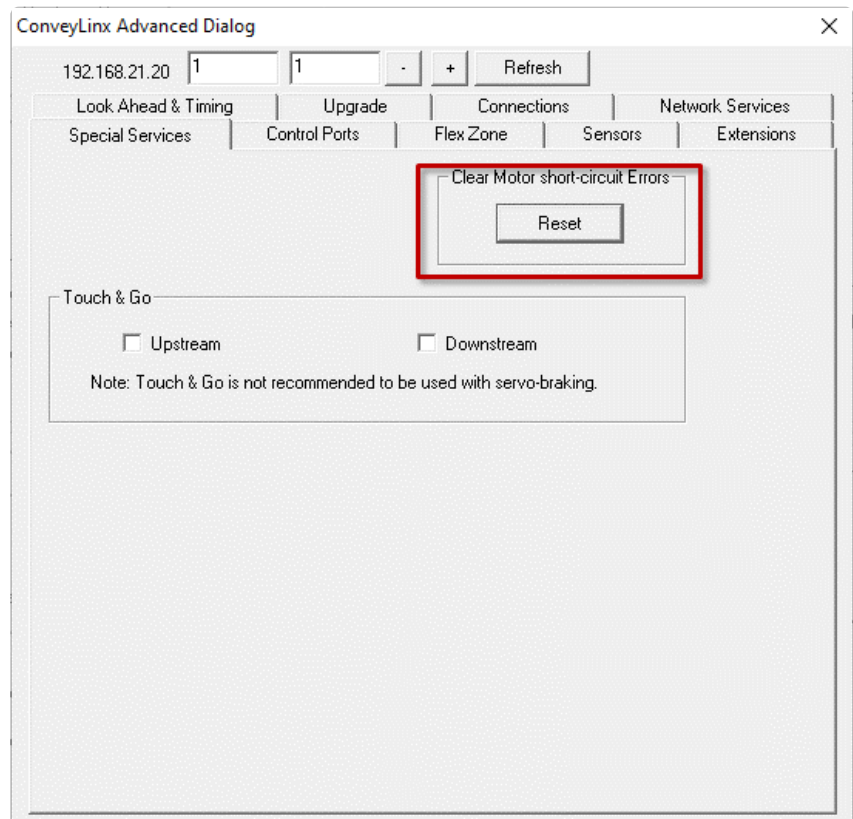
There is not a separate *Reset* button for left and right motors. One button will perform the function for either or both motors.

Touch & Go

The *Touch & Go* function is available in ZPA mode and when activated causes the MDR in the activated zone to sense rotational movement of the MDR in its default direction. If this rotational movement (such as someone pushing a carton onto the zone) is of sufficient duration and speed; the zone will "wake up" as if its upstream interlock had been energized. You enable this function by checking the appropriate Upstream or Downstream checkbox in the *Touch & Go* area.

You can set this functionality for either or both the Upstream and Downstream Zones in the direction of flow. In this example we checked the Upstream zone
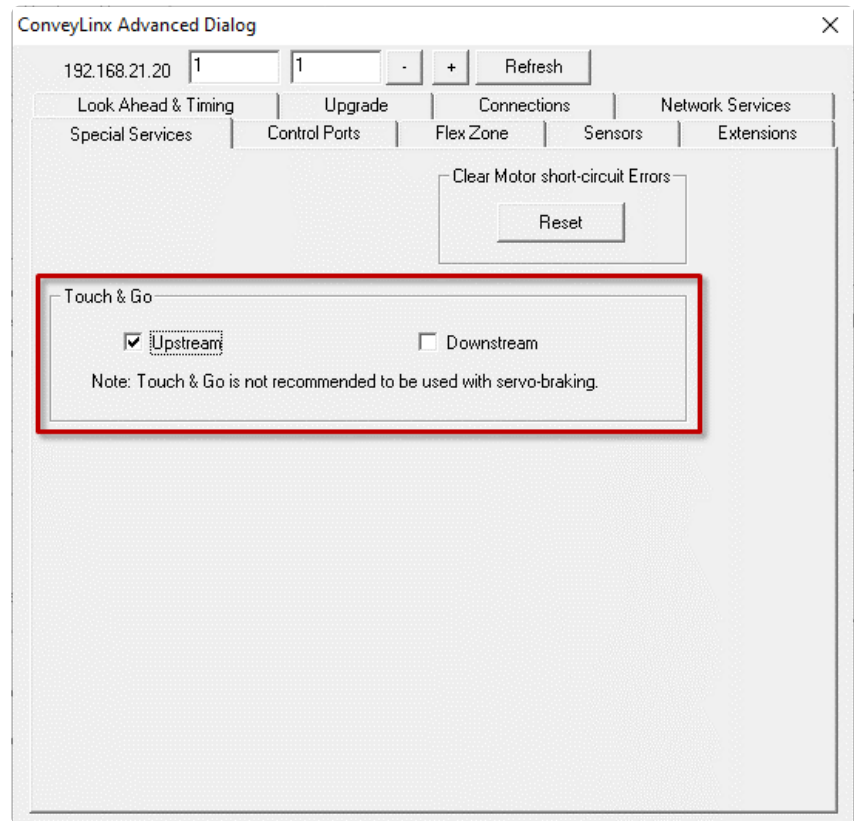
> ❗ Touch & Go is only applicable for ZPA mode. Do NOT use Touch & Go with Servo Braking

7.3.6. Flex Zone Tab

The *Flex Zone* feature is enabled by default when you perform the *Auto Configure Procedure*. There can be certain situations such as higher speed applications and/or applications where a large percentage of cartons are at a length very close to the zone length where a "false triggering" of a flex zone condition can occur. In these situations, when product needs to accumulate, you may see several zones unoccupied because of this false triggering. For these applications where accumulation density is paramount, you can disable *Flex Zone* to eliminate this false triggering situation.

> ❗ Please note that Flex Zone function has to be enabled or disabled for the entire subnet. It cannot be disabled or enabled on a per zone basis or for a group of zones within the same subnet.

Click the appropriate button to either *Disable* or *Enable* Flex Zone based upon the *Subnet's* current status as indicated. Because this function applies to all modules in the *Subnet*, it does not matter what module you were connected to when you invoked the *Advanced Dialog* in order to disable or enable *Flex Zone*
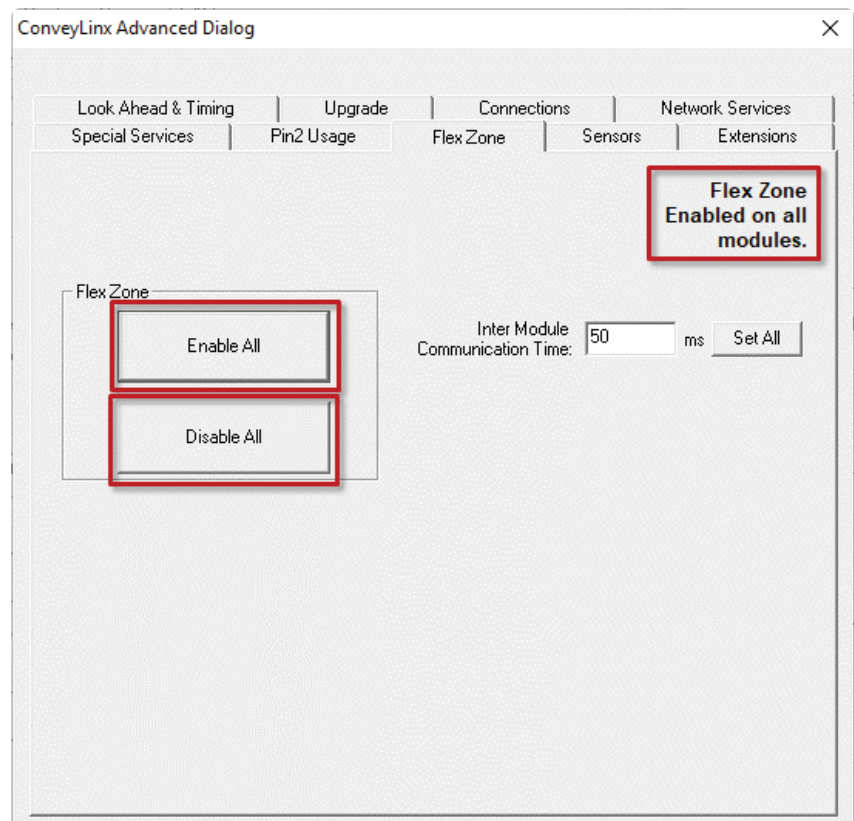
Inter Module Communication Time

The *Inter Module Communication Time* value is used to adjust the behavior of the Flex Zone operation. In situations where carton lengths can be close to the length of the zones (i.e. distance between photoeyes), you may want to adjust Flex Zone operation so that it either engages or not in these situations.

> Typically there are one or two driven rollers past the sensor in a Zone. If you have cartons that encroach upon these rollers when accumulated in the downstream zone, you may want *Flex Zone* to engage so that the next upstream carton will not enter the zone (Zone A in the example), thus not running these rollers underneath the accumulated carton.
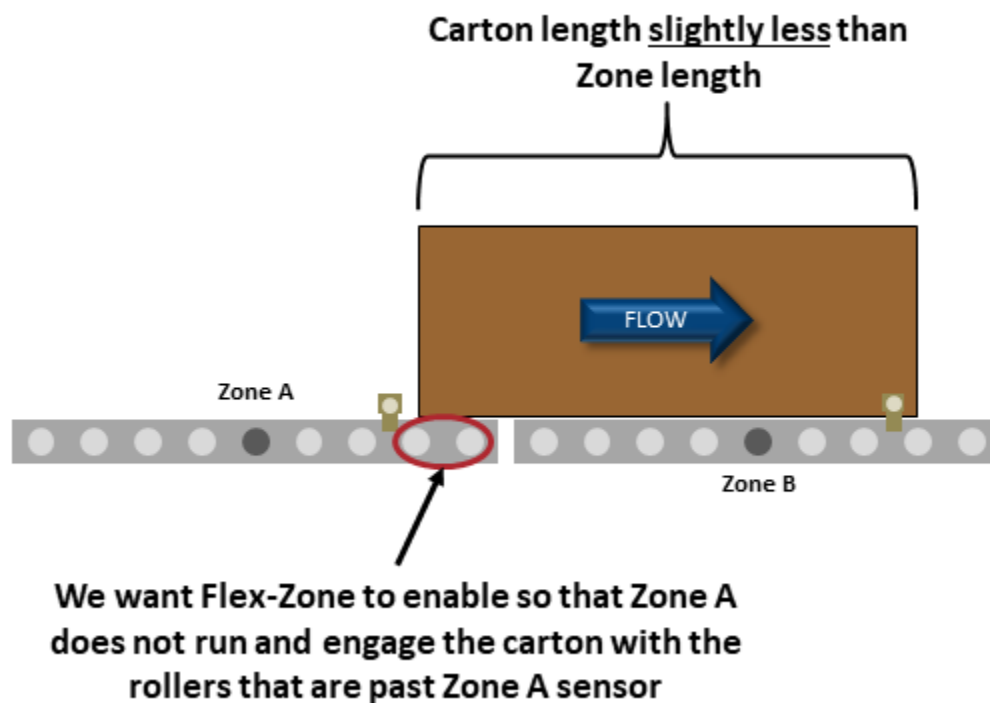
The value entered is in msec and it can be thought of as an amount of time added to the logic after the trailing edge of the sensor in order to simulate increasing the carton length. Please note that this value will be set the same for each module in the *Subnet*.
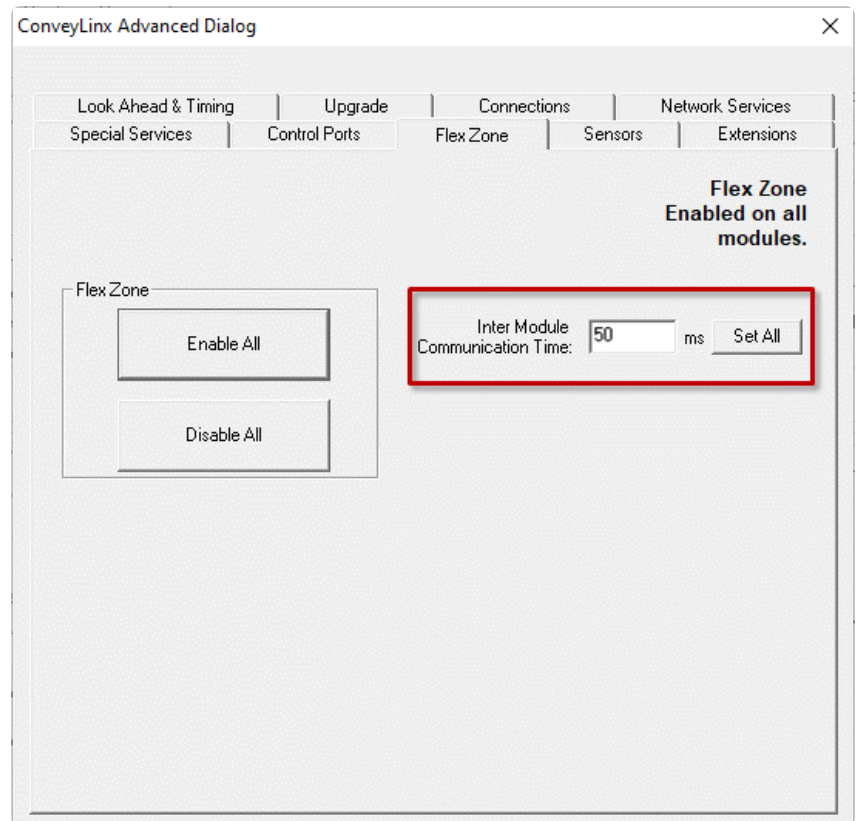
✱ Please note this value has nothing to do with speeding up or slowing down the actual speed of communications between modules

7.3.7. Sensors Tab

The *Sensors* tab displays the status of how the two sensor signals were configured during the Auto Configuration Procedure. If for some reason one or more of the sensors was not configured properly during the Auto Configuration Procedure; for example they were not aligned with their reflector or there was an obstruction blocking the sensor at the time the procedure was performed, the *Sensors* Tab will allow you to change the sensor's configuration without requiring you to re-perform the *Auto Configuration Procedure* for the entire subnet. For example, if all the sensors on the system are light energized normally open then the corresponding zone's sensor graphic on this tab will show *Off Blocked*.

Click the appropriate icon to change either or both sensor's blocked signal state. Note that you can make the same change for a group of modules by clicking the *Set All Sensors this Way* button
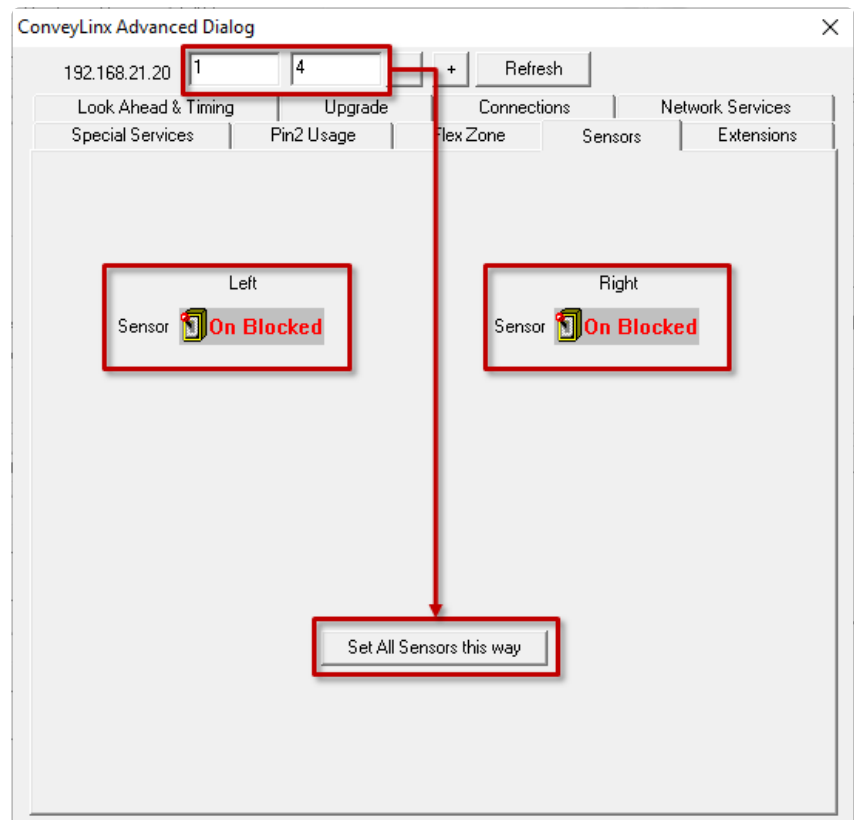
7.3.8. Extensions Tab

The *Extensions* Tab allows you to extend or "slave" single or multiple zones to one "master" zone. An example situation could be that you have for example certain length zones and on a given conveyor line there is a need for an extra partial zone that is too short to be a functional zone that accumulates a carton, but at the same time it is long enough that mechanically you need to have an MDR in that zone. In this situation you would like to provide a module and an MDR but no sensor and you want this zone to run its MDR when either its adjacent upstream or downstream zone is also running. In essence you want to make this partial zone a logical extension to one of its neighbors.

Example for Using Extensions for a Lift Gate

A common use of the Extension mode configuration available from the Connections tab selection is for a powered lift gate. In this example the module on the lifting or gate portion of conveyor has an MDR and no photo-sensors. Normal operation when the gate is down is for the MDR on the gate to run when its immediate downstream zone runs so as to create "one long logical zone". This means that if a carton is accumulated on the upstream zone of Node 192.168.21.23; a carton arriving at the downstream zone of Node 192.168.21.21 will stop and accumulate and no loads will ever be logically accumulated or stopped on the gate portion. In order to accomplish this, all we need to do is instruct Node 192.168.21.22 to be an Extension of its downstream neighbor Node 192.168.21.23
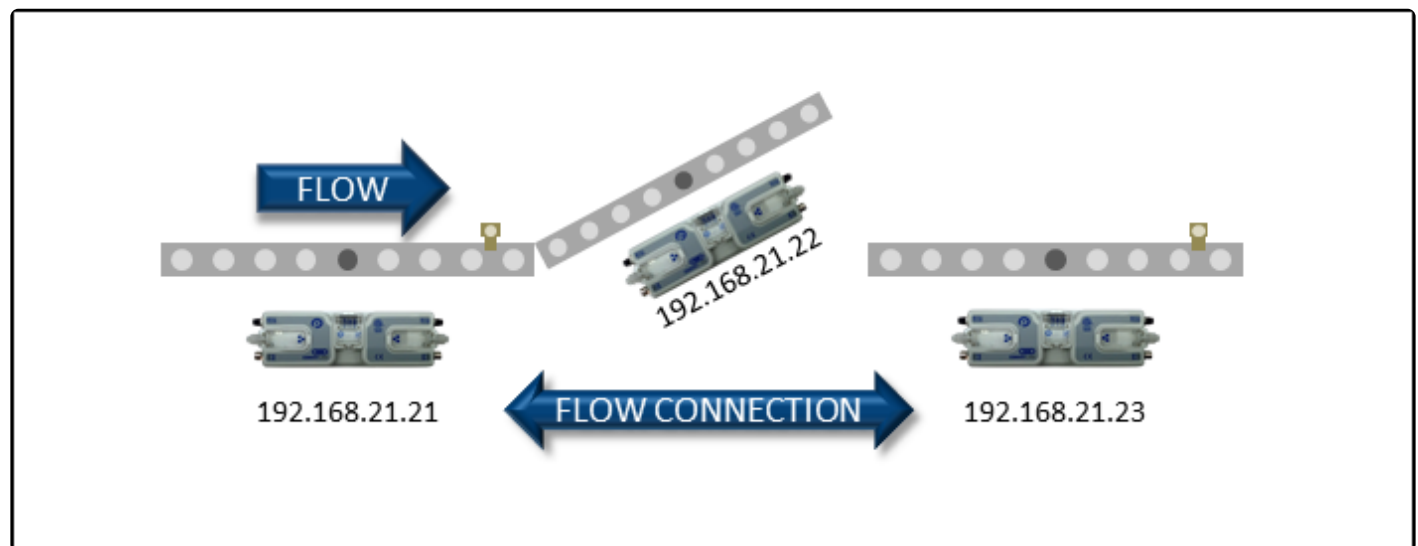
Lift Gate Example

In the *Advanced Dialog*, navigate to **Node 3** and click the *Extensions* Tab. Notice that Node 3's IP address 192.168.21.22 is filled in. In the drop-down box, select *Downstream* and click *Apply*. Note that this may take a few seconds to complete
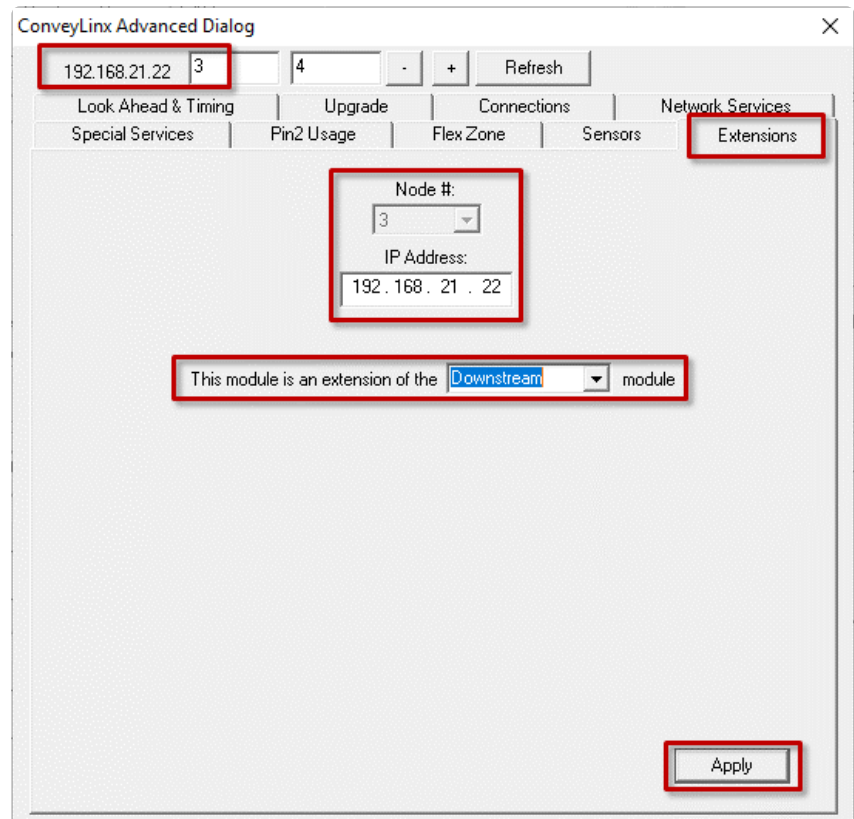
After clicking *Apply*, the screen will update and show you that the module has been configured as an extension or "slave" of its downstream neighbor
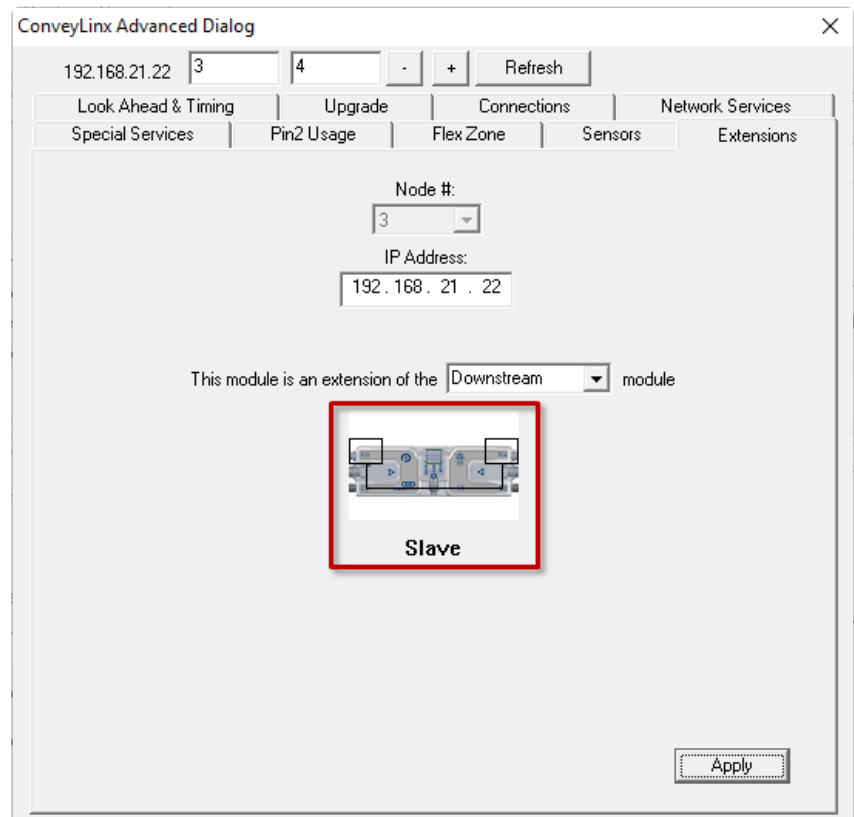
Now if you navigate back to **Node 2** and look at the *Connections* Tab, you can see that the downstream connection of **Node 2** is now **Node 4** instead of **Node 3**
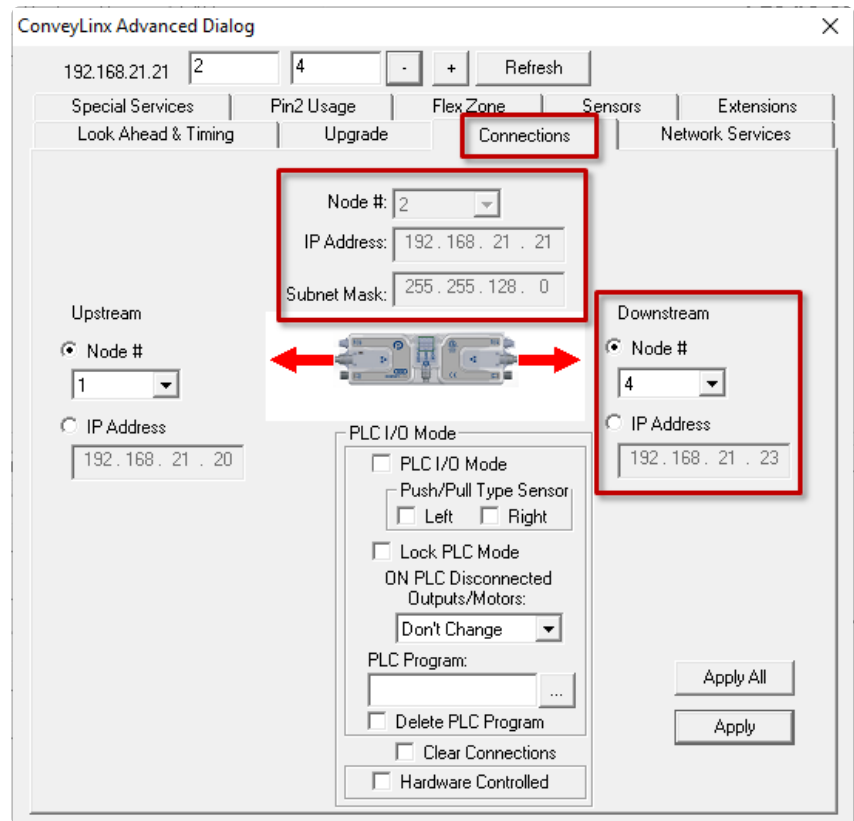
And then when you navigate to **Node 4**, you can see that its upstream connection is now **Node 2** instead of **Node 3**
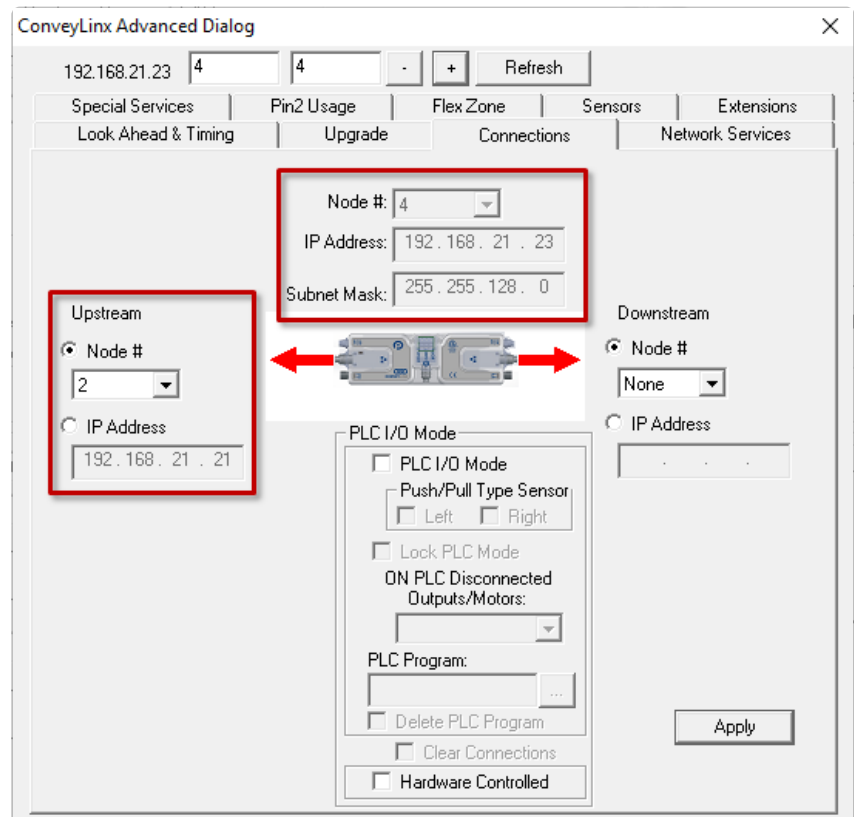
And when you navigate to **Node 3**, you can see that it is configured as an Extension of **Node 4**

7.3.9. Pin 2 Usage Tab

Pin 2 on each of the two Sensor port's M8 connector on the Module is configurable as to its potential function. This pin can function as either an input or an output. The default "out-of-the-box" usage for Pin 2 is "None" or "External Control". Using the Pin 2 Usage screen from the ConveyLinx Advanced Dialog allows you to select one of the following functions for each zone on the module:
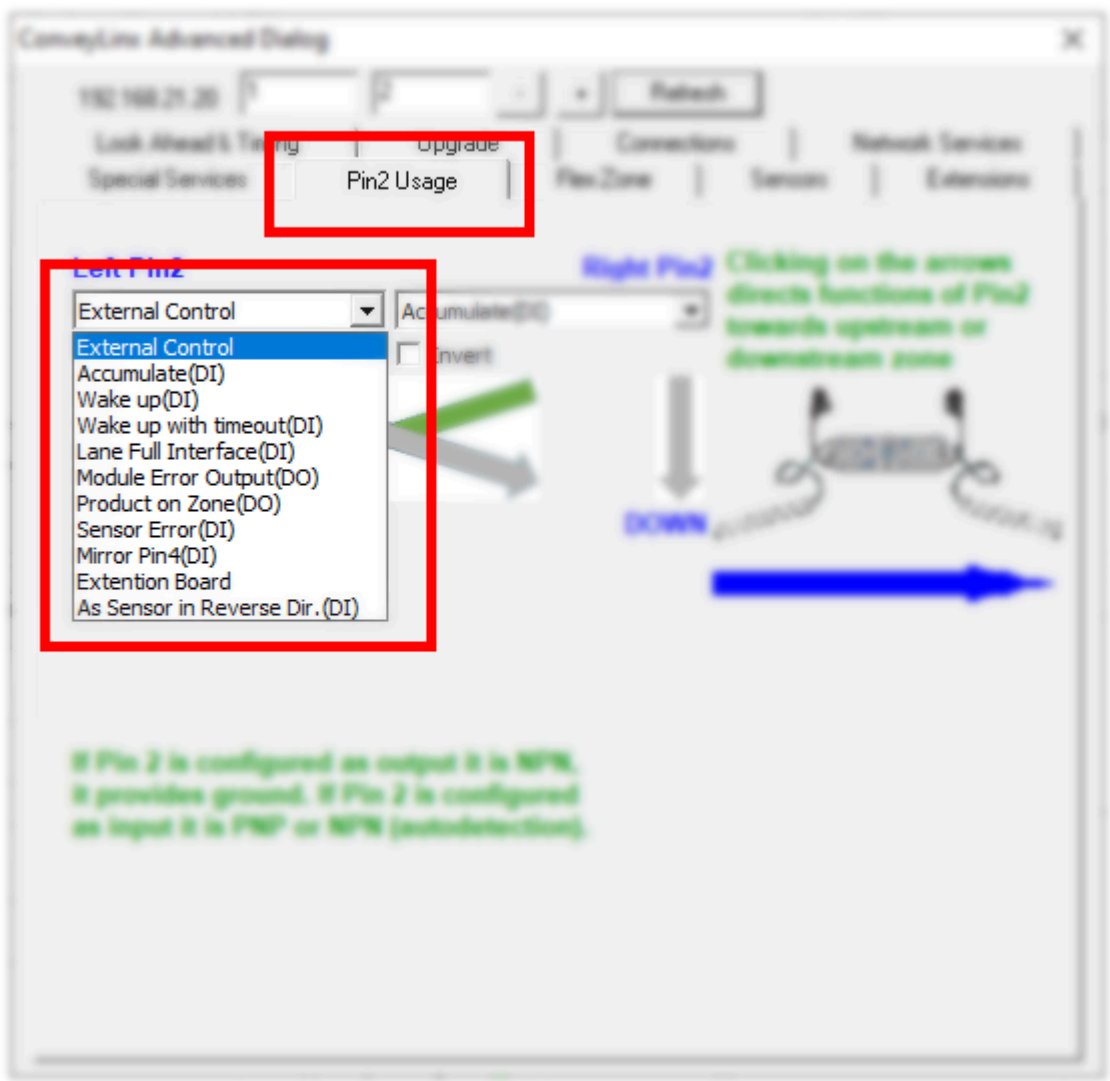
Item	Signal	Description
External Control	Input or OUT	Signal does not affect ZPA control and is configurable and accessible by external PLC control
Accumulate(DI)	Input	Used to cause local ZPA zone to accumulate

Wake-up(DI)	Input	Used to cause causes local zone to wake-up and run its motor to accept a product
Wake-up with Timeout(DI)	Input	Same as Wake-up(DI) except the running zone will stop after a timeout if signal remains on
Lane Full Interface(DI)	Input	Used to operate the Lane Full Interface discharge control function
Module Error Output(DO)	Output	Energizes when there is a module error condition active
Product on Zone(DO)	Output	Energizes when the local ZPA zone is occupied
Sensor Error(DI)	Input	Used for sensors equipped with a health or error output signal
Mirror Pin4(DI)	Input	
Extension Board	-	Not Used
As Sensor in Reverse Dir.(DI)	Input	Used to when reverse direction sensor is connected for reversing operation

Depending on how the module is configured, you can independently select which of these functions you want to reside on which of the two available sensor port Pin 2 signals.

> ✱ *For EasyRoll versions prior to 4.21, the drop down selector will indicate "None" instead of "External Control". The function of the selection is the same either way.

7.3.9.1. Most Downstream Zone

Please note that by default and without any intervention, the most downstream zone will always try to discharge product. To control this, you need to utilize the Aux I/O Pin 2 signal for the most downstream zone as an *Accumulate* input.
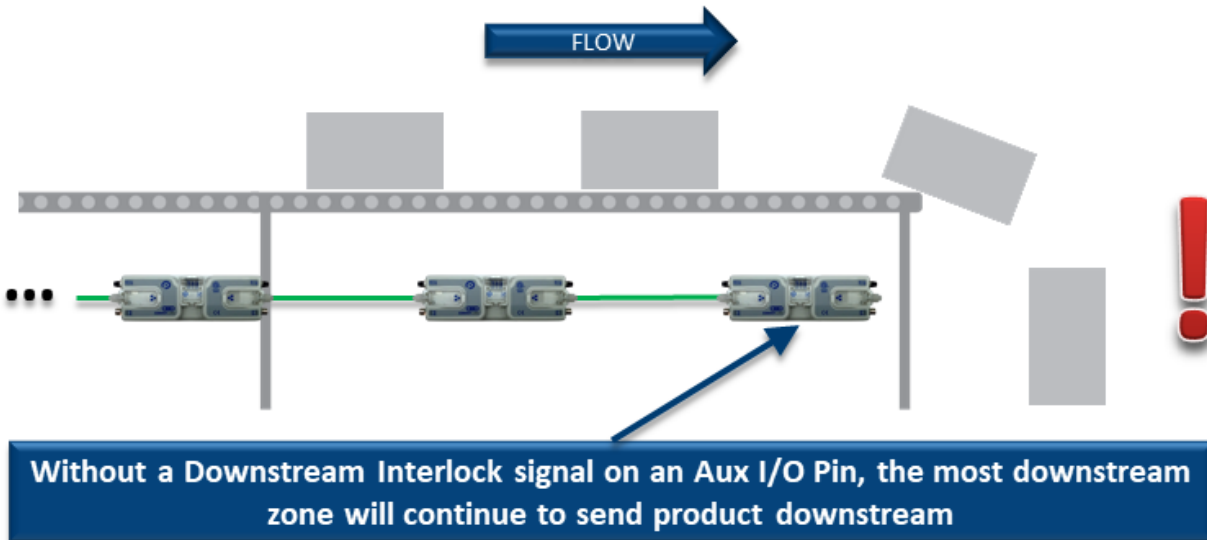
Without a Downstream Interlock signal on an Aux I/O Pin, the most downstream zone will continue to send product downstream

Assuming the most downstream zone is connected to the right side of the module; we set the Right Aux I/O Pin 2 to *Accumulate* from the drop-down box and make sure we click the "DOWN" arrow to indicate that the Right Pin 2 signal is to be associated with the Downstream Zone. When the Right Pin 2 signal is energized, the downstream zone will stop when an item arrives on its sensor
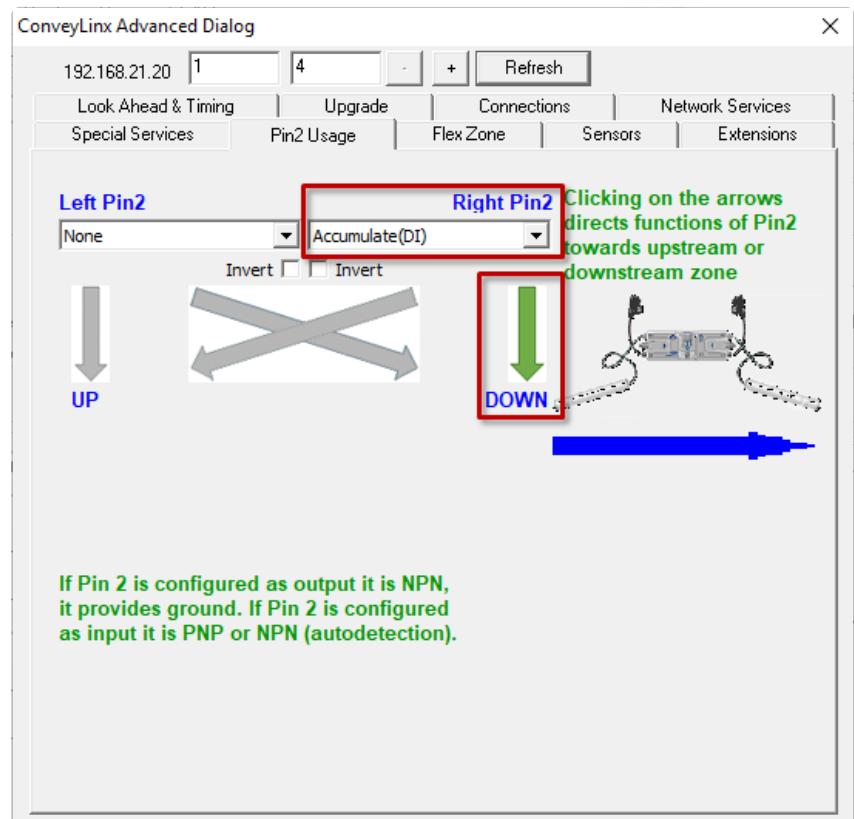
✱ For EasyRoll version 4.21, the drop down selector will indicate "External Control" instead of "None"

7.3.9.2. Most Upstream Zone

By default and without any intervention, the most upstream zone will never turn on to accept new product. To cause the most upstream zone to run to accept an item with a wired signal, you need to utilize the Aux I/O Pin 2 for the upstream zone as a *Wake-up* input.
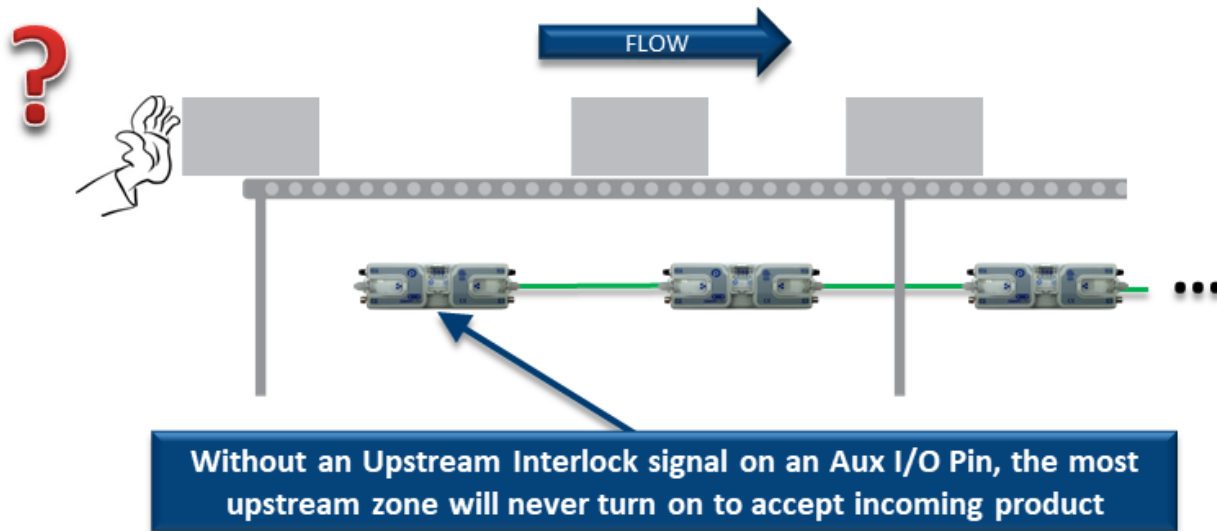
Without an Upstream Interlock signal on an Aux I/O Pin, the most upstream zone will never turn on to accept incoming product

Assuming the most upstream zone is connected to the left side of the module; we set the Left Pin 2 to *Wake up* from the drop-down box and make sure we click the UP arrow to indicate that the Left Pin 2 signal is to be associated with the Upstream Zone. When the Left Pin 2 signal is energized, the upstream zone will run to accept an item
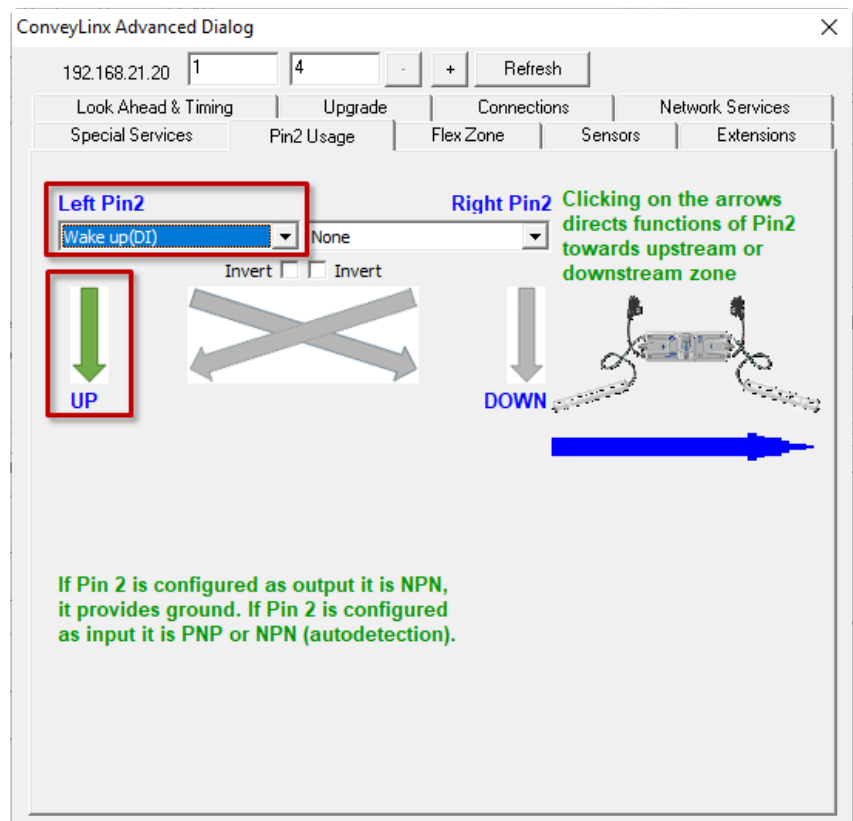
✱ For EasyRoll version 4.21, the drop down selector will indicate "External Control" instead of "None"

7.3.9.3. Accumulate Intermediate Zone

By default, and without intervention; all zones in between the most upstream and most downstream zones always try to convey items downstream if the next downstream zone is clear. To cause an intermediate zone to accumulate based upon a wired signal, you need to utilize the Aux I/O Pin 2 signal for the zone in questions as an *Accumulate* input.

Without a Local Zone Accumulation Control signal, all intermediate zones will always try to convey product downstream when conditions are clear
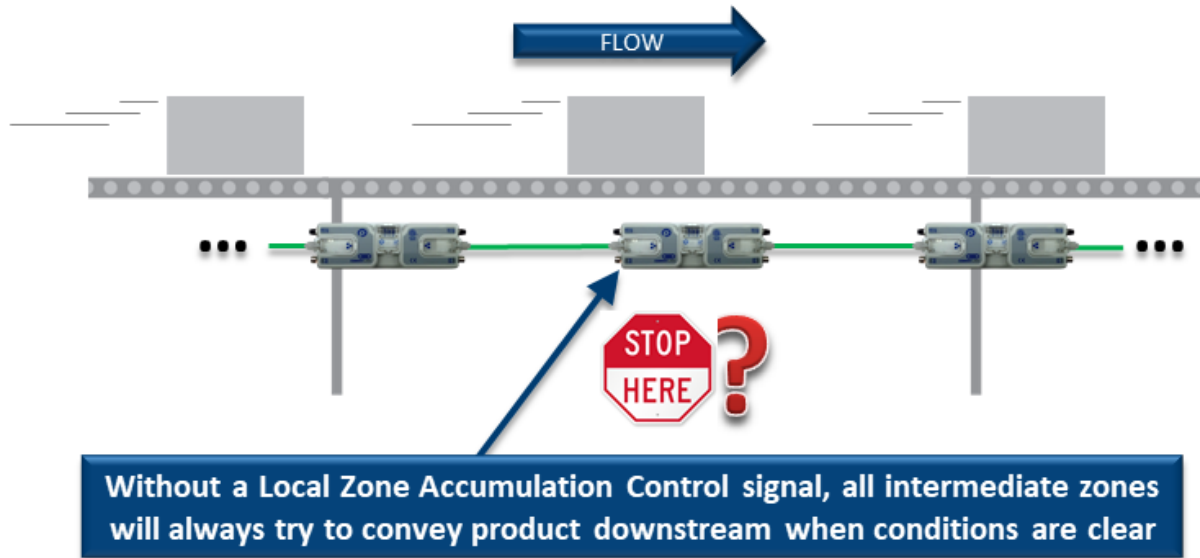
Assuming the intermediate zone we want to accumulate is the upstream zone and the upstream zone is connected to the Left side of the module; we set the Left Aux I/O Pin 2 to *Accumulate* from the drop down box and make sure we click the "UP" arrow to indicate that the Left Pin 2 signal is to be associated with the Upstream Zone. When the Left Pin 2 is energized, the upstream zone will accumulate the next item that arrives
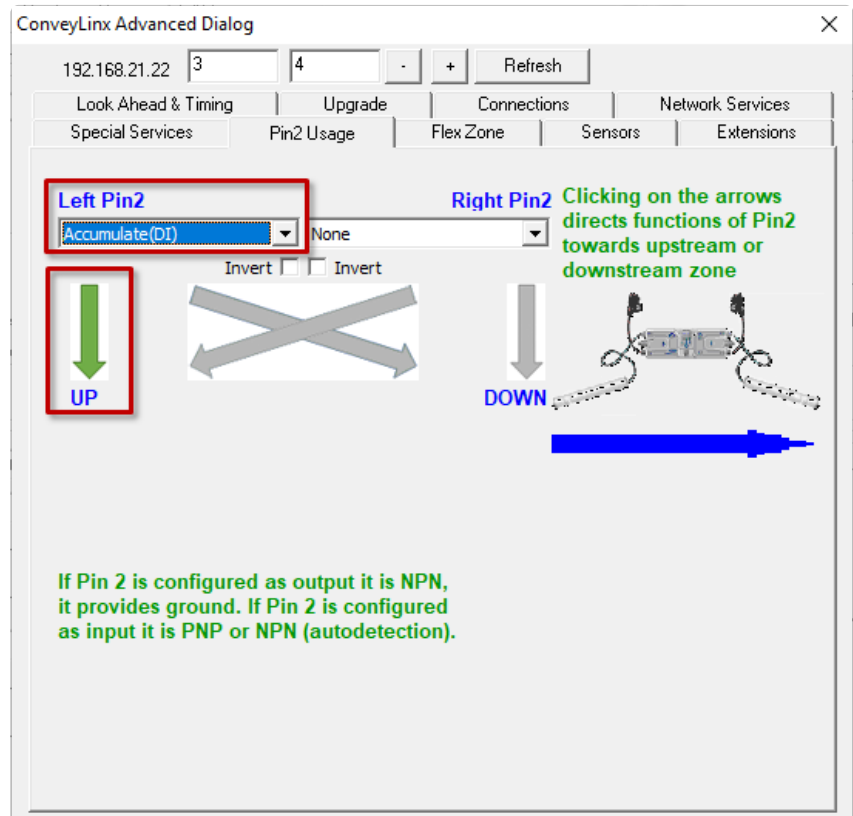
✱ For EasyRoll version 4.21, the drop down selector will indicate "External Control" instead of "None"

7.3.9.4. Lane Full Interface

A special case of the accumulate signal for the most downstream zone is referred to as *Lane Full Interface*. When Lane Full Interface is used on the most downstream zone; the input signal is treated with a block and clear timer such that when the signal is energized, the signal must remain energized for the "block" timer duration. When the block timer expires, then the zone is set to accumulate. Similarly, when the signal is de-energized, the signal must remain de-energized for the "clear" timer duration. When the clear time expires, the zone is set to release as normal. The Lane Full Interface also disables the [Arrival Jam](#) detection logic such that no arrival verification signal from downstream is required. This means that if the signal is off (and the clear time has expired) items arriving at the most downstream sensor will immediately release.

Assuming the most downstream zone is connected to the right side of the module; we set the Right Aux I/O Pin 2 to *Lane Full Interface* from the drop-down box and make sure we click the "DOWN" arrow to indicate that the Right Pin 2 signal is to be associated with the Downstream Zone. The signal must be energized for the Block time period before cartons will stop and accumulate and the signal must be de-energized for the Clear time period before cartons will be allowed to release. In this example, we entered 5 seconds for the Block Time and 6 seconds for the Clear Time
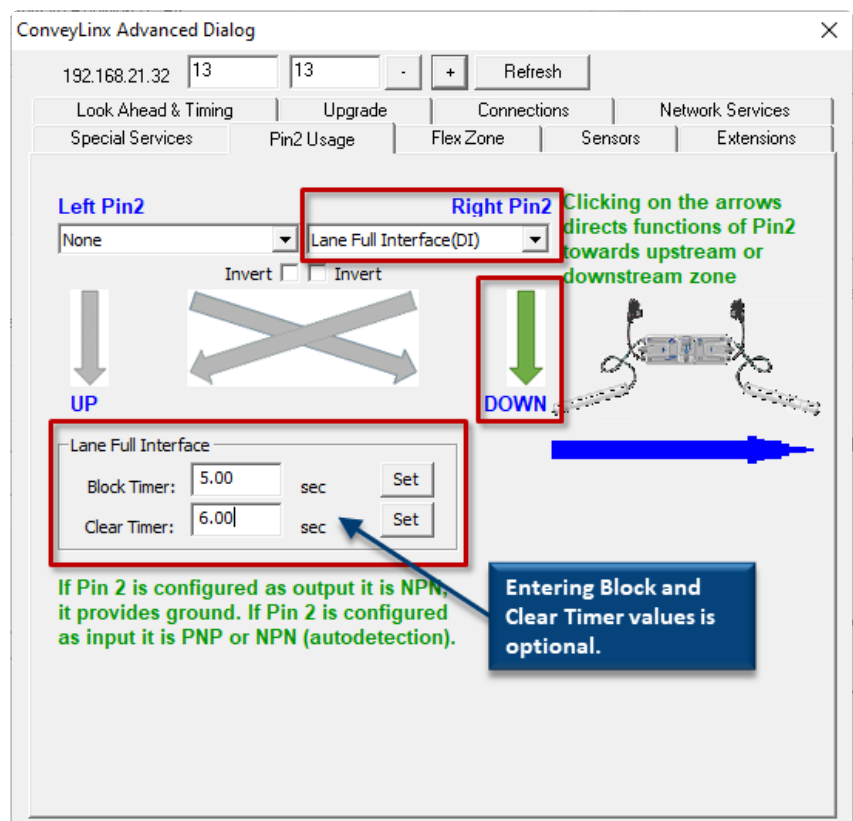
For EasyRoll version 4.21, the drop down selector will indicate "External Control" instead of "None"

7.3.9.5. Most Upstream Zone Handshake Interlock

In applications where the equipment feeding the most upstream zone of ConveyLinx controlled conveyor requires a handshake interlock to know when the most upstream zone of the ConveyLinx controlled conveyor is empty and ready to accept a new item, this can be achieved by utilizing both the Left and Right Aux I/O Pin signals. One of the Aux I/O Pin 2 signals needs to be set as an input to *Wake Up* the zone and the other signal needs to be configured as an output to indicate whether there is *Product on Zone*. When this *Product on Zone* output is energized, then the feeding equipment knows that the most upstream zone is occupied and is not ready to accept a new item.

Assuming the most upstream zone is connected to the left side of the module; we set the Left Pin 2 to *Wake up* from the drop-down box and make sure we click the UP arrow to indicate that the Left Pin 2 signal is to be associated with the Upstream Zone. When the Left Pin 2 signal is energized, the upstream zone will run to accept an item
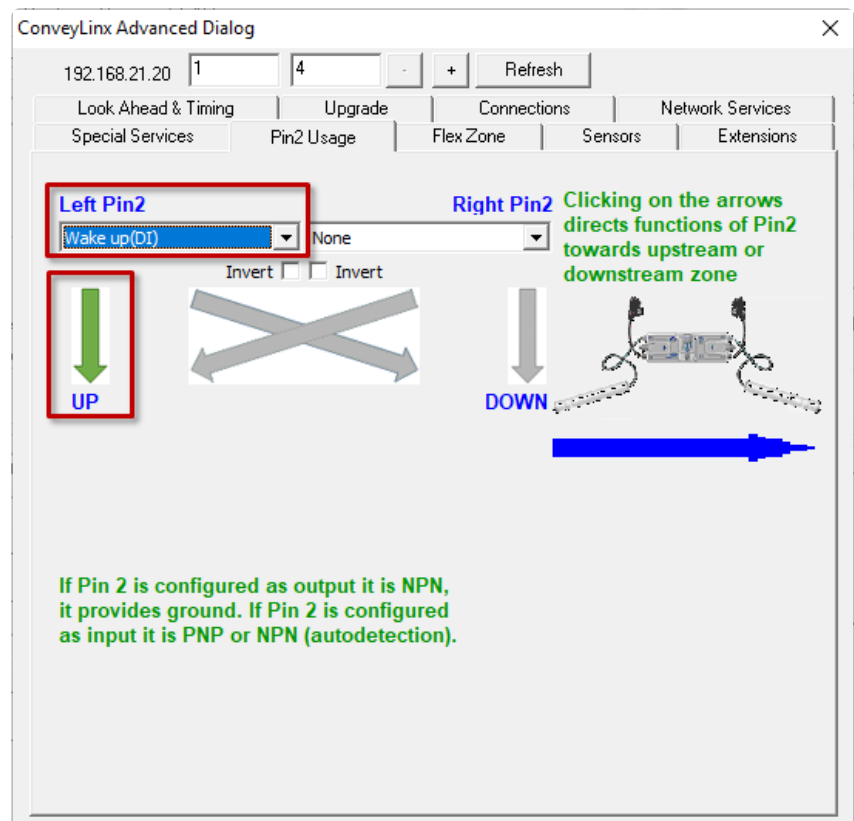
✱ For EasyRoll version 4.21, the drop down selector will indicate "External Control" instead of "None"

Then we select *Product on Zone* for the Right Pin 2 signal and click the diagonal arrow to indicate that this signal is to reflect the status of the Upstream zone
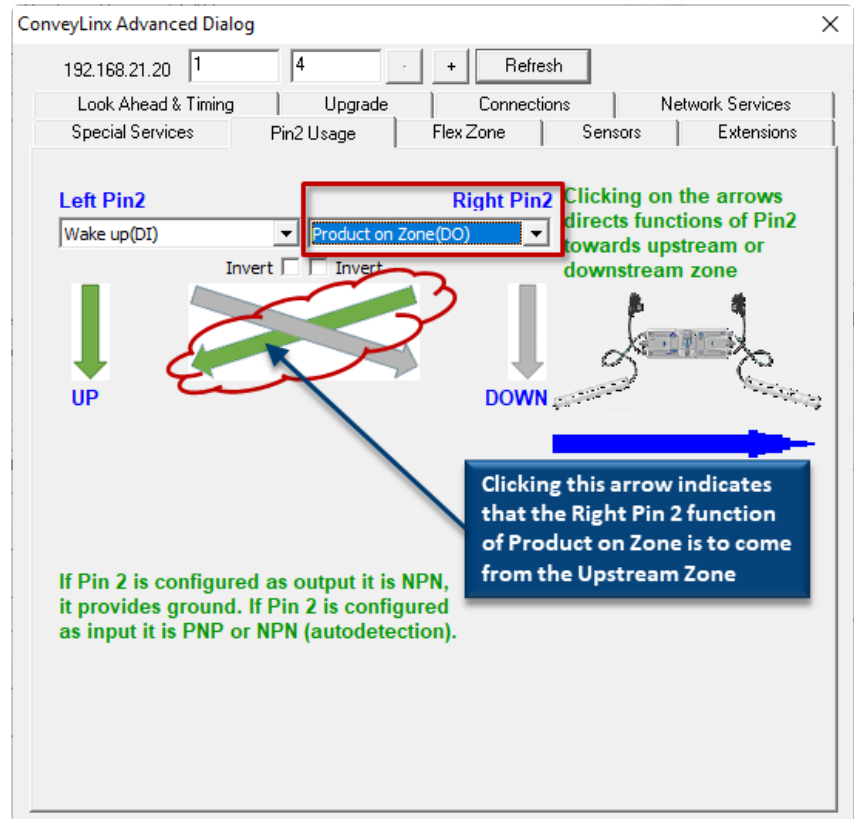
7.3.9.6. Most Downstream Zone Handshake Interlock

In applications where the equipment accepting product from the most downstream zone of ConveyLinx controlled conveyor requires a handshake interlock to know when the most downstream zone of the ConveyLinx controlled conveyor is occupied and ready to discharge the item, this can be achieved by utilizing both the Left and Right Aux I/O Pin signals. One of the Aux I/O Pin 2 signals needs to be set as an input to *Accumulate* the zone and the other signal needs to be configured as an output to indicate whether there is *Product on Zone*. When this *Product on Zone* output is energized, then the accepting equipment knows that the most downstream zone is occupied and is ready to discharge the item.

Assuming the most downstream zone is connected to the right side of the module; we set the Right Pin 2 to *Accumulate* from the drop-down box and make sure we click the "DOWN" arrow to indicate that the Right Pin 2 signal is to be associated with the Downstream Zone. Similarly, we select *Product on Zone* for the Left Pin 2 signal and click the diagonal arrow to indicate that this signal is to reflect the status of the Downstream zone
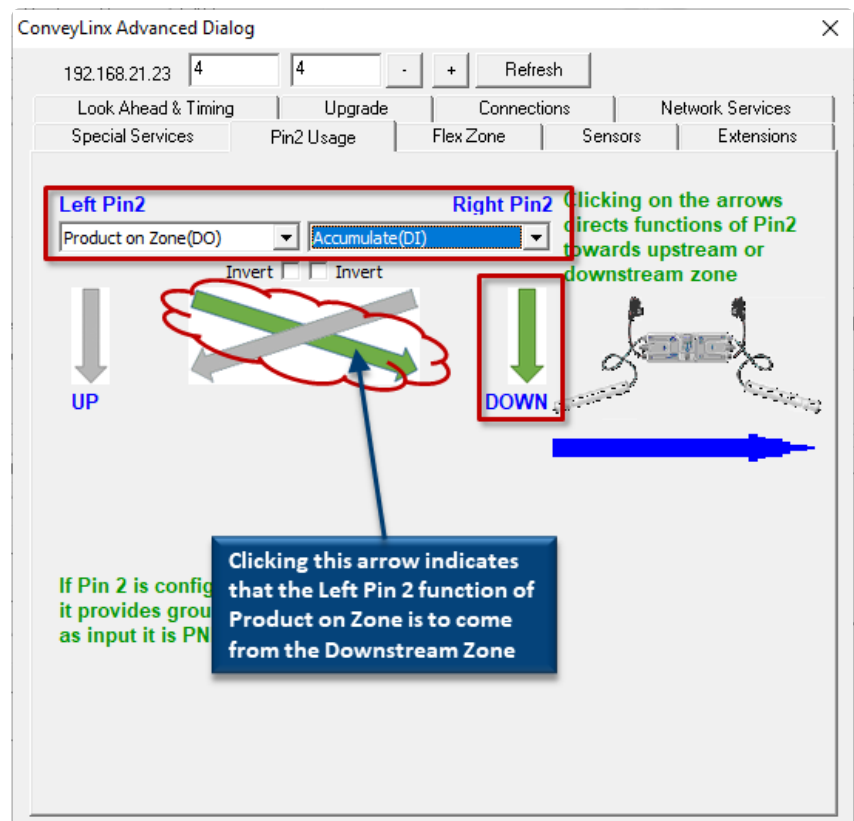
7.3.9.7. Inverting the Pin 2 Signals

On this dialog, you can also invert the meaning of the electrical signal by checking the Invert box for either or both Pins. In this example, because we have the *Invert* box checked for both Pin signals, their respective functions will be activated when their signal is electrically OFF.

Note that the *Invert* function works similarly for the output signals. If Product on Zone is selected as a function, with the Invert box checked when the zone is occupied, the electrical signal will be OFF. With the Invert box checked, when the zone is clear, the electrical signal will be ON
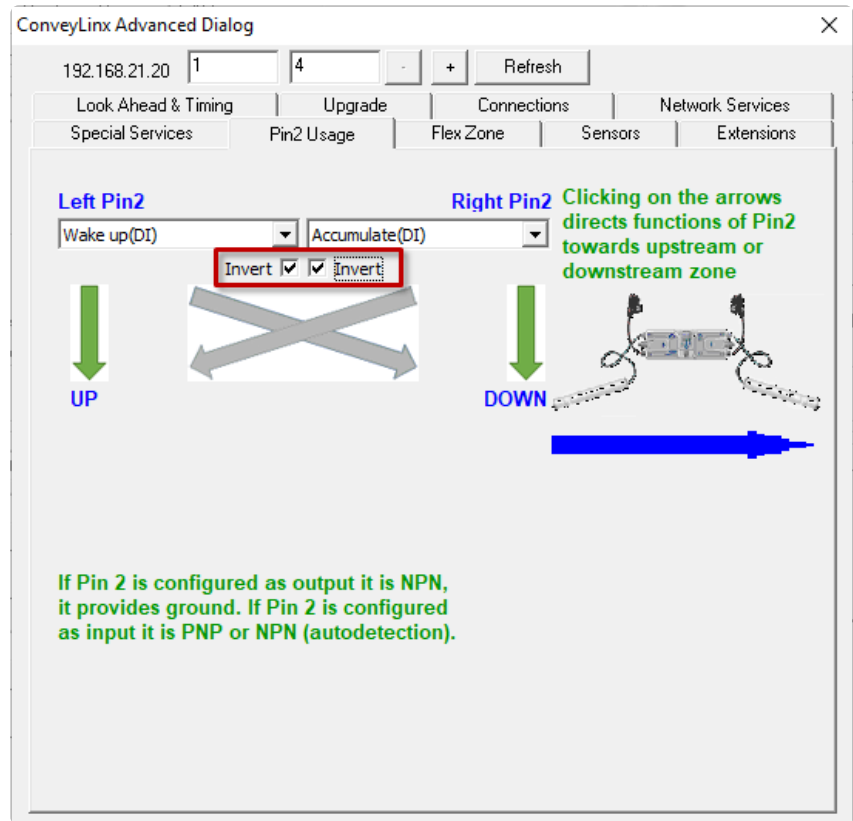
8. EasyRoll+ Software

* EasyRoll+ works for **both 24V** based and **48V** based ConveyLinx controllers. However, EasyRoll+ is <u>required</u> for **any 48V** based ConveyLinx controller

* EasyRoll+ will support a mix of 24V and 48V ConveyLinx controllers on the same network

EasyRoll+ provides all of the functionality of EasyRoll with some added features and a more intuitive user interface.

EasyRoll+ Features:

- Offline configuration of ConveyLinx Networks
- Support for PCs with more than one NIC
- Data point watch and capture including visual graphing
- Support for ConveyLinx 48V based controllers
- Power Management and control for 48V based controllers

[Download EasyRoll+](#)

9. ConveyMerge

> ✳ *ConveyMerge* is available on all ConveyLinx and ConveyLinx-Ai Family controllers. Check all prerequisites here

ConveyMerge is an optional functionality layer built on top of *ConveyLinx's ZPA* logic. Users can enable and configure it using *EasyRoll* software. *ConveyMerge* behaves the same as ZPA logic, but allows you to define a special *Merge Zone* that can have up to three upstream zones connected to it. *ConveyMerge* also has simple priority schemes for the multiple upstream zones as well as the ability to dynamically change those priorities on-the-fly by utilizing the *Merge Line Full* option. It also takes care of passing tracking information properly and has *Jam Condition* detection.

ConveyMerge Components

ConveyMerge can accommodate product merge control from one or two conveyor paths onto a single take-away conveyor path as shown:
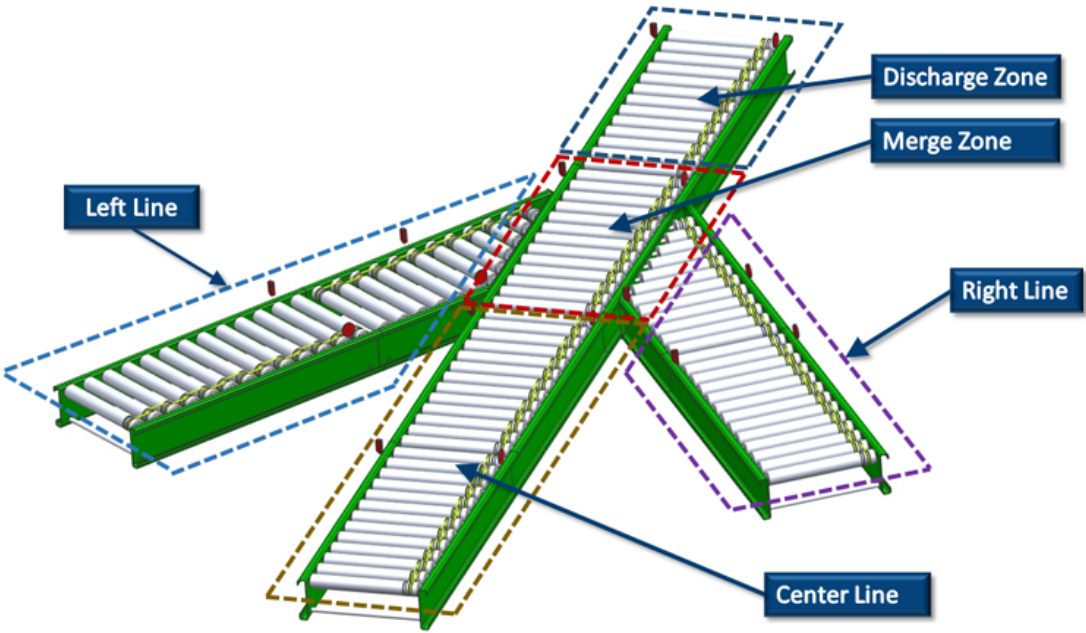
Component	Description
Merge Zone	Powered conveyor section that accepts an item from any one of the 3 possible upstream sources
Main Line	Term to describe the 3 in-line conveyor sections – *Center Line*, *Merge Zone* and *Discharge Zone*

Center Line	Part of *Center Line* that is one of three possible sources of items that can flow into the *Merge Zone*. This term is used in the configuration screen of *EasyRoll* to designate the network that serves as the trunk line that contains the *Merge Zone*
Discharge Zone	Part of the *Center Line* that is downstream of the *Merge Zone*. This conveyor section accepts the item from the *Merge Zone*
Left Line	One of the three possible conveyor sections that can discharge items into the *Merge Zone*
Right Line	One of the three possible conveyor sections that can discharge items into the *Merge Zone*

9.1. ConveyMerge Prerequisites and Requirements

Please Note – ConveyMerge works for ALL ConveyLinx Family Modules

Module	Firmware
ConveyLinx-ERSC	4.25 and later
ConveyLinx-HTF	4.25 and later
ConveyLinx-Ai2	4.20 and later
ConveyLinx-Ai3-24-xx	4.20 and later
ConveyLinx-Ai2-48	Any
ConveyLinx-Ai3-48-FC	Any

The following items are the prerequisites and constraints required to provide a fully functioning automatic conveyor merge:

- *EasyRoll* version 4.3 or greater or *EasyRoll+* version 1.2.9 or greater (**EasyRoll+** is required for any **48V** based ConveyLinx controller)
- Only (1) one *Merge Zone* is allowed for any given single module
- *Merge Zone* cannot have any *Extension (slaved) zone*
- *Merge Zone* must be in *ZPA* mode. *Left Line* discharge, *Right Line* discharge, and *Center Line* discharge can be in either *ZPA* mode or *PLC I/O* mode
- Left, Right, and Center Lines discharges cannot also be included in another instance of *ConveyMerge*

✱ If *Left Line* discharge, *Right Line* discharge, or *Center Line* discharge modules are in PLC I/O mode, you are responsible for programming logic to recognize and respond to ZPA zone status values between the *Merge Zone* module for proper jam free singulation control

9.2. Network Architecture

All lines (*Center*, *Left*, and *Right*) should be on separate *subnets* of modules, as shown. Each *subnet* should be *Auto-Configured* separately and then connected together with an Ethernet switch. Keep in mind that the separate *subnets* will have to be able to communicate with each other so special care must be taken when setting up the *IP Network Mask* before performing the *Auto-Configuration Procedure*.

> ✻ A good starting point is the default mask of 255.255.128.0. With this setting you can vary the third octet in the ranges 0-127 and 128-254 and all three lines will be able to communicate with each other as long as the third octets of their IP addresses are in the same range (either 0-127 or 128-254).

> ✻ Even though *ConveyMerge* can work with all lines being in the *same subnet*, care must be taken during *Auto-Configuration* to assure proper zone flow for all lines. We recommend *separate subnets* in order to assure proper zone flow for all lines.
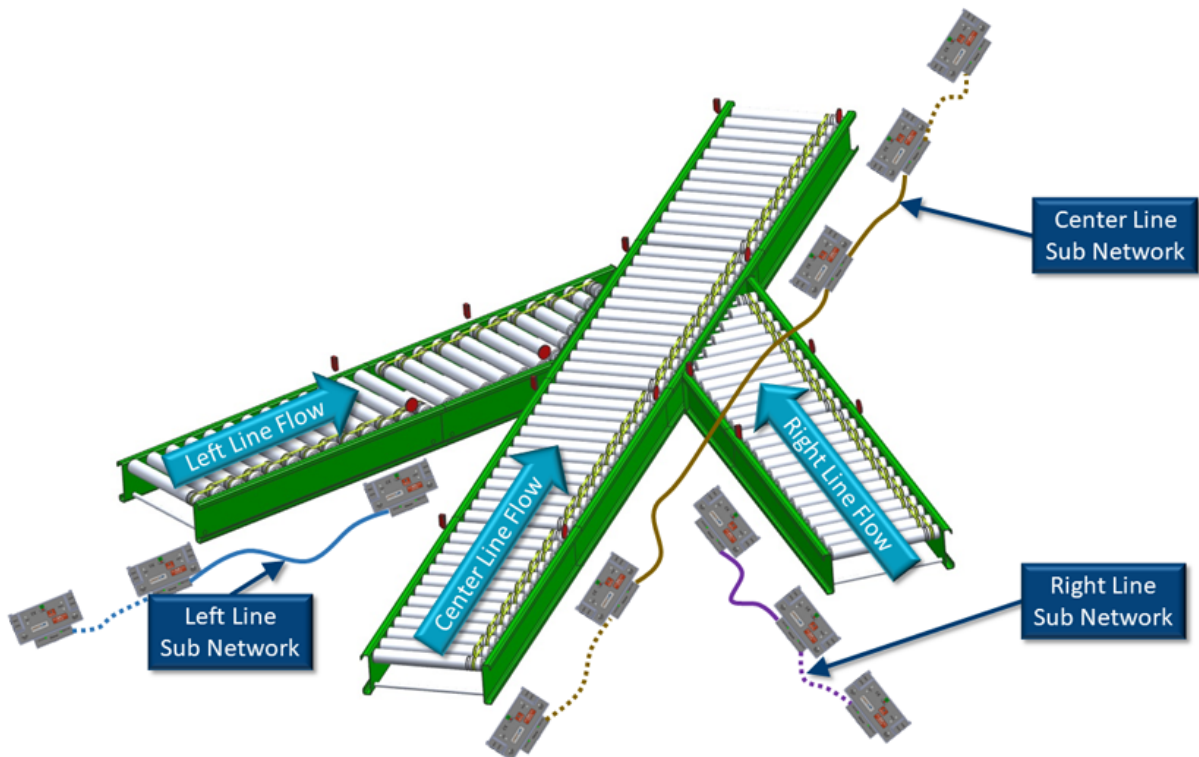
Architecture Summary

- The *Center Line* is a separate *subnet* of modules that have been *Auto-Configured*. The

Merge Zone and the *Discharge Zone* are part of this *subnet*. The *Center Line* can consist of any number of *ConveyLinx-Ai Family Controllerss* and the actual position of the *Merging Zone* can be on any *ConveyLinx-Ai Family Controllers* within the *subnet*

- The *Discharge Zone* has to be the zone immediately downstream of the *Merge Zone*. However, the *Discharge Zone* can be on either the same ConveyLinx-Ai Family Controllers as the *Merge Zone* or on the adjacent downstream ConveyLinx-Ai Family Controllers from the *Merge Zone's* ConveyLinx-Ai Family Controllers

- The *Left Line* is a separate *subnet* of modules and its most downstream zone has to physically discharge on to the *Merge Zone*

- The *Right Line* is a separate_* subnet*_ of modules and its most downstream zone has to physically discharge on to the *Merge Zone*

- All three *subnets* must be connected to a common Ethernet switch <u>only after</u> each *subnet* has been *Auto-Configured*.

9.3. Sensor Placement

> ❗ It is very important for the internal *ConveyMerge* logic, that the product is seen by only one sensor at a time

Conventional Spur Merge Sensor Placement

90° T-Merge Sensor Placement

Line Discharge Zone Sensor

Merge Zone Sensor T-Zone Configuration

Line Discharge Zone Sensor
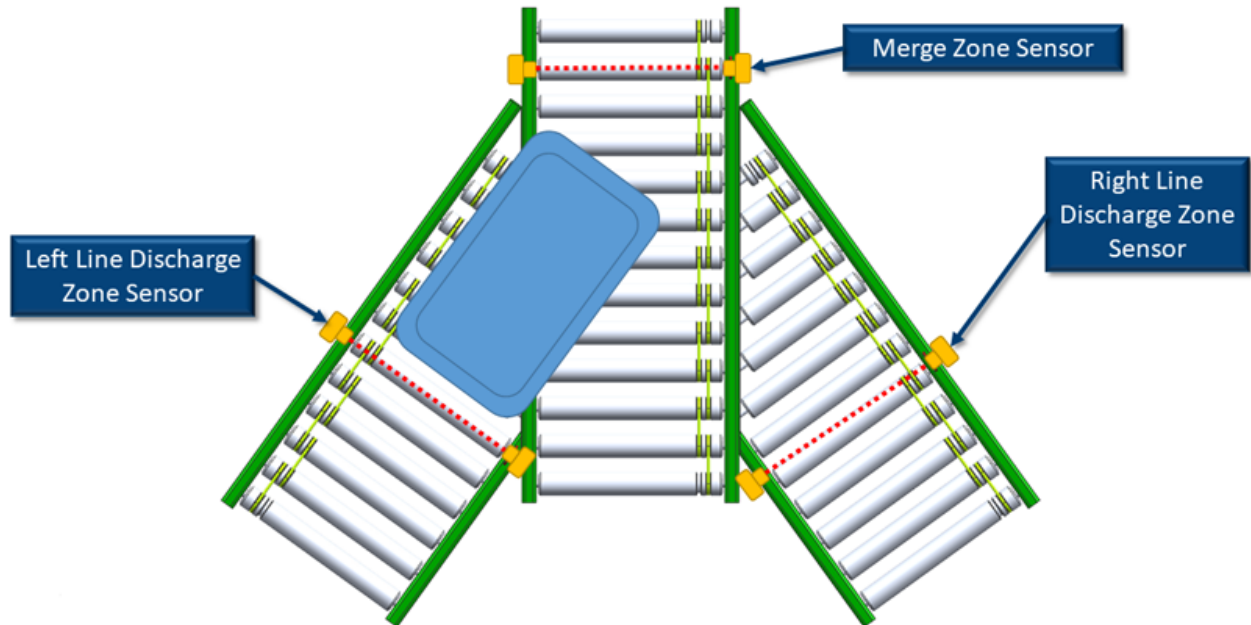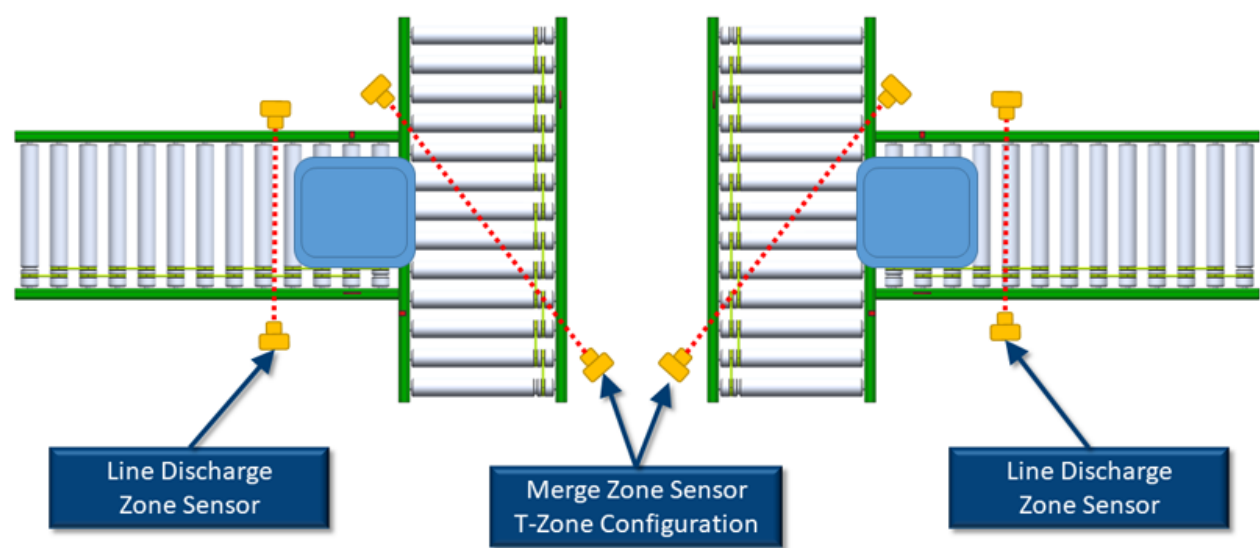
9.4. Merge Zone Module

Depending on the physical arrangement of actual zones on the *Center Line*, the *Merge Zone* can be assigned to either the Upstream or Downstream zone of its particular ConveyLinx-Ai Family Controllers. It can also be assigned to a ConveyLinx-Ai Family Controllers that is configured as a single zone ZPA controller.

In all 4 of these examples, *Node 2* is the Merge Zone's controlling ConveyLinx-Ai Family Controllers

> ✳ Images show ConveyLinx-ERSC modules but functions are applicable to all ConveyLinx Family modules

Merge Zone on Upstream Zone
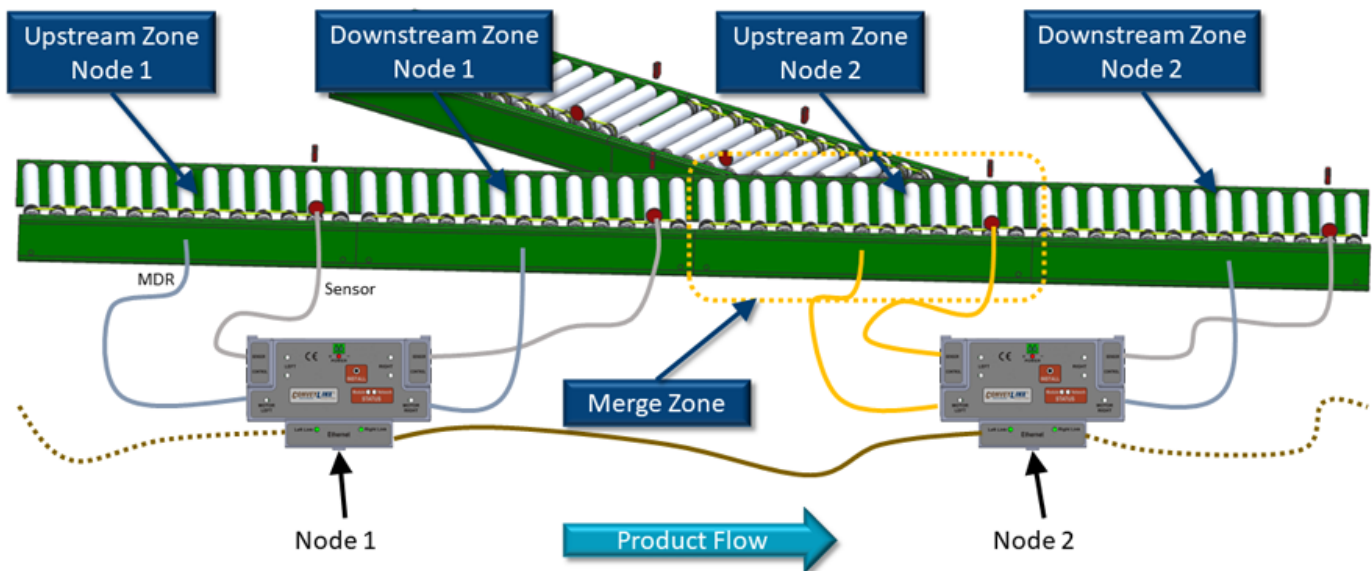
Merge Zone on Downstream Zone
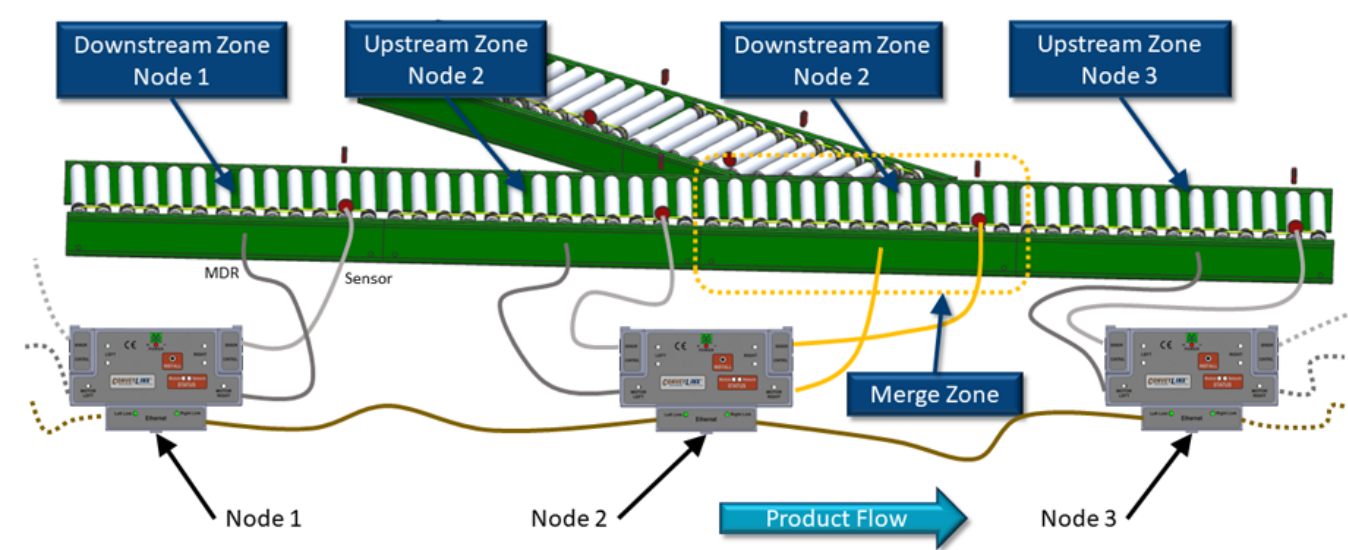
Single Zone Module

Whenever an ConveyLinx-Ai Family Controllers has been *Auto-Configured* as a **single zone**; regardless of whether the single zone is connected to the right side or left side of the ConveyLinx-Ai Family Controllers, the logical designation for the zone on the this module is always the Upstream Zone. Shown below are examples of a single zone module using the left and right sides of the module respectively.
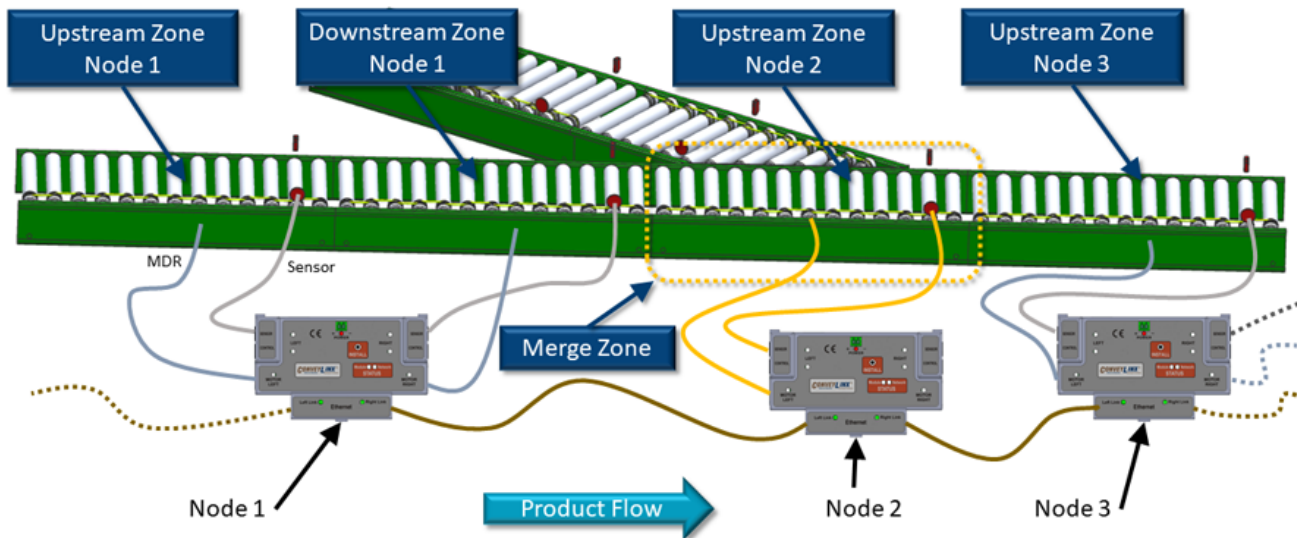
Merge Zone on a Single Zone Module's Left Side
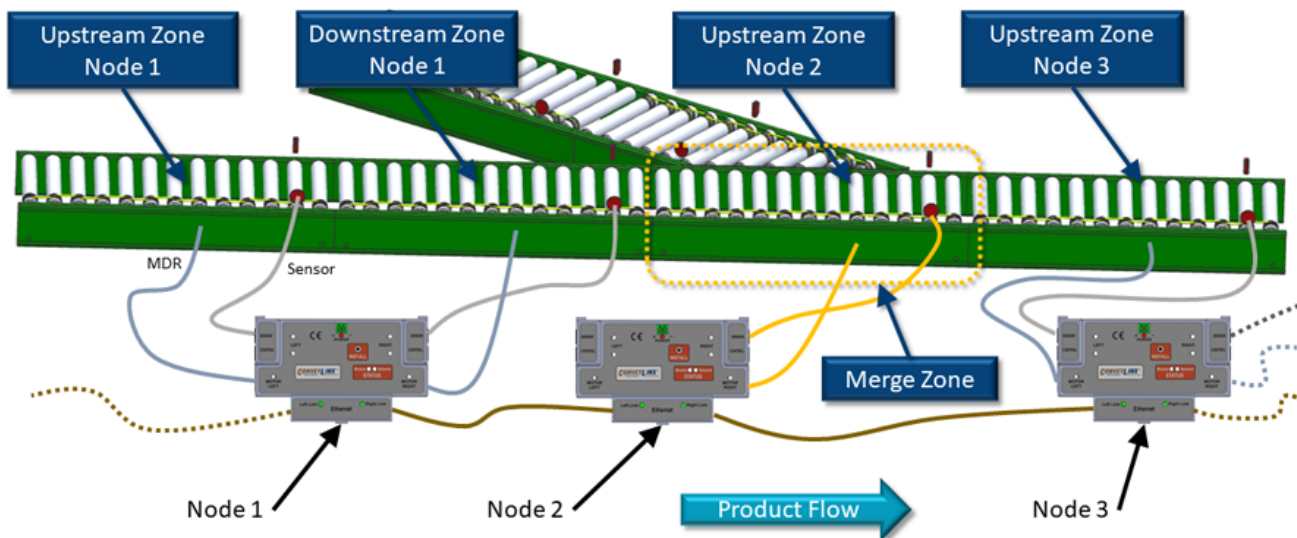
Merge Zone on a Single Zone Module's Right Side

9.5. Merging Lines

The *Left Line* and *Right Line* are set up in a similar fashion to the *Center Line*. Each of these lines are in their own separate *subnet* that is both different from each other and different from the *Center Line*. Regardless of whether these merging lines are an Conventional Spur arrangement or a 90° T-Merge arrangement; the conveyor zone that discharges product onto the *Merge Zone* <u>must be the most downstream zone if its particular subnet</u>. The following figures illustrate this for an *Conventional* Spur Left Line and a 90° *T-Merge* Left Line respectively. The same would apply if these were *Right Line* examples as well.

> ✱ Images show ConveyLinx-ERSC modules but functions are applicable to all ConveyLinx Family modules

Most Downstream Zone of the Last Node on the Left Line Sub Network

Last Node of the Left Line Sub Network

Merge Zone
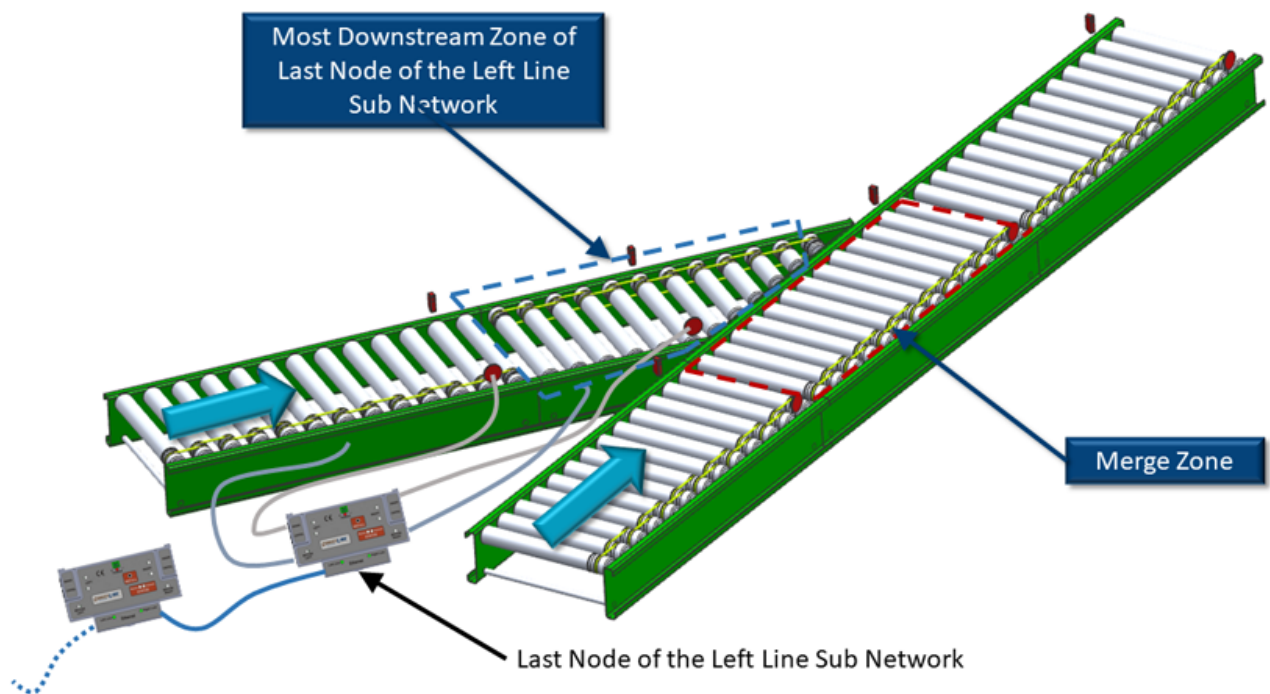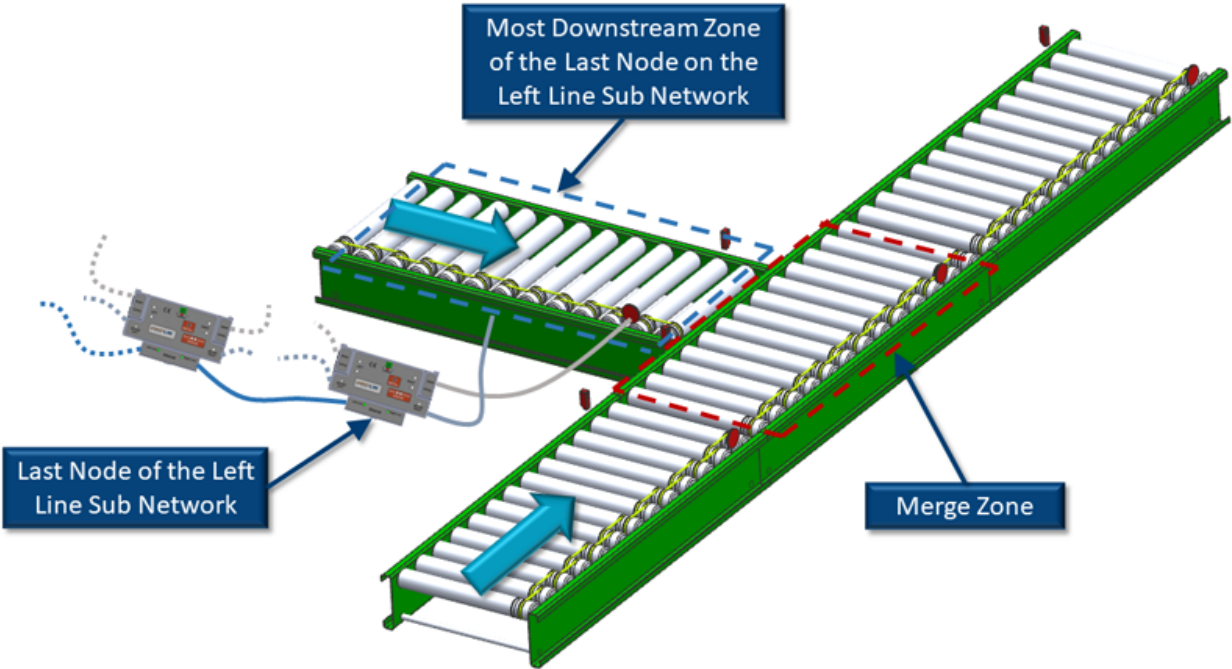
9.6. Merge Configurations

The *Merge Configurations* available from within in *EasyRoll* are:

- *Left Line* Only with either *Conventional* (spur discharge) or *T-Merge* (90˚ discharge)
- *Right Line* Only with either *Conventional* (spur discharge) or *T-Merge* (90˚ discharge)
- Both *Left* and *Right* Lines with either *Conventional* (spur discharge) or *T-Merge* (90˚ discharge) for each

9.7. Merge Priority

EasyRoll provides the ability to adjust which of the merging lines receives priority.

EasyRoll Priority Selection	Description
First Come First Served	The logic monitors the arrival status in each of the zones that discharge in to the *MergeZone*. The first product that arrives at any of these lines will be allowed to convey into the *MergeZone*. When the *MergeZone* is clear, the next line that has had a product arrive will be allowed to convey into the *MergeZone*
Center Line Priority	The *Center Line* will be allowed to release as long as product arrives at the *Center Line's* zone sensor prior to the previously released product having not cleared the *MergeZone*. When the *Merge Zone* is clear and if there is no product sensed on the center line zone; then the *Left* and/or *Right Lines* will be allowed to release if product has arrived at their respective sensors. As long as no product is arriving on the *Center Line*, if *Left* and *Right Lines* are enabled, these will operate as *First Come First Served* between them
Left Line Priority	The *Left Line* will be allowed to release as long as product arrives at the *Left Line's* most downstream sensor prior to the previously released product having not cleared the *MergeZone*. When the *Merge Zone* is clear and if there is no product sensed on the *Left Line's* most downstream sensor; then the *Center* and/or *Right Line* will be allowed to release if product has arrived at their respective sensors. As long as no product is arriving on the *Left Line*, if the *Right Line* is enabled, the *Center* and *Right* will operate as *First Come First Served* between them
Right Line Priority	The *Right Line* will be allowed to release as long as product arrives at the *Right Line's* most downstream sensor prior to the previously released product having not cleared the *MergeZone*. When the *Merge Zone* is clear and if there is no product sensed on the *Right Line's* most downstream sensor; then the *Center* and/or *Left Line* will be allowed to release if product has arrived at their respective sensors. As long as no product is arriving on the *Right Line*, if the *Left Line* is enabled, the *Center* and *Left* will operate as *First Come First Served* between them

9.8. T-Merge Settings

For a *T-Merge*, the discharging conveyor is arranged perpendicular to the *Merge Zone* and this discharging conveyor needs to "throw" the product across the Merge Zone's rollers. In order to mitigate carton alignment issues, the logic controls need to make sure that it keeps the *Merge Zone's* rollers stopped until the discharging conveyor has completely finished "throwing" the product onto the *Merge Zone*. In order to accommodate various conveyor speeds and material handling situations; *EasyRoll* provides some timer adjustments to facilitate proper *T-Merge* functionality.

Accept Timeout

This timer specifies how much time the *Merge Zone* will wait after its sensor is blocked until the logic allows it's motor to run. Once the timer has expired, the Merge Zone's motor will be allowed to run. Keep in mind that conditions downstream of the *Merge Zone* also dictate if the motor is allowed to run.
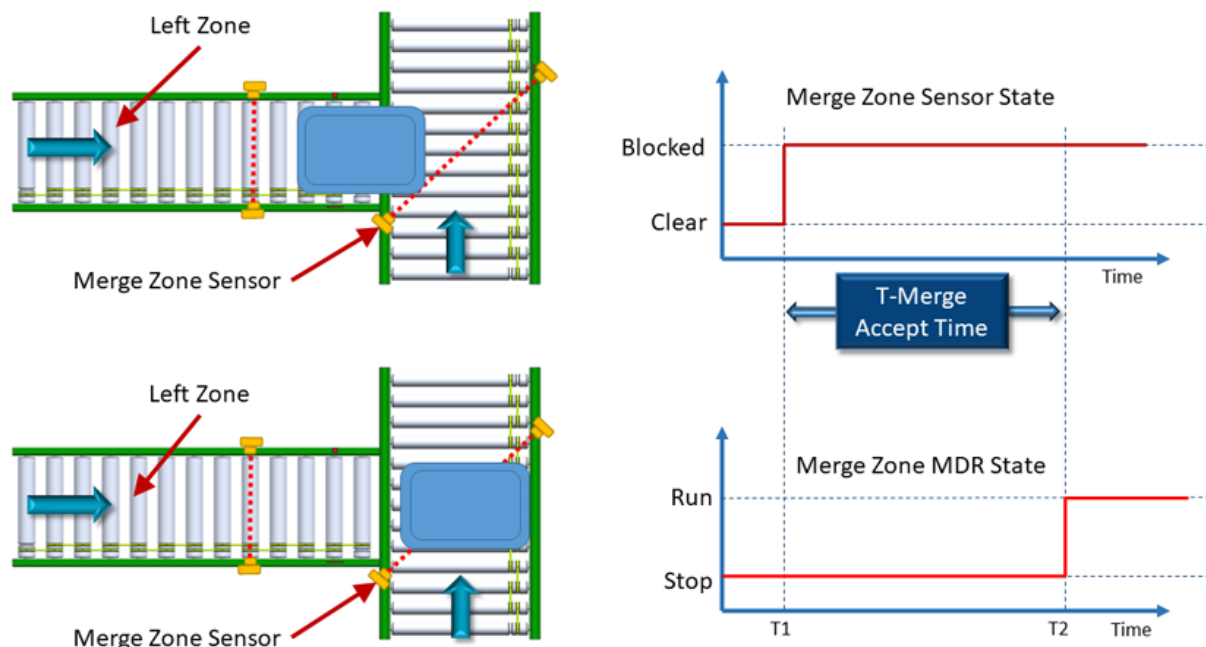
Send Timeout

This timer specifies how much time the discharging zone has to wait after the *Merge Zone* sensor is clear before it is allowed to release the next product into the *Merge Zone*. This timer helps ensure that the *Merge Zone* is clear when sensor placement cannot guarantee that the zone is clear.
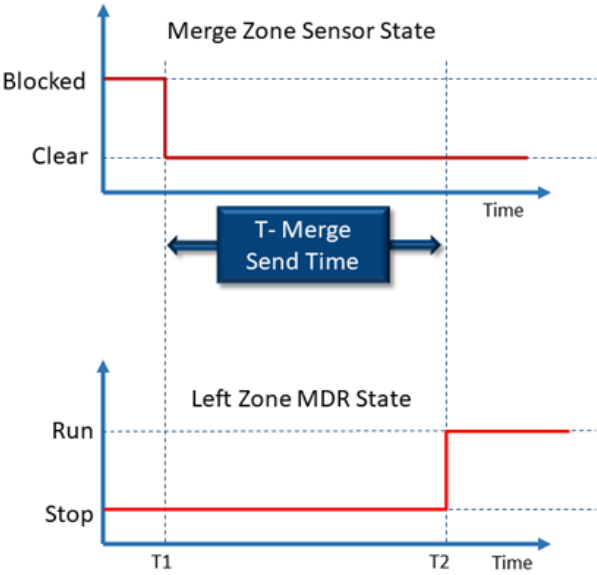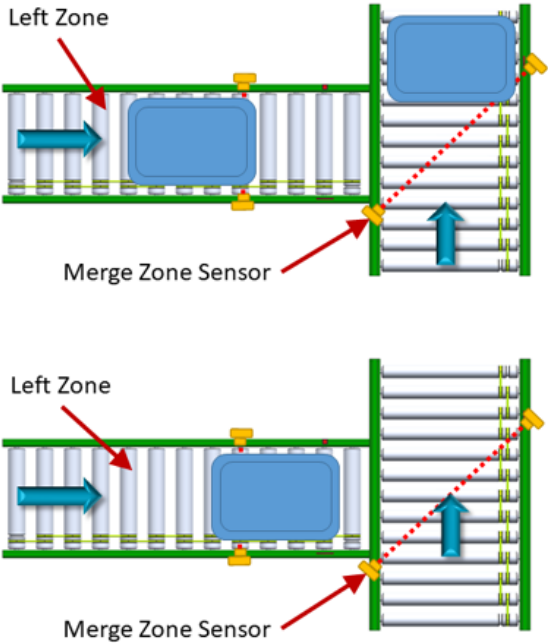
9.9. Configuring Dynamic Priority Release

In situations where the full status of a given merge line needs to determine merge priority, *ConveyMerge* provides the ability to monitor the status of one or all merging lines and base the release priority accordingly. *EasyRoll* provides the option to select a given upstream zone on a given line as the *Monitor Zone*. When the *Monitor Zone* becomes occupied for a given time value; its line will be given release priority into the *Merge Zone*. There are two adjustable time values for this function:

EasyRoll Selection	Description
Block Timeout	Specified the amount of time the *Monitor Zone* must be occupied before the Monitor Zone's line will receive priority to release into the *Merge Zone*. This timer starts when the *Monitor Zone* first becomes accumulated (sensor blocked and motor stopped) and continues to time as long as the *Monitor Zone* stays accumulated
Clear Timeout	Specified amount of time that the priority release will continue to be active once the priority release has begun. When this time value expires, the priority setting will return to its original *EasyRoll* configured setting

✳ If multiple lines are configured for *Dynamic Priority* and each of their respective *Block Timers* have expired; *ConveyMerge* logic will revert to the *First Come First Serve* release mode until one or more of the releasing lines' *Clear Timer* has expired

Dynamic Priority Release Example

Please refer to Figure A and Figure B below. In *EasyRoll*, Node 2 upstream zone of the *Center Line* has been selected as the *Merge Zone* and the merge type is Up Left. The *EasyRoll* setting for priority is *Center Line Priority*. Also in EasyRoll, the *Merge Line Full* function has been enabled as assigned to Node 3 downstream zone of the *Left Line*.
With the *Merge Line Full* feature enabled on the *Left Line*; once Left Line Node 3 downstream zone remains accumulated for the *Block Timer's* timeout (Figure A); priority will dynamically change to the *Left Line*. This Left Line priority will remain in effect until the *Clear Timer's* timeout has expired (Figure B). When this occurs, the priority will automatically revert to the *EasyRoll* configured priority (*Center Line* in this example).

In this example, if we did not enable the *Merge Line Full* function for the *Left Line* and if there was a steady flow of product on the *Center Line*; the *Left Line* would never be able to release because the default priority in EasyRoll was set to *Center Line*.

✳ Images show ConveyLinx-ERSC modules but functions are applicable to all ConveyLinx Family modules
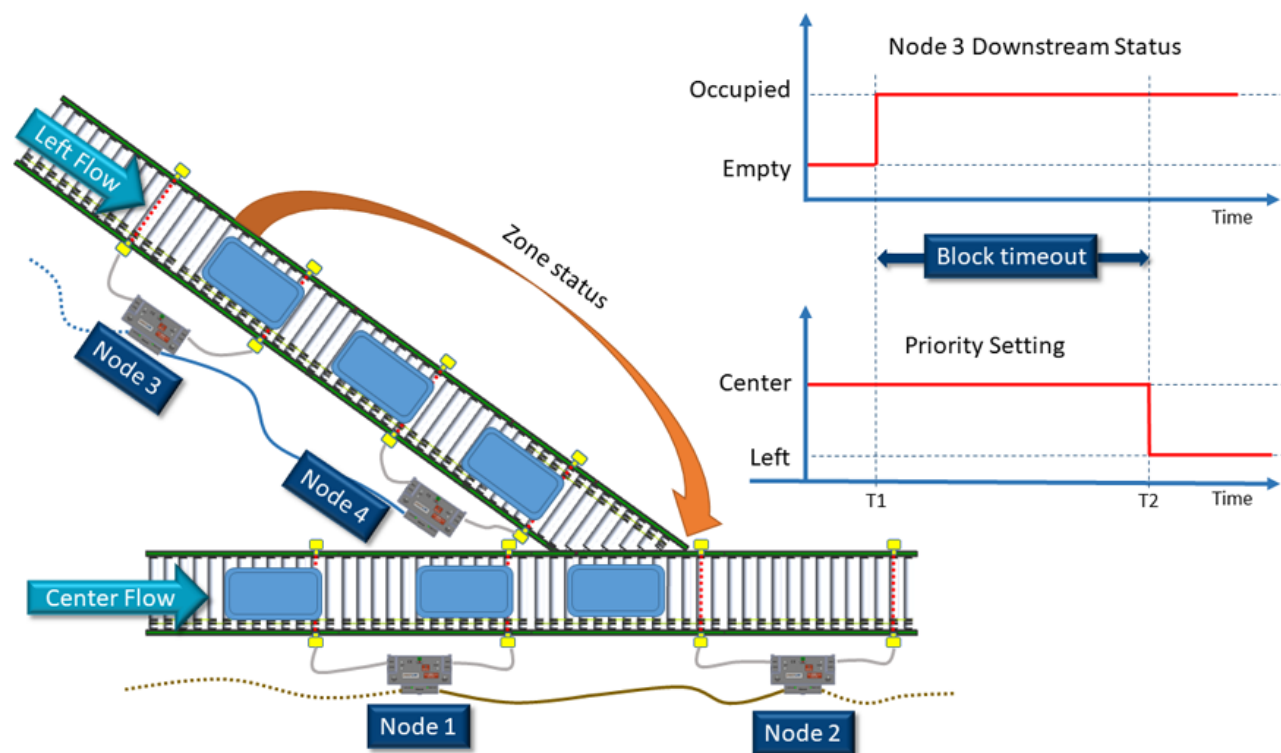
Figure A - Merge Line Full Block Timer
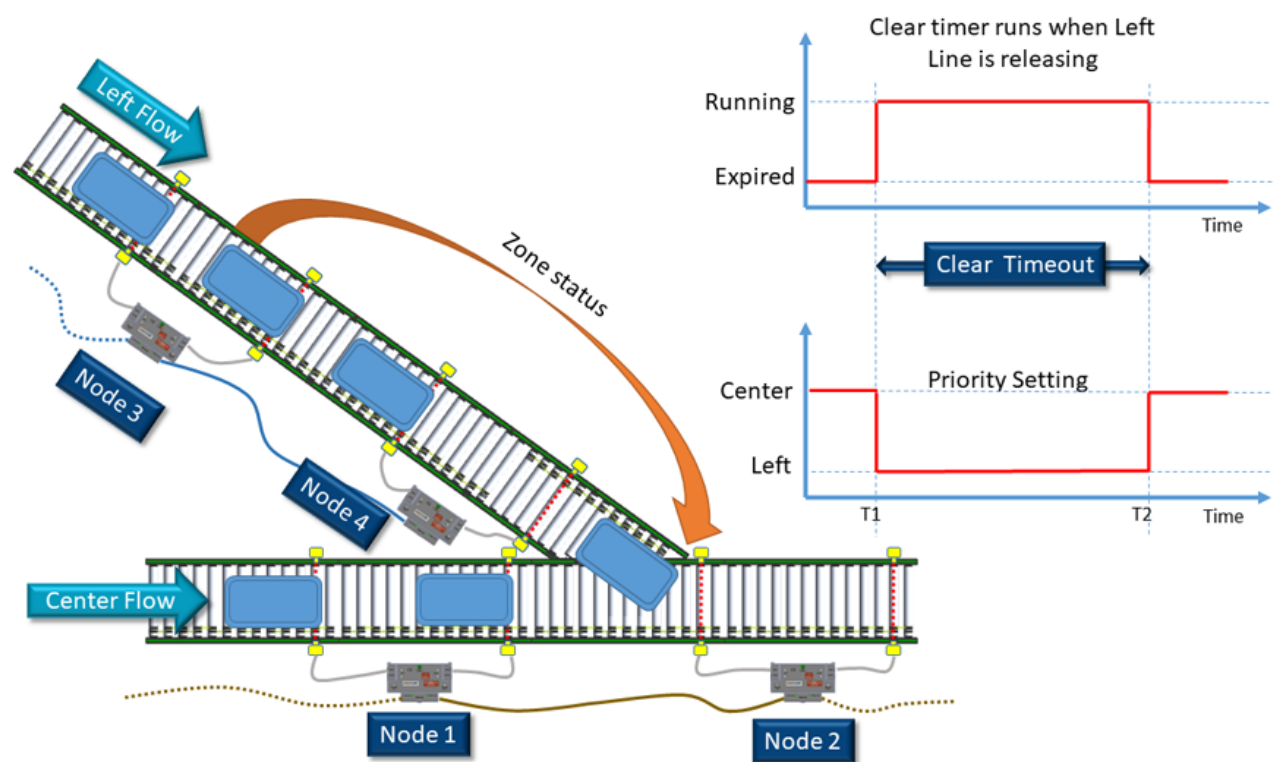
Figure B - Merge Line Full Clear Timer

9.10. Enabling ConveyMerge from EasyRoll

• Open *EasyRoll*
• Connect and navigate to the ConveyLinx-Ai Family Controllers that will control the *Merge Zone*.
• Press *CTRL+SHIFT+M* to open the *ConveyMerge Configuration Screen*
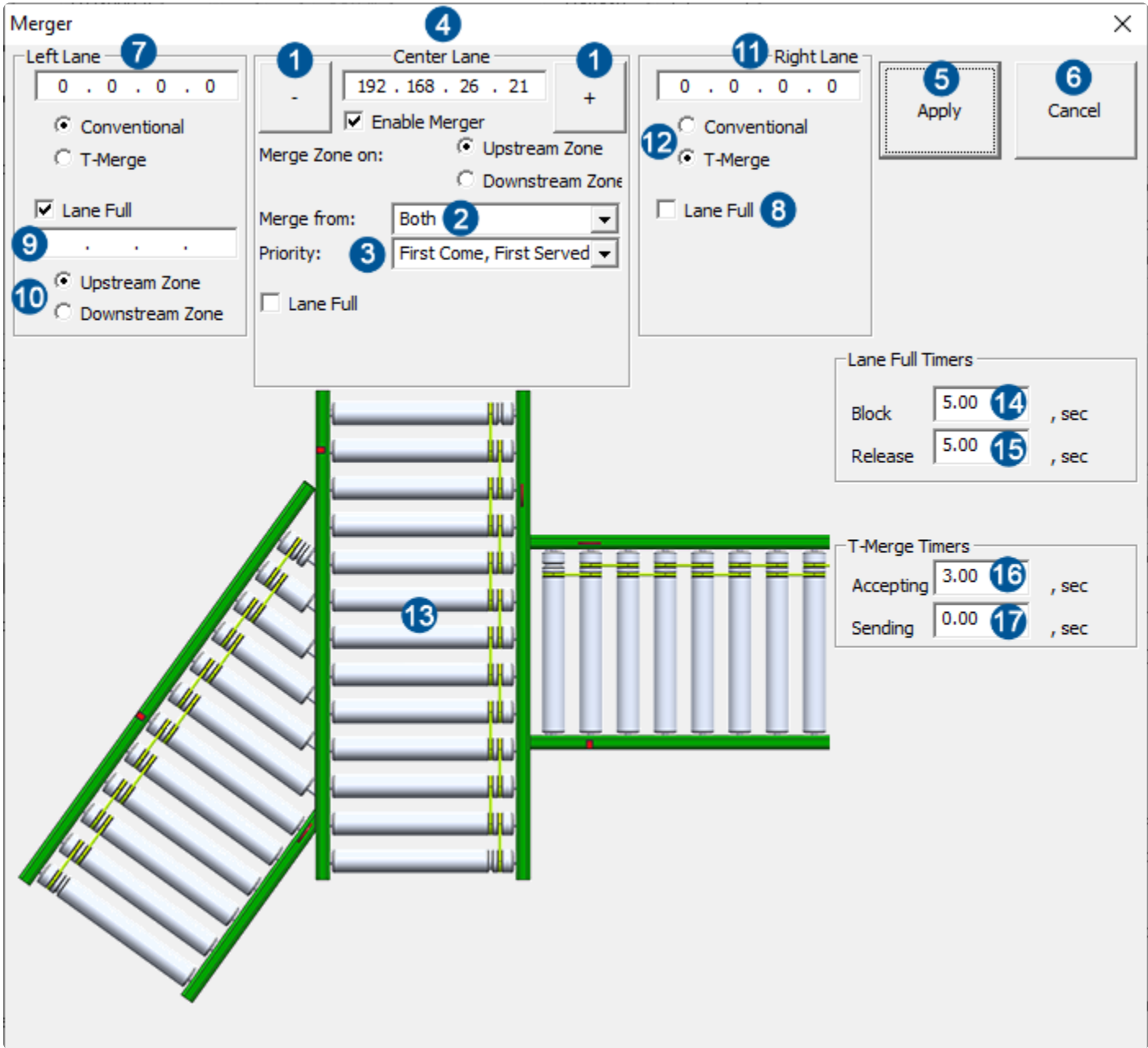
Item	Description
1	Navigation Buttons for Node selection

2	Drop down selector for *Merge Type* – Left, Right, or Both
3	Drop down for *Merge Priority*
4	Display of IP Address for the ConveyLinx-Ai Family Controllers selected that will contain the *Merge Zone*
5	User button to click to apply the settings to the affected modules
6	User button to cancel any changes made and close the screen
7	Field to enter the IP Address of the *Left Line's* most downstream ConveyLinx-Ai Family Controllers
8	Checkbox to enable *Merge Line Full* functions
9	IP Address of the ConveyLinx-Ai Family Controllers that contains the zone being defined as the *Monitor Zone* for the *Merge Line Full* function
10	Radio buttons to select whether the upstream or downstream zone of the ConveyLinx-Ai Family Controllers identified in Item 9 will be used as the *Monitor Zone*
11	Field to enter the IP Address of the *Right Line's* most downstream ConveyLinx-Ai Family Controllers
12	User checkbox to select *T-Merge* type
13	Graphical representation of the merge. Please note this will update and change as settings and configuration is modified
14	*Merge Line Full Block Timer* preset value
15	*Merge Line Full Clear Timer* preset value
16	*T-Merge Accept Timer* preset value
17	*T-Merge Send Timer* preset value

9.11. Conventional Spur Merge Example

In this example we are going to configure the *Left Line* to be standard spur angle merge onto the *Center Line* and we will set the *Merge Priority* to be *First Come First Served*

✱ Images show ConveyLinx-ERSC modules but functions are applicable to all ConveyLinx Family modules
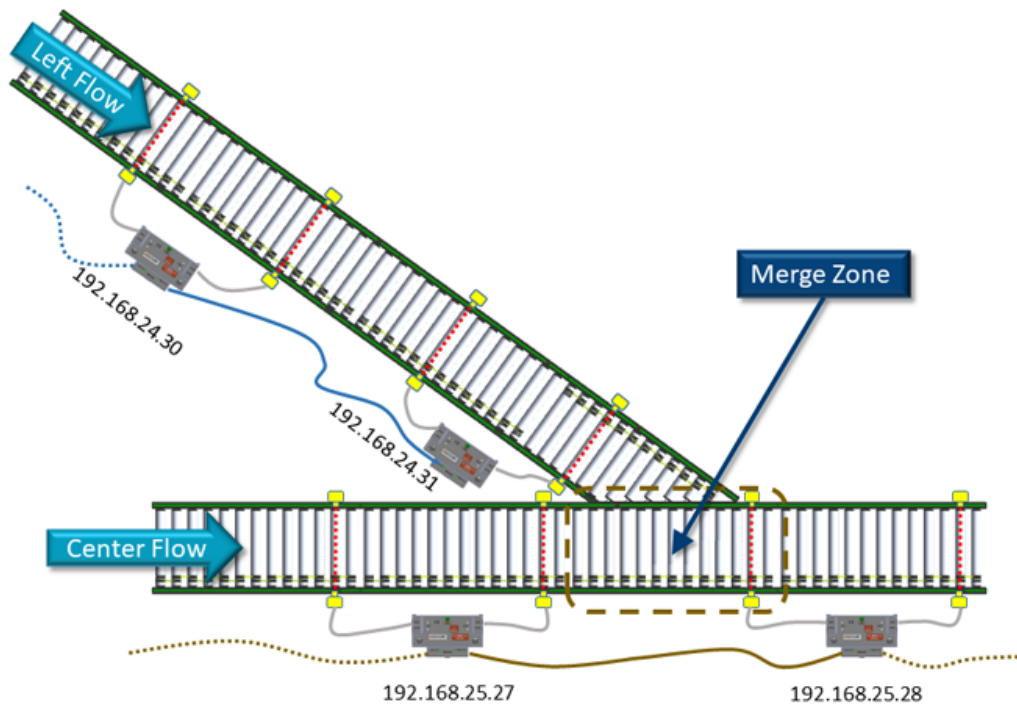
The *Merge Zone* is the upstream zone of the module at 192.168.25.28 and the most downstream ConveyLinx-Ai Family Controllers for the *Left Line* is at 192.168.24.31. Here is the *ConveyMerge Configuration Screen* with these values entered
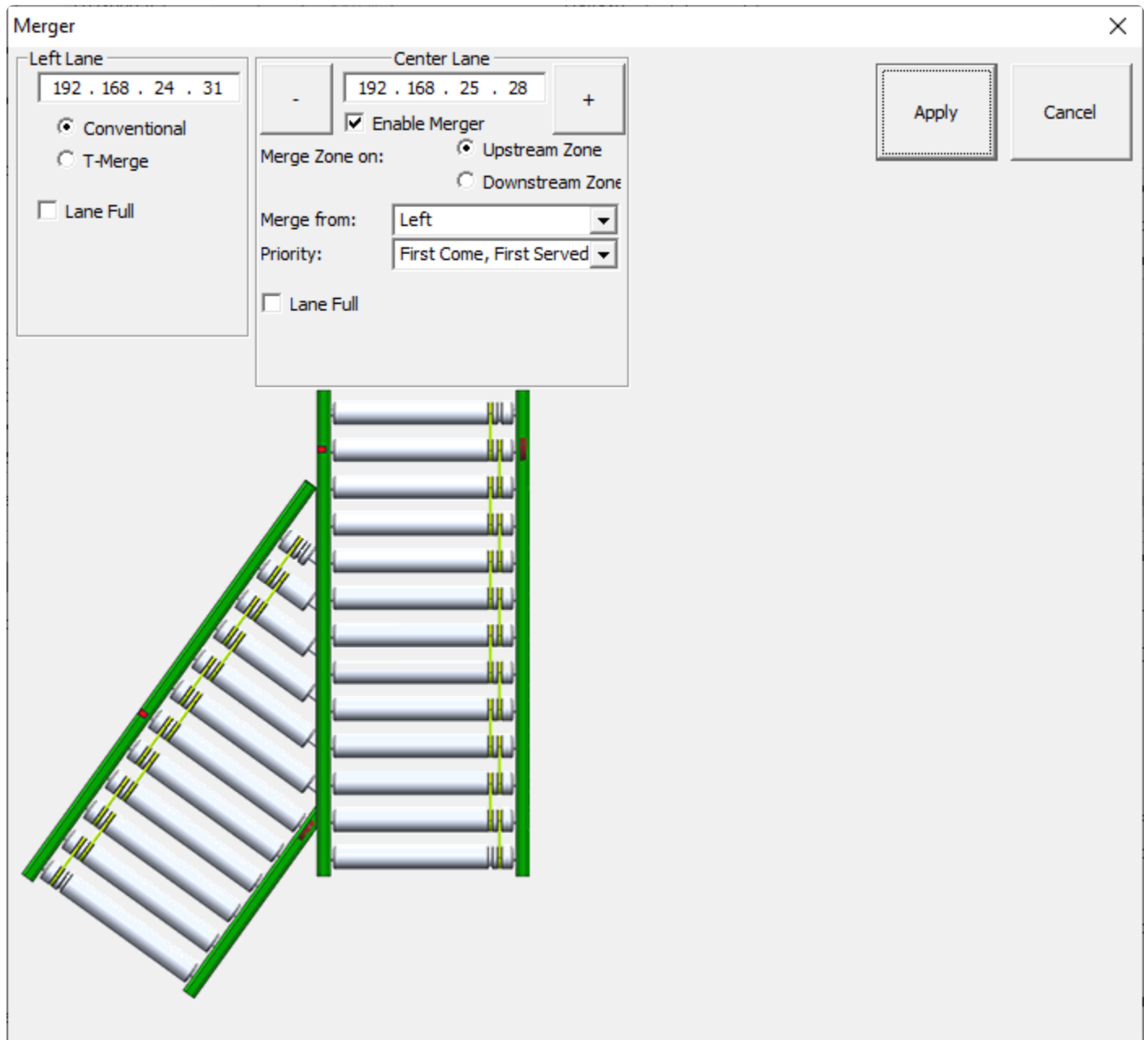
The last step is to click the *Apply* button to send the changes to the affected ConveyLinx-Ai Family Controllerss

✱ Please note that after clicking the "Apply" button, if the configuration has changed, the affected modules will reboot. It will take a few seconds for the reboot to finish and some or all of the affected zones may briefly run their motors

9.12. T-Merge Example

In this example we are going to configure the *Right Line* to be a *T-Merge* onto the *Center Line* and we will set the *Merge Priority* to be *First Come First Served*

✱ Images show ConveyLinx-ERSC modules but functions are applicable to all ConveyLinx Family modules
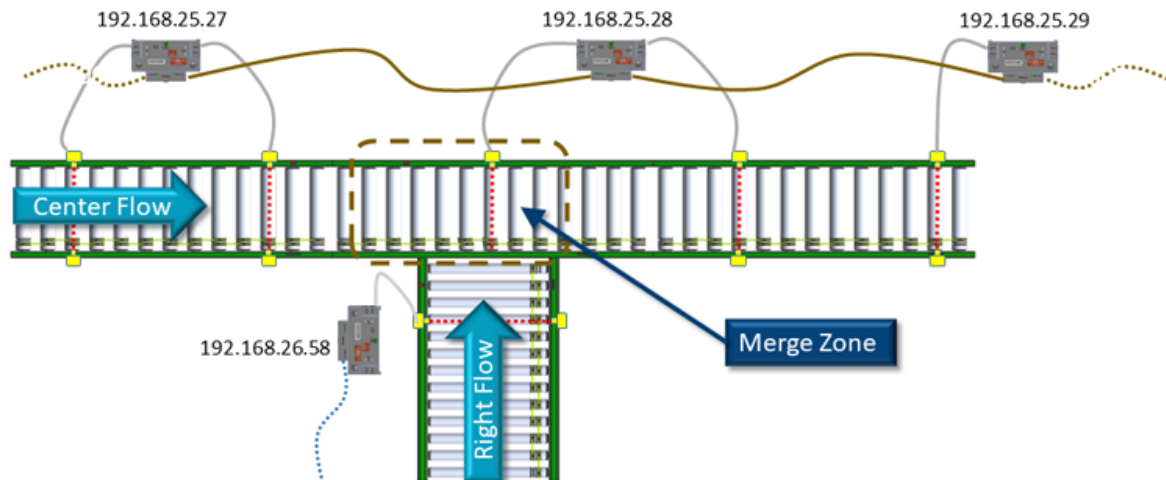
The *Merge Zone* is the upstream zone of the module at 192.168.25.28 and the most downstream ConveyLinx-Ai Family Controllers for the *Right Line* is at 192.168.26.58. Here is the *ConveyMerge Configuration Screen* with these values entered

Merger   ✕

Center Lane

[-]   192 . 168 . 25 . 28   [+]

☑ Enable Merger

Merge Zone on: ◉ Upstream Zone
                      ○ Downstream Zone

Merge from: Right ▼

Priority: First Come, First Served ▼

☐ Lane Full

Right Lane

192 . 168 . 26 . 58

○ Conventional
◉ T-Merge

☐ Lane Full

[Apply]   [Cancel]

T-Merge Timers

Accepting 3.00   , sec

Sending   0.00   , sec

9.13. Merge Line Full Example

In this example we have the *Left Line* configured as standard spur angle merge and the *Merge Priority* to be *Center Line*

> ✱ Images show ConveyLinx-ERSC modules but functions are applicable to all ConveyLinx Family modules
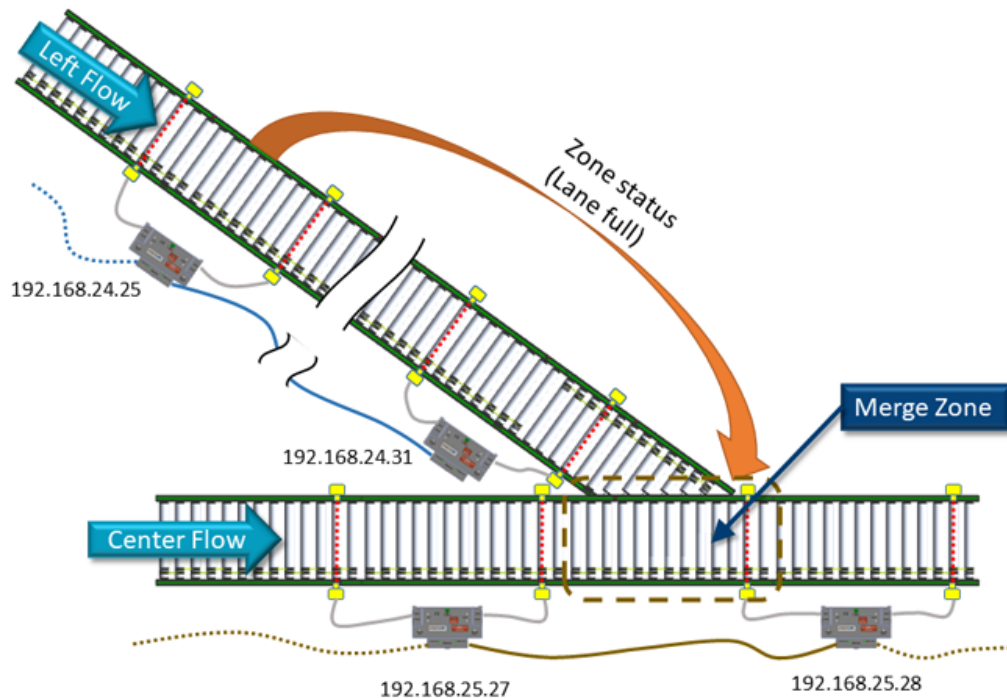
The *Merge Zone* is the upstream zone of ConveyLinx-Ai Family Controllers 192.168.25.28 and the *Left Line's* most downstream node is 192.168.24.31 and we want this line to have the *Merge Line Full* function enabled. On the *Left Line*, we would like for the upstream zone of ConveyLinx-Ai Family Controllers 192.168.24.25 to be the Merge Line Full function's *Monitor Zone*. We would also like that once the *Monitor Zone* has been accumulated (or blocked) for 5 seconds, we want the *Left Line* to release for 5 seconds and then return the priority to the Center Line. Here is the *ConveyMerge Configuration Screen* with the values entered for this example:

Merger ✕

Left Lane

192 . 168 . 24 . 31

◉ Conventional
○ T-Merge

☑ Lane Full

192 . 168 . 24 . 25

◉ Upstream Zone
○ Downstream Zone

Center Lane

`-` 192 . 168 . 25 . 28 `+`

☑ Enable Merger

Merge Zone on: ◉ Upstream Zone
○ Downstream Zone

Merge from: Left ▼
Priority: Center Line Priority ▼

☐ Lane Full

Apply   Cancel

Lane Full Timers

Block 5.00 , sec

Release 5.00 , sec

10. IOX Interface Module

In the section [Electrical Connections for Sensor Port Aux I/O](#), it is shown how to use off the shelf commercially available M8 cable accessories to gain access to the Aux I/O pin on the sensor ports. Pulseroller offers a Breakout/Splitter Module Accessory that offers the same cable splitter functionality as the commercially available M8 units along with breaking out some of the signals to convenient wire terminal access for flying lead devices.

The *IOX-2 Interface Module* provides a convenient plug and play means to separate the zone sensor and Aux I/O signals on the Module's sensor port. The *IOX-2* utilizes M8 style connection headers so you can connect your M8 style zone sensor as well as M8 style cable (or additional sensor) for access to the Aux I/O Pin 2 signal. The *IOX-2* also contains wire terminal access for the Aux I/O Pin 2 signals for complete flexibility.

Mounting Dimensions
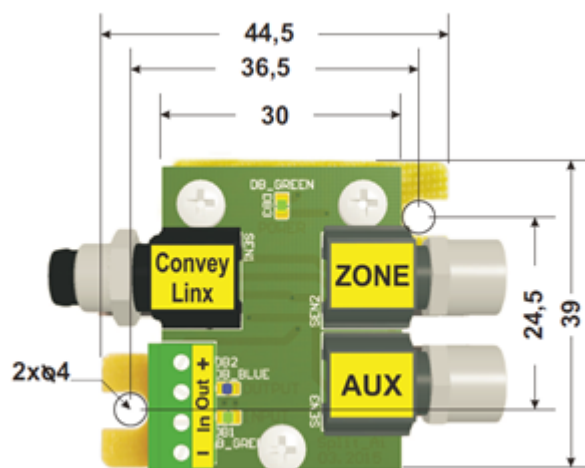
Wiring Diagrams
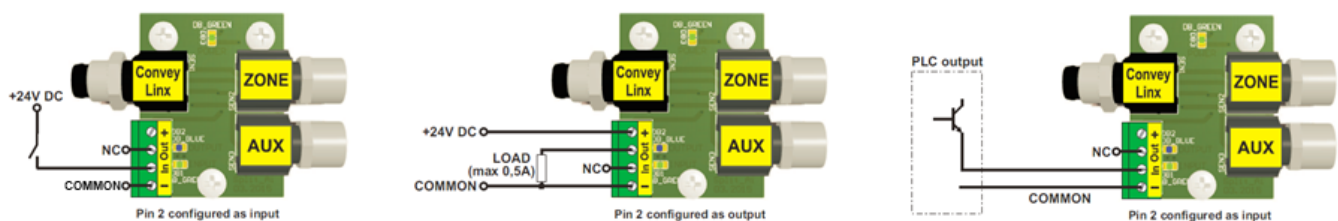
Electrical Schematic

Electrical Schematic of IOX-2 Module
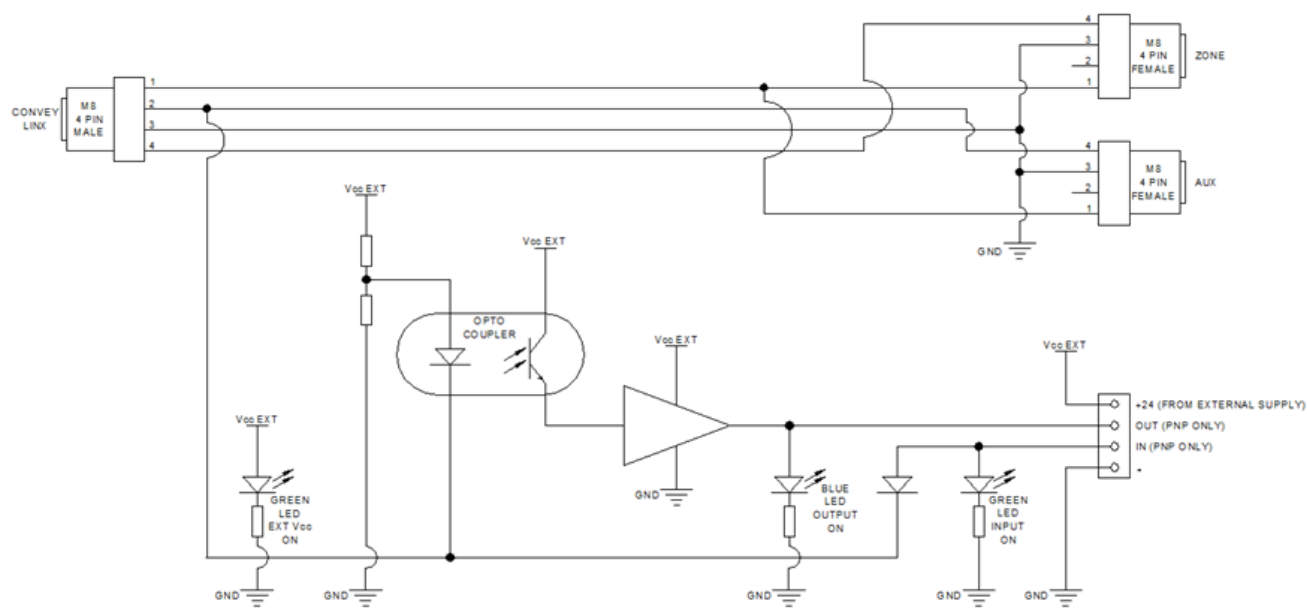
10.1. Wake up and/or Lane Full Interface
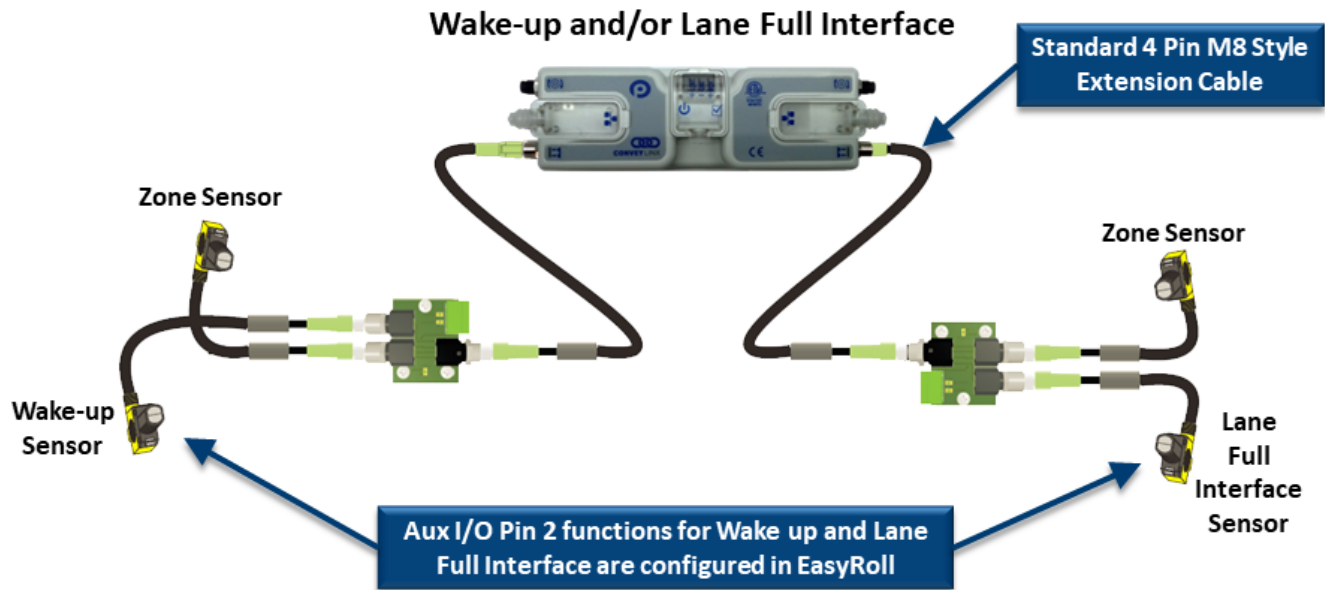
10.2. Wake up/Lane Full with Wired Terminals

Wake-up and/or Lane Full Interface using Wired Terminals

Zone Sensor

Zone Sensor

Sensor on Wired Terminal Block must provide PNP signal to IN terminal

Wake-up Sensor

Lane Full Interface Sensor

Sensor connected to Wired Terminal Block require external 24V to be connected to "+" terminal.
"+" terminal DOES NOT provide 24V from Ai2 Sensor Port

Aux I/O Pin 2 functions for Wake up/Lane Full Interface are configured in EasyRoll
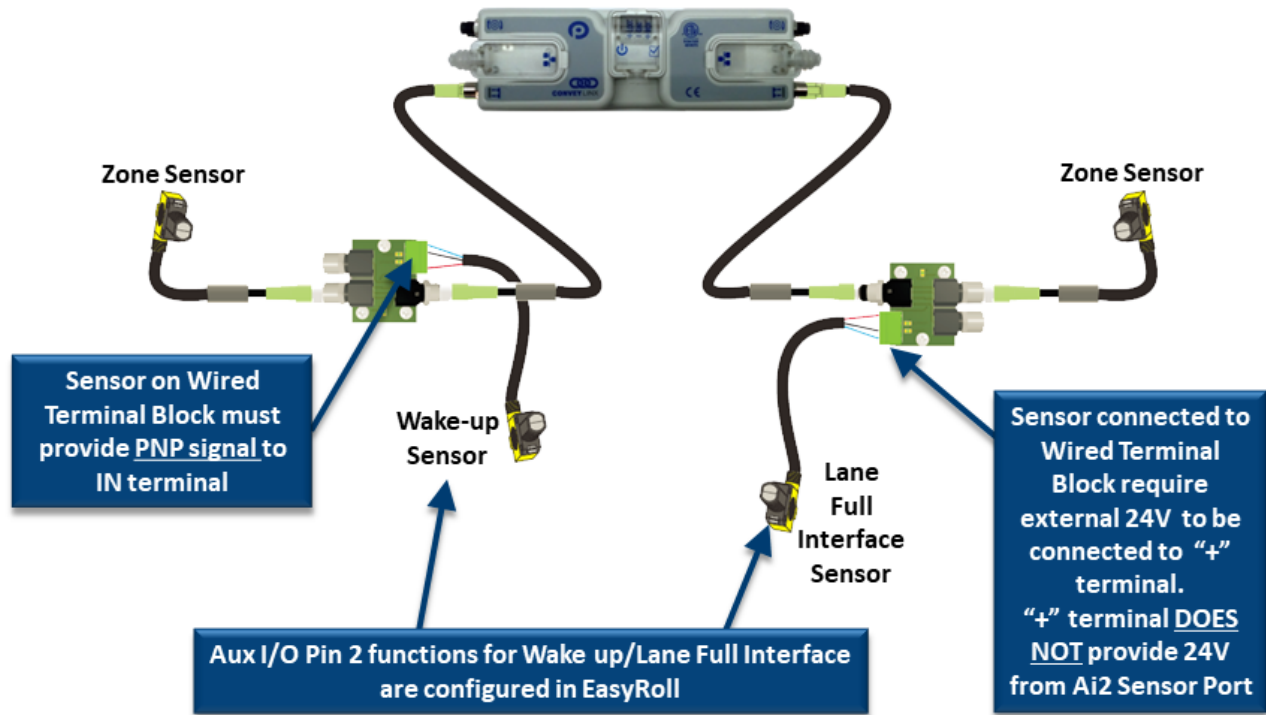
10.3. Wake up/Lane Full with Discrete Signals

Wake-up/Lane Full Interface/Accumulate using Discrete Signals to Wired Terminals
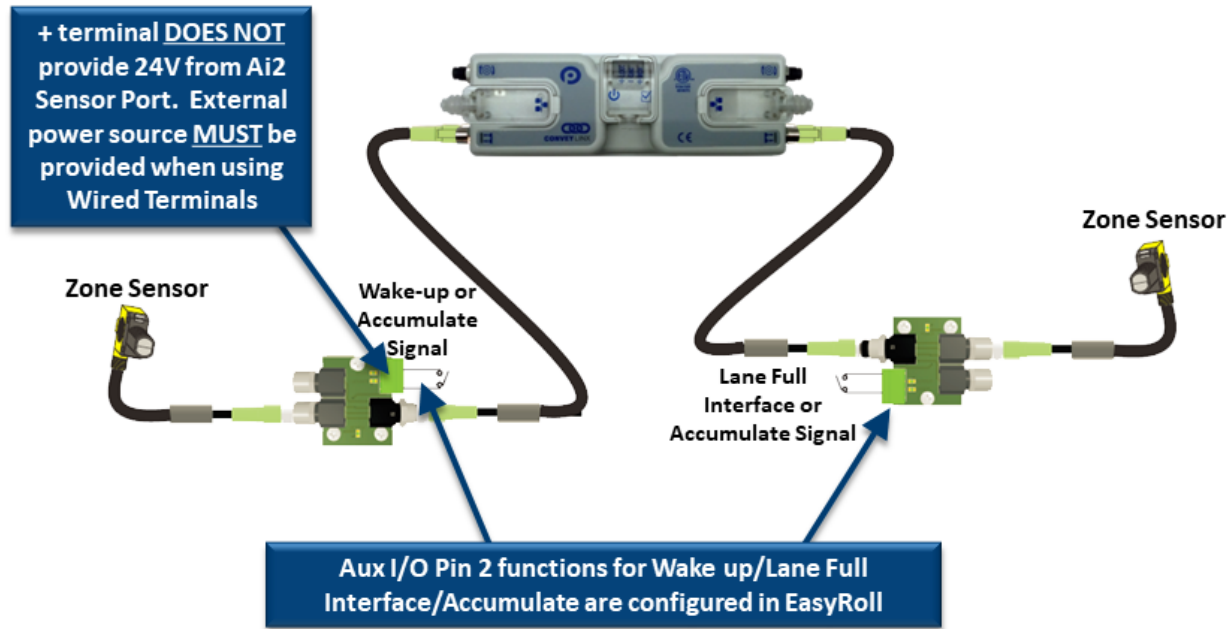
10.4. Wake up/Product on Zone Handshake Interlock

Most Upstream Zone Wake Up with Product On Zone Handshake Output

Zone Sensor Most Upstream Zone

Wake Up Sensor

Product On Zone Output

Zone Sensor

Right Aux I/O Pin 2 configured as Product On Zone Output assigned to the Upstream Zone of the module. When Zone Sensor for Most Upstream Zone is blocked, Product On Zone output is energized
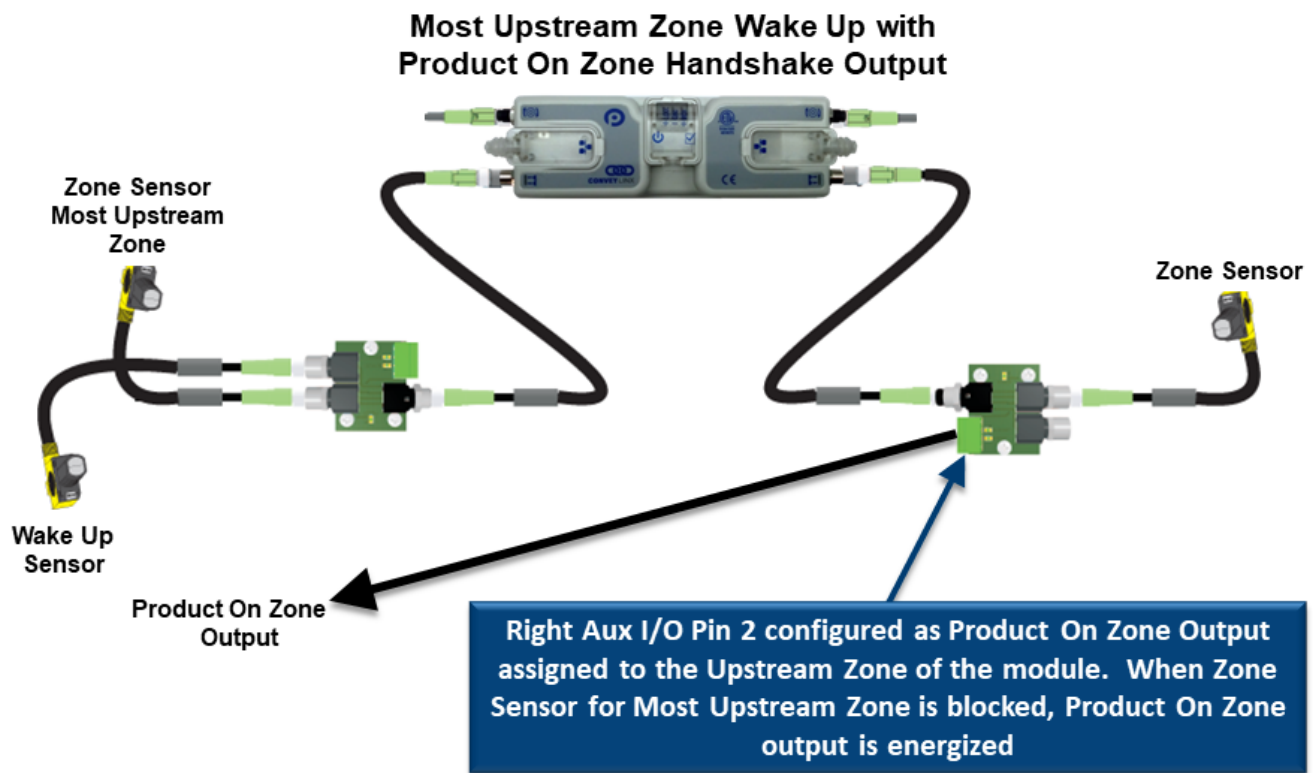
10.5. Downstream/Product on Zone Handshake Interlock

Most Downstream Zone Lane Full Interface with Product On Zone Handshake Output

Zone Sensor

Zone Sensor Most Downstream Zone

Product On Zone Output

Lane Full Sensor

Left Aux I/O Pin 2 configured as *Product On Zone* Output and is assigned to the Downstream Zone of the module. When Zone Sensor for Most Downstream Zone is blocked, *Product On Zone* output is energized
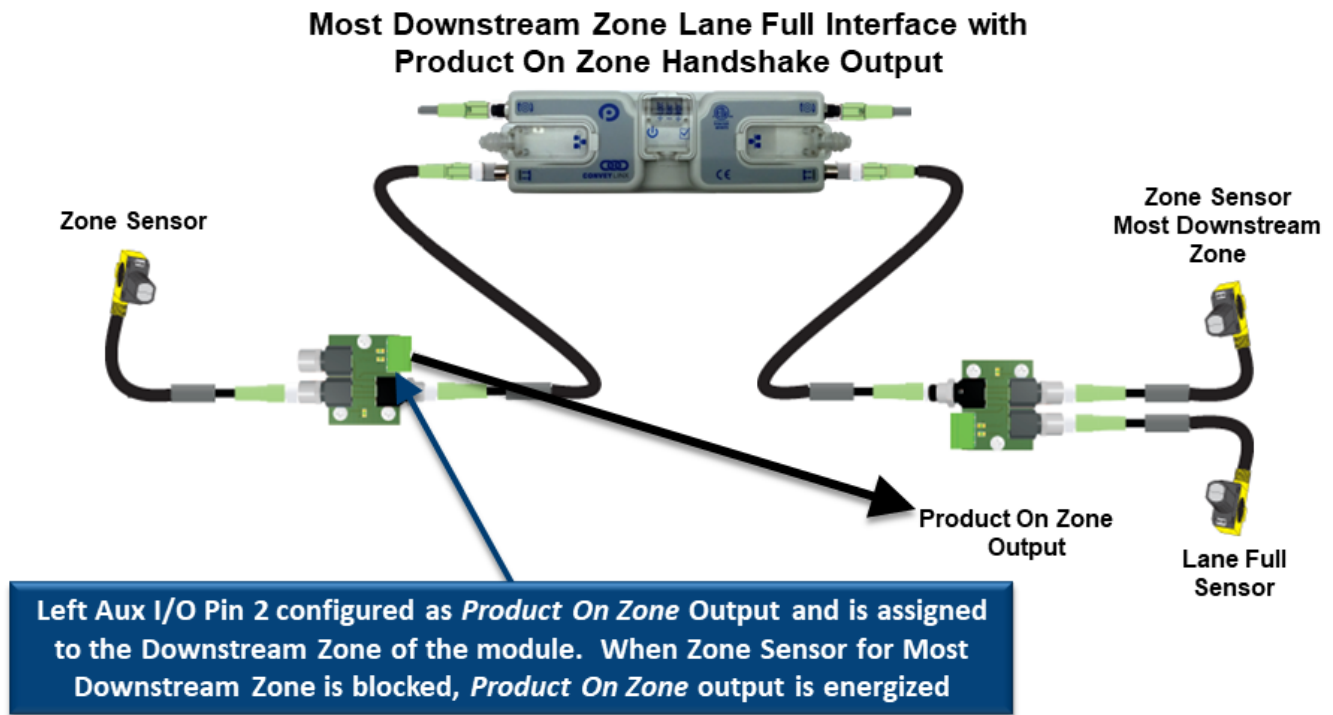
10.6. Pin 2 Output on Aux I/O M8

Using AUX M8 Port to Connect Sensor Port AUX I/O Pin when Aux I/O is configured as an OUTPUT

Sensor Port Aux I/O Pin 2 configured as OUTPUT

Zone Sensor

AUX Port connection on IOX-2

+ LOAD -

Pin 4

Pin 1
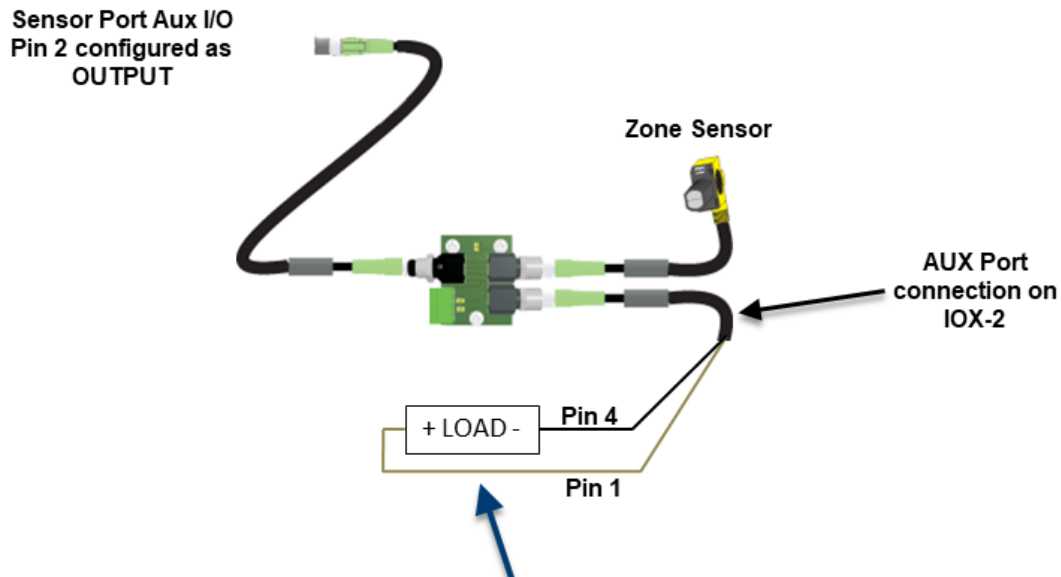
When the AUX M8 Port on the IOX-2 is used to connect the Sensor Port's AUX I/O Pin 2 signal as an output, the output signal is NPN and appears on PIN 4 of the AUX M8 Port

10.7. Pin 2 Output on Wired Terminals

Using Wired Terminals to Connect Sensor Port AUX I/O Pin when Aux I/O is configured as an OUTPUT
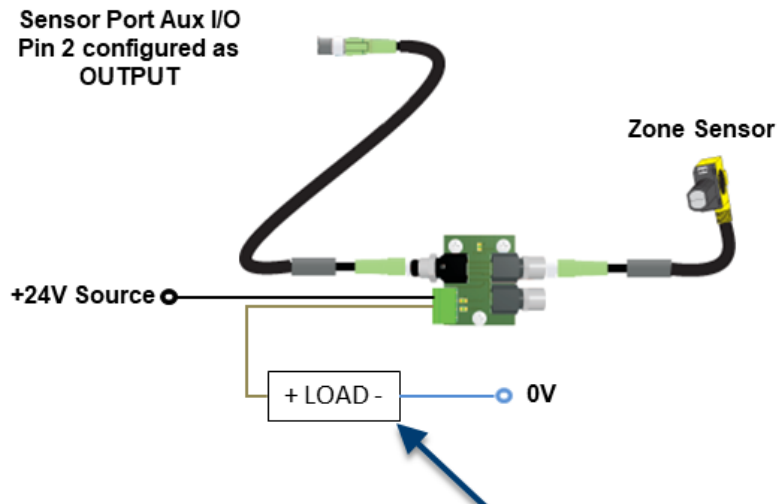
When the Wired Terminals on the IOX-2 are used to connect the Sensor Port's AUX I/O Pin 2 signal as an output, 24V source must be connected to the "+" terminal and a PNP signal is provided on the "OUT" terminal.

11. PLC Developers Guide

This section defines the data connectivity available between ConveyLinx-Ai Family Controllerss and remote control devices such as *PLCs* and *PCs*

Click Links to Learn More

Network Architecture
Understanding Assemblies
ZPA Mode Control
PLC I/O Mode Control
ConveyLogix Interface
Assemblies with Reset Protection
Motor Port as Digital I/O

11.1. Network Architecture

Each *ConveyLinx-Ai Family Controllers* communicates to its adjacent modules and to any connected PC or PLC via Ethernet physical media. ConveyLinx-Ai Family Controllerss recognize (3) TCP/IP based protocols: *Modbus TCP, Profinet I/O, and Ethernet I/P*. Modbus TCP is the "native" protocol for communications between *ConveyLinx-Ai Family Controllerss* and the *EasyRoll* PC software. When ConveyLinx-Ai Family Controllerss are used even for basic ZPA control with no external connections to a PC or PLC, they utilize Modbus TCP for inter-module communication. *Ethernet I/P* is recognized by ConveyLinx-Ai Family Controllerss and any given ConveyLinx-Ai Family Controllers can be attached to an Ethernet I/P capable PLC (Allen-Bradley ControlLogix or CompactLogix platforms) and be recognized as a *Generic Ethernet Module* or explicitly as an ConveyLinx-Ai Family Controllers by using a Pulseroller provided *EDS file*. *Profinet I/O* is recognized by ConveyLinx-Ai Family Controllerss and can be attached to any Profinet I/O capable PLC (applicable Siemens platforms) using a Pulseroller provided *GSDXML file*.

> ✳ All protocols access the same internal data locations on a given ConveyLinx-Ai Family Controllers.

Each ConveyLinx-Ai Family Controllers'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 ConveyLinx-Ai Family Controllers is configured (MDR type, speed, direction, etc.) for its local controls. Other registers are used for inter-module communications for conveyor operation. For example, when an upstream ConveyLinx-Ai Family Controllers has a Carton ready to discharge to its neighboring downstream ConveyLinx-Ai Family Controllers, the upstream ConveyLinx-Ai Family Controllers will write a specific value to a specific address in the downstream ConveyLinx-Ai Family Controllers's internal registers. The downstream ConveyLinx-Ai Family Controllers's on board logic monitors these registers and knows that a specific value means that an upstream Carton is coming and to engage the proper control logic to convey the Carton.
Because ConveyLinx utilizes an open architecture (Modbus TCP) for inter-module communications; with proper definition and expected usage of certain register address locations, external control devices (PC's and PLC's) can easily interact with ConveyLinx-Ai Family Controllerss to monitor and control various points along the conveyor path.

11.2. Understanding Assemblies

The Modbus Holding Registers described above will be defined as *Module Register Addresses* and are used for the function of the ConveyLinx-Ai Family Controllers regardless of if any remote PLC or PC is connected. There are 512 *Module Register Addresses* that are used for the operation of the ConveyLinx-Ai Family Controllers and these can be thought of as "physical" module address locations. However, many of these are not applicable, required, or should even be made available to a remote PLC.

An *Assembly* is a grouping of some subset of these 512 Module registers based upon the relevance of the data. For example, the *ZPA Input Assembly* groups together 21 Module registers out of the 512 that are relevant for ZPA Inputs. This relevant data from within the Module's 512 registers are not necessarily in consecutive address locations and are scattered throughout the 512 addresses. The *Assembly* groups them together so they can be read efficiently all at once.
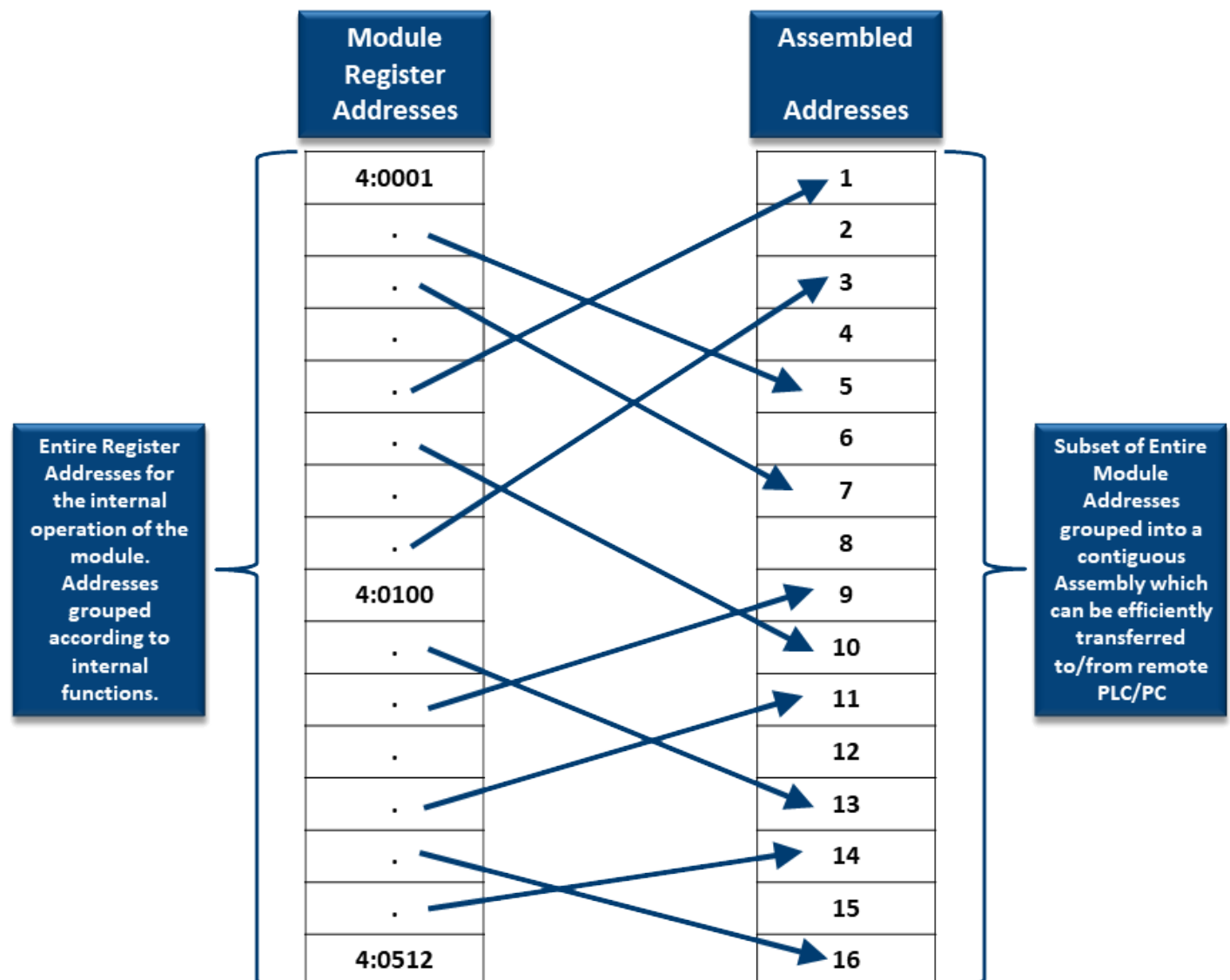
11.2.1. Modbus Assembly Instance Structure

Each ConveyLinx-Ai Family Controllers 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 ConveyLinx-Ai Family Controllers's memory structure contains a fixed array of internal data locations that are constructed as *Modbus Holding Registers*. Each ConveyLinx-Ai Family Controllers 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 ConveyLinx-Ai Family Controllers'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 such as MDR speed, direction, type, etc.

> ✱ 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.

Supported Modbus TCP Service Codes

Any Modbus TCP capable PC or PLC can connect to any ERSC visible on its network and access Input and Output Holding Register Assemblies. The ERSC 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)

Properly Addressing Assembly Registers

In the ConveyLinx-Ai Family Controllers, the Assembly Registers are "virtual" in that they are not "physical" Module Register Addresses and as such are not directly addressable. The ConveyLinx-Ai Family Controllers firmware intercepts the request and builds the assembly data in order to transmit it to the requesting device. Because of this, you must always address an Assembly by using the first address in the assembly in order for the ConveyLinx-Ai Family

Controllers to return any data.

> ❗ 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 ConveyLinx-Ai Family Controllers to return an error

For example, for ZPA Mode Assembly Inputs, if you only need to read register 4:1504; your I/O connection set-up in your PLC must use the starting address of 4:1500 and a sufficient length of registers to read (in this example at least 5 registers) in order to get to the desired register. If you set up your PLC to start reading at 4:1504, the ConveyLinx-Ai Family Controllers will return an error

11.2.2. Ethernet I/P Assembly Instance Structure

When an ConveyLinx-Ai Family Controllers is attached to an external Ethernet I/P controller (Logix 5000 based PLC), it is done so as a Generic Ethernet I/O device or by installing EDS file(s) provided by Pulseroller.

> ✱ For purposes of working with assemblies, we will be describing them in terms of them connected as Generic Ethernet I/O devices. For details on how to use EDS files and Add On Instructions (AOIs), you can go to Connecting to Rockwell PLCs section

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:

:O.Data[Index].Bit
:I.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
• "[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 Assembly Instances group the appropriate *Module Register Address* registers into contiguous Input and Output array images that fit into the Allen-Bradley Logix 5000 controller tags

Learn more about connecting to Rockwell PLCs

11.2.3. Profinet IO Assembly Instance Structrure

Prior to connecting any modules to a Profinet IO capable (Siemens) 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.

[Learn more about connecting to Siemens PLCs](#)

11.2.4. Assembly Register Chart Legend

All of the Assemblies are organized in charts that show:

- The Name of the register and its Module Address
- The Assembled Address for Modbus TCP, Ethernet IP, and Profinet
- Description of the register's usage

The Assembled Addresses are shown in specific notation familiar to the specific protocol per the following examples:

M: 4:1500	Indicates Modbus TCP addressing notation for Holding Registers
E: I.Data [0]	Indicates Ethernet IP addressing notation for Rockwell RSLogix Tags
P: Byte 0 (Hi) Byte 1 (Lo)	Indicates Profinet I/O addressing notation for Profinet IO memory location offset

11.3. ZPA Mode Control

When an ConveyLinx-Ai Family Controllers is in its default *ZPA* mode, an external networked PLC or PC controller can connect to the ConveyLinx-Ai Family Controllers and perform the following:

- Instruct either or both the upstream and downstream zone to accumulate the next Carton that arrives
- Receive indication that a new Carton has arrived at either zone
- Receive indication that a Carton has departed from either zone
- Read tracking data associated with Carton at accumulated zone
- Update tracking data associated with Carton at accumulated zone
- Instruct accumulated zone to release Carton and accumulate on next Carton arrival
- Change the MDR speed for either zone
- Remove accumulation control and return zone to normal operation
- Read fault and error status of either zone or motor
- Optionally take over control of either or both Control Ports

Notes about Assemblies in ZPA Mode

When an ConveyLinx-Ai Family Controllers is in *ZPA* Mode, its primary task is to operate its local conveyor zones and respond to its immediate upstream and downstream conditions. External PLC controller interaction with an ConveyLinx-Ai Family Controllers in *ZPA* mode is intended to be for decision point monitoring and general status data gathering. Upstream and Downstream zones work exactly the same for zone control, only the register addresses are different depending on which zone (or both) that need to be controlled

> ✲ In general, when utilizing *ZPA* Mode registers; "upstream" and "downstream" registers are logically determined by conveyor flow after the system has been *Auto-Configured* and will not necessarily be associated with the module's physical "left" or "right" side's connections. For motor and port specific I/O items, register's description will explicitly indicate "left" or "right".

> ✲ For modules that are auto-configured as single zone, regardless of whether the left or right side is physically used as the single zone; external controller must use the "Upstream" control registers to interface with the single zone

11.3.1. PLC Inputs for ZPA Mode

This Assembly contains the output from the ERSC that is provided as input to the PLC. You can select which area of the assembly to view below.

Local Zone Status
Arrival/Departure Counts
Module Status
Tracking and Release Counts
Forward and Reverse Tracking
Port Inputs and ConveyStop Status

11.3.1.1. Local Zone Status

Register Name / Module Address	Assembled Address for PLC	Description
Local Status Upstream Zone Forward Direction 4:0116	M: 4:1500 (Lo Byte) E: I.Data [0] (Lo Byte) P: Byte 1	Unsigned Value of Byte:
Local Status Upstream Zone Reverse Direction 4:0116	M: 4:1500 (Hi Byte) E: I.Data [0] (Hi Byte) P: Byte 0	0×01 = Zone sensor clear and motor stopped 0×02 = Zone sensor clear, motor running, accepting from upstream zone
Local Status Downstream Zone Forward Direction 4:0196	M: 4:1501 (Lo Byte) E: I.Data [1] (Lo Byte) P: Byte 3	0×04 = Zone sensor blocked, motor running, discharging to downstream zone 0×05 = Zone sensor blocked and motor stopped
Local Status Downstream Zone Reverse Direction 4:0196	M: 4:1501 (Hi Byte) E: I.Data [1] (Hi Byte) P: Byte 2	0×06 = Busy (state during ConveyStop active mode)

The values 0xXX01 thru 0xXX06 are shown because these are the possible logical values used for inter-module communication. External networked devices (PLC or PC) monitoring these registers may; depending on their scan rate, not actually see each of these values change in sequence as a Carton is conveyed from zone to zone, even though the inter-module communications and ZPA is functioning normally.

> ✱ IMPORTANT NOTE: Status register values utilize both the HIGH BYTE and the LOW BYTE of the 16-Bit integer value. The HIGH BYTE is used for zone status for reversing conveyor applications and MAY CONTAIN DATA. PLC/PC programmers working with single direction conveyor applications who also are working with the entire 16-bit register data MUST MASK THE HIGH BYTE or otherwise ignore the high byte in processing status data from these registers.

> ✱ For PLC/PC programming purposes, you can only depend on seeing values 0xXX01 and 0xXX05 in program logic for determining zone status. The values 0xXX02 and 0xXX04 may not always be visible to PLC/PC from inter-module

communication depending upon speed of the conveyor, length of the zone, and/or location of the zone sensors.

11.3.1.2. Arrival/Departure Counts

Register Name / Module Address	Assembled Address for PLC	Description
Arrival Count Local Upstream Zone 4:0106	M: 4:1502 E: I.Data [2] P: Byte 4 (Hi) Byte 5 (Lo)	Unsigned Integer Value: • Increments by 1 each time a Carton **arrives** in the zone • Value rolls over from 65,535 back to 0
Arrival Count Local Downstream Zone 4:0186	M: 4:1504 E: I.Data [4] P: Byte 8 (Hi) Byte 9 (Lo)	
Departure Count Local Upstream Zone 4:0107	M: 4:1503 E: I.Data [3] P: Byte 6 (Hi) Byte 7 (Lo)	Unsigned Integer Value: • Increments by 1 each time a Carton **departs** the zone • Value rolls over from 65,535 back to 0
Departure Count Local Downstream Zone 4:0187	M: 4:1505 E: I.Data [5] P: Byte 10 (Hi) Byte 11 (Lo)	

11.3.1.3. Module Status

Register Name / Module Address	Assembled Address for PLC	Description
Module Status Word 1 4:0088	M: 4:1506 E: I.Data [6] P: Byte 12 (Hi) Byte 13 (Lo)	Bitwise Value – Read Only bit 0 = Module Reset Flag – 1 when module resets, 0 when PLC connected bit 1 = Reserved bit 2 = Over-Voltage – Input Power > 30V bit 3 = Left Motor Error – bits 7 thru 15 indicate specific error bit 4 = Ethernet Connections NOT OK bit 5 = Upstream Jam Error bit 6 = Left Sensor Error bit 7 = Low Voltage Error – Input Power < 18V bit 8 = Left Motor Over-heated – Calculated temperature over 105°C bit 9 = Left Motor Over-current – Over limit for selected MDR bit 10 = Left Motor Short Circuit bit 11 = Left Motor Not Connected bit 12 = Left Motor Overload –Motor has been stalled for more than 20 seconds bit 13 = Left Motor Stalled – MDR slower than 10% of selected speed bit 14 = Left Motor Hall Sensor Error bit 15 = Left Motor Not Used
Module Status Word 2 4:0089	M: 4:1507 E: I.Data [7] P: Byte 14 (Hi) Byte 15 (Lo)	Bitwise Value – Read Only bit 0 = Reserved bit 1 = Reserved bit 2 = Over-Voltage – Input Power > 30V bit 3 = Right Motor Error – bits 7 thru 15 indicate specific error bit 4 = Reserved bit 5 = Downstream Jam Error bit 6 = Right Sensor Error bit 7 = Low Voltage Error – Input Power < 18V bit 8 = Right Motor Over-heated – Calculated

		temperature over 105°C bit 9 = Right Motor Over-current – Over limit for selected MDR bit 10 = Right Motor Short Circuit bit 11 = Right Motor Not Connected bit 12 = Right Motor Overload –Motor has been stalled for more than 20 seconds bit 13 = Right Motor Stalled – MDR slower than 10% of selected speed bit 14 = Right Motor Hall Sensor Error bit 15 = Right Motor Not Used

11.3.1.4. Tracking and Release Counts

Register Name / Module Address	Assembled Address for PLC	Description
Current Upstream Zone Tracking Word 1 4:0119	M: 4:1508 E: I.Data [8] P: Byte 16 (Hi) Byte 17 (Lo)	When Carton is accumulated in UPSTREAM zone: Value = Tracking data word #1 (16-bit integer) for the Carton currently accumulated and stopped in the module's Upstream zone
Current Upstream Zone Tracking Word 2 4:0120	M: 4:1509 E: I.Data [9] P: Byte 18 (Hi) Byte 19 (Lo)	When Carton is accumulated in UPSTREAM zone: Value = Tracking data word #2 (16-bit integer) for the Carton currently accumulated and stopped in the module's Upstream zone
Current Downstream Zone Tracking Word 1 4:0199	M: 4:1510 E: I.Data [10] P: Byte 20 (Hi) Byte 21 (Lo)	When Carton is accumulated in DOWNSTREAM zone: Value = Tracking data word #1 (16-bit integer) for the Carton currently accumulated and stopped in the module's Downstream zone
Current Downstream Zone Tracking Word 2 4:0200	M: 4:1511 E: I.Data [11] P: Byte 22 (Hi) Byte 23 (Lo)	When Carton is accumulated in DOWNSTREAM zone: Value = Tracking data word #2 (16-bit integer) for the Carton currently accumulated and stopped in the module's Downstream zone
Current Release Count for Upstream Zone 4:0105	M: 4:1512 E: I.Data [12] P: Byte 24 (Hi) Byte 25 (Lo)	Copy of the current value in the *Release Upstream* output register which can be used by PLC logic to confirm release count prior to writing new data to the *Release Upstream* output register
Current Release Count for Downstream Zone 4:0185	M: 4:1513 E: I.Data [13] P: Byte 26 (Hi) Byte 27 (Lo)	Copy of the current value in the *Release Downstream* output register which can be used by PLC logic to confirm release count prior to writing new data to the *Release Downstream* output register

11.3.1.5. Forward and Reverse Tracking

Register Name / Module Address	Assembled Address for PLC	Description
Get Tracking Forward Direction Word 1 4:0201	M: 4:1514 E: I.Data [14] P: Byte 28 (Hi) Byte 29 (Lo)	When ERSC is discharging to Non-ConveyLinx controlled conveyor: Value = Tracking data word #1 (16-bit integer) for the Carton that has just discharged from the local downstream zone when local conveyor is operating in default or "forward" direction
Get Tracking Forward Direction Word 2 4:0202	M: 4:1515 E: I.Data [15] P: Byte 30 (Hi) Byte 31 (Lo)	When ERSC is discharging to Non-ConveyLinx controlled conveyor: Value = Tracking data word #2 (16-bit integer) for the Carton that has just discharged from the local downstream zone when local conveyor is operating in default or "forward" direction
Get Tracking Reverse Direction Word 1 4:0121	M: 4:1516 E: I.Data [16] P: Byte 32 (Hi) Byte 33 (Lo)	When ERSC is discharging to Non-ConveyLinx controlled conveyor: Value = Tracking data word #1 (16-bit integer) for the Carton that has just discharged from the local downstream zone when local conveyor is operating in opposite of default or "reverse" direction
Get Tracking Reverse Direction Word 2 4:0122	M: 4:1517 E: I.Data [17] P: Byte 34 (Hi) Byte 35 (Lo)	When ERSC is discharging to Non-ConveyLinx controlled conveyor: Value = Tracking data word #2 (16-bit integer) for the Carton that has just discharged from the local downstream zone when local conveyor is operating in opposite of default or "reverse" direction

11.3.1.6. Port Inputs and ConveyStop Status

Register Name / Module Address	Assembled Address for PLC	Description
Sensor Port Inputs 4:0035	M: 4:1518 E: I.Data [18] P: Byte 36 (Hi) Byte 37 (Lo)	Bitwise Value – Read Only: bit 0 = Left Sensor Port – Aux I/O (M8 Pin 2) bit 2 = Right Sensor Port – Aux I/O (M8 Pin 2) bit 4 = Left Sensor Port – Sensor Signal (M8 Pin4) bit 6 = Right Sensor Port – Sensor Signal (M8 Pin 4) bit 15 = 2 sec on / 2 sec off heatbeat All other bits reserved
Reserved	M: 4:1519 E: I.Data [19] P: Byte 38 (Hi) Byte 39 (Lo)	Used for Reset Protection connection See topic Assemblies with Reset Protection for details
ConveyStop Status 4:0019	M: 4:1520 E: I.Data [20] P: Byte 40 (Hi) Byte 41 (Lo)	Bitwise Value – Read Only: bit 0 – 4 Reserved bit 5 = Stop active on another module in Stop Group bit 6 = Stop active due to lost communication connection bit 7 = Stop active due to lost PLC connection bit 8 = Reserved bit 9 = Reserved bit 10 = Stop active due to Stop Command from PLC bit 11 – 15 Reserved

✱ The values for the signals on the Sensor ports (bit 0, bit 2, bit 4, and bit 6) are determined by the module's initial Auto-Configuration results and the bit values are set to a 1 when sensor is blocked and 0 when sensor is clear regardless of the sensor type used. For example, each Sensor port Pin 4 signal is for the sensor's output. If the sensor is light energized, N.O. then the electrical signal on Pin 4 is ON when the sensor is clear and OFF when the sensor is blocked. However, when reading bit 4 or bit 6 in this register, the bit will be a 1 when the sensor is blocked and 0 when the sensor is clear

11.3.2. PLC Outputs for ZPA Mode

This Assembly contains the output from the PLC that is provided as input to the ERSC. You can select which area of the assembly to view below.

Set Local Tracking
Accumulation Control
Speed Control
Release and Status
Induct Tracking Forward and Reverse
Set Outputs and Motor Clear
ConveyStop and Clear Jams
Direction and Accumulation Mode
ConveyMerge Interface

11.3.2.1. Set Local Tracking

Register Name / Module Address	Assembled Address for PLC	Description
Set Local Upstream Zone Tracking Word 1 4:0132	M: 4:1600 E: O.Data [0] P: Byte 0 (Hi) Byte 1 (Lo)	Write value for 16-bit integer tracking data word #1 for the Carton accumulated in the Upstream Zone
Set Local Upstream Zone Tracking Word 2 4:0133	M: 4:1601 E: O.Data [1] P: Byte 2 (Hi) Byte 3 (Lo)	Write value for 16-bit integer tracking data word #2 for the Carton accumulated in the Upstream Zone
Set Local Downstream Zone Tracking Word 1 4:0212	M: 4:1602 E: O.Data [2] P: Byte 4 (Hi) Byte 5 (Lo)	Write value for 16-bit integer tracking data word #1 for the Carton accumulated in the Downstream Zone
Set Local Downstream Zone Tracking Word 2 4:0213	M: 4:1603 E: O.Data [3] P: Byte 6 (Hi) Byte 7 (Lo)	Write value for 16-bit integer tracking data word #2 for the Carton accumulated in the Downstream Zone

Writing Tracking Details

Because the ConveyLinx-Ai Family Controllers is connected as I/O, the PLC inherently is always trying to update the Output image on (at least) RPI intervals. In order to prevent the PLC from inadvertently overwriting the "real" tracking data registers; the Assembly Output implementation utilizes the holding register locations shown and automatically updates the "real" tracking registers with this new data only upon release of the Carton from the zone. Included in this automatic functionality are two special reserved values that can be used for convenience:

- **Set both tracking registers shown to 0:** This will instruct the module to not modify the existing "real" tracking data and allow it to continue downstream "as-is" when the Carton is released.

- **Set both tracking registers shown to 0xFFFF:** This will instruct the module to clear the "real" tracking data and when the Carton is released, the "real" tracking data will be "0" in both registers.

! Both word 1 and word 2 of the tracking data in question must have 0xFFFF written to it in order to signal the module to clear the tracking data. If only one word has 0xFFFF written to it and the other does not, the value of 0xFFFF will be the new tracking data for that word.

11.3.2.2. Accumulation Control

Register Name / Module Address	Assembled Address for PLC	Description
Accumulation Control for Local Upstream Zone 4:0104	M: 4:1604 E: O.Data [4] P: Byte 8 (Hi) Byte 9 (Lo)	Bitwise Values: bit 00 = Set/Clear Accumulation Mode for Local Zone bit 08 = Accumulate adjacent upstream zone bit 09 = Set Arrival Confirmation for Local zone bit 10 = Jog zone in default direction bit 11 = Jog zone in opposite of default direction bit 12 = Wake up Local Zone bit 13 = Enable Maintenance Mode All other bits reserved
Accumulation Control for Local Downstream Zone 4:0184	M: 4:1605 E: O.Data [5] P: Byte 10 (Hi) Byte 11 (Lo)	

Notes for Accumulation Control Bits

Bit 8 – Accumulate Adjacent Upstream Zone

Setting this bit will cause the next upstream zone of the local module to accumulate. This next upstream zone can be either on the local module or the downstream zone of the adjacent upstream module.

Bit 9 – Set Arrival Confirmation for Downstream Zone

By default, for ZPA operation, ConveyLinx requires a confirmation from the downstream zone when a Carton is discharged. Without this confirmation, the releasing zone will detect a jam condition. This bit is used in applications where the Carton is removed from the conveyor (either manually or say by a PLC controlled external mechanism such as a pusher or diverter) and the PLC needs to "confirm" the removal of the Carton in order to satisfy the ZPA confirmation logic.

Bits 10 & 11 – Jog Controls

These bits can be used by the PLC to jog the local zone for specialized applications when local movement of the Carton on a zone is required. An example would be once a Carton has arrived in the local zone, the PLC determines that the Carton needs to be re-positioned or perhaps squared up against a PLC controlled pop-up stop.

> ! Jog control bits over-ride ZPA logic control and should be used with caution! Improper usage of jog controls can produce unexpected results and/or damage to

Bit 12 – Wake Up Local Zone

Setting this bit will cause the local zone to "wake up" and run to accept a carton the same as if it's upstream ConveyLinx-Ai Family Controllers had written a status value of "4". This function would be useful for a merge onto a main line of ZPA conveyor.

Bit 13 – Enable Maintenance Mode

Setting this bit will place the local zone in maintenance mode. In this mode the motor will not run regardless of zone conditions. The zone upstream of this local zone will receive a "busy" status to inhibit release of any item into this local zone. While in this state, the SEN and Motor LEDs will flash on and off in green color.

11.3.2.3. Speed Control

Register Name / Module Address	Assembled Address for PLC	Description
Set Left Motor Speed 4:0040	M: 4:1606 E: O.Data [6] P: Byte 12 (Hi) Byte 13 (Lo)	Value in mm/sec for MDR or RPM x 10 for PGD: Range: Depends upon the Ai-MDR or PGD connected *MDR Example: 400 = 0.4 m/s* *PGD Example: 400 = 40 RPM* 0 = Remain at last non-zero value entered
Set Right Motor Speed 4:0064	M: 4:1607 E: O.Data [7] P: Byte 14 (Hi) Byte 15 (Lo)	

✱ Leaving these registers at "0" will instruct the module to use its configured speed. Any non-zero value will instruct the nodule to use this non zero value as the speed reference. The speed will stay at this reference until this register is changed to a new non zero value or set to "0". When this register is set to "0", the module will continue to use the last non zero value it was given. **Setting this value to "0" will not cause the motor to run at "0" speed**

11.3.2.4. Release and Status

Register Name / Module Address	Assembled Address for PLC	Description
Release and Accumulate on Next Arrival for Local Upstream Zone 4:0105	M: 4:1608 E: O.Data [8] P: Byte 16 (Hi) Byte 17 (Lo)	When bit 0 of Accumulation Control for Local Upstream Zone is set: Changing the value in this register will cause the Carton accumulated in this zone to release and the zone will be armed to automatically accumulate the next Carton that arrives
Release and Accumulate on Next Arrival for Local Downstream Zone 4:0185	M: 4:1609 E: O.Data [9] P: Byte 18 (Hi) Byte 19 (Lo)	When bit 0 of Accumulation Control for Local Downstream Zone is set: Changing the value in this register will cause the Carton accumulated in this zone to release and the zone will be armed to automatically accumulate the next Carton that arrives
Set Status for Upstream Induct 4:0134	M: 4:1610 E: O.Data [10] P: Byte 20 (Hi) Byte 21 (Lo)	Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor: Set value to 4 to cause the local upstream zone to run to accept the Carton being delivered by the non-ConveyLinx controlled conveyor. Set value to 1 to cause the local upstream zone to accept the tracking data written in *Set Induct Tracking Word 1 / Word 2*
Set Status for Downstream Discharge 4:0232	M: 4:1611 E: O.Data [11] P: Byte 22 (Hi) Byte 23 (Lo)	Only used when local module is discharging loads to Non-ConveyLinx controlled conveyor: • Set value to 5 to cause the local downstream zone to accumulate and hold any Carton that arrives • Set value to 1 to allow the local downstream zone to release the Carton

11.3.2.5. Induct Tracking Forward and Reverse

Register Name / Module Address	Assembled Address for PLC	Description
Set Induct Tracking Forward Direction Word 1 4:0139	M: 4:1612 E: O.Data [12] P: Byte 24 (Hi) Byte 25 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor:</u> Set value for 16-bit integer tracking word #1 for Carton leaving non-ConveyLinx controlled conveyor that is in transit to arrive on the local upstream zone when conveyor is running in default or "forward" direction
Set Induct Tracking Forward Direction Word 2 4:0140	M: 4:1613 E: O.Data [13] P: Byte 26 (Hi) Byte 27 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor:</u> Set value for 16-bit integer tracking word #2 for Carton leaving non-ConveyLinx controlled conveyor that is in transit to arrive on the local upstream zone when conveyor is running in default or "forward" direction
Set Induct Tracking Reverse Direction Word 1 4:0237	M: 4:1614 E: O.Data [14] P: Byte 28 (Hi) Byte 29 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor:</u> Set value for 16-bit integer tracking word #1 for Carton leaving non-ConveyLinx controlled conveyor that is in transit to arrive on the local upstream zone when conveyor is running in opposite of default or "reverse" direction
Set Induct Tracking Reverse Direction Word 2 4:0238	M: 4:1615 E: O.Data [15] P: Byte 30 (Hi) Byte 31 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor:</u> Set value for 16-bit integer tracking word #2 for Carton leaving non-ConveyLinx controlled conveyor that is in transit to arrive on the local upstream zone when conveyor is running in opposite of default or "reverse" direction

11.3.2.6. Set Outputs and Motor Clear

Register Name / Module Address	Assembled Address for PLC	Description

Clear Motor Error
4:0022
	M: 4:1616
	E: O.Data [16]
	P: Byte 32 (Hi)
	Byte 33 (Lo)
		Logical 1 or 0:
		   1 = send reset command
		   0 = clear reset command

Set Aux I/O Outputs
4:0063
	M: 4:1617
	E: O.Data [17]
	P: Byte 34 (Hi)
	Byte 35 (Lo)
		Bitwise Value:
		   bit 08 = Left Aux I/O (Pin 2) Usage: 0 = Use as Input / 1 = Use as Output
		   bit 09 = Right Aux I/O (Pin 2) Usage: 0 = Use as Input / 1 = Use as Output
		   bit 12 = Left Aux I/O (Pin 2) as Output: 1 = ON, 0 = OFF (only if bit 8 is ON)
		   bit 13 = Right Aux I/O (Pin 2) as Output: 1 = ON, 0 = OFF (only if bit 9 is ON)
		All other bits reserved

Reserved	M: 4:1618 E: O.Data [18] P: Byte 36 (Hi) Byte 37 (Lo)	

Clear Motor Error

Errors deemed "fatal" for the module (motor short circuit and Hall Effect sensor fault) require either removal of power to reset or remote reset by PLC. Setting bit 0 of this register to 1 will initiate this remote error reset from the PLC to the local module. Setting this bit will reset a fatal error on either (or both) the left or right MDR.

> **!** PLC must continuously set bit 0 = 1 in the *Clear Motor Error* register for at least 500 msec for the module to recognize the reset command

Aux I/O Outputs

In certain applications, it may be desired to have the PLC actuate a device along the conveyor (i.e. illuminate a light or energize a solenoid coil, etc.). With the ConveyLinx network this can be done without installing a separate PLC I/O system to do this. In *EasyRoll*, simply select NONE from the Aux I/O Pin 2 Usage tab and this allows the PLC to have access to the Aux I/O pin and configure it to be either an input or an output and use it accordingly.

✱ If you make any selection other than "None" for the Aux I/O Pin 2 Usage, the PLC has no access to use or configure the Aux I/O Pin 2.

11.3.2.7. ConveyStop and Clear Jams

Register Name / Module Address	Assembled Address for PLC	Description
ConveyStop Command Word 4:0020	M: 4:1619 E: O.Data [19] P: Byte 38 (Hi) Byte 39 (Lo)	Integer Value: 0 = No Command 1 = Command local module's Stop Group to go to Stopped State 2 = Command Local Module's Stop Group to Clear Stopped State
Clear Sensor Jam for Local Upstream Zone 4:0109	M: 4:1620 E: O.Data [20] P: Byte 40 (Hi) Byte 41 (Lo)	Use when PLC detects a local Upstream jam in Module Status Word #1 Bit 5: PLC creates transition from 0 to 1 to send command to local upstream zone to clear the jam condition
Clear Sensor Jam for Local Downstream Zone 4:0189	M: 4:1621 E: O.Data [21] P: Byte 42 (Hi) Byte 43 (Lo)	Use when PLC detects a local Downstream jam in Module Status Word #2 Bit 5: PLC creates transition from 0 to 1 to send command to local downstream zone to clear the jam condition

✱ Please note that for the *Clear Sensor Jam* registers, the function requires the PLC to make a transition from 0 to 1 in the register to initiate another auto-clear cycle. Holding the value to 1 will not cause the retry to continue indefinitely. Each attempt requires a new transition from 0 to 1

11.3.2.8. Direction and Accumulation Mode

Register Name / Module Address	Assembled Address for PLC	Description
Direction & Accumulation Mode Control for Local Upstream Zone 4:0365	M: 4:1622 E: O.Data [22] P: Byte 44 (Hi) Byte 45 (Lo)	Used to change direction of flow or accumulation mode for a contiguous group of zones beginning with the local upstream / downstream zone

Value for Low Byte of Register:

0 = Normal Function

1 = Accumulate Zones

2 = Accumulate Zones

3 = Change Accumulation Release Mode

4 = Return Release Mode to Configured Default

5 = Run All FWD Direction |
| Direction & Accumulation Mode Control for Local Downstream Zone 4:0375 | M: 4:1623 E: O.Data [23] P: Byte 46 (Hi)   Byte 47 (Lo) | 6 = Run All REV Direction

7 = Place All in Maintenance Mode

10 = Set Direction to Configured Default (Forward)

11 = Set Direction to opposite of Configured default (Reverse)

Value for High Byte of Register:

Number of ZONES beginning with the local Upstream / Downstream Zone for which the Low Byte value is applied – from 1 to 220. If ALL ZONES in the subnet need to be controlled then leave the High Byte = "0" |

Notes on Accumulate Zones Functionality

The *Accumulate Zones* function (Low Byte values 1 or 2) expects all zones to be empty or in the process of releasing downstream when the function is initiated. Once the *Accumulate Zones* function has been initiated, the logic is expecting to see the zone status change from clear to blocked before the zone accumulates.

This can produce underlined unexpected results if the *Accumulate Zones* function is initiated on a line where some or all zones are already occupied and accumulated.

Let's say that we have a conveyor line where all cartons are stopped and accumulated. If the *Accumulate Zones* function is initiated; nothing will happen and it will appear that all zones are being commanded to accumulate.
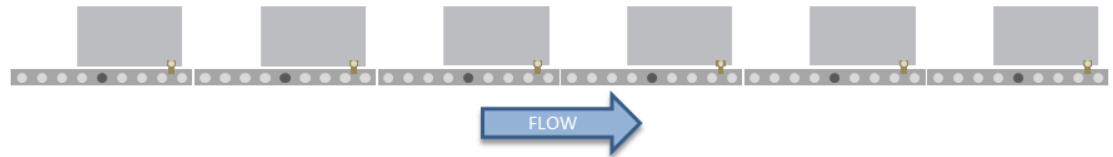
However, each zone needs to see a transition from empty to full again for the function to be initiated. So in this case, if someone removes a carton from a zone, the carton upstream will still convey to fill the empty zone, as well as any other upstream cartons. If a given zone was already empty when the function was initiated; the next carton that arrives will accumulate.

FLOW
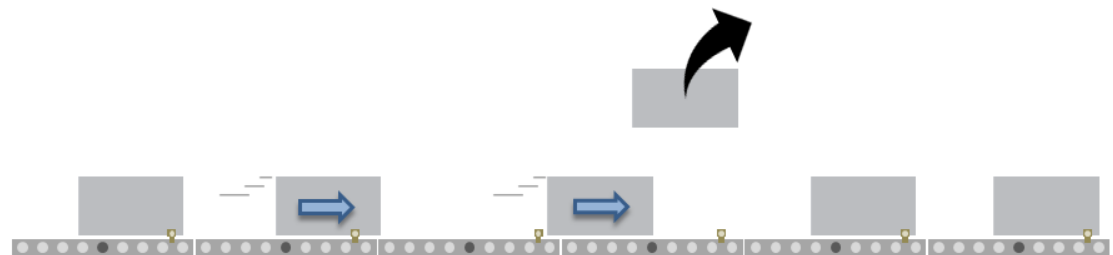
11.3.2.9. ConveyMerge Interface

Register Name / Module Address	Assembled Address for PLC	Description
ConveyMerge Interface 4:0387	M: 4:1624 E: O.Data [24] P: Byte 48 (Hi) 　　Byte 49 (Lo)	ONLY AVAILABLE for FIRMWARE 4.25 and HIGHER Used to dynamically modify a ConveyMerge configuration already established with EasyRoll 　Bitwise Values: 　　Bit 15: Set to enable PLC control of ConveyMerge 　　Bit 4: Set to disable release from Center line 　　Bit 5: Set to disable release from Left line 　　Bit 6: Set to disable release from Right line Bits 0 thru 3 are interpreted as a numerical value to change the release priority: 　Value: 　　0 = First come, first served release 　　1 = Center Line has priority 　　2 = Left Line has priority 　　3 = Right Line has priority

11.3.3. ZPA Examples

Basic Accumulate and Release with Tracking Data
Conveyor Setup for Simple Bar Code Reader
Upstream Accept Interface
Downstream Discharge Interface
Simple Divert Example
Merge onto ZPA Main Line

11.3.3.1. Basic Accumulate and Release with Tracking Data

Here is a typical arrangement of an upstream or downstream zone on an ConveyLinx-Ai Family Controllers in ZPA mode that is not the most upstream or most downstream zone in a given network. This example will show how to cause a Carton to accumulate, how to detect a Carton has arrived, how to write tracking data, and finally how to release the Carton. For this example, the PLC must establish ZPA Mode Assembly Input/Output connections to Module B as shown

Upstream Zone Example

First, let's assume we want to accumulate any Carton that arrives on the upstream zone of Module B. With the PLC:

1. Set bit 0 in _Accumulation Control for Local Upstream Zone_ register to instruct this zone to accumulate any Carton that arrives.
2. Monitor _Arrival Count Local Upstream Zone_ and _Departure Count Local Upstream Zone_ registers. On the leading edge when these two values become not equal, the PLC knows there is a new arrival. Note that as long as a Carton is physically occupying the upstream zone, these two values will not be equal.
3. Upon a successful arrival of a Carton in the upstream zone, then tracking data in _Current Upstream Zone Tracking Word 1_ and _Current Upstream Zone Tracking Word 2_

will be valid for the newly arrived Carton.

4. The PLC may then decide that this tracking data is to be updated. The PLC can then write new tracking data to registers *Set Local Upstream Zone Tracking Word 1* and *Set Local Upstream Zone Tracking Word 2*

5. When the PLC is ready to release the Carton in the upstream zone, it should read the value in the *Current Release Count for Upstream Zone* register, add 1 to this value, and then write this new value to *Release and Accumulate on Next Arrival for Local Upstream Zone* register. When Module B sees this new value in this register, it will release the Carton in the upstream zone and automatically accumulate the next new Carton that arrives. Please note that if the downstream conditions from Module B are full when this new value is written, Module B will remember that it was instructed to release and will release the Carton when downstream conditions become clear without any further signal from the PLC.

6. The PLC can detect when the Carton has departed the sensor on Module B upstream zone by examining the values in *Arrival Count Local Upstream Zone* and *Departure Count Local Upstream Zone* registers. On the leading edge of when these two values are equal, the PLC will know that the Carton has departed the zone sensor in Module B upstream zone.

If the PLC wants to cancel the accumulation control for Module B upstream zone:
PLC can reset bit 0 in *Accumulation Control for Local Upstream Zone* register. This will signal Module B to release any Carton accumulated and not accumulate the next Carton that arrives at Module B upstream zone.

> ✳ Please note that if any new tracking data has been written to *Set Local Upstream Zone Tracking Word 1* and/or *Set Local Upstream Zone Tracking Word 2* and accumulation control is then canceled by resetting bit 0, this data will NOT be assigned to the Carton when it is released. **The ONLY way to pass tracking data to a Carton is by following Step 5 above**

Downstream Zone Example

First, let's assume we want to accumulate any Carton that arrives on the upstream zone of Module B. With the PLC:

1. Set bit 0 in *Accumulation Control for Local Downstream Zone* register to instruct this zone to accumulate any Carton that arrives.

2. Monitor *Arrival Count Local Downstream Zone* and *Departure Count Local Downstream Zone* registers. On the leading edge when these two values become not equal, the PLC knows there is a new arrival. Note that as long as a Carton is physically occupying the downstream zone, these two values will not be equal.

3. Upon a successful arrival of a Carton in the downstream zone, then tracking data in *Current Downstream Zone Tracking Word 1* and *Current Downstream Zone Tracking Word 2* will be valid for the newly arrived Carton.

4. The PLC may then decide that this tracking data is to be updated. The PLC can then write new tracking data to registers *Set Local Downstream Zone Tracking Word 1* and *Set Local Downstream Zone Tracking Word 2*.

5. When the PLC is ready to release the Carton in the downstream zone, it should read the value in the *Current Release Count for Downstream Zone* register, add 1 to this value, and then write this new value to *Release and Accumulate on Next Arrival for Local Downstream Zone* register. When Module B sees this new value in this register, it will release the Carton in the upstream zone and automatically accumulate the next new Carton that arrives. Please note that if the downstream conditions from Module B are full when this new value is written, Module B will remember that it was instructed to release and will release the Carton when downstream conditions become clear without any further signal from the PLC.

6. The PLC can detect when the Carton has departed the sensor on Module B downstream zone by examining the values in *Arrival Count Local Downstream Zone* and *Departure Count Local Downstream Zone* registers. On the leading edge of when these two values are equal, the PLC will know that the Carton has departed the zone sensor in Module B downstream zone.

If the PLC wants to cancel the accumulation control for Module B downstream zone:
PLC can reset bit 0 in *Accumulation Control for Local Downstream Zone* register. This will signal Module B to release any Carton accumulated and not accumulate the next Carton that arrives at Module B downstream zone.

> ✱ Please note that if any new tracking data has been written to *Set Local Downstream Zone Tracking Word 1* and/or *Set Local Downstream Zone Tracking Word 2* and accumulation control is then canceled by resetting bit 0, this data will NOT be assigned to the Carton when it is released. **The ONLY way to pass tracking data to a Carton is by following Step 5 above**

11.3.3.2. Conveyor Setup for Simple Bar Code Reader

This example shows how to set up the conveyor control to easily singulate cartons through a bar code scanning region
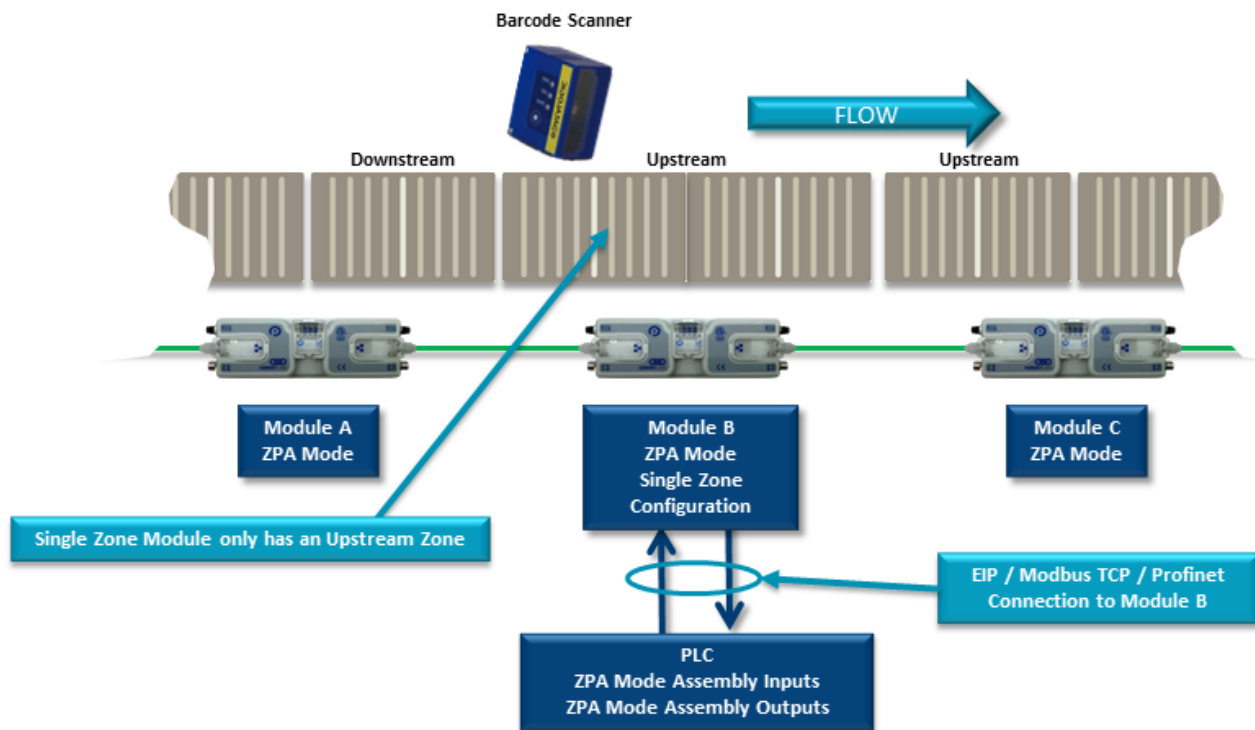
In this example, *Module B* must be logically configured as a single zone. This can be either a single long zone with one MDR and one sensor, or a single long zone with 2 MDRs and one sensor.

For this example to work properly; the discharging zone of *Module A* and the single zone of *Module B* must be in the default *Singluation Release* mode (not *Train mode*). *Module A* must also <u>not</u> have its *Arrival Jam* disabled />

Because of built-in ZPA functionality, when a Carton leaves *Module A* discharge zone, *Module A* waits until it receives confirmation from *Module B* that the Carton arrived. If a new Carton arrives at *Module A*, it will accumulate until this arrival confirmation occurs. Because *Module B* is a single long zone, the space for what would have been the upstream zone for *Module B* (if it was configured as two zones) will now essentially be left clear when a Carton is accumulated on *Module B*. This is the area where the bar code scanner is located. In this configuration, any new arrival at *Module B* will be assured to be the Carton associated with the last scan from the bar code reader. For this example, the PLC must establish a connection with *Module B* and then simply follow the zone control as described for [basic accumulate and release](#)

11.3.3.3. Upstream Accept Interface

This example describes how to use a PLC to control the "wake-up" and passing of tracking data to the most upstream zone of a ConveyLinx controlled conveyor
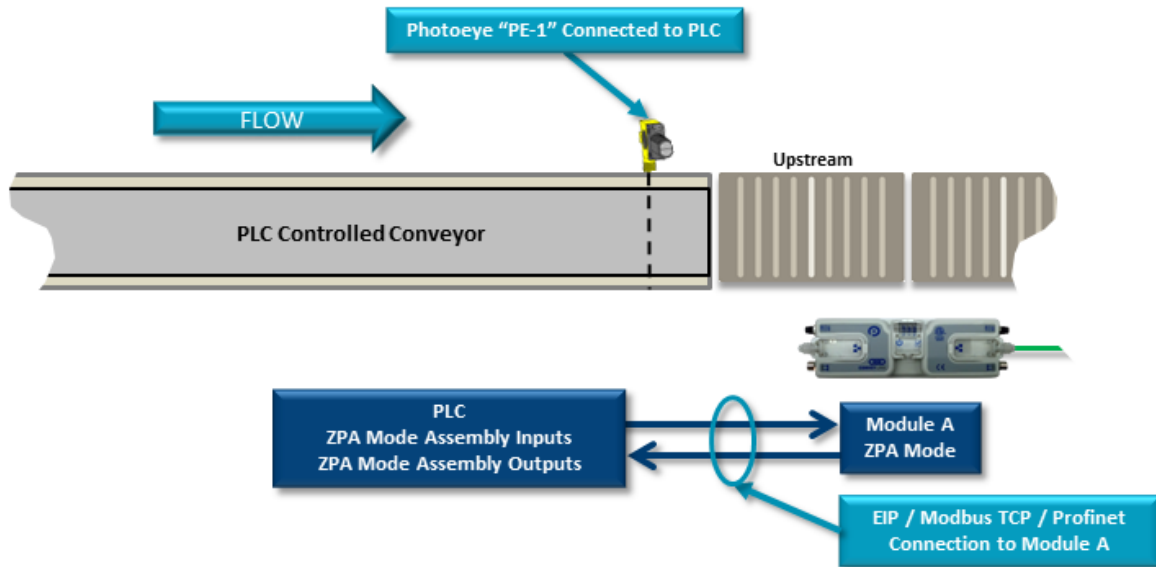
For this example, the PLC must establish a connection with **Module A**. When PLC is ready to transfer a Carton from the PLC controlled conveyor to the Upstream Zone of the MDR Conveyor, the PLC logic must:

1. Write a "4" into *Set Status for Upstream Induct* register. This will cause the upstream zone of **Module A** to run to accept Carton.
2. When Carton clears **PE-1**, write tracking data to *Set Induct Tracking Forward Direction Word 1* and *Set Induct Tracking Forward Direction Word 2*
3. When Carton clears **PE-1**, write a "1" to *Set Status for Upstream Induct* register. When **Module A** sees this register change to "1", it knows that the tracking data written to *Set Induct Tracking Forward Word 1* and *Set Induct Tracking Forward Word 2* is valid and when the Carton reaches the sensor in its upstream zone, the tracking data will be correctly inducted

✱ Module A must see the *Set Status for Upstream Induct* register change from 4 to 1 in order to recognize the tracking data written to *Set Induct Tracking Forward Word* registers. If using the clearing of PE-1 to initiate the write of "1" the *Set Status for Upstream Induct* register in the PLC logic; the physical length of the Carton must be less that the distance between PE-1 and Module A's upstream zone sensor because **Module A** inducts the tracking data upon the Carton arriving at its upstream sensor and if *Set Status for Upstream Induct* has not changed from 4 to 1, the tracking data will not be valid.

! If a value of "4" is in the *Set Status for Upstream Induct* register at the same time the Carton reaches the local upstream zone's sensor; the local upstream zone will assume the Carton is physically longer than one zone and will invoke it's on board **Flex Zone** logic and behave accordingly. Please refer to the ConveyLinx User's Guide for description of Flex Zone operation. To reliably transfer tracking data and ensure single Carton induction to the local upstream zone, the PLC programmer must ensure proper use of the *Set Status for Upstream Induct* register

11.3.3.4. Downstream Discharge Interface

This example shows how to control the discharge of a Carton from the most downstream zone of a ConveyLinx controlled conveyor. This example also shows how to properly accept the tracking data from the ConveyLinx module. For this example, the PLC must establish a connection with **Module X**.
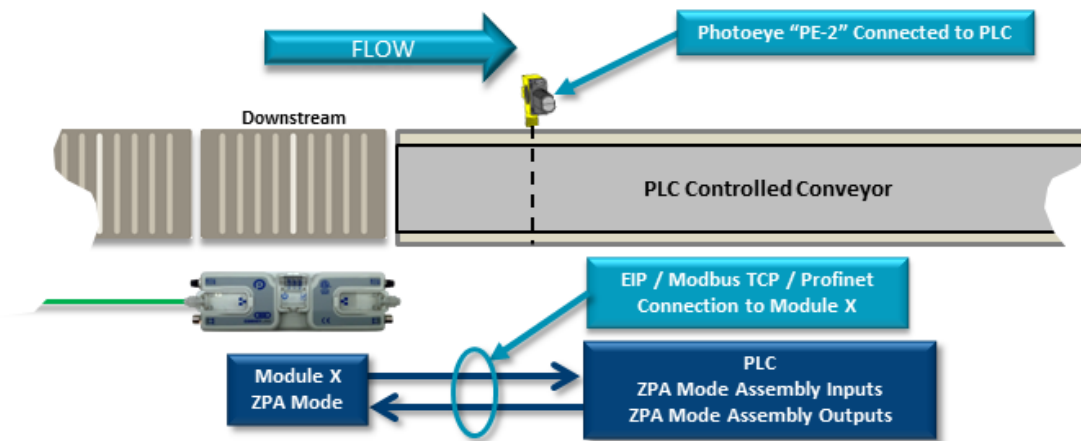
1. To hold any Carton that arrives at **Module X** downstream zone, the PLC writes a "5" to *Set Status for Downstream Discharge* register. This tells **Module X** that downstream conditions are "full and stopped" so it will not try to release any loads downstream.
2. When the PLC controlled conveyor is ready to accept a Carton from the MDR downstream zone, it writes a "1" to *Set Status for Downstream Discharge* register. This tells **Module X** that downstream conditions are clear and if it has a Carton on its downstream zone, it will run to release it downstream.
3. When the Carton clears the local sensor in **Module X** Downstream Zone, the tracking data for that Carton will be populated in *Get Tracking Forward Direction Word 1* and *Get Tracking Forward Direction Word 2* registers.
4. When Carton arrives at PE-2, the PLC writes a "5" to *Set Status for Downstream Discharge* register. This tells **Module X** that the Carton successfully transferred. If this is not done, then **Module X** will produce an **Arrival Jam** condition.

Optional Step:

If the application is such that the PLC would like to inhibit a new Carton from entering **Module X** downstream zone for whatever reason, there is a way to accumulate the next upstream zone to the **Module X** downstream zone. The PLC can set bit 8 in the *Accumulation Control for Local Downstream Zone* and this will cause the adjacent upstream zone to accumulate (whether this zone is on **Module X** or next adjacent upstream ConveyLinx Module). To return the accumulated upstream zone to normal operation; the PLC simply resets bit 8 in this register

✴ Please note that if **Module X** is configured as a **single zone**, then you must set bit 8 in the *Accumulation Control for Local Upstream Zone* register because **a single zone module only has an upstream zone and no downstream zone.**

11.3.3.5. Simple Divert Example

This example shows how to use a PLC to control a simple divert mechanism to divert a Carton from a ZPA zone and capture its tracking. This example also illustrates how to use the set downstream arrival function with the PLC to keep the discharging module from generating a jam condition when the Carton is moved from the ZPA zone to the divert lane. The PLC will be connected to *Module B*.

> ✱ This example assumes that the tracking data arriving from each new Carton arrival at *Module B* Upstream Zone contains a value that will indicate to the PLC that a Carton does or does not need to divert. However, it is not required that the PLC use Carton tracking data be used to determine when to divert a Carton. The PLC can use whatever criteria it needs to make a divert decision.
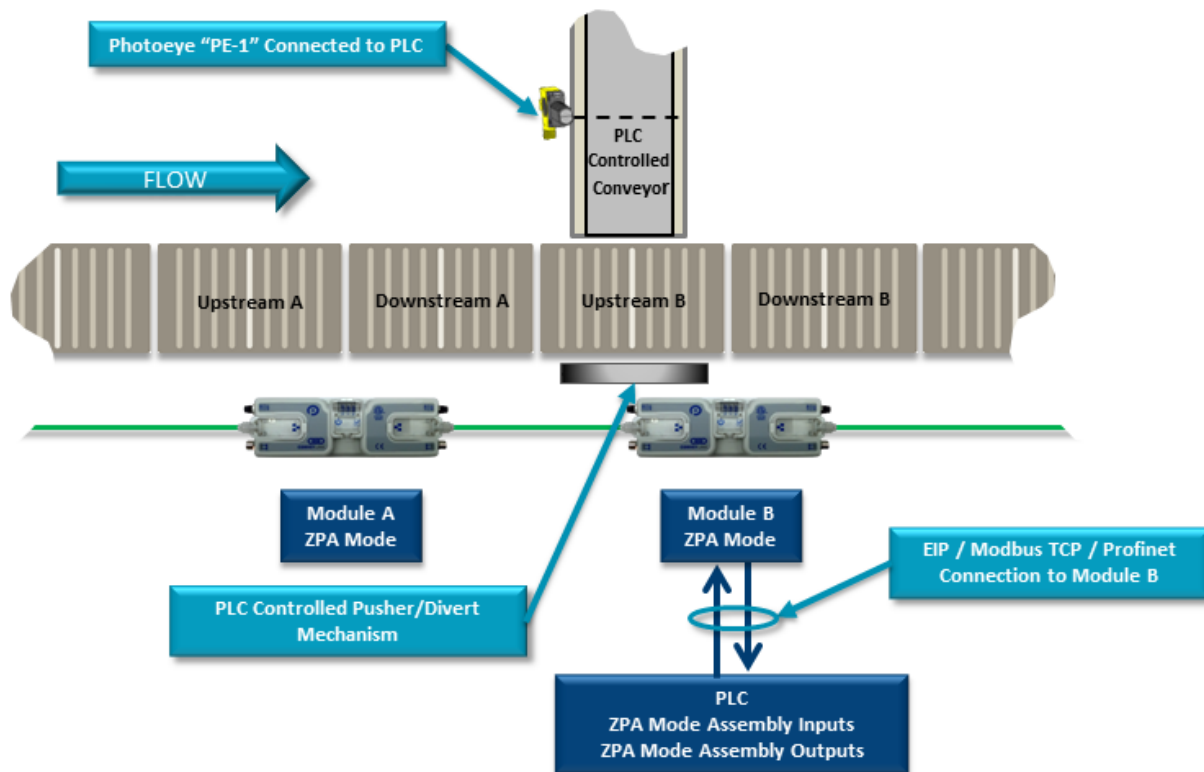
Prepare Module B Upstream Zone:

1. Set **bit 0** in *Accumulation Control for Local Upstream Zone* register to instruct this zone to accumulate any Carton that arrives.
2. Monitor *Arrival Count Local Upstream Zone* and *Departure Count Local Upstream Zone* registers. On the leading edge when these two values become not equal, the PLC knows there is a new arrival. Note that as long as a Carton is physically occupying the upstream zone, these two values will not be equal.

3. Upon a successful arrival of a Carton in the upstream zone, then tracking data in *Current Upstream Zone Tracking Word 1* and *Current Upstream Zone Tracking Word 2* will be valid for the newly arrived Carton

If the PLC determines that the newly arrived carton needs to divert:

1. **Set bit 8** of *Accumulation Control for Local Upstream Zone* register. This will keep *Module A* from releasing a Carton to *Module B* while the divert operation is in progress.
2. PLC will then initiate it's divert mechanism.
3. When carton reaches **PE-1**, the PLC will then **set bit 9** of the *Accumulation Control for Local Upstream Zone* register. This will tell *Module B* that the carton "successfully arrived at its downstream position".
4. Once carton has cleared **PE-1**, then the PLC should **clear bit 9** of the *Accumulation Control for Local Upstream Zone* register.
5. PLC then can **clear bit 8** of the *Accumulation Control for Local Upstream Zone* register to instruct **Module A** that it is OK to allow the next carton to enter *Module B's upstream zone*.

If the PLC determines that the newly arrived carton does not need to divert:

The PLC can simply modify Carton's tracking data if desired and release the Carton as described in Basic Accumulate and Release section

> ✳ Do not release the Carton by clearing bit 0 of *Accumulation Control for Local Upstream Zone* in your PLC program. Follow the procedure in the "**Basic Accumulate and Release**" example

11.3.3.6. Merge onto ZPA Main Line

This example shows how to perform a simple merge with only a single module connection to the PLC. In this example the PLC only needs to connect to *Module C* and from this connection, the PLC can monitor conditions on *Module A* as well as "wake-up" **Module C / Module B** zone to accept a carton from the merging curve section.
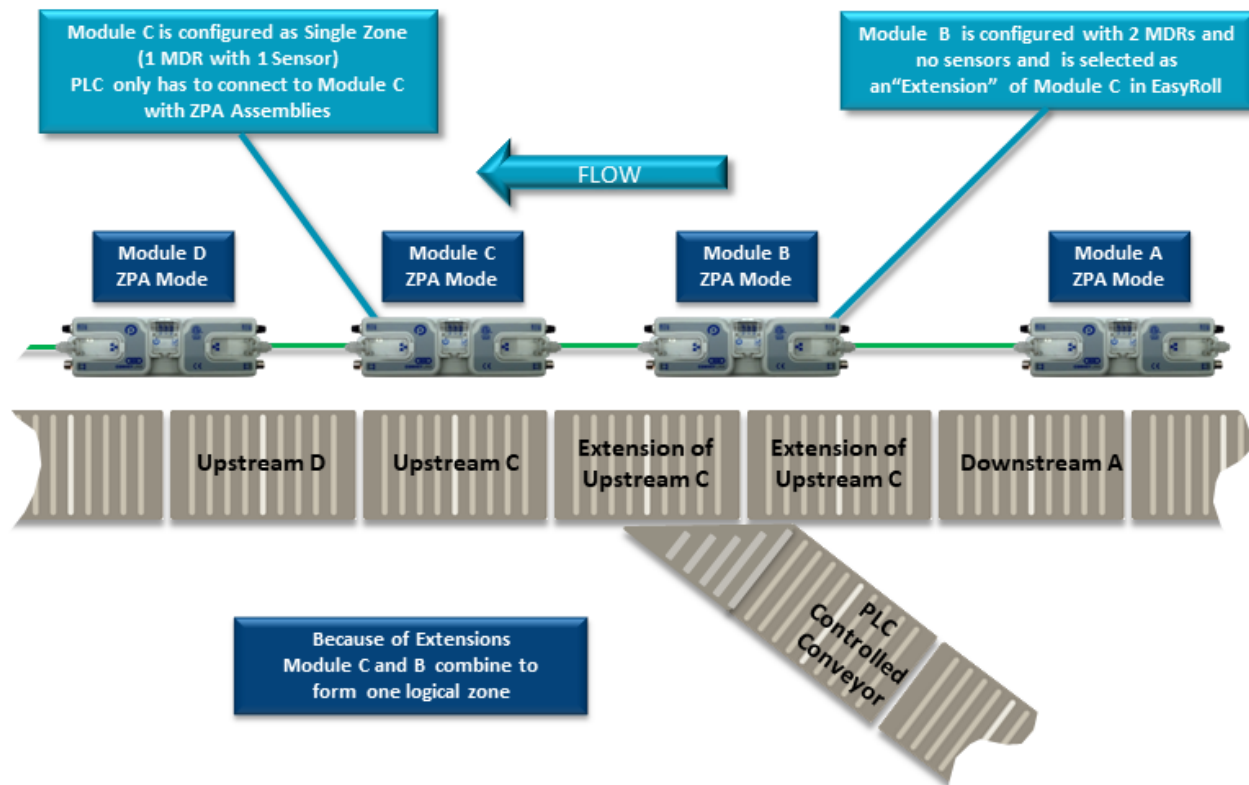
1. With no PLC intervention, as long as cartons arrive into *Module A* they will be conveyed to **Module C/B** and onto **Module D** in normal ZPA fashion.
2. To verify that *Module C* (and by extension *Module B*) is ready to accept a carton from the merge curve; the PLC needs to know that *Module C*'s zone is clear and stopped by examining *Local Status Upstream Zone Forward Direction* register (Low Byte). When the low byte of this register equals 1, the zone is clear and stopped.
3. When the PLC needs to convey a carton from the merging curve, the PLC needs to **set bit 8** in *Accumulation Control for Local Upstream Zone* register to accumulate the adjacent upstream zone, in this example **setting bit 8** will cause *Module A* **downstream zone** to accumulate any carton that arrives.
4. When the PLC is ready to release a carton from the **PLC Controlled Conveyor** spur, the PLC **sets bit 12** in the *Accumulation Control for Local Upstream Zone* register to wake up the *Module C* zone (along with Module B's zones because *Module B* is an extension of *Module C*).
5. When the PLC sees the low byte of the *Local Status Upstream Zone Forward Direction* register change to a value of 4 or 5, the PLC knows that the carton made it to Module C's

zone sensor and must then reset bit 12 in the *Accumulation Control for Local Upstream Zone* register. Either of these values will indicate a successful arrival at *Module C*

6. Upon reset of bit 12 of the *Accumulation Control for Local Upstream Zone* register, the PLC can reset bit 8 of the same register to indicate to *Module A* that it is OK to continue normal ZPA function

11.3.4. Reduced Size ZPA Mode Assemblies

For some PLC controllers, the data size footprint required for a given assembly can be a limiting factor on how many devices can connect to a given controller. ConveyLinx includes input/output assemblies with fewer registers that contain basic functions in applications where the full functionality of the Standard ZPA Mode Assemblies is not required. The Reduced Size Assemblies provide basic accumulation and release control along with module diagnostic data while removing the interfaces for reading/writing tracking data.

ZPA Mode Inputs to PLC

Module Address	Register Name	Assembled Modbus Address	Assembled EIP Address	Assembled Profinet Address
4:0116	Local Status Upstream Zone	4:2900	I.Data [0]	Byte 0 (Hi) Byte 1 (Lo)
4:0196	Local Status Downstream Zone	4:2901	I.Data [1]	Byte 2 (Hi) Byte 3 (Lo)
4:0106	Arrival Count Local Upstream Zone	4:2902	I.Data [2]	Byte 4 (Hi) Byte 5 (Lo)
4:0107	Departure Count Local Upstream Zone	4:2903	I.Data [3]	Byte 6 (Hi) Byte 7 (Lo)
4:0186	Arrival Count Local Downstream Zone	4:2904	I.Data [4]	Byte 8 (Hi) Byte 9 (Lo)
4:0187	Departure Count Local Downstream Zone	4:2905	I.Data [5]	Byte 10 (Hi) Byte 11 (Lo)
4:0088	Module Status Word 1	4:2906	I.Data [6]	Byte 12 (Hi) Byte 13 (Lo)
4:0089	Module Status Word 2	4:2907	I.Data [7]	Byte 14 (Hi) Byte 15 (Lo)
4:0105	Current Release Count for Upstream Zone	4:2908	I.Data [8]	Byte 16 (Hi) Byte 17 (Lo)
4:0185	Current Release Count for Downstream Zone	4:2909	I.Data [9]	Byte 18 (Hi) Byte 19 (Lo)
4:0035	Sensor Port Inputs	4:2910	I.Data [10]	Byte 20 (Hi) Byte 21 (Lo)

	Reserved	4:2911	I.Data [11]	Byte 22 (Hi) Byte 23 (Lo)

ZPA Mode Outputs from PLC

Module Address	Register Name	Assembled Modbus Address	Assembled EIP Address	Assembled Profinet Address
4:0104	Accumulation Control for Local Upstream Zone	4:3000	O.Data [0]	Byte 0 (Hi) Byte 1 (Lo)
4:0184	Accumulation Control for Local Downstream Zone	4:3001	O.Data [1]	Byte 2 (Hi) Byte 3 (Lo)
4:0040	Set Left MDR Speed	4:3002	O.Data [2]	Byte 4 (Hi) Byte 5 (Lo)
4:0064	Set Right MDR Speed	4:3003	O.Data [3]	Byte 6 (Hi) Byte 7 (Lo)
4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	4:3004	O.Data [4]	Byte 8 (Hi) Byte 9 (Lo)
4:0185	Release and Accumulate on Next Arrival for Local Downstream Zone	4:3005	O.Data [5]	Byte 10 (Hi) Byte 11 (Lo)
4:0134	Set Status for Upstream Induct	4:3006	O. Data[6]	Byte 12 (Hi) Byte 13 (Lo)
4:0232	Set Status for Downstream Discharge	4:3007	O.Data [7]	Byte 14 (Hi) Byte 15 (Lo)
4:0022	Clear Motor Error	4:3008	O.Data [8]	Byte 16 (Hi) Byte 17 (Lo)
4:0063	Set Aux I/O Outputs	4:3009	O.Data [9]	Byte 18 (Hi) Byte 19 (Lo)
	Reserved	4:3010	O.Data [10]	Byte 20 (Hi) Byte 21 (Lo)
4:0109	Clear Sensor Jam Command for Local Upstream Zone	4:3011	O.Data [11]	Byte 22 (Hi) Byte 23 (Lo)
4:0189	Clear Sensor Jam Command for Local Downstream Zone	4:3012	O.Data [12]	Byte 24 (Hi) Byte 25 (Lo)
4:0365	Direction & Accumulation Mode Control for	4:3013	O.Data	Byte 26 (Hi)

	Local Upstream Zone		[13]	Byte 27 (Lo)
4:0375	Direction & Accumulation Mode Control for Local Downstream Zone	4:3014	O.Data [14]	Byte 28 (Hi) Byte 29 (Lo)

11.4. PLC I/O Mode Control

When an ConveyLinx-Ai Family Controllers is in PLC I/O mode, all automatic functions of detecting loads and running motors are suspended by the local ConveyLinx-Ai Family Controllers's on-board logic and the external controller must explicitly read inputs and write data output to cause motors to run. The following items are available for external controller when the ConveyLinx-Ai Family Controllers is in PLC I/O Mode:

- Status of all available digital inputs on Sensor Ports (2 dedicated as inputs and 2 configurable as inputs)
- Module voltage reading
- Left and Right motor status of frequency, current, and calculated temperature
- Left and Right motor diagnostic error status word
- Control of Aux I/O digital outputs
- Ability to independently run both Left and Right motors
- Ability to set speed, acceleration, deceleration, and Braking method for Left and Right motors
- Ability to configure one or both motor ports to digital output mode
- Ability to remotely clear fatal motor error condition
- Ability to instruct module to stop motor outputs
- Ability to interface with upstream and downstream ZPA modules for status and tracking

✱ When an ConveyLinx-Ai Family Controllers is placed in PLC I/O mode; it suspends all of its internal ZPA logic control. Any sensors or motors connected to the module will require explicit interaction with an external controller. The external controller will have typical Ethernet-based remote I/O performance from an ConveyLinx-Ai Family Controllers when in PLC I/O mode

11.4.1. Setting PLC I/O Mode in EasyRoll

Individual modules must be placed into PLC I/O Mode from the *EasyRoll* software tool. This is done by invoking the *Advanced Dialog* and using the *Connections* tab

From the main screen, first enter the correct *Subnet* into the "Network IP" boxes and the correct *Node* you want to connect. Invoke the *ConveyLinx Advanced Dialog* and select the *Connections* tab. Note that the Node is being viewed is in the center and it is greyed out. Select the "PLC I/O Controlled" checkbox. With this checked the "Clear Connections" checkbox becomes enabled. **Check or Uncheck the "Clear Connections" checkbox depending upon your application**. Click "Apply" to initiate the change. The module will restart and this may take several seconds to complete
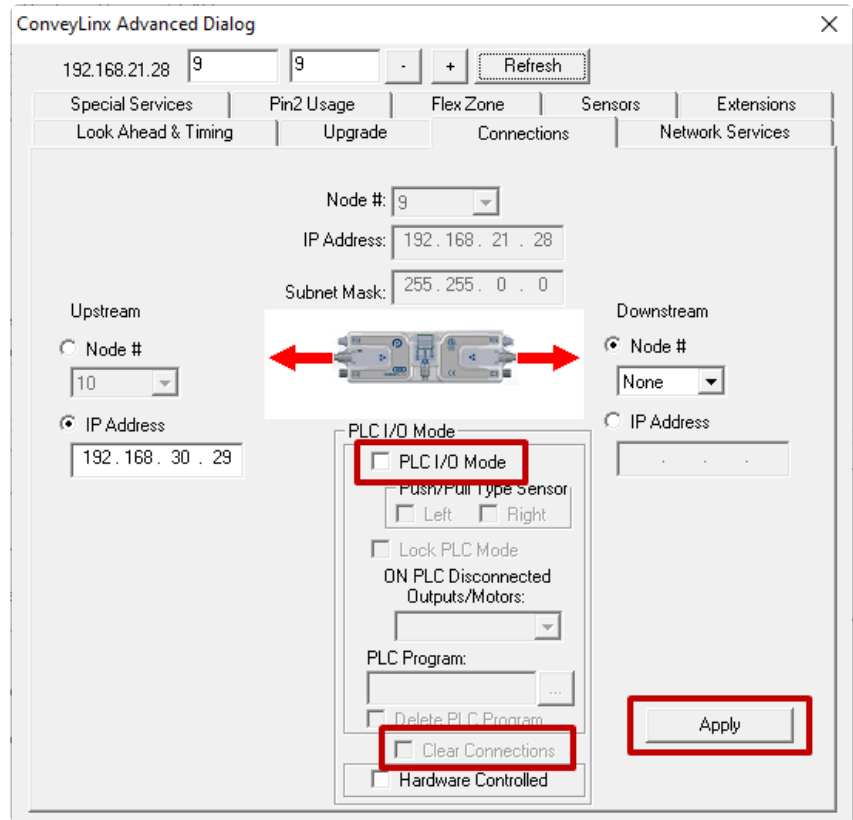
> **!** IMPORTANT NOTE: Once a given module has been placed in PLC I/O mode, the ONLY way to return it to ZPA mode is to perform an Auto-Configuration procedure or Restore from a backup file with EasyRoll. There is no "undo" or "reset" function for this action

11.4.2. Optional Clear Connections Choice

The decision to "Clear Connections" is based upon the application. When a string of modules are Auto-Configured, each successive module in the string establishes a logical upstream / downstream connection with its neighbor module's. These connections provide the basis for the logical flow of inter-module status data for ZPA functionality. However, if a single module node within a string of ZPA configured nodes needs to be utilized in PLC I/O mode, these logical connections can remain in place and be used to PLC programmer's advantage.

When You *DO NOT* Clear Connections
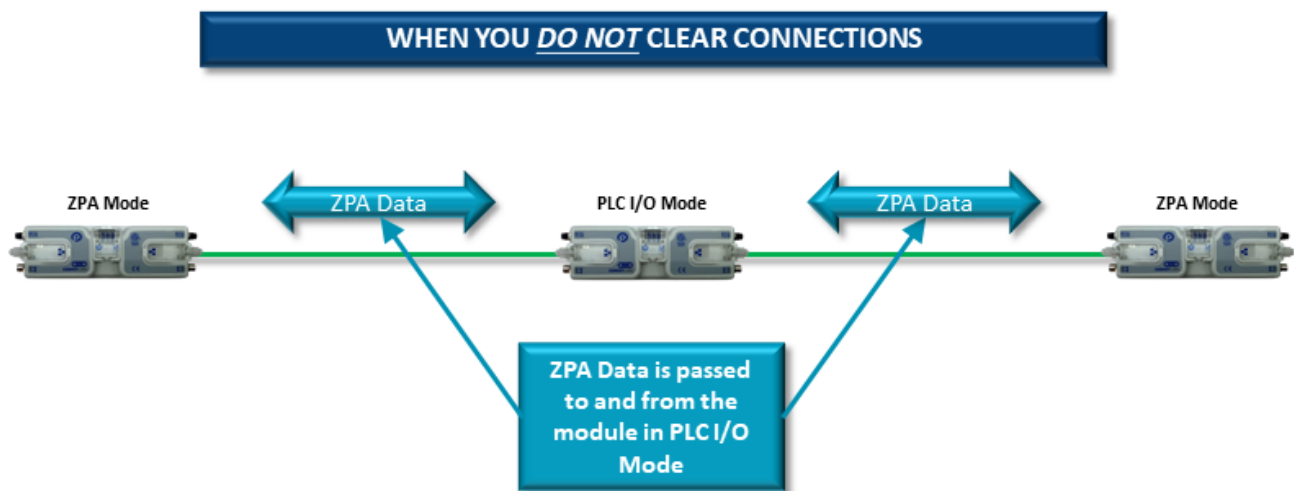
By NOT clearing the connections, the module in PLC I/O will maintain its inter-module data exchange. This could be advantageous for applications where you want to control a specialized conveyor section such as a right angle transfer or merge conveyor with a ConveyLinx-Ai Family Controllers in PLC I/O mode. For example, if the PLC I/O mode module is being either fed or feeds conveyors controlled by modules in standard ZPA mode, these modules will populate PLC I/O configured module's registers with their respective status data. Likewise, the PLC can manipulate the PLC I/O configured module's zone status registers and these registers will automatically be written to the adjacent modules by virtue of these already established connections without requiring the PLC to explicitly perform the messaging.

When You *DO* Clear Connections
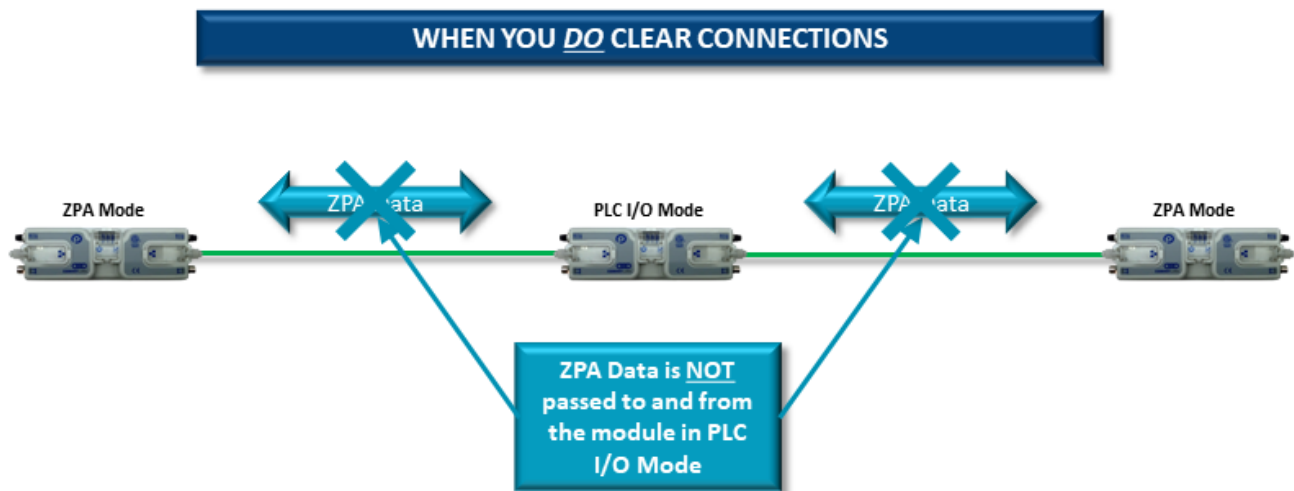
If you choose the option to "Clear Connections", this automatic data transfer of status is inhibited. This means that for a module in PLC I/O mode whose connections have been cleared; its status registers are not automatically written to its adjacent neighbors. Clearing the connections could be advantageous when utilizing several modules in a row configured as PLC I/O where having this additional inter-module communication is not required and would only add to unnecessary communication bandwidth usage.

11.4.3. Configuring Action for Loss of Communication

When changing the mode of a given ConveyLinx-Ai Family Controllers to PLC I/O mode in *EasyRoll*, you are given the option to select the behavior of the ConveyLinx-Ai Family Controllers's outputs upon loss of communications with the PLC.

Select "Don't Change" if you want module's logical outputs and MDR's to remain in the state they were in at the time of the communication loss

Select "All OFF" to instruct the ConveyLinx-Ai Family Controllers to turn off all logical outputs and stop all MDR's at the time of communication loss.

Upon re-establishing communications with the PLC, the ConveyLinx-Ai Family Controllers will automatically resume having its outputs and MDR's controlled by PLC command
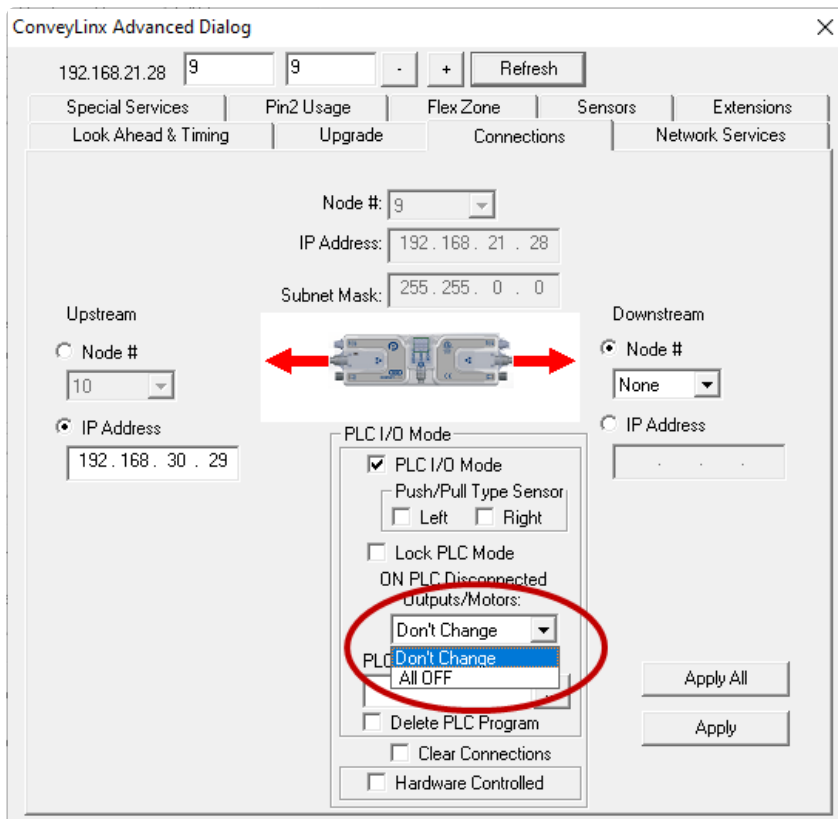
11.4.4. PLC Inputs for PLC I/O Mode

These Topics show the data coming from the ConveyLinx-Ai Family Controllers and provided as input to the PLC:

ConveyStop Status

Sensor & Control Ports

Left Motor Status

Right Motor Status

Motor Ports Digital Status

Upstream / Downstream Status & Tracking

Servo Control Status

11.4.4.1. ConveyStop Status

Register Name / Module Address	Assembled Address for PLC	Description

ConveyStop Status Word 4:0019	M: 4:1700
	E: I.Data [0]
	P: Byte 0 (Hi)
	    Byte 1 (Lo)
		Bitwise Value – Read Only:
		   bit 00 – 04 Reserved
		   bit 05 = Stop active on another module in Stop Group
		   bit 06 = Stop active due to lost communication connection
		   bit 07 = Stop active due to lost PLC connection
		   bit 08 = Reserved
		   bit 09 = Reserved
		   bit 10 = Stop active due to Stop Command from PLC
		   bit 11 – 15 Reserved

11.4.4.2. Sensor Ports

Register Name / Module Address	Assembled Address for PLC	Description
Sensor Port Inputs 4:0035	M: 4:1701 E: I.Data [1] P: Byte 2 (Hi) Byte 3 (Lo)	<u>Bitwise Value – Read Only</u> bit 00 = Left Sensor Port – Aux I/O (M8 Pin 2) bit 02 = Right Sensor Port – Aux I/O (M8 Pin 2) bit 04 = Left Sensor Port – Sensor Signal (M8 Pin 4) bit 06 = Right Sensor Port – Sensor Signal (M8 Pin 4) bit 15 = 2 sec on / 2 sec off heartbeat All other bits Reserved
Sensor Detect 4:0036	M: 4:1702 E: I.Data [2] P: Byte 4 (Hi) Byte 5 (Lo)	<u>Bitwise Value – Read Only</u> bit 0 = Device is connected to Right Sensor Port bit 1 = Device is connected to Left Sensor Port
Module Voltage 4:0024	M: 4:1703 E: I.Data [3] P: Byte 6 (Hi) Byte 7 (Lo)	Value in mV of Module Power Supply Range: 0 to 35000 *Example: 23500 = 23.5 Volts*

✱ The electrical logic state of the signal on the Sensor/Control Port pins are bit-wise exclusive OR (XOR) with the bits set in the *Sensor Port Input Signal Condition Mask* register to arrive at the bit values seen in the *Sensor Port Inputs* register. This allows the PLC programmer to control whether an electrically energized condition results in a logical 1 or a logical 0 in the *Sensor Port Inputs* register. This same relationship is also mirrored by the state of the input's corresponding LED indicator

11.4.4.3. Left Motor Status

Register Name / Module Address	Assembled Address for PLC	Description
Left Motor Current 4:0055	M: 4:1704 E: I.Data [4] P: Byte 8 (Hi) Byte 9 (Lo)	Integer Value in mA for the current the motor is drawing *Example: 1900 = 1.9 Amps*
Left Motor Frequency 4:0056	M: 4:1705 E: I.Data [5] P: Byte 10 (Hi) Byte 11 (Lo)	Integer value in Hz for the electrical frequency the motor is running *Example: 300 = 300 Hz*
Left Motor Temperature 4:0057	M: 4:1706 E: I.Data [6] P: Byte 12 (Hi) Byte 13 (Lo)	High Byte / Low Byte of temperatures in °C High Byte = Motor temperature Low Byte = Module temperature
Left Motor Status 4:0058	M: 4:1707 E: I.Data [7] P: Byte 14 (Hi) Byte 15 (Lo)	Bitwise Value – Read Only: bit 0 = Motor Rotation Status bit 1 = Motor Rotation Status bit 2 = Port in Digital Mode bit 3 = Reserved bit 4 = Reserved bit 5 = Board Overheat bit 6 = Over-voltage bit 7 = Low Voltage bit 8 = Over-heated bit 9 = Over-current bit 10 = Short Circuit bit 11 = Motor Not Connected bit 12 = Overloaded bit 13 = Stalled bit 14 = Sensor Error bit 15 = Motor Not Used

Left Motor Real Speed 4:0507	M: 4:1723
	E: I.Data [23]
	P: Byte 46 (Hi)
	Byte 47 (Lo)
		Motor Speed and Speed Status
		bit 00 thru bit 13 = Numerical value or current speed
		   * For MDR value is in mm/sec
		   * For PGD value is in RPM x 10
		bit 14: 1 = Set Speed is above motor's maximum speed capability

		bit 15: 1 = Set Speed is below motor's minimum speed capability

Motor Rotation Status

Bits 0 and 1 are used in combination to provide 4 possible states. The following chart defines the bit values for these states:

Bit 1	Bit 0	Description
0	0	Motor not running, Normal or Servo Braking Mode applied
0	1	Motor running in CCW Direction
1	0	Motor running in CW Direction
1	1	Motor not running and Free Braking Mode applied

Motor Frequency

This contains the current value of the electrical frequency of the motor if it is running. For Senergy motors, electrical frequency divided by 5 = mechanical frequency. The mechanical frequency is the number of motor rotations per second. You can use this value along with the mechanical gear ratio of the gearbox to calculate the actual running motor RPM. Please consult your Pulseroller catalog for the gear ratio value for the particular part number you are using.

Motor Real Speed

The Ai motor technology encodes the specific motor data into a memory chip inside the motor. This data is read by the ConveyLinx-Ai Family Controllers. Included in this data are the mechanical characteristics for the motor including the minimum and maximum speed settings. You can enter values in the speed reference registers that are outside the allowable range for the connected motor. If you enter a value greater than the maximum; the speed will be set to the maximum. If you enter a value lower than the minimum, the speed will be set to the minimum.

For the *Motor Real Speed* input registers, you can know if the value entered in the Speed Reference register is outside the allowable range by monitoring bits 14 and 15 of these registers.
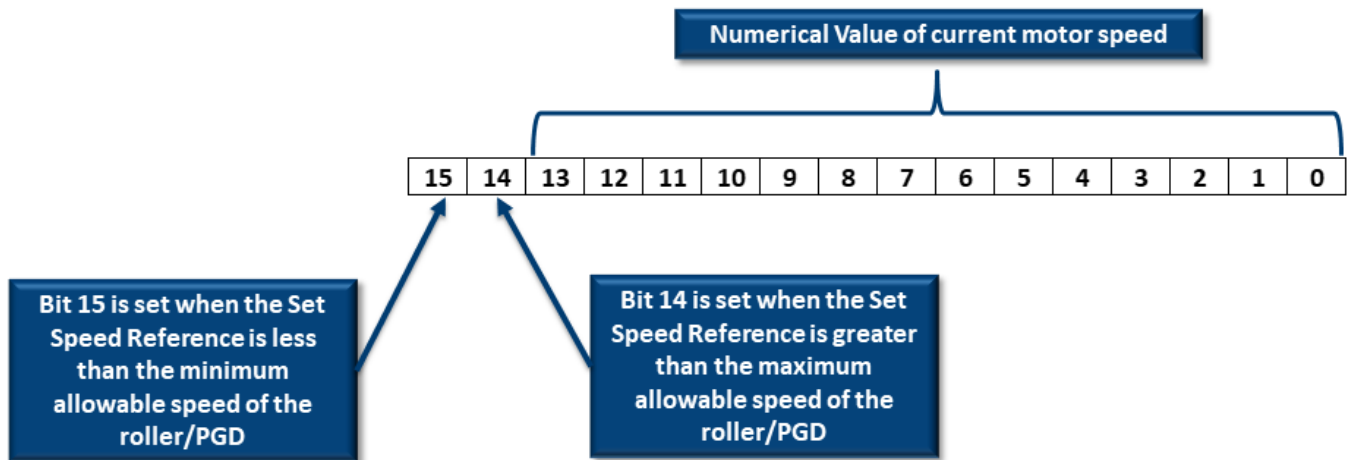
An Example

For example, let's say we have an MDR whose maximum speed is 1.34 m/s. At this speed, the value in the Real Speed input register will show a decimal value very close or equal to 1340 (which is the speed in mm/s).

0000010100111100 = 1340

If we enter a value of 2.0 m/s for the speed reference (value of 2000 in *Speed Reference* register) and then run the motor, in the *Real Speed* input register for that motor you will see a decimal value very close or equal to 17724. Analyzing this value at the bit level, you will see that *bit 14 is set* indicating that the *Speed Reference* value is above the maximum value.

10001010011100 = 17724

If you strip off bit 14 from this value, the remaining decimal value (i.e. the binary value of bits

0 thru 13) will be very close or equal to 1340; which is the maximum speed allowed for the MDR. Similarly, if the *Speed Reference* value entered is below the minimum value, bit 15 will be set in the motor's *Real Speed* register. Stripping out bit 15, the binary value of bits 0 thru 13 will indicate a value very close or equal to the motor's minimum allowed speed.

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	0	0	1	1	1	1	0	0

XX00010100111100 = 1340

11.4.4.4. Right Motor Status

Register Name / Module Address	Assembled Address for PLC	Description
Right Motor Current 4:0079	M: 4:1708 E: I.Data [8] P: Byte 16 (Hi) Byte 17 (Lo)	Integer Value in mA for the current the motor is drawing *Example: 1900 = 1.9 Amps*
Right Motor Frequency 4:0080	M: 4:1709 E: I.Data [9] P: Byte 18 (Hi) Byte 19 (Lo)	Integer value in Hz for the electrical frequency the motor is running *Example: 300 = 300 Hz*
Right Motor Temperature 4:0081	M: 4:1710 E: I.Data [10] P: Byte 20 (Hi) Byte 21 (Lo)	High Byte / Low Byte of temperatures in °C High Byte = Motor temperature Low Byte = Module temperature

Right Motor Status 4:0082	M: 4:1711 E: O.Data [11] P: Byte 22 (Hi) Byte 23 (Lo)	Bitwise Value – Read Only:
		    bit 0 = Motor Rotation Status
		    bit 1 = Motor Rotation Status
		    bit 2 = Port in Digital Mode
		    bit 3 = Reserved
		    bit 4 = Reserved
		    bit 5 = Board Overheat
		    bit 6 = Over-voltage
		    bit 7 = Low Voltage
		    bit 8 = Over-heated
		    bit 9 = Over-current
		    bit 10 = Short Circuit
		    bit 11 = Motor Not Connected
		    bit 12 = Overloaded
		    bit 13 = Stalled
		    bit 14 = Sensor Error
		    bit 15 = Motor Not Used

Right Motor Real Speed 4:0508	M: 4:1724 E: I.Data [24] P: Byte 48 (Hi) Byte 49 (Lo)	Motor Speed and Speed Status
		    bit 00 thru bit 13 = Numerical value or current speed
		        * For MDR value is in mm/sec
		        * For PGD value is in RPM x 10
		    bit 14: 1 = Set Speed is above motor's maximum speed capability
		    bit 15: 1 = Set Speed is below motor's minimum speed

		capability

Motor Rotation Status

Motor Frequency

Motor Real Speed

11.4.4.5. Motor Ports Digital Status

Register Name / Module Address	Assembled Address for PLC	Description
Left Motor Port Digital I/O Status 4:0060	M: 4:1712 E: I.Data [12] P: Byte 24 (Hi) Byte 25 (Lo)	Bitwise Value – Read Only: bit 12 = Short Circuit Error on one or more outputs bit 14 = Over Current – More than 1A detected on one or more outputs
Right Motor Port Digital I/O Status 4:0084	M: 4:1713 E: I.Data [13] P: Byte 26 (Hi) Byte 27 (Lo)	

How to Use Motor Ports as Digital Outputs

11.4.4.6. Upstream / Downstream Status & Tracking

Register Name / Module Address	Assembled Address for PLC	Description
Upstream Module Status 4:0134	M: 4:1714 E: I.Data [14] P: Byte 28 (Hi) Byte 29 (Lo)	<u>Interger Value of Low or High Byte:</u> 0×01 = Zone sensor clear and motor stopped 0×02 = Zone sensor clear, motor running, accepting from upstream zone 0×04 = Zone sensor blocked, motor running, discharging to downstream zone 0×05 = Zone sensor blocked and motor stopped 0×06 = Busy (state during ConveyStop active mode)
Downstream Module Status 4:0232	M: 4:1715 E: I.Data [15] P: Byte 30 (Hi) Byte 31 (Lo)	
Current Tracking Word 1 for Adjacent Upstream Module 4:0139	M: 4:1716 E: I.Data [16] P: Byte 32 (Hi) Byte 33 (Lo)	Value = Tracking data word #1 (16-bit integer) for the Carton that has just discharged from the module adjacent to this local module
Current Tracking Word 2 for Adjacent Upstream Module 4:0140	M: 4:1717 E: I.Data [17] P: Byte 34 (Hi) Byte 35 (Lo)	Value = Tracking data word #2 (16-bit integer) for the Carton that has just discharged from the module adjacent to this local module
Reserved	M: 4:1718 E: I.Data [18] P: Byte 36 (Hi) Byte 37 (Lo)	Used for Reset Protection connection See topic Assemblies with Reset Protection for details

Understanding Adjacent Modules

These registers only contain meaningful data if the ConveyLinx connections between upstream and/or downstream modules are preserved when placing the local module into PLC I/O mode from within *EasyRoll*. If connections are cleared in *EasyRoll*, these registers will not contain any pertinent data and will not be updated by adjacent modules. Learn about clearing connections.

11.4.4.7. Servo Control Status

Register Name / Module Address	Assembled Address for PLC	Description
Left Motor Servo Position 4:0062	M: 4:1719 E: I.Data [19] P: Byte 38 (Hi) Byte 39 (Lo)	Signed integer value that indicates the current position of the Left Motor in relation to its "0" position
Right Motor Servo Position 4:0086	M: 4:1720 E: I.Data [20] P: Byte 40 (Hi) Byte 41 (Lo)	Signed integer value that indicates the current position of the Right Motor in relation to its "0" position
Left Motor Servo Status 4:0011	M: 4:1721 E: I.Data [21] P: Byte 42 (Hi) Byte 43 (Lo)	Bit 0: Servo Command Status 1 = Last Servo Run Command Complete 0 = Servo Command in Process Bit 1: Servo Reset Status Echoes state of Motor Servo Command bit 0
Right Motor Servo Status 4:0016	M: 4:1722 E: I.Data [22] P: Byte 44 (Hi) Byte 45 (Lo)	Bit 2: Servo Command Status Echoes state of Motor Servo Command bit 1

See Servo Move example

11.4.5. PLC Outputs for PLC I/O Mode

11.4.5.1. ConveyStop Command & Clear Motor Error

Register Name / Module Address	Assembled Address for PLC	Description

ConveyStop Command Word 4:0020	M: 4:1800
	E: O.Data [0]
	P: Byte 0 (Hi)
	Byte 1 (Lo)
		Integer Value
		  0= No Command
		  1 = Command local module's Stop Group to
		     go to Stopped State
		  2 = Command local module's Stop Group to
		     Clear Stopped State

Clear Motor Error 4:0022	M: 4:1816
	E: O.Data [16]
	P: Byte 32 (Hi)
	Byte 33 (Lo)
		Logical 1 or 0:
		  1 = send reset command
		  0 = clear reset command

Learn more about Clear Motor Error

11.4.5.2. Motor & Sensor Port Digital Output

Register Name / Module Address	Assembled Address for PLC	Description

Set Left Motor Port Digital Control 4:0060	M: 4:1801 E: O.Data [1] P: Byte 2 (Hi) Byte 3 (Lo)	Bitwise values
		   bit 00 = Reserved
		   bit 01 = Energize Motor Port Pin 4
		   bit 02 = Energize Motor Port Pin 3
		   bit 03 thru bit 14 = Reserved
		   bit 15 = Digital Output Enable
		        0 = Use Port as Motor Control
		        1 = Use Port as Digital Outputs

Set Right Motor Port Digital Control 4:0084	M: 4:1802 E: O.Data [2] P: Byte 4 (Hi) Byte 5 (Lo)	Bitwise values
		   bit 00 = Reserved
		   bit 01 = Energize Motor Port Pin 2
		   bit 02 = Energize Motor Port Pin 3
		   bit 03 thru bit 14 = Reserved
		   bit 15 = Digital Output Enable
		        0 = Use Port as Motor Control
		        1 = Use Port as Digital Outputs

Sensor Port Digital Output Control 4:0037	M: 4:1803 E: O.Data [3] P: Byte 6 (Hi) Byte 7 (Lo)	Bitwise Values
		   bit 00: 1 = Left Pin 2 ON, 0 = Left Pin 2 OFF
		   bit 01: 1 = Right Pin 2 ON, 0 = Right Pin 2 OFF
		   bit 02 thru bit 04 = Reserved
		   bit 05: 1 = Enable Left Pin 2 as Output, 0 = Enable Left Pin 2 as Input
		   bit 06: 1 = Enable Right Pin 2 as Output, 0 = Enable Right Pin 2 as Input
		   All other bits reserved

Learn more about using Motor Ports for Digital Outputs

11.4.5.3. Left Motor Control

Register Name / Module Address	Assembled Address for PLC	Description

Left Motor Run / Reverse
4:0260
	M: 4:1804
	E: O.Data [4]
	P: Byte 8 (Hi)
	Byte 9 (Lo)
		Bit 0:
		   1 = Run Command
		   0 = Stop Command
		Bit 8:
		   0 = Run in Configured Direction
		   1 = Run opposite of Configured Direction

Left Motor Brake Method
4:0261
	M: 4:1805
	E: O.Data [5]
	P: Byte 10 (Hi)
	Byte 11 (Lo)
		Integer Value:
		   1 = Use Standard Brake Method
		   2 = Use Free Coast Brake Method
		   3 = Use Servo Brake Method
		   4 = Use Continuous Torque
		   0 = Remain at last configured or last value entered

Left Motor Slave Mode
4:0262
	M: 4:1806
	E: O.Data [6]
	P: Byte 12 (Hi)
	Byte 13 (Lo)
		Integer Value:
		   0 = Ignore
		   1 = Slave Mode OFF – Left motor independently controlled from Right
		   2 = Slave Mode ON – Left motor mirrors Right motor control
		   3 = Slave Mode ON – Left motor runs in opposite direction of Right motor

Left Motor Speed Reference
4:0040
	M: 4:1810
	E: O.Data [10]
	P: Byte 20 (Hi)
	Byte 21 (Lo)
		Integer Value
		   For MDR value is in mm/s
		  For PGD value is in RPM X 10
		   0 = Remain at last non zero value entered

Left Motor Acceleration
4:0043
	M: 4:1812
	E: O.Data [12]
	P: Byte 24 (Hi)
	Byte 25 (Lo)
		Integer Value
		   For MDR value is in mm
		   For PGD value is in motor pulses

Left Motor Deceleration 4:0044	M: 4:1813 E: O.Data [13] P: Byte 26 (Hi) Byte 27 (Lo)	Deceleration Range: 0 to 10000 (Both MDR and PGD) Acceleration Range: 30 to 10000 (Both MDR and PGD) 0 = Remain at last non-zero value entered

11.4.5.4. Right Motor Control

Register Name / Module Address	Assembled Address for PLC	Description
Right Motor Run / Reverse 4:0270	M: 4:1807 E: O.Data [7] P: Byte 14 (Hi) 　Byte 15 (Lo)	Bit 0: 　1 = Run Command 　0 = Stop Command Bit 8: 　0 = Run in Configured Direction 　1 = Run opposite of Configured Direction
Right Motor Brake Method 4:0271	M: 4:1808 E: O.Data [8] P: Byte 16 (Hi) 　Byte 17 (Lo)	Integer Value: 　1 = Use Standard Brake Method 　2 = Use Free Coast Brake Method 　3 = Use Servo Brake Method 　4 = Use Continuous Torque 　0 = Remain at last configured or last value entered
Right Motor Slave Mode 4:0272	M: 4:1809 E: O.Data [9] P: Byte 18 (Hi) 　Byte 19 (Lo)	Integer Value: 　0 = Ignore 　1 = Slave Mode OFF – Right motor independently controlled from Left 　2 = Slave Mode ON – Right motor mirrors Left motor control 　3 = Slave Mode ON – Right motor runs in opposite direction of Left motor
Right Motor Speed Reference 4:0064	M: 4:1811 E: O.Data [11] P: Byte 22 (Hi) 　Byte 23 (Lo)	Integer Value 　For MDR value is in mm/s 　For PGD value is in RPM X 10 　0 = Remain at last non zero value entered
Right Motor Acceleration 4:0067	M: 4:1814 E: O.Data [14] P: Byte 28 (Hi) 　Byte 29 (Lo)	Integer Value 　For MDR value is in mm 　For PGD value is in motor pulses Deceleration Range: 0 to 10000 (Both MDR and PGD) Acceleration Range: 30 to 10000 (Both MDR and PGD) 　0 = Remain at last non-zero value entered
Right Motor Deceleration 4:0068	M: 4:1815 E: O.Data [15] P: Byte 30 (Hi) 　Byte 31 (Lo)	

11.4.5.5. Set Status & Tracking

Register Name / Module Address	Assembled Address for PLC	Description
Set Status to Downstream Module 4:0196	M: 4:1817 E: O.Data [17] P: Byte 34 (Hi) Byte 35 (Lo)	Used to write this module's ZPA Status to its neighboring Downstream module: 4 = Instruct Downstream module to "wake-up" and run its most upstream zone 1 = Instructs Downstream module that carton has exited local zone and to accept any tracking data written in Set Discharge Tracking Word 1 / Word 2 registers when carton arrives
Set Status to Upstream Module 4:0116	M: 4:1818 E: O.Data [18] P: Byte 36 (Hi) Byte 37 (Lo)	Used to write this module's ZPA Status to its neighboring Upstream module: 5 = Instructs Upstream module's discharge zone to accumulate and hold any carton that arrives at its discharge zone. 1 = Instructs Upstream module's discharge zone to release any carton that arrives at its discharge zone
Set Discharge Tracking Word 1 4:0201	M: 4:1820 E: O.Data [20] P: Byte 40 (Hi) Byte 41 (Lo)	Only used when local PLC I/O Mode module needs to pass tracking data to a downstream connected ConveyLinx-Ai Family Controllers. Used in conjunction with Set Status to Downstream Module register
Set Discharge Tracking Word 2 4:0202	M: 4:1821 E: O.Data [21] P: Byte 42 (Hi) Byte 43 (Lo)	
Reserved	M: 4:1822 E: O.Data [22] P: Byte 44 (Hi) Byte 45 (Lo)	

✳ For registers involving status and tracking to upstream or downstream modules that are in ZPA mode, the connections must NOT be cleared when placing this module into PLC I/O mode from within *EasyRoll*. Learn more about Clearing Connection Options

11.4.5.6. Set Sensor Port Input Mask

Register Name / Module Address	Assembled Address for PLC	Description

Sensor Port Input Signal Condition Mask 4:0034	M: 4:1819 E: O.Data [19] P: Byte 38 (Hi) Byte 39 (Lo)	<u>Bitwise Value:</u>
		   bit 00 = Left Sensor Port – Aux I/O (M8 Pin 2)
		   bit 02 = Right Sensor Port – Aux I/O (M8 Pin 2)
		   bit 04 = Left Sensor Port – Sensor Signal (M8 Pin 4)
		   bit 06 = Right Sensor Port – Sensor Signal (M8 Pin 4)
		   All other bits reserved

How Input signals are conditioned in ZPA Mode

For a module in standard ZPA mode, the *Auto-Configuration* procedure sets values in this register to allow the module to correctly display the Sensor Port's LEDs to facilitate diagnostics. This is done, for example, to make visual LED diagnostics the same for "zone blocked" regardless of the sensor type.

For example, suppose the zone photo sensors used are "Light Operate, Normally Open". This means that the sensor's output is energizing the Sensor Port input pin 4 when the zone is clear. The module's Sensor Port LED indicator for pin 4 (green) should illuminate when the zone is blocked; so the *Auto-Configuration* procedure sets a bit in the *Sensor Port Input Signal Condition Mask* register to correspond to the pin 4 signal on the appropriate sensor port. If the sensor is electrically opposite such that its output energizes pin 4 of the sensor port when the zone is blocked, then the bit corresponding to pin 4 for this sensor port is clear such that the sensor port's LED illuminates green when pin 4 is energized.

How Input signals are conditioned in PLC I/O Mode

When a module is placed in PLC I/O mode; the *Sensor Port Input Signal Condition Mask* register is cleared of the values set during the *Auto-Configure* procedure. The *Sensor Port Input Signal Condition Mask* register is made available for PLC I/O mode Sensor Port inputs to give the PLC programmer the same flexibility for configuring which electrical state (on or off) of the input will cause a logical 1 to appear in the *Sensor Port Inputs* register and illuminate the pin's corresponding LED. By setting or clearing the corresponding bit for a given port's pin 3 or 4 signal, the PLC programmer can determine which physical state (on or off) of the input signal will cause its corresponding pin's bit in the Sensor Port Inputs register to be set and its corresponding LED to illuminate. The following are two examples that show the bit patterns and signals for one of the module's inputs. The same pattern applies to all available inputs:

Example: Right Sensor Port – Pin 4 Signal

Electrical Signal	*Sensor Port Input Signal Condition Mask* Register bit 6	*Sensor Port Inputs* Register bit 6	LED State
OFF	0	0	Green = OFF
ON	0	1	Green = ON
OFF	1	1	Green = ON
ON	1	0	Green = OFF

Example: Left Sensor Port – Pin 2 Signal			
Electrical Signal	*Sensor Port Input Signal Condition Mask* Register bit 0	*Sensor Port Inputs* Register bit 0	LED State
OFF	0	0	Red = OFF
ON	0	1	Red = ON
OFF	1	1	Red = ON
ON	1	0	Red = OFF

> **!** Be careful when changing the *Sensor Port Input Signal Condition Mask* in your Ethernet I/P PLC program. The input bit values in the *Sensor Port Inputs* register can show unexpected or opposite values from expected until the PLC has updated the *Sensor Port Input Signal Condition Mask* data. This update could take several program scans depending upon the Ethernet I/P RPI settings. Take care to be sure that the *Sensor Port Input Signal Condition Mask* data is written to the module before acting on any input values in the *Sensor Port Inputs* register

11.4.5.7. Servo Control

Register Name / Module Address	Assembled Address for PLC	Description
Left Motor Servo Command Word 4:0009	M: 4:1824 E: O.Data [24] P: Byte 48 (Hi) Byte 49 (Lo)	Bit 0: Reset Command 1 = Set Current Servo Position Value to "0" Bit 1: Servo Run Command 1 = Run in Motor from current position value to set position value in Left Motor Servo Command Value Register
Right Motor Servo Command Word 4:0014	M: 4:1826 E: O.Data [26] P: Byte 52 (Hi) Byte 53 (Lo)	
Left Motor Servo Command Value 4:0008	M: 4:1823 E: O.Data [23] P: Byte 46 (Hi) Byte 47 (Lo)	Signed integer value of the position to move to on the next Servo Run Command Valid values are from -32767 to +32767 Value in mm for MDR and in pulses for PGD
Right Motor Servo Command Value 4:0013	M: 4:1825 E: O.Data [25] P: Byte 50 (Hi) Byte 51 (Lo)	

> **!** For the ConveyLinx-Ai2 module, Servo Motor Control requires firmware 4.14 or later

> ***** The Servo Motor Control function utilizes the existing motor speed, acceleration, and deceleration registers. The existing starting, stopping, and direction control registers are not used and these functions are incorporated into the Servo Control registers

11.4.5.7.1. Servo Control Example

For our example, we want to perform the following cycle using PLC control of the Left MDR on an ConveyLinx-Ai Family Controllers:

- Establish a zero or home position by external input to the module form a PLC (sensor or operator button)
- Rotate in the CCW direction for 7000 mm at a speed of 0.8 m/s with acceleration ramp of 100 mm and a deceleration ramp of 50 mm
- Rotate the CW direction for 9000 mm at a speed of 1.0 m/s with acceleration ramp of 50 mm and a deceleration ramp of 10 mm.
- Rotate CCW back to the zero or home position at a speed of 0.8 m/s with acceleration ramp of 50 mm and a deceleration ramp of 10 mm.
- Wait for a cycle dwell time of 4 seconds and then repeat the rotation cycles

> ❗ Rotation from 0 to a positive value always causes rotation in CCW direction and rotation from 0 to a negative value always causes rotation in CW direction. Servo Motor Control ignores the CW/CCW direction setting in EasyRoll and/or any configured or default direction that may exist in the ConveyLinx-Ai Family Controllers's configuration

Define PLC Tags

Tag Name	Data Type	Assembly Register (/bit)
ZERO	Boolean	Left Servo Command Word – bit 0
ZERO_ACK	Boolean	Left Motor Servo Status – bit 1
RUN	Boolean	Left Servo Command Word – bit 1
READY	Boolean	Left Motor Servo Status – bit 0
POSITION	Integer	Left Servo Command Pulses
SPEED	Integer	Left Motor Speed Reference
ACCEL	Integer	Left Motor Acceleration Ramp
DECEL	Integer	Left Motor Deceleration Ramp

Sequence of Operation

Step #1: Upon external signal from sensor or button, set *ZERO* bit to establish zero or "home" position. When PLC sees that the *ZERO_ACK* bit is set, then the PLC resets the *ZERO* bit.

Step #2: To make the first rotation, we need write the speed, ramp values, and distance to rotate to the appropriate registers:

- Write 800 to *SPEED*
- Write 100 to *ACCEL*
- Write 50 to *DECEL*
- Write 7000 to *POSITION*

Step #3: If *READY* is reset, then the PLC can set the *RUN* bit to begin the rotation. When the rotation is complete, the module sets the *READY* bit. This will be the signal to the PLC to reset the *RUN* bit. Once the module sees that the *RUN* bit has been reset, it will reset the *READY* bit.

Step #4: For the second rotation, we need to write the speed and ramp values to the appropriate registers:

- Write 1000 to SPEED
- Write 50 to ACCEL
- Write 10 to DECEL

Because we want to rotate in the opposite direction, we need to determine the new location based upon the zero or "home" position. In this case, we know we went 7000 mm "forward" and we want to move 9000 mm "backward". The position we want to end up is 7000-9000 = -2000.

- Write -2000 to POSITION

Step #5: Repeat **Step #4**

Step #6: For the 3rd rotation, we keep the ramp values from the 2nd rotation, but we need to set the speed and the position to rotate. In this case we want to go to the zero or "home position.

- Write 800 to SPEED
- Write 0 to POSITION

Step #7: Repeat **Step #4**

PLC Ladder Diagram

0

ZERO THE LEFT MOTOR POSITION FROM EXTERNAL SIGNAL

READY
External_Reset   <ERSC_1:I.Data[21].0>

ZERO
<ERSC_1:O.Data[24].0>   External_Reset

1

ZERO   ZERO_ACK
<ERSC_1:O.Data[24].0>   <ERSC_1:I.Data[21].1>

ZERO
<ERSC_1:O.Data[24].0>   Zero_Complete

PERFORM FIRST ROTATION

READY
Zero_Complete   Cycle_Step_1   Cycle_Step_2   Cycle_Step_3   Cycle_Step_4   <ERSC_1:I.Data[21].0>

2

MOV
Move
Source   7000

Dest   POSITION
<ERSC_1:O.Data[23]>
-2000

MOV
Move
Source   800

Dest   SPEED
<ERSC_1:O.Data[10]>
1000

MOV
Move
Source   100

Dest   ACCEL
<ERSC_1:O.Data[12]>
50

MOV
Move
Source   50

Dest   DECEL
<ERSC_1:O.Data[13]>
10

RUN
<ERSC_1:O.Data[24].1>

Cycle_Step_1

ACKNOWLEDGE FIRST ROTATION AS COMPLETE

3

Zero_Complete Cycle_Step_1 RUN <ERSC_1:O.Data[24].1> READY <ERSC_1:I.Data[21].0> Cycle_Step_1 (U) RUN <ERSC_1:O.Data[24].1> (U) Cycle_Step_2 (L)

PERFORM 2nd ROTATION

4

Zero_Complete Cycle_Step_2 READY <ERSC_1:I.Data[21].0>

MOV
Move Source   -2000
Dest   POSITION
    <ERSC_1:O.Data[23]>
                   -2000◦

MOV
Move Source   1000
Dest   SPEED
    <ERSC_1:O.Data[10]>
                   1000◦

MOV
Move Source   50
Dest   ACCEL
    <ERSC_1:O.Data[12]>
                     50◦

MOV
Move Source   10
Dest   DECEL
    <ERSC_1:O.Data[13]>
                     10◦

RUN <ERSC_1:O.Data[24].1> (L)

ACKNOWLEDGE 2ND ROTATION AS COMPLETE

5

Zero_Complete Cycle_Step_2 RUN <ERSC_1:O.Data[24].1> READY <ERSC_1:I.Data[21].0> Cycle_Step_2 (U) Cycle_Step_3 (L) RUN <ERSC_1:O.Data[24].1> (U)

PERFORM 3rd ROTATION

6

Zero_Complete Cycle_Step_3 READY <ERSC_1:I.Data[21].0>

MOV
Move Source   0
Dest   POSITION
    <ERSC_1:O.Data[23]>
                   -2000◦

MOV
Move Source   800
Dest   SPEED
    <ERSC_1:O.Data[10]>
                   1000◦

MOV
Move Source   50
Dest   ACCEL
    <ERSC_1:O.Data[12]>
                     50◦

MOV
Move Source   10
Dest   DECEL
    <ERSC_1:O.Data[13]>
                     10◦

RUN <ERSC_1:O.Data[24].1> (L)

ACKNOWLEDGE 3RD ROTATION AS COMPLETE

7
Zero_Complete Cycle_Step_3 RUN <ERSC_1:O.Data[24].1> READY <ERSC_1:I.Data[21].0> RUN <ERSC_1:O.Data[24].1> (U) Cycle_Step_3 (U) Cycle_Step_4 (L)

HOLD FOR 4 SECOND DUTY CYCLE

8
Zero_Complete Cycle_Step_4 Cycle_Delay.DN /

TON
Timer On Delay   —(EN)—
Timer   Cycle_Delay
Preset   4000+   —(DN)—
Accum   0+

9
Cycle_Delay.DN   Cycle_Step_4 (U)

EXTERNAL SIGNAL TO STOP CYCLE ALTOGETHER

10
Zero_Complete External_Stop Zero_Complete (U) External_Stop (U) Cycle_Step_1 (U) Cycle_Step_2 (U) Cycle_Step_3 (U) Cycle_Step_4 (U) RUN <ERSC_1:O.Data[24].1> (U)

(End)

11.4.6. Reduced Size PLC I/O Mode Assemblies

For some PLC controllers, the data size footprint required for a given assembly can be a limiting factor on how many devices can connect to a given controller. ConveyLinx includes input/ output assemblies with fewer registers that contain basic functions in applications where the full functionality of the PLC I/O Mode Assemblies is not required. The *Reduced Size Assemblies* provide basic I/O and motor control while removing the interfaces for servo moves, upstream/ downstream data, tracking data, etc.

PLC I/O Mode Inputs to PLC

Module Address	Register Name	Assembled Modbus Address	Assembled EIP Address	Assembled Profinet Address
4:0035	Sensor & Control Port Inputs	4:2700	I.Data [0]	Byte 0 (Hi) Byte 1 (Lo)
4:0036	Sensor Detect	4:2701	I.Data [1]	Byte 2 (Hi) Byte 3 (Lo)
4:0057	Left Motor Temperature	4:2702	I.Data [2]	Byte 4 (Hi) Byte 5 (Lo)
4:0058	Left Motor Status	4:2703	I.Data [3]	Byte 6 (Hi) Byte 7 (Lo)
4:0081	Right Motor Temperature	4:2704	I.Data [4]	Byte 8 (Hi) Byte 9 (Lo)
4:0082	Right Motor Status	4:2705	I.Data [5]	Byte 10 (Hi) Byte 11 (Lo)
4:0060	Left Motor Port Digital I/O Status	4:2706	I.Data [6]	Byte 12 (Hi) Byte 13 (Lo)
4:0084	Right Motor Port Digital I/O Status	4:2707	I.Data [7]	Byte 14 (Hi) Byte 15 (Lo)
	Reserved	4:2708	I.Data [8]	Byte 16 (Hi) Byte 17 (Lo)

PLC I/O Mode Outputs from PLC

Module Address	Register Name	Assembled Modbus Address	Assembled EIP Address	Assembled Profinet Address
4:0060	Set Left Motor Port Digital Control	4:2800	O.Data [0]	Byte 0 (Hi) Byte 1 (Lo)
4:0084	Set Right Motor Port Digital Control	4:2801	O.Data [1]	Byte 2 (Hi) Byte 3 (Lo)
4:0037	Control Port Digital Output Control	4:2802	O.Data [2]	Byte 4 (Hi) Byte 5 (Lo)
4:0260	Left Motor Run / Reverse	4:2803	O.Data [3]	Byte 6 (Hi) Byte 7 (Lo)
4:0270	Right Motor Run / Reverse	4:2804	O.Data [4]	Byte 8 (Hi) Byte 9 (Lo)
4:0040	Left Motor Speed Reference	4:2805	O.Data [5]	Byte 10 (Hi) Byte 11 (Lo)
4:0064	Right Motor Speed Reference	4:2806	O.Data [6]	Byte 12 (Hi) Byte 13 (Lo)
4:0022	Clear Motor Error	4:2807	O.Data [7]	Byte 14 (Hi) Byte 15 (Lo)
	Reserved	4:2808	O.Data [8]	Byte 16 (Hi) Byte 17 (Lo)

11.5. ConveyLogix Interface

ConveyLinx-Ai Family Controllerss in PLC I/O mode allow for download of a user program generated with the ConveyLogix programming tool. There may be instances when you would like for an external PLC to be able to interface with an ConveyLinx-Ai Family Controllers that is running a ConveyLogix program.

> ✳ With ConveyLogix, because a custom user generated program is running in the ConveyLinx-Ai Family Controllers, the previously defined existing Assemblies for ZPA and PLO I/O modes are no longer valid and if used, could produce unexpected results

When using ConveyLogix programming software version 2.2 and later, the environment provides predefined controlled input and output register tags that can be used by the programmer to interface with an external PLC. The Assemblies described in this section are those that the external PLC would connect and use to access the data in the ConveyLogix program.

This data is divided into 16 Integer Input words and 16 Integer output words. It is left to the ConveyLogix programmer and the PLC programmer to utilize as much or as little of these data blocks as needed for their particular application.

11.5.1. ConveyLogix Assembly Inputs to PLC

ConveyLogix Controller Tag	Ethernet I/P Tag	Modbus PLC Holding Register	Profinet I/O Byte Offset
ToPLCArray [0]	E: I.Data [0]	M: 4:13100	P: Byte 0 (Hi) Byte 1 (Lo)
ToPLCArray [1]	E: I.Data [1]	M: 4:13101	P: Byte 2 (Hi) Byte 3 (Lo)
ToPLCArray [2]	E: I.Data [2]	M: 4:13102	P: Byte 4 (Hi) Byte 5 (Lo)
ToPLCArray [3]	E: I.Data [3]	M: 4:13103	P: Byte 6 (Hi) Byte 7 (Lo)
ToPLCArray [4]	E: I.Data [4]	M: 4:13104	P: Byte 8 (Hi) Byte 9 (Lo)
ToPLCArray [5]	E: I.Data [5]	M: 4:13105	P: Byte 10 (Hi) Byte 11 (Lo)
ToPLCArray [6]	E: I.Data [6]	M: 4:13106	P: Byte 12 (Hi) Byte 13 (Lo)
ToPLCArray [7]	E: I.Data [7]	M: 4:13107	P: Byte 14 (Hi) Byte 15 (Lo)
ToPLCArray [8]	E: I.Data [8]	M: 4:13108	P: Byte 16 (Hi) Byte 17 (Lo)
ToPLCArray [9]	E: I.Data [9]	M: 4:13109	P: Byte 18 (Hi) Byte 19 (Lo)
ToPLCArray [10]	E: I.Data [10]	M: 4:13110	P: Byte 20 (Hi) Byte 21 (Lo)
ToPLCArray [11]	E: I.Data [11]	M: 4:13111	P: Byte 22 (Hi) Byte 23 (Lo)
ToPLCArray [12]	E: I.Data [12]	M: 4:13112	P: Byte 24 (Hi) Byte 25 (Lo)
ToPLCArray [13]	E: I.Data [13]	M: 4:13113	P: Byte 26 (Hi) Byte 27 (Lo)
ToPLCArray [14]	E: I.Data [14]	M: 4:13114	P: Byte 28 (Hi) Byte 29 (Lo)
ToPLCArray [15]	E: I.Data [15]	M: 4:13115	P: Byte 30 (Hi) Byte 31 (Lo)

11.5.2. ConveyLogix Assembly Outputs from PLC

ConveyLogix Controller Tag	Ethernet I/P Tag	Modbus PLC Holding Register	Profinet I/O Byte Offset
FromPLCArray [0]	E: O.Data [0]	M: 4:13200	P: Byte 0 (Hi) Byte 1 (Lo)
FromPLCArray [1]	E: O.Data [1]	M: 4:13201	P: Byte 2 (Hi) Byte 3 (Lo)
FromPLCArray [2]	E: O.Data [2]	M: 4:13202	P: Byte 4 (Hi) Byte 5 (Lo)
FromPLCArray [3]	E: O.Data [3]	M: 4:13203	P: Byte 6 (Hi) Byte 7 (Lo)
FromPLCArray [4]	E: O.Data [4]	M: 4:13204	P: Byte 8 (Hi) Byte 9 (Lo)
FromPLCArray [5]	E: O.Data [5]	M: 4:13205	P: Byte 10 (Hi) Byte 11 (Lo)
FromPLCArray [6]	E: O.Data [6]	M: 4:13206	P: Byte 12 (Hi) Byte 13 (Lo)
FromPLCArray [7]	E: O.Data [7]	M: 4:13207	P: Byte 14 (Hi) Byte 15 (Lo)
FromPLCArray [8]	E: O.Data [8]	M: 4:13208	P: Byte 16 (Hi) Byte 17 (Lo)
FromPLCArray [9]	E: O.Data [9]	M: 4:13209	P: Byte 18 (Hi) Byte 19 (Lo)
FromPLCArray [10]	E: O.Data [10]	M: 4:13210	P: Byte 20 (Hi) Byte 21 (Lo)
FromPLCArray [11]	E: O.Data [11]	M: 4:13211	P: Byte 22 (Hi) Byte 23 (Lo)
FromPLCArray [12]	E: O.Data [12]	M: 4:13212	P: Byte 24 (Hi) Byte 25 (Lo)
FromPLCArray [13]	E: O.Data [13]	M: 4:13213	P: Byte 26 (Hi) Byte 27 (Lo)
FromPLCArray [14]	E: O.Data [14]	M: 4:13214	P: Byte 28 (Hi) Byte 29 (Lo)
FromPLCArray [15]	E: O.Data [15]	M: 4:13215	P: Byte 30 (Hi) Byte 31 (Lo)

11.6. Assemblies with Reset Protection

For control system applications where the Ethernet I/P PLC needs to take specific action to recover from a loss of communications due to a module that has had its power cycled off and on; there is an additional set of instances implemented that provides 2 new registers that allows the PLC to manipulate the function of the module for recovery.

When a ConveyLinx-Ai Family Controllers in ZPA Mode has lost power and then is powered back up, due to perhaps a system E-stop that disconnects control power, some of the module's working register values are reset to 0. Among these are the arrival counters, departure counters, and the accumulate commands for each configured zone(s) on the module. Upon power cycle to the ConveyLinx-Ai Family Controllers, if the PLC can establish its full Ethernet I/P connection prior to the ZPA task becoming fully functional, the preceding PLC programming examples would still function as expected. However, because PLC Ethernet I/P connection time is variable and not fixed; a robust control system design cannot count on the PLC establishing Ethernet I/P connection prior to the ConveyLinx-Ai Family Controllers's ZPA task commanding the module as if no PLC was connected. A consequence of this in the previous programming example is that if a load happens to be accumulated in a PLC controlled zone at the time of power loss, upon powering back up, the load can release without the PLC commanding it to do so. This release could be caused by the PLC logic detecting a change in arrival count and thus incrementing the release or it could be caused by the fact that the accumulate command is cleared in the ConveyLinx-Ai Family Controllers due to power cycle and because the PLC has not established communications to set the accumulate command bit, the module releases the zone because there is no command present to accumulate.

For ConveyLinx-Ai Family Controllers's in PLC I/O mode, *Reset Protection* may not be as much a concern, however reset protection assemblies are available for PLC I/O mode. Some items such as current servo position, etc. are reset upon restoration of power and thus the PLC programmer may want to detect this condition and act accordingly.

> ✳ *Reset Protection* assemblies are used for applications where the ConveyLinx-Ai Family Controllers remains in a "hold" state until the PLC has established communications. Otherwise, the register mappings for these assemblies are the same as their non-protected counterparts.

> ❗ Please note that the Modbus TCP starting addresses for each assembly with reset protection is different from their non-protected counterparts.

11.6.1. ZPA Mode Assembly Inputs with Reset Protection

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0116	Local Status Upstream Zone	E: I.Data [0]	M: 4:3500
4:0196	Local Status Downstream Zone	E: I.Data [1]	M: 4:3501
4:0106	Arrival Count Local Upstream Zone	E: I.Data [2]	M: 4:3502
4:0107	Departure Count Local Upstream Zone	E: I.Data [3]	M: 4:3503
4:0186	Arrival Count Local Downstream Zone	E: I.Data [4]	M: 4:3504
4:0187	Departure Count Local Downstream Zone	E: I.Data [5]	M: 4:3505
4:0088	Module Status Word 1	E: I.Data [6]	M: 4:3506
4:0089	Module Status Word 2	E: I.Data [7]	M: 4:3507
4:0119	Current Upstream Zone Tracking Word 1	E: I.Data [8]	M: 4:3508
4:0120	Current Upstream Zone Tracking Word 2	E: I.Data [9]	M: 4:3509
4:0199	Current Downstream Zone Tracking Word 1	E: I.Data [10]	M: 4:3510
4:0200	Current Downstream Zone Tracking Word 2	E: I.Data [11]	M: 4:3511
4:0105	Current Release Count for Upstream Zone	E: I.Data [12]	M: 4:3512
4:0185	Current Release Count for Downstream Zone	E: I.Data [13]	M: 4:3513
4:0201	Get Tracking Forward Direction Word 1	E: I.Data [14]	M: 4:3514
4:0202	Get Tracking Forward Direction Word 2	E: I.Data [15]	M: 4:3515
4:0237	Get Tracking Reverse Direction Word 1	E: I.Data [16]	M: 4:3516
4:0238	Get Tracking Reverse Direction Word 2	E: I.Data [17]	M: 4:3517
4:0035	Sensor & Control Port Inputs	E: I.Data [18]	M: 4:3518
N/A	**Current Module Reset Count**	E: I.Data [19]	M: 4:3519
4:0019	ConveyStop Status	E: I.Data [20]	M: 4:3520

11.6.2. ZPA Mode Assembly Outputs with Reset Protection

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0132	Set Local Upstream Zone Tracking Word 1	E: O.Data [0]	M: 4:3600
4:0133	Set Local Upstream Zone Tracking Word 2	E: O.Data [1]	M: 4:3601
4:0212	Set Local Downstream Zone Tracking Word 1	E: O.Data [2]	M: 4:3602
4:0213	Set Local Downstream Zone Tracking Word 2	E: O.Data [3]	M: 4:3603
4:0104	Accumulation Control for Local Upstream Zone	E: O.Data [4]	M: 4:3604
4:0184	Accumulation Control for Local Downstream Zone	E: O.Data [5]	M: 4:3605
4:0040	Set Left MDR Speed	E: O.Data [6]	M: 4:3606
4:0064	Set Right MDR Speed	E: O.Data [7]	M: 4:3607
4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	E: O.Data [8]	M: 4:3608
4:0185	Release and Accumulate on Next Arrival for Local Upstream Zone	E: O.Data [9]	M: 4:3609
4:0134	Set Status for Upstream Induct	E: O.Data [10]	M: 4:3610
4:0232	Set Status for Downstream Discharge	E: O.Data [11]	M: 4:3611
4:0139	Set Induct Tracking Forward Direction Word 1	E: O.Data [12]	M: 4:3612
4:0140	Set Induct Tracking Forward Direction Word 2	E: O.Data [13]	M: 4:3613

4:0237	Set Induct Tracking Reverse Direction Word 1	E: O.Data [14]	M: 4:3614
4:0238	Set Induct Tracking Reverse Direction Word 2	E: O.Data [15]	M: 4:3615
4:0022	Clear Motor Error	E: O.Data [16]	M: 4:3616
4:0063	Set Control Port Outputs	E: O.Data [17]	M: 4:3617
N/A	Set Module Reset Count	E: O.Data [18]	M: 4:3618
4:0020	ConveyStop Command	E: O.Data [19]	M: 4:3619
4:0109	Clear Sensor Jam Command for Local Upstream Zone	E: O.Data [20]	M: 4:3620
4:0189	Clear Sensor Jam Command for Local Downstream Zone	E: O.Data [21]	M: 4:3621
4:0365	Direction & Accumulation Mode Control for Local Upstream Zone	E: O.Data [22]	M: 4:3622
4:0375	Direction & Accumulation Mode Control for Local Downstream Zone	E: O.Data [23]	M: 4:3623
4:0387	ConveyMerge Interface	E: O.Data [24]	M: 4:3624

11.6.3. Reduced Size ZPA Mode Assemblies with Reset Protection

Inputs to PLC

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0116	Local Status Upstream Zone	E: I.Data [0]	M: 4:4900
4:0196	Local Status Downstream Zone	E: I.Data [1]	M: 4:4901
4:0106	Arrival Count Local Upstream Zone	E: I.Data [2]	M: 4:4902
4:0107	Departure Count Local Upstream Zone	E: I.Data [3]	M: 4:4903
4:0186	Arrival Count Local Downstream Zone	E: I.Data [4]	M: 4:4904
4:0187	Departure Count Local Downstream Zone	E: I.Data [5]	M: 4:4905
4:0088	Module Status Word 1	E: I.Data [6]	M: 4:4906
4:0089	Module Status Word 2	E: I.Data [7]	M: 4:4907
4:0105	Current Release Count for Upstream Zone	E: I.Data [8]	M: 4:4908
4:0185	Current Release Count for Downstream Zone	E: I.Data [9]	M: 4:4909
4:0035	Sensor & Control Port Inputs	E: I.Data [10]	M: 4:4010
N/A	Current Module Reset Count	E: I.Data [11]	M: 4:4911

Outputs from PLC

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0104	Accumulation Control for Local Upstream Zone	E: O.Data [0]	M: 4:5000
4:0184	Accumulation Control for Local Downstream Zone	E: O.Data [0]	M: 4:5001
4:0040	Set Left MDR Speed	E: O.Data [2]	M: 4:5002
4:0064	Set Right MDR Speed	E: O.Data [3]	M: 4:5003
4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	E: O.Data [4]	M: 4:5004

4:0185	Release and Accumulate on Next Arrival for Local Upstream Zone	E: O.Data [5]	M: 4:5005
4:0134	Set Status for Upstream Induct	E: O.Data [6]	M: 4:5006
4:0232	Set Status for Downstream Discharge	E: O.Data [7]	M: 4:5007
4:0022	Clear Motor Error	E: O.Data [8]	M: 4:5008
4:0063	Set Control Port Outputs	E: O.Data [9]	M: 4:5009
N/A	**Set Module Reset Count**	E: O.Data [10]	M: 4:5010
4:0109	Clear Sensor Jam Command for Local Upstream Zone	E: O.Data [11]	M: 4:5011
4:0189	Clear Sensor Jam Command for Local Downstream Zone	E: O.Data [12]	M: 4:5012
4:0365	Direction & Accumulation Mode Control for Local Upstream Zone	E: O.Data [13]	M: 4:5013
4:0375	Direction & Accumulation Mode Control for Local Downstream Zone	E: O.Data [14]	M: 4:5014

11.6.4. PLC I/O Mode Assembly Inputs with Reset Protection

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0019	ConveyStop Status	E: I.Data [0]	M: 4:3700
4:0035	Sensor & Control Port Inputs	E: I.Data [1]	M: 4:3701
4:0036	Sensor Detect	E: I.Data [2]	M: 4:3702
4:0024	Module Voltage	E: I.Data [3]	M: 4:3703
4:0055	Left Motor Current	E: I.Data [4]	M: 4:3704
4:0056	Left Motor Frequency	E: I.Data [5]	M: 4:3705
4:0057	Left Motor Temperature	E: I.Data [6]	M: 4:3706
4:0058	Left Motor Status	E: I.Data [7]	M: 4:3707
4:0079	Right Motor Current	E: I.Data [8]	M: 4:3708
4:0080	Right Motor Frequency	E: I.Data [9]	M: 4:3709
4:0081	Right Motor Temperature	E: I.Data [10]	M: 4:3710
4:0082	Right Motor Status	E: I.Data [11]	M: 4:3711
4:0060	Left Motor Port Digital I/O Status	E: I.Data [12]	M: 4:3712
4:0084	Right Motor Port Digital I/O Status	E: I.Data [13]	M: 4:3713
4:0134	Upstream Module Status	E: I.Data [14]	M: 4:3714
4:0232	Downstream Module Status	E: I.Data [15]	M: 4:3715
4:0139	Current Tracking Word 1 for Adjacent Upstream Module	E: I.Data [16]	M: 4:3716
4:0140	Current Tracking Word 2 for Adjacent Upstream Module	E: I.Data [17]	M: 4:3717
N/A	**Current Module Reset Counter**	E: I.Data [18]	M: 4:3718
4:0062	Left Motor Servo Position	E: I.Data [19]	M: 4:3719
4:0086	Right Motor Servo Position	E: I.Data [20]	M: 4:3720
4:0011	Left Motor Servo Status	E: I.Data [21]	M: 4:3721

| 4:0016 | Right Motor Servo Status | E: I.Data [22] | M: 4:3722 |

11.6.5. PLC I/O Mode Assembly Outputs with Reset Protection

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0020	ConveyStop Command	E: O.Data [0]	M: 4:3800
4:0060	Set Left Motor Port Digital Control	E: O.Data [1]	M: 4:3801
4:0084	Set Right Motor Port Digital Control	E: O.Data [2]	M: 4:3802
4:0037	Control Port Digital Output Control	E: O.Data [3]	M: 4:3803
4:0260	Left Motor Run / Reverse	E: O.Data [4]	M: 4:3804
4:0261	Left Motor Brake Method	E: O.Data [5]	M: 4:3805
4:0262	Left Motor Speed Control Method	E: O.Data [6]	M: 4:3806
4:0270	Right Motor Run / Reverse	E: O.Data [7]	M: 4:3807
4:0271	Right Motor Brake Method	E: O.Data [8]	M: 4:3808
4:0272	Right Motor Speed Control Method	E: O.Data [9]	M: 4:3809
4:0040	Left Motor Speed Reference	E: O.Data [10]	M: 4:3810
4:0064	Right Motor Speed Reference	E: O.Data [11]	M: 4:3811
4:0043	Left Motor Acceleration Ramp	E: O.Data [12]	M: 4:3812
4:0044	Left Motor Deceleration Ramp	E: O.Data [13]	M: 4:3813
4:0067	Right Motor Acceleration Ramp	E: O.Data [14]	M: 4:3814
4:0068	Right Motor Deceleration Ramp	E: O.Data [15]	M: 4:3815
4:0022	Clear Motor Error	E: O.Data [16]	M: 4:3816
4:0196	Set Status to Downstream Module	E: O.Data [17]	M: 4:3817
4:0116	Set Status to Upstream Module	E: O.Data [18]	M: 4:3818
4:0034	Sensor Port Input Signal Condition Mask	E: O.Data [19]	M: 4:3819
4:0201	Set Discharge Tracking Word 1	E: O.Data [20]	M: 4:3820
4:0202	Set Discharge Tracking Word 2	E: O.Data [21]	M: 4:3821
N/A	Set Module Reset Count	E: O.Data [22]	M: 4:3822

4:0008	Left Motor Servo Command Pulses	E: O.Data [23]	M: 4:3823
4:0009	Left Motor Servo Command Word	E: O.Data [24]	M: 4:3824
4:0013	Right Motor Servo Command Pulses	E: O.Data [25]	M: 4:3825
4:0014	Right Motor Servo Command Word	E: O.Data [26]	M: 4:3826

11.6.6. Reduced Size PLC I/O Mode Assemblies with Reset Protection

Inputs to PLC

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0035	Sensor & Control Port Inputs	E: I.Data [0]	M: 4:4700
4:0036	Sensor Detect	E: I.Data [1]	M: 4:4701
4:0057	Left Motor Temperature	E: I.Data [2]	M: 4:4702
4:0058	Left Motor Status	E: I.Data [3]	M: 4:4703
4:0081	Right Motor Temperature	E: I.Data [4]	M: 4:4704
4:0082	Right Motor Status	E: I.Data [5]	M: 4:4705
4:0060	Left Motor Port Digital I/O Status	E: I.Data [6]	M: 4:4706
4:0084	Right Motor Port Digital I/O Status	E: I.Data [7]	M: 4:4707
N/A	**Current Module Reset Count**	E: I.Data [8]	M: 4:4708

Outputs from PLC

Module Address	Register Name	Ethernet I/P Tag	Modbus PLC Holding Register
4:0060	Set Left Motor Port Digital Control	E: O.Data [0]	M: 4:4800
4:0084	Set Right Motor Port Digital Control	E: O.Data [1]	M: 4:4801
4:0037	Control Port Digital Output Control	E: O.Data [2]	M: 4:4802
4:0260	Left Motor Run / Reverse	E: O.Data [3]	M: 4:4803
4:0270	Right Motor Run / Reverse	E: O.Data [4]	M: 4:4804
4:0040	Left Motor Speed Reference	E: O.Data [5]	M: 4:4805
4:0064	Right Motor Speed Reference	E: O.Data [6]	M: 4:4806
4:0022	Clear Motor Error	E: O.Data [7]	M: 4:4807
N/A	**Set Module Reset Count**	E: O.Data [8]	M: 4:4808

11.6.7. How to use Assemblies with Reset Protection

When the PLC determines that it has a valid connection to the ConveyLinx-Ai Family Controllers in question and is ready for it to respond to output data being written by the PLC:

- Your PLC program needs to move the value in the *Current Module Reset Counter* register into the *Set Module Reset Counter* register

- When the ConveyLinx-Ai Family Controllers detects that the value in *Set Module Reset Counter* register is equal to the value in the *Current Module Reset Counter* register; the module will respond to data being written by the PLC to the remaining output assembly registers.

> ✱ Please note that when the value in *Current Module Reset Counter* register is <u>not equal</u> to the value in the *Set Module Reset Counter* register; the values in the Input registers will updated by the ConveyLinx-Ai Family Controllers and will be valid in your PLC program. In this state, even though the PLC may be writing data to Output registers, the <u>module will ignore it</u>

> ✱ To make sure a given ZPA zone accumulates upon power up, use *EasyRoll* configuration tool software to set the zone to "Accumulate". When set from *EasyRoll*, this setting will be retained in the flash memory of the module so that the zone will initially accumulate if a load happens to be in the zone at the time of power up. <u>Learn about Accumulate control from EasyRoll</u>

> ✱ When a ZPA zone is commanded to accumulate with a PLC and you happen to connect to that module with EasyRoll; the "Accumulate" switch icon on the main screen will visibly indicate that the zone is accumulated. This visible icon will look the same as if it was clicked ON from EasyRoll.
> Keep in mind that a <u>PLC command to accumulate IS NOT retained upon power loss to the module</u>. Only if the "Accumulate" switch is toggled "ON" with EasyRoll is the accumulate condition for the zone retained in flash memory for use upon power-up

11.7. Motor Port as Digital I/O

> ❗ Motor port as digital I/O is only applicable for a ConveyLinx-Ai Family Controllers in PLC I/O Mode

Each motor port can provide 1 or 2 independently controlled digital outputs and each of these outputs can be energized simultaneously for a total of 4 outputs available per ConveyLinx-Ai module. Each individual output has a 0.75A load capacity.

External controller must first set bit 15 = 1 in the *Set Left/Right Motor Port Digital Control* register for the motor port (Left or Right) that is to be used as digital output. If bit 15 = 0, then the ConveyLinx-Ai Family Controllers ignores the bit 0 thru bit 2 commands and will not provide meaningful status on bits 12 and 14 in the corresponding *Left/Right Motor Port Digital I/O Status* register for the port in question

> ❗ Please note that for both the Left and Right Motor Ports, the each digital output switches to GND to complete the circuit

[See Wiring Diagrams for Motor Port Digital Outputs](#)

> ❗ *Short Circuit Error* on bit 12 in the *Left/Right Motor Port Digital I/O Status* register is classified as a "fatal" error that will require interaction. You can (1) cycle of power on the ConveyLinx-Ai Family Controllers, (2) Send Reset command from EasyRoll, or (3) Send a Motor Fault Reset command from the PLC

12. Setting Up Dual Motor Zones

This section describes how to set up conveyor zones with two motor rollers in scenarios involving ZPA Mode, PLC I/O Mode, and ConveyLogix program control. The details of the set up are also governed by the mechanical construction of the conveyor zone.

Solution Matrix for ConveyLinx-Ai Family Modules

Zone Type	Operational Mode	Solution
2 MDRs – Coupled	ZPA	Automatic – No Adjustments required
	PLC I/O	Requires program code to set up proper PLC Output Assembly values
	ConveyLogix	Requires program code to set up proper internal register values
2 MDRs – Non-Coupled	ZPA	**Will Not Work Properly**
	PLC I/O	No special adjustments required – Use program code to run independently or together as desired
	ConveyLogix	No special adjustments required – Use program code to run independently or together as desired
2 Motors in 1 MDR Tube	ZPA	**Not Possible**
	PLC I/O	Requires program code to set up proper PLC Output Assembly values
	ConveyLogix	Requires program code to set up proper internal register values

12.1. Two Ai MDRs Mechanically Coupled

This scenario is perhaps the most common in that this would be the case for a two-motor roller used in an incline or decline application as shown. Please note that mechanically the two motor rollers can be coupled with just O-bands and with a full width belt. The concept is that the upstream motor roller is there to assist the downstream motor roller.
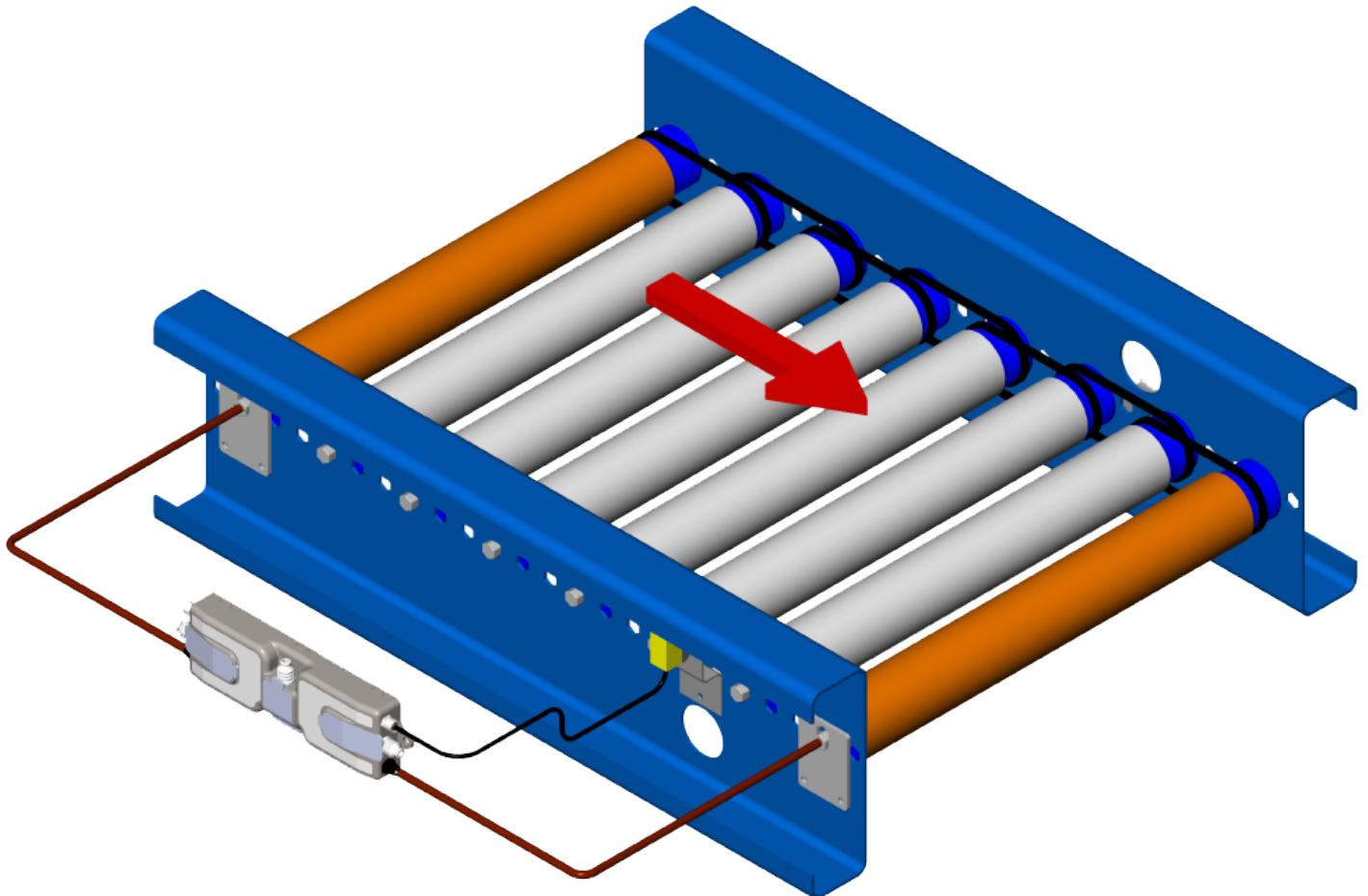
12.1.1. ZPA Mode

In general, when you perform an *Auto-Configure* procedure, ConveyLinx-Ai2 modules will configure in ZPA mode. If one sensor is connected (either Left or Right) and two motors are connected, the module is automatically configured as a *single logical zone* and will run both motors at the same time. The side (Left or Right) that the sensor is connected will automatically configure that side's motor as the *master* and the other motor is the *slave*. The master motor's PI regulation and PWM output is applied to the "slave" motor.

> ✳ Please note that any settings for the slave motor that may have been in place in EasyRoll are ignored and only the *master motor settings* are recognized for <u>both</u> motors

Slave MDR will ignore settings in EasyRoll and will assume the settings of the Master MDR

Master MDR because it is on the same side as the Sensor
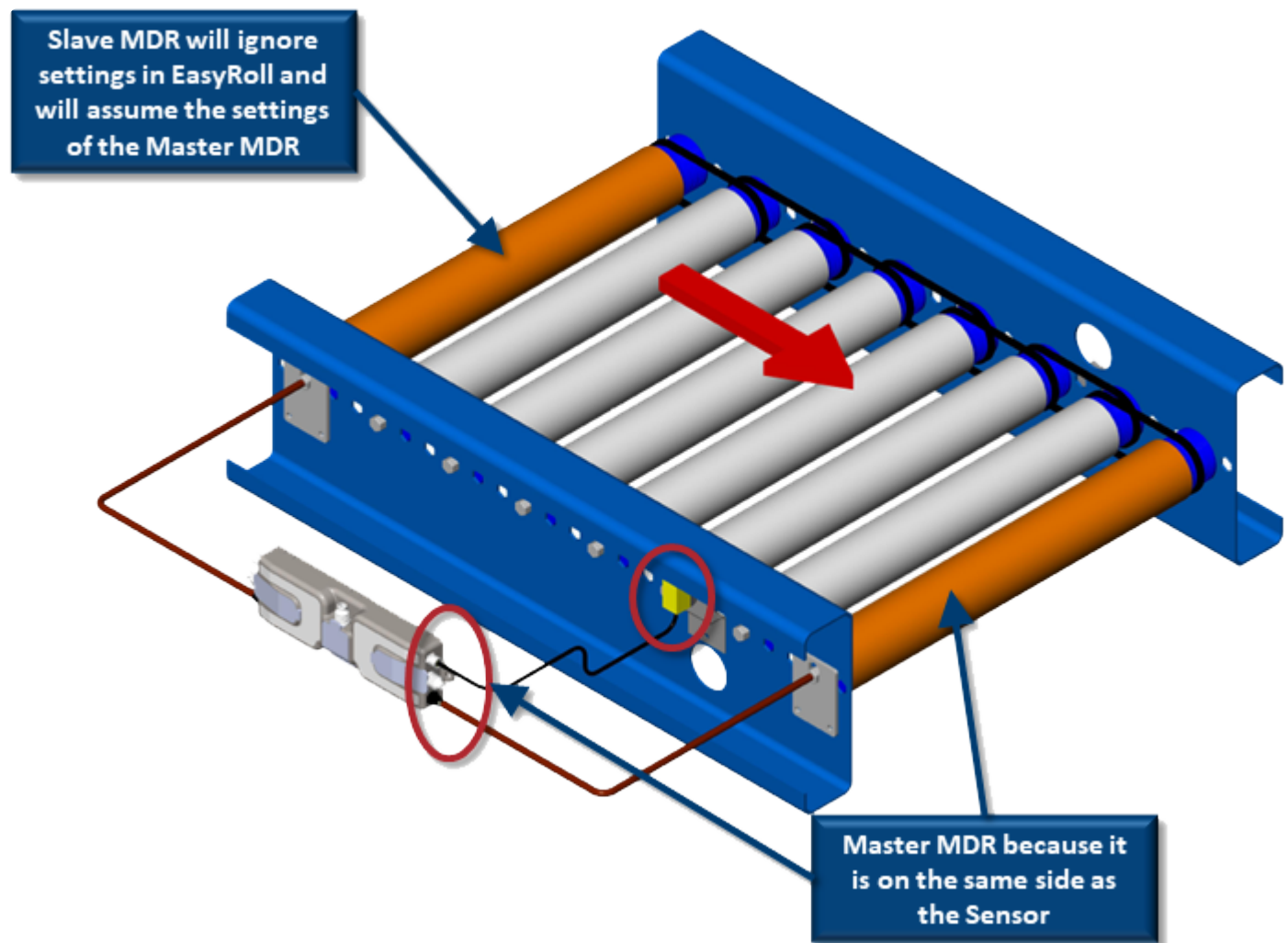
12.1.2. PLC I/O Mode

In *PLC I/O* mode there is no *ZPA logic* operating in the module, so the motors are independently controlled by default when the module is placed in PLC I/O mode. With a remote PLC you can connect to the PLC I/O Assemblies and configure <u>either</u> the Left or the Right motor to be the *master* and command only the *master* and the *slave* motor will operate along with the *master*.

In the PLC I/O Output Assembly, there is a *Left Motor Slave Mode (4:0262)* register and a *Right Motor Slave Mode (4:0272)* register. Depending on which single motor (**Left** or **Right**) you want to command, you select the *opposite motor's corresponding Slave Mode register* to instruct the opposite motor to follow the master motor's commands. Also, when you do this, the Slave Motor takes on the Master Motor's speed and settings.

PLC I/O Mode Example

In this example, let's say you want to use the *Left Motor* as the *Master* and the *Right Motor* as the *Slave*. In this case you populate the *Right Motor Slave Mode* register with a **value of 2** to tell the *Right Motor* to use the settings of the *Left Motor* and operate when the *Left Motor* is instructed to operate by the PLC program logic.

Left Motor is selected as the Master – all motor settings and control will be applied to the Left Motor in your PLC program

Location of Sensor does not matter in PLC I/O mode

Right Motor is the Slave – it is configured with the *Right Motor Slave Mode* register to follow the Left Motor. EasyRoll settings are ignored
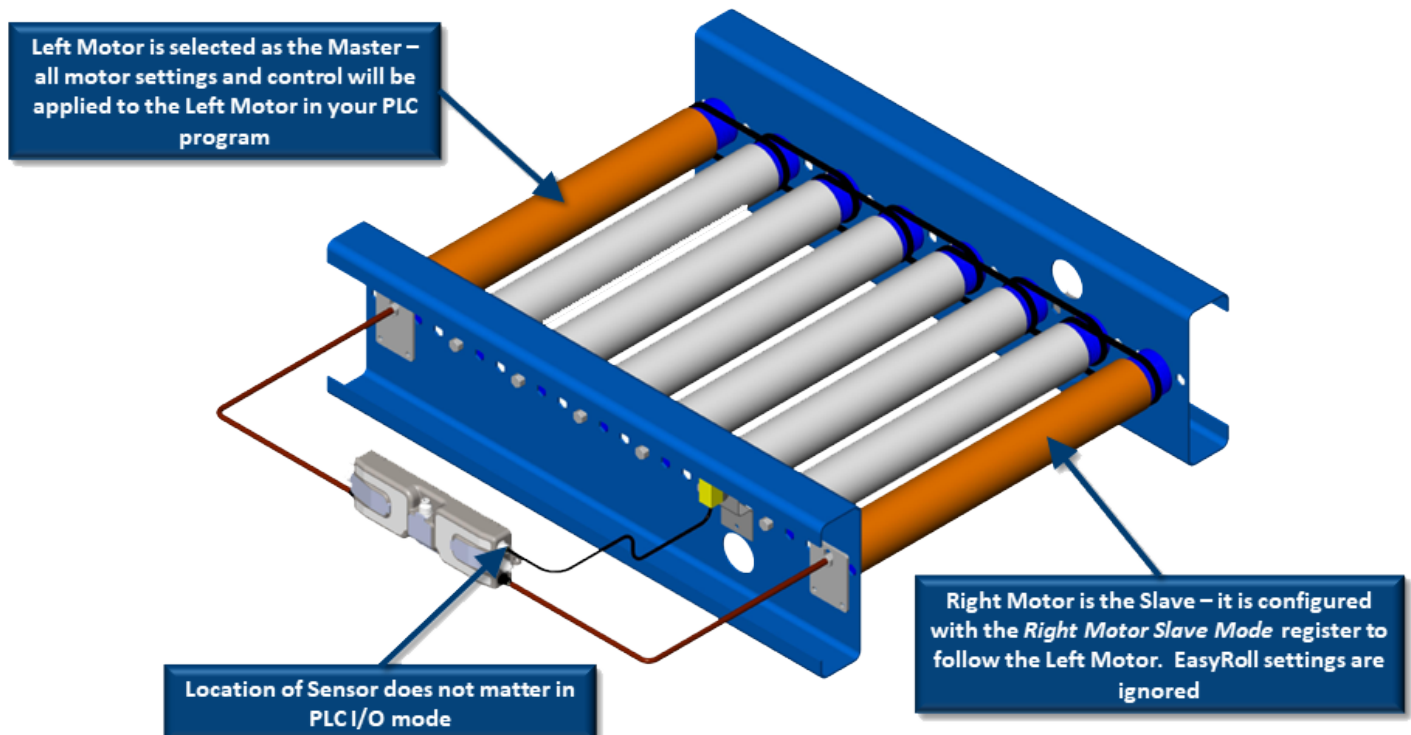
12.1.3. ConveyLogix Program

Similar to *PLC I/O Mode* example, you populate either the *Left Motor Slave Mode (4:0262)* register or the *Right Motor Slave Mode (4:0272)* register according to the desired control. It is recommended you use the *FirstLadderExec* controller tag Boolean in a rung that executes a *WRC* block that writes the desired data to Register *262* or *272*. By using the *FirstLadderExec* N.O. contact in this rung makes sure the data is written upon any power up or re-boot of the ConveyLinx-Ai2 module and that it is not continuously writing the data every program scan and using up unnecessary module resources.

12.2. Two Ai Motors in One Roller Tube (Dual Drive)

Some Pulseroller-Ai models are available with two motors in one tube as shown.
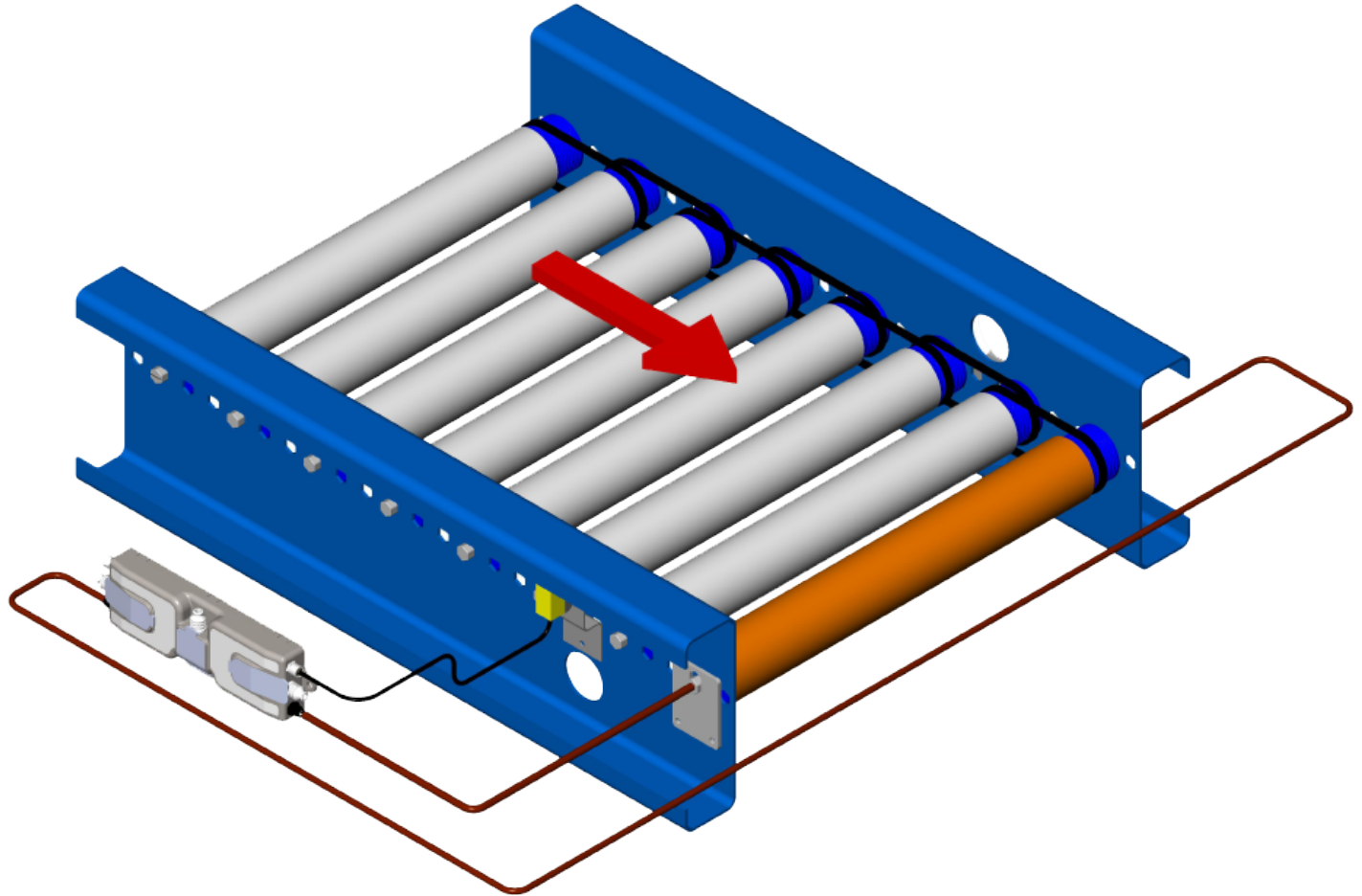
ZPA Mode

This mechanical configuration is not supported with ConveyLinx-Ai Family controls.

> ❗ Attempting to use a Dual Drive MDR in ZPA Mode with a ConveyLinx-Ai Family controller could result in damage to the roller assembly

PLC I/O Mode

During the *Auto-Configure Procedure*, all zones become *ZPA mode* and motors will run. The ConveyLinx-Ai Family Controllers does not know the opposite motor is in the same tube and will try to rotate the motors in the same direction and they will be in a mechanical bind. Unplugging the opposite motor will keep this from happening. Once the Auto-Configuration is complete, place the module in PLC I/O mode and you can reconnect the motor to the module.

Unplug opposite side motor prior to Auto-Configuration
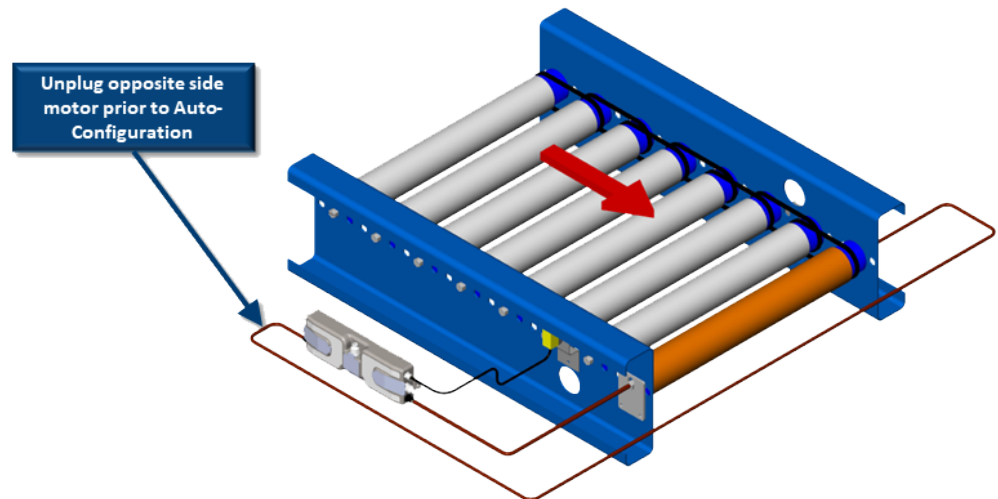
Example

In this example, the Right Motor is the Master and the Left Motor is the Slave. After you have performed Auto-Configuration and then placed the module in PLC I/O Mode and reconnected the motor, you write a *value of 3* set up the [Left Motor Slave Mode (4:0262)](#) register. This value tells the Left Motor to be in Slave Mode to the Right Motor and always run in the opposite direction of the Right Motor.

Left Motor is the Slave – it is configured with the *Left Motor Slave Mode* register to follow the Right Motor. EasyRoll settings and direction are ignored

Left Motor Slave Mode register needs a value of 3 to instruct the Left Motor to be a Slave to the Right Motor and to run in the opposite direction of the Right Motor

Right Motor is selected as the Master and all motor settings and control will be applied to the Right Motor in your PLC program

ConveyLogix Program

Follow the same procedure as in this PLC I/O Mode example as far as unplugging the Slave motor prior to Auto-Configuration. The follow the procedure described for the *Two Ai MDRs Mechanically Coupled – ConveyLogix Program* section.

> ! Be sure to remember to write the proper value in the proper Motor Slave Mode register so that the Slave Motor always runs in the opposite direction to the Master Motor

12.3. Two Motor Rollers in One Logical Zone Not Coupled

In this mechanical scenario, the motor rollers are not coupled together as shown.
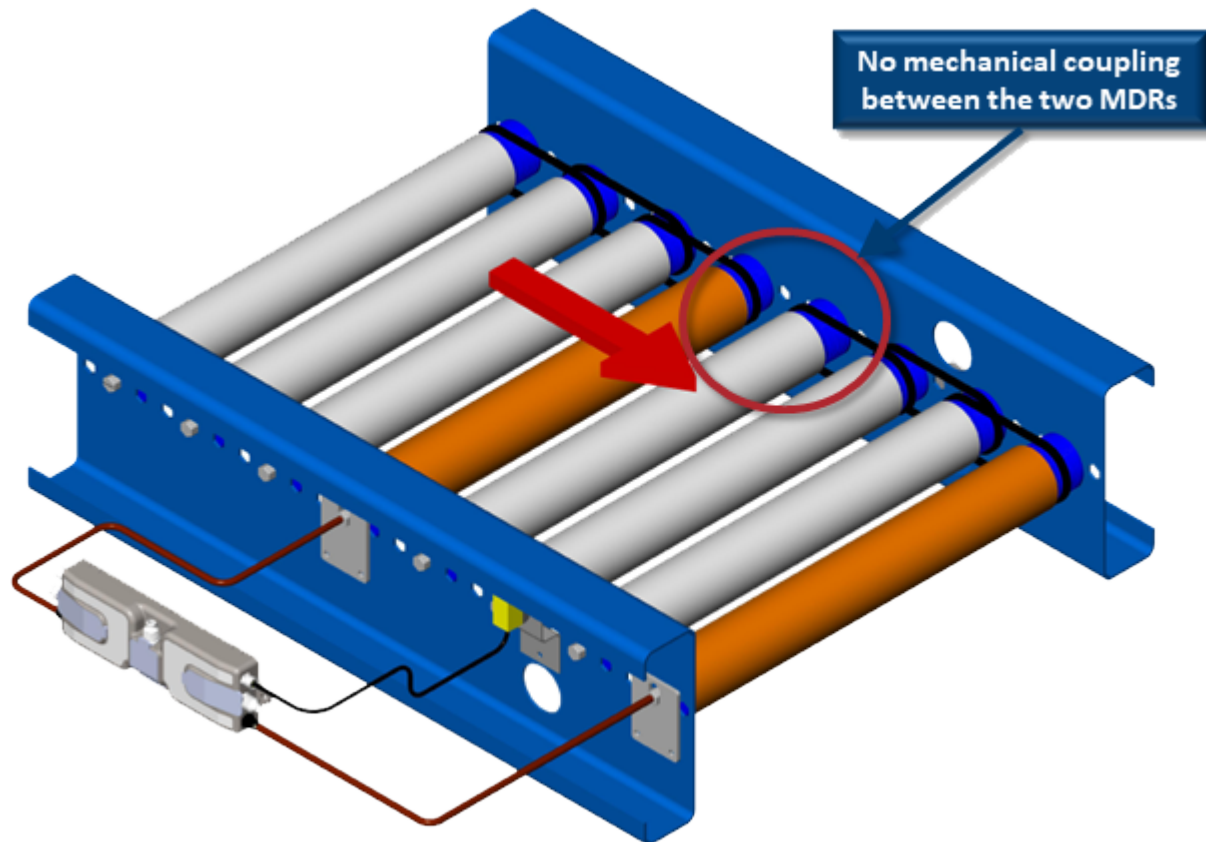
ZPA Mode

In *ZPA Mode* with this mechanical configuration, it will not work properly because the module will never monitor and adjust the Slave Motor's speed. Because they are un-coupled; both motors will vary their speed based upon their respective loads while the module is only paying attention to the load on the master.

> ! Even though this configuration is logically allowed, it is recommended that you do not use ZPA mode for this mechanical configuration

PLC I/O Mode and ConveyLogix Program

You can set the motors to the same settings in *EasyRoll* and logically run both motors at the same time in your PLC logic program or ConveyLogix program. Even though you can attempt to use the Master/Slave functionality as previously described, you will have the same situation as

just described for ZPA mode where uncoupled motors will vary their speed.

> **!** It is not recommended to use Master/Slave motor control for this mechanical configuration

13. Connecting to Rockwell PLCs with Ethernet I/P

This manual will provide instructions on how to connect your Rockwell Ethernet I/P capable PLC to a network of ConveyLinx modules. There are three basic methods for connecting ConveyLinx to Rockwell PLCs:

- Use Generic Ethernet Device
- Import EDS and optionally import and use AOIs
- Use MSG Instruction

All three methods can be used for ConveyLinx modules in ZPA mode and in PLC I/O mode. However, the MSG Instruction method does not maintain a constant connection to a ConveyLinx module and should not be used for "time critical" operations.

Ethernet I/P Guidelines

Each Allen-Bradley PLC has 2 metrics for limiting maintained Ethernet I/P communications to remote devices:

- Fixed quantity of TCP connections available on its Ethernet Port
- Fixed quantity of I/O data table memory available for connected devices

If the limit of either of these quantities is reached, the PLC processor will indicate I/O communications fault on one or more instances of device declaration. For ConveyLinx device declarations utilizing either ZPA or PLC I/O Mode instances, in general the PLC limitation on TCP connections will be reached before I/O data table memory limit is realized.
For example, for a CompactLogix L3x series processor, the documented quantity of TCP connections available on its Ethernet Port is 32. The processor always keeps one TCP connection in reserve for programming terminal access, etc. An L3x series processor can accept 31 full-time ConveyLinx Connections as generic I/O modules utilizing any combination of ZPA mode and PLC I/O Mode instances.
When a ConveyLinx module is attached as a "full-time generic I/O module" to the PLC, the connection is continually maintained and data is exchanged at a minimum RPI value (referred to as an *implicit connection*). If the PLC cannot communicate with the ConveyLinx module for any reason, the PLC's I/O tree will register a fault. It is possible for the PLC to communicate via Ethernet I/P with any ConveyLinx module it can physically reach over its Ethernet port without the module being "full-time connected as a generic I/O module". This is accomplished with a Logix5000 MSG instruction (referred to as *explicit connection*).

> ✳ Reserve Ethernet I/P TCP connections for ConveyLinx modules in PLC I/O Mode

and for key ZPA Mode modules where permanent accumulate/query/release functionality is required

✻ Use MSG Instruction to gather less time-critical data for things such as status and diagnostics

13.1. Selecting Your Connection Method based upon Assembly

As described in the [PLC Developer's Guide](#) section, the data that you exchange with your PLC and a given ConveyLinx-Ai Family Controllers depends on the mode of the module and how you want to use it. The I/O data to be exchanged are arranged in register *Assemblies* and depending on the assembly, will dictate whether you can connect using the EDS file method or the Generic Ethernet Module method.

> ✳ All available assemblies can be connected utilizing the Generic Ethernet Module method and only selected assemblies are available from the EDS file installation

Assembly Pair	Available from EDS File Installation	Available as Generic Ethernet Module
ZPA Mode Assembly Inputs ZPA Mode Assembly Outputs	✔	✔
ZPA Mode Assembly Inputs with Reset Protection ZPA Mode Assembly Outputs with Reset Protection	✔	✔
Reduced Size ZPA Mode Assembly Inputs Reduced Size ZPA Mode Assembly Outputs		✔
Reduced Size ZPA Mode Assembly Inputs with Reset Protection Reduced Size ZPA Mode Assembly Outputs with Reset Protection		✔
PLC I/O Mode Assembly Inputs PLC I/O Mode Assembly Outputs	✔	✔
PLC I/O Mode Assembly Inputs with Reset Protection PLC I/O Mode Assembly Outputs with Reset Protection	✔	✔
Reduced Size PLC I/O Mode Assembly Inputs Reduced Size PLC I/O Mode Assembly Outputs		✔
Reduced Size PLC I/O Mode Assembly Inputs with Reset Protection Reduced Size PLC I/O Mode Assembly Outputs with Reset Protection		✔
ConveyLogix Assembly Inputs ConveyLogix Assembly Outputs	✔	✔

13.2. Using Generic Ethernet Module Method

When using the *Generic Ethernet Module* construct in RSLogix 5000, you must supply configuration information about the device you are trying to connect. The following sections show the step by step procedure to connect a ConveyLinx-Ai Family Controllers for each set of Input and Output Assemblies described in the PLC Developers Guide section.

Assembly	Recognized Instance Values
ZPA Mode Assembly Inputs	105
ZPA Mode Assembly Outputs	106
ZPA Mode Assembly Inputs with Reset Protection	305
ZPA Mode Assembly Outputs with Reset Protection	306
Reduced Size ZPA Mode Assembly Inputs	119
Reduced Size ZPA Mode Assembly Outputs	120
Reduced Size ZPA Mode Assembly Inputs with Reset Protection	319
Reduced Size ZPA Mode Assembly Outputs with Reset Protection	320
PLC I/O Mode Assembly Inputs	107
PLC I/O Mode Assembly Outputs	108
PLC I/O Mode Assembly Inputs with Reset Protection	307
PLC I/O Mode Assembly Outputs with Reset Protection	308
Reduced Size PLC I/O Mode Assembly Inputs	117
Reduced Size PLC I/O Mode Assembly Outputs	118
Reduced Size PLC I/O Mode Assembly Inputs with Reset Protection	317
Reduced Size PLC I/O Mode Assembly Outputs with Reset Protection	318
ConveyLogix Assembly Inputs	121
ConveyLogix Assembly Outputs	122

13.3. Procedure for Connecting using Generic Ethernet Module

All assembly pairs can be connected to a single ConveyLinx-Ai Family Controllers using the same procedure within RSLogix 5000 environment:

1. Create a New Module in your Ethernet Tree
2. Select Generic Ethernet Module from the list of devices
3. Enter name and I.P. Address
4. Select the correct Comm Data type
5. Enter Input Assembly Instance Value and Size
6. Enter Output Assembly Instance Value and Size
7. Enter desired RPI value

For example if you need to attach to 5 ConveyLinx-Ai Family Controllerss that are in ZPA Mode, each module will have to have a unique name and I.P. address (step 3) and steps 4, 5, 6, and 7 will use the same values for each ConveyLinx-Ai Family Controllers.

13.3.1. Parameters for Each Assembly

✳ Please note that for all Assemblies and all versions of firmware the Instance value for the "Configuration" parameter is always "1" and its size is always "0"

Assembly	Type	Instance Value	Size Value
ZPA Mode Assembly	Input	105	21
	Output	106	25
ZPA Mode Assembly with Reset Protection	Input	305	21
	Output	306	25
Reduced Size ZPA Mode Assembly	Input	119	12
	Output	120	15
Reduced Size ZPA Mode Assembly with Reset Protection	Input	319	12
	Output	320	15
PLC I/O Mode Assembly	Input	107	25
	Output	108	27
PLC I/O Mode Assembly with Reset Protection	Input	307	25
	Output	308	27
Reduced Size PLC I/O Mode Assembly	Input	117	9
	Output	118	9
Reduced Size PLC I/O Mode Assembly with Reset Protection	Input	317	9
	Output	318	9
ConveyLogix Assembly	Input	121	16
	Output	122	16

13.4. Procedure for using EDS Method

Pulseroller.com website contains all EDS files for download including older versions. Please visit the Downloads section of Pulseroller.com to get the latest EDS file for the ConveyLinx-Ai Family of modules

> **!** For best results, you should remove any previous ConveyLinx-Ai Family EDS file(s) you may have installed in your RSLogix 5000 environment before installing a new version

> **!** Delete all unused module data types from your program especially if you are modifying or starting with an existing program

Installing the EDS file provided by Pulseroller into your RSLogix 5000 environment will allow you to select the ConveyLinx-Ai Family Controllers module from your list of known devices without having to use the Generic Ethernet Module method. The EDS file contains the Instance and size parameters so you do not have to fill in this information. When you connect to a ConveyLinx-Ai Family Controllers, the data is arranged in assembled registers with the data appearing in your Controller Tags similarly to how the data appears when you connect using the *Generic Ethernet Module* method.

https://www.youtube.com/embed/z28iBdmeRyg?rel=0

13.5. Using Add On Instructions (AOI) with RSLogix 5000

Pulseroller has authored and made available *Add On Instructions (AOI)* in order to make your programming easier to follow. In this document up until this section, when connecting to a ConveyLinx-Ai Family Controllers regardless of mode; your PLC program needs to directly access the register data array tags created when you created the ConveyLinx-Ai Family Controllers instance. The *AOIs* attach to the created ConveyLinx-Ai Family Controllers's register data arrays and maps the data into user tags and functions with meaningful names. There are two separate *AOIs* for use depending on the mode of the ConveyLinx-Ai Family Controllers you want to connect: a *ZPA mode AOI* and a *PLC I/O mode AOI*

> ✳ Please note that the use of <u>AOI</u> is <u>purely optional</u>. However, you <u>must install the EDS file</u> as previously described before you can use any AOI

Selecting the Proper AOI Instruction

AOI files are imported to your specific PLC program file and not into the RSLogix 5000 environment like an EDS file. The following chart provides a cross-reference for selecting the proper *AOI* file based upon the mode of operation

ConveyLinx-Ai Family Controllers Mode	EDS File	AOI File
ZPA Mode	ConveyLinxAi_V5_1.eds	Ai2_ZPA_AOI_4_5.L5X
PLC I/O Mode	ConveyLinxAi_V5_1.eds	Ai2_PLCIO_AOI_4_5.L5X

> ✳ Please refer to the downloads section of Pulseroller.com for the latest updates to EDS and/or AOI files

> ❗ Please note that in the video below that the AOI is installed before the Module is created in the Ethernet Tree. For some Rockwell processors, the Module must be created first before the AOI can be installed

https://www.youtube.com/embed/dvkgMVEocoI?rel=0

13.5.1. AOI Tag Descriptions

The following links take you to the listings for each tag made available in the both the ERSC-ZPA AOI and the PLC I/O AOI.

ZPA Mode Inputs

ZPA Mode Outputs

PLC I/O Inputs

PLC I/O Outputs

13.5.1.1. ZPA Mode Inputs

Tag Name	Data Type	Module Register	Bit	Description
I_Arrival_DnZone	BOOL	4:0116	AOI Logic	Accumulated Arrival Downstream – Only Active When Local Accumulate Bit is Set
I_Arrival_UpZone	BOOL	4:0196	AOI Logic	Accumulated Arrival Upstream – Only Active When Local Accumulate Bit is Set
I_ConveyStopByLostConnection	BOOL	4:0019	6	ConveyStop Activated Because Of Lost Connection
I_ConveyStopByPLCCmd	BOOL	4:0019	10	ConveyStop Activated Because Of PLC Command
I_ConveyStopByPLCDisconnect	BOOL	4:0019	7	ConveyStop Activated Because Of Lost PLC Connection
I_ConveyStopByRemoteModule	BOOL	4:0019	5	ConveyStop Activated By Another Module in ConveyStop Group
I_Get_Tracking_FWD	DINT	4:0201 (MSW) 4:0202 (LSW)	AOI Logic	Get Forward Tracking Value at Induct to Local Upstream Zone
I_Get_Tracking_REV	DINT	4:0121 (MSW) 4:0122 (LSW)	AOI Logic	Get Reverse Tracking Value at Induct to Local Downstream Zone
I_Heartbeat	BOOL	4:0035	15	Module Heartbeat, 2s On / 2s Off
I_Jam_at_DnZone	BOOL	4:0089	5	Sensor Jam Local Downstream Zone
I_Jam_at_UpZone	BOOL	4:0088	5	Sensor Jam Local Upstream Zone
I_Module_Enabled	BOOL	-	AOI Logic	Local Outputs are Enabled (Reset Protection Mode)
I_Module_Fault_Left	BOOL	4:0088	3	Left Motor Error Active
I_Module_Fault_Right	BOOL	4:0089	3	Right Motor Error Active

I_Module_Sts_Left	INT	4:0088	-	Left Module Status
I_Module_Sts_Right	INT	4:0089	-	Right Module Status
I_Sensor_Port_Left_Pin2	BOOL	4:0035	0	Left Sensor Port Pin 2 Energized (if used as input)
I_Sensor_Port_Left_Pin4	BOOL	4:0035	4	Left Sensor Port Pin 4 Energized
I_Sensor_Port_Right_Pin2	BOOL	4:0035	2	Right Sensor Port Pin 2 Energized (if used as input)
I_Sensor_Port_Right_Pin4	BOOL	4:0035	6	Right Sensor Port Pin 4 Energized
I_Tracking_DnZn	DINT	4:0199 (MSW) 4:0200 (LSW)	AOI Logic	For Load Accumulated in Downstream Zone
I_Tracking_UpZn	DINT	4:0119 (MSW) 4:0120 (LSW)	AOI Logic	For Load Accumulated in Upstream Zone
I_Zone_Status_DnZn_FWD	SINT	4:0196	Lo Byte	Local Downstream Status Forward Direction
I_Zone_Status_DnZn_REV	SINT	4:0196	Hi Byte	Local Downstream Status Reverse Direction
I_Zone_Status_UpZn_FWD	SINT	4:0116	Lo Byte	Local Upstream Status Forward Direction
I_Zone_Status_UpZn_REV	SINT	4:0116	Hi Byte	Local Upstream Status Reverse Direction

13.5.1.2. ZPA Mode Outputs

Tag Name	Data Type	Module Register	Bit	Description
O_Acc_Adj_UpStr_to_DnZn	BOOL	4:0184	8	Accumulate Adlacent Upstream to Local Upstream Zone
O_Acc_Adj_UpStr_to_UpZn	BOOL	4:0104	8	Accumulate Adlacent Upstream to Local Downstream Zone
O_Acc_Arr_DnZn	BOOL	4:0184	0	Set Local Downstream Zone to Accumulate
O_Acc_Arr_UpZn	BOOL	4:0104	0	Set Local Upstream Zone to Accumulate
O_Clear_Jam_DnZn	BOOL	4:0189	0	Clear Jam at local Downstream Zone
O_Clear_Jam_UpZn	BOOL	4:0109	0	Clear Jam at local Upstream Zone
O_Clear_Motor_Error	BOOL	4:0022	0	Clear Motor Error Left & Right
O_ConfArrivalAdjDownstreamToDnZn	BOOL	4:0184	9	Confirm Downstream Arrival for Adjacent Downstream Zone
O_ConfArrivalAdjDownstreamToUpZn	BOOL	4:0104	9	Confirm Downstream Arrival for Adjacent Downstream Zone
O_ConveyMerge_Disable_Center	BOOL	4:0387	4	ConveyMerge: Disable Center Line from Releasing
O_ConveyMerge_Disable_Left	BOOL	4:0387	5	ConveyMerge: Disable Left Line from Releasing
O_ConveyMerge_Disable_Right	BOOL	4:0387	6	ConveyMerge: Disable Right Line from Releasing
O_ConveyMerge_Enable_PLC_Cntrl	BOOL	4:0387	15	Enable PLC to Control ConveyMerge Priorty & Release
O_ConveyMerge_Priority	SINT	4:0387	-	0 = First come first served, 1

				= Center line has priority, 2 = Left line has priority, 3 = Right line has priority
O_ConveyStop_Command	INT	4:0020	-	Set Local ConveyStop Command
O_DA_Mode_Cmd_DnZn	SINT	4:0375	Lo Byte	Direction & Accumulation Mode Command Byte for Downstream Zone
O_DA_Mode_Cmd_UpZn	SINT	4:0365	Lo Byte	Direction & Accumulation Mode Command Byte for Upstream Zone
O_DA_Mode_Value_DnZn	SINT	4:0375	Hi Byte	Direction & Accumulation Mode Number of Zones
O_DA_Mode_Value_UpZn	SINT	4:0365	Hi Byte	Direction & Accumulation Mode Number of Zones
O_Enable_Module	BOOL	-	AOI Logic	Enable Output to Module (Reset Protection Mode)
O_Jog_FWD_DnZn	BOOL	4:0184	10	Jog Forward for Local Downstream Zone
O_Jog_FWD_UpZn	BOOL	4:0104	10	Jog Forward for Local Upstream Zone
O_Jog_REV_DnZn	BOOL	4:0184	11	Jog Reverse for Local Downstream Zone
O_Jog_REV_UpZn	BOOL	4:0104	11	Jog Reverse for Local Upstream Zone
O_Maintenance_Stop_DnZn	BOOL	4:0184	13	Force Stop Local Downstream Zone
O_Maintenance_Stop_UpZn	BOOL	4:0104	13	Force Stop Local Upstream Zone
O_Release_DnZn	BOOL	4:0185	AOI Logic	Release and Accumulate on Next at Downstream Zone
O_Release_UpZn	BOOL	4:0105	AOI Logic	Release and Accumulate on Next at Upstream Zone
O_Sensor_Prt_P2_Left_Output_Enable	BOOL	4:0063	12	Enable Sensor Port Pin 2 Left to be an Output

O_Sensor_Prt_P2_Left_Output_Energize	BOOL	4:0063	8	Energize Sensor Port Pin 2 Left, Output Enable must be Energized
O_Sensor_Prt_P2_Right_Output_Enable	BOOL	4:0063	13	Enable Sensor Port Pin 2 Right to be an Output
O_Sensor_Prt_P2_Right_Output_Energize	BOOL	4:0063	9	Energize Sensor Port Pin 2 Right, Output Enable must be Energized
O_Speed_Motor_Left	INT	4:0040	-	Value in mm/sec for MDR or RPM x 10 for PGD
O_Speed_Motor_Right	INT	4:0064	-	Value in mm/sec for MDR or RPM x 10 for PGD
O_Sts_DnZn_Discharge_FWD	SINT	4:0232	AOI Logic	Set Downstream Discharge Zone Forward Status Value
O_Sts_DnZn_Discharge_REV	SINT	4:0232	AOI Logic	Set Downstream Discharge Zone Reverse Status Value
O_Sts_UpZn_Induct_FWD	SINT	4:0134	AOI Logic	Set Upstream Induct Zone Forward Status Value
O_Sts_UpZn_Induct_REV	SINT	4:0134	AOI Logic	Set Upstream Induct Zone Reverse Status Value
O_Tracking_DnZn	DINT	4:0212 (MSW) 4:0213 (LSW)	AOI Logic	Write 32 Bit Tracking Data for Local DnZn When Accum.
O_Tracking_Induct_FWD	DINT	4:0139 (MSW) 4:0140 (LSW)	AOI Logic	Set Forward Induct Tracking Value
O_Tracking_Induct_REV	DINT	4:0237 (MSW) 4:0238 (LSW)	AOI Logic	Set Reverse Induct Tracking Value
O_Tracking_UpZn	DINT	4:0132 (MSW) 4:0133 (LSW)	AOI Logic	Write 32 Bit Tracking Data for Local UpZn When Accum.

| O_WakeUp_DnZn | BOOL | 4:0184 | 12 | WakeUp Local Downstream Zone |
| O_WakeUp_UpZn | BOOL | 4:0104 | 12 | WakeUp Local Upstream Zone |

13.5.1.3. PLC I/O Mode Inputs

Tag Name	Data Type	Module Register	Bit	Description
I_ConveyStopByLostConnection	BOOL	4:0019	6	Stop active due to lost communication connection
I_ConveyStopByPLCCmd	BOOL	4:0019	10	Stop active due to Stop Command from PLC
I_ConveyStopByPLCDisconnect	BOOL	4:0019	7	Stop active due to lost PLC connection
I_ConveyStopByRemoteModule	BOOL	4:0019	5	Stop active on other module in Stop Group
I_Current_Left_MDR	REAL	4:0055	-	Current Draw of Left MDR in Amps
I_Current_Right_MDR	REAL	4:0079	-	Current Draw of Right MDR in Amps
I_Dig_Mtr_Over_Cur_Left	BOOL	4:0060	14	Left: More than 1A detected on one or more outputs
I_Dig_Mtr_Over_Cur_Right	BOOL	4:0084	14	Right: More than 1A detected on one or more outputs
I_Dig_Mtr_Short_Cir_Left	BOOL	4:0060	12	Left: Short Circuit Error on one or more outputs
I_Dig_Mtr_Short_Cir_Right	BOOL	4:0084	12	Right: Short Circuit Error on one or more outputs
I_Frequency_Left_MDR	INT	4:0056	-	Frequency of Left MDR in Hz
I_Frequency_Right_MDR	INT	4:0080	-	Frequency of Right MDR in Hz
I_Heartbeat	BOOL	4:0035	15	Module Heartbeat, 2s On / 2s Off
I_Module_Enabled	BOOL	-	AOI Logic	Local Outputs are Enabled (Reset Protection Mode)
I_Motor_Sts_Left	INT	4:0058	-	Left Module Status
I_Motor_Sts_Right	INT	4:0082	-	Right Module Status
I_Sensor_Detect_Left_Port	BOOL	4:0036	1	Sensor Detected on Left Pin 4
I_Sensor_Detect_Right_Port	BOOL	4:0036	0	Sensor Detected on Right Pin 4
I_Sensor_Port_Left_Pin2	BOOL	4:0035	0	Left Sensor Port Pin 2 Energized (if

				used as input)
I_Sensor_Port_Left_Pin4	BOOL	4:0035	4	Left Sensor Port Pin 4 Energized
I_Sensor_Port_Right_Pin2	BOOL	4:0035	2	Right Sensor Port Pin 2 Energized (if used as input)
I_Sensor_Port_Right_Pin4	BOOL	4:0035	6	Right Sensor Port Pin 4 Energized
I_Servo_CmdComplete_Left	BOOL	4:0011	0	1 = Last Servo Run Command Complete
I_Servo_CmdComplete_Right	BOOL	4:0016	0	1 = Last Servo Run Command Complete
I_Servo_CmdStatus_Left	BOOL	4:0011	2	1 = Running to Pulse Command Position
I_Servo_CmdStatus_Right	BOOL	4:0016	2	1 = Running to Pulse Command Position
I_Servo_Position_Left	INT	4:0062	-	Signed Integer Value for Position from Position "0"
I_Servo_Position_Right	INT	4:0086	-	Signed Integer Value for Position from Position "0"
I_Speed_Left	INT	4:0507	-	Value in mm/sec for MDR or RPM x 10 for PGD Range: depends upon the Ai MDR or PGD connected MDR Example: 400 = 0.40 m/s PGD Example : 400 = 40 RPM 0 = Remain at last non zero value entered
I_Speed_Left_Max	BOOL	4:0507	14	Left Motor Set Speed is Greater Than Motor Maximum
I_Speed_Left_Min	BOOL	4:0507	15	Left Motor Set Speed is Less Than Motor Minimum
I_Speed_Right	INT	4:0508	-	Value in mm/sec for MDR or RPM x 10 for PGD Range: depends upon the Ai MDR or PGD connected MDR Example: 400 = 0.40 m/s PGD Example : 400 = 40 RPM 0 = Remain at last non zero value entered
I_Speed_Right_Max	BOOL	4:0508	14	Right Motor Set Speed is Greater

				Than Motor Maximum
I_Speed_Right_Min	BOOL	4:0508	15	Right Motor Set Speed is Less Than Motor Minimum
I_Temp_MDR_Left	INT	4:0057	Hi Byte	Left Motor Temperature Reading
I_Temp_MDR_Right	INT	4:0081	Hi Byte	Right Motor Temperature Reading
I_Temp_On_Board_Left	INT	4:0057	Lo Byte	Left On Board Temperature Reading
I_Temp_On_Board_Right	INT	4:0081	Lo Byte	Right On Board Temperature Reading
I_Tracking_UpZn	DINT	4:0139 (MSW) 4:0140 (LSW)	AOI Logic	Tracking for Load Discharged from Adjacent Upstream Module
I_Voltage_MDR	REAL	4:0024	-	Voltage Measured on MDR Power Connection
I_Zone_Status_DnModule	INT	4:0232	-	Local Downstream Status
I_Zone_Status_UpModule	INT	4:0134	-	Local Upstream Status

13.5.1.4. PLC I/O Mode Outputs

Tag Name	Data Type	Module Register	Bit	Description
O_Clear_Motor_Error	BOOL	4:0022	0	Clear Motor Error Left & Right
O_ConveyStop_Command	INT	4:0020	-	Set Local ConveyStop Command
O_Digital_Mode_Enable_Left	BOOL	4:0060	15	Enable Digital Output Mode for Left Motor Port
O_Digital_Mode_Enable_Right	BOOL	4:0084	15	Enable Digital Output Mode for Right Motor Port
O_Digital_Out_Left_P3_Energize	BOOL	4:0060	2	Energize Digital Output Pin 3 Left Motor Port
O_Digital_Out_Left_P4_Energize	BOOL	4:0060	1	Energize Digital Output Pin 4 Left Motor Port
O_Digital_Out_Right_P2_Energize	BOOL	4:0084	2	Energize Digital Output Pin 2 Right Motor Port
O_Digital_Out_Right_P3_Energize	BOOL	4:0084	1	Energize Digital Output Pin 3 Right Motor Port
O_Enable_Module	BOOL	-	AOI Logic	Enable Output to Module (Reset Protection Mode)
O_Mtr_Accel_Ramp_Left	INT	4:0043	-	Value in mm for MDR or Pulses for PGD
O_Mtr_Accel_Ramp_Right	INT	4:0067	-	Value in mm for MDR or Pulses for PGD
O_Mtr_Brake_Method_Left	INT	4:0261	-	Left Motor Control
O_Mtr_Brake_Method_Right	INT	4:0271	-	Right Motor Control
O_Mtr_Decel_Ramp_Left	INT	4:0044	-	Value in mm for MDR or Pulses for PGD
O_Mtr_Decel_Ramp_Right	INT	4:0068	-	Value in mm for MDR or Pulses for PGD
O_Mtr_Dir_Left	BOOL	4:0260	8	On = Run in Opposite of Config. Dir.

O_Mtr_Dir_Right	BOOL	4:0270	8	On = Run in Opposite of Config. Dir.
O_Mtr_Run_Left	BOOL	4:0260	0	On = Run
O_Mtr_Run_Right	BOOL	4:0270	0	On = Run
O_Mtr_Slave_Mode_Left	INT	4:0262	-	0 = Ignore 1 = OFF: Left motor independently controlled 2 = ON: Left motor mirrors Right motor control
O_Mtr_Slave_Mode_Right	INT	4:0272	-	0 = Ignore 1 = OFF: Right motor independently controlled 2 = ON: Right motor mirrors Left motor control
O_Mtr_Speed_Left	INT	4:0040	-	Value in mm/sec for MDR or RPM x 10 for PGD
O_Mtr_Speed_Right	INT	4:0064	-	Value in mm/sec for MDR or RPM x 10 for PGD
O_Sensor_Prt_Mask_P2_Left	BOOL	4:0034	0	Invert the Active State for the Pin
O_Sensor_Prt_Mask_P2_Right	BOOL	4:0034	2	Invert the Active State for the Pin
O_Sensor_Prt_Mask_P4_Left	BOOL	4:0034	4	Invert the Active State for the Pin
O_Sensor_Prt_Mask_P4_Right	BOOL	4:0034	6	Invert the Active State for the Pin
O_Sensor_Prt_P2_Left_Output_Enable	BOOL	4:0037	5	Enable Sensor Port Pin 2 Left to be an Output
O_Sensor_Prt_P2_Left_Output_Energize	BOOL	4:0037	0	Energize Sensor Port Pin 2 Left, Output Enable must be Energized
O_Sensor_Prt_P2_Right_Output_Enable	BOOL	4:0037	6	Enable Sensor Port Pin 2 Right to be an Output
O_Sensor_Prt_P2_Right_Output_Energize	BOOL	4:0037	1	Energize Sensor Port Pin 2 Right, Output Enable must be Energized

O_Servo_CmdPosition_Left	INT	4:0008	-	Value in mm for MDR or Pulses for PGD
O_Servo_CmdPosition_Right	INT	4:0013	-	Value in mm for MDR or Pulses for PGD
O_Servo_GoCmd_Left	BOOL	4:0009	1	Go to Commanded Position from Position "0"
O_Servo_GoCmd_Right	BOOL	4:0014	1	Go to Commanded Position from Position "0"
O_Servo_Zero_Left	BOOL	4:0009	0	Set Current Pulse Count as "0"
O_Servo_Zero_Right	BOOL	4:0014	0	Set Current Pulse Count as "0"
O_Tracking_Discharge	DINT	4:0201 (MSW) 4:0202 (LSW)	AOI Logic	Tracking for Load Being Discharged
O_Zone_Sts_DnZn	INT	4:0196	-	Zone Status to Downstream Module
O_Zone_Sts_UpZn	INT	4:0116	-	Zone Status to Upstream Module

13.6. Using Logix 5000 MSG Instruction

Access to ConveyLinx-Ai Family Controllerss is also available utilizing the Logix 5000 *MSG* instruction. The *MSG* instruction utilizes *CIP Explicit Messaging*. This means that the connection is not maintained as an implicit connection. *Generic Ethernet Module* and *EDS* connections are *implicit* and thus must be maintained at all times or there will be a communication fault. *Explicit Messaging* opens the connection, reads/writes data, and then closes the connection thus freeing up communications resources for the PLC.

When to use MSG Instructions

Because the *MSG* instruction is executed asynchronous to program scan and is not subject to implicit messaging RPI restrictions; the response time between requesting data and receiving data is not deterministic and can vary between separate requests for the same data from the same device. Therefore, **we recommend that** *MSG* **instructions should** <u>not</u> **be used for dedicated "real time" control of equipment**. For ConveyLinx-Ai Family Controllerss, *MSG* instructions are intended to gather "low priority" status information and/or to send infrequent parameter changes. Please note that this is only a recommendation. Your particular application's specifics, PLC's capacity, available network bandwidth, etc. may allow you to get expected results with "real time" control utilizing *MSG* instructions to interface with ConveyLinx-Ai Family Controllerss.

Module VS. Assembly Addresses with MSG Instruction

Module Addresses are the 512 "actual" registers in the module and [Assemblies](#) are "virtual" registers grouped based upon function. There are certain restrictions on what you can do with a MSG Instruction:

- You can us a single MSG instruction to **read** one and up to 30 consecutive ConveyLinx-Ai Family Controllers registers
- You can use a single MSG instruction to **write** to one (an only one) of the ConveyLinx-Ai Family Controllers registers
- You can use a single MSG instruction to **read** any of the available *Input Assembly* registers in their entirety
- You **CANNOT** use a MSG instruction to **write** to an *Output Assembly* array of registers.

13.6.1. Read MSG Instruction

For this example, we have created a ConveyLinx-Ai Family Controllers in our Ethernet Tree and added a MSG Instruction to a ladder rung. We named this MSG block "ReadStatus" and created the Controller Tag for it. With this MSG Instruction we are going to read Module Address 4:0088 and 4:0089 which are Module Status Word 1 and 2. We also created a Controller Tag of data type DINT we named "TempDiagStatus" to hold this data. Data from 4:0088 will be in the High Word of the DINT and 4:0089 will be in the low word of the DINT.

1	From the Message Type drop down, select *CIP Generic*
2	From the Service Type drop down, select *Get Attribute Single*
3	For the Instance Field, enter "88"

④	For the Class Field, enter "64"
⑤	For the Attribute Field, enter "2" because we want to read 4:0088 and 4:0089
⑥	For the Destination Element, use the Tag Browser drop down to select *TempDiagStatus*

13.6.2. Write MSG Instruction

For this example, we have created a ConveyLinx-Ai Family Controllers in our Ethernet Tree and added a MSG Instruction to a ladder rung. We named this MSG block "WriteSpeed" and created the Controller Tag for it. With this MSG Instruction we are going to write a value for the Left Motor Speed at Module Address 4:0040. We also created a Controller Tag of data type INT we named "NewSpeed" which will hold the speed value we want to write to the module when we execute the MSG Instruction.

❶	From the *Message Type* drop down, select *CIP Generic*
❷	From the *Service Type* drop down, select *Set Attribute Single*
❸	For the *Instance* Field, enter "40"

4	For the *Class* Field, enter "64"
5	For the *Attribute* Field, enter "1"
6	For the *Source Element*, use the Tag Browser drop down to select *NewSpeed*
7	For the *Source Length*, enter "2" (2 bytes = 1 INT)

! Please note that when using a MSG Instruction to write data that the ConveyLinx registers are <u>NOT</u> zero protected!

13.6.3. Reading an Input Assembly with MSG Instruction

For this example, we have created a ConveyLinx-Ai Family Controllers in our Ethernet Tree and added a MSG Instruction to a ladder rung. We named this MSG block "ReadZPA" and created the Controller Tag for it. With this MSG Instruction we are going to read all 21 registers from PLC Inputs for ZPA Mode. We also created a Controller Tag called "ZPA_Inputs" which is an INT array of 21 elements in order to hold the data.

> ✱ When reading any input Assembly with a MSG Instruction, be sure you create the Destination Element Controller Tag an array of INT with the number of elements that match the number of registers in the Assembly

1	From the Message Type drop down, select *CIP Generic*
2	From the Service Type drop down, select *Get Attribute Single*
3	For the Instance Field, enter "105" which is the Assembly vale for the ZPA Mode inputs
4	For the Class Field, enter "4"
5	For the Attribute Field, enter "3"
6	For the Destination Element, use the Tag Browser drop down to select *ZPA_Inputs* and expand to select the [0] element

13.7. EDS Module Data Type Cross Reference

When you create an instance of a device from an EDS file in your RSLogix 5000 environment; the EDS file provides a *Module Defined Data Type* for the inputs and outputs of the device. This *Module Defined Data Type's* name is automatically generated by the EDS file creation's software and is based upon (among other things) a checksum of the items in the file. This often creates a somewhat cryptic alpha-numeric string for the name.

In situations where you may inherit an existing program and your RSLogix 5000 environment is missing the EDS file used for this program; you will need to determine which version of EDS file was used and then go find it on our Pulseroller.com web site. Similarly, you may also have the situation where you need to be able to match your AOI version to its correct EDS file. The following chart cross references the most common EDS files, AOI files, and Module Data Types

EDS File	AOI File	Module Data Type
ConveyLinxAi_V5_1.eds	Ai2_ZPA_AOI_4_5.L5X	_055C:Ai2_84EED4D3:I:0
		_055C:Ai2_3015BAF1:O:0
	Ai2_PLCIO_AOI_4_5.L5X	_055C:Ai2_1F7F96C5:I:0
		_055C:Ai2_7DDD1BFA:O:0

✱ For versions or Module Data Types not shown, please contact support@pulseroller.com

14. Connecting to Siemens PLC with Profinet IO

> **!** This section assumes you have a solid working knowledge of both Siemens PLC's and the TIA Portal or Step7 manager development environments.

> **!** This section applies to all ConveyLinx Family modules. Figures and software screenshots may only show ConveyLinx-Ai Family examples, but the concepts are applicable to all ConveyLinx modules

S7 PLCs from Siemens can use *ConveyLinx* modules because they support **PROFINET IO** communication protocol and can act as PROFINET IO-devices with the S7 PLC acting as a PROFINET IO controller. The PLC can connect in different ways to the same *ConveyLinx* module based upon how you want to use it. Each of these different ways that the PLC can connect to a given module is called a <u>Device Access Point (DAP)</u>. For a given *ConveyLinx* module type or family, your PLC programming environment needs a <u>GSDML file</u> installed so your environment can display each of the *DAP's* available for that given module type or family. Each *ConveyLinx* module family (ConveyLinx-ERSC, ConveyLinx-Ai, and ConveyLinx-ECO) has their own *GSDML file* that contains all of its *DAP* information. Each *DAP* corresponds to a **Mode of Operation** of the ConveyLinx module.

Preparing your Programming Environment
Understanding the ConveyLinx Module's Modes of Operation

Device Access Points (DAP)

The ProfiNet protocol allows for multiple access ways for a single physical device. This is called a *Device Access Point* or *DAP*. In essence each DAP allows the PLC to connect to the same physical device in a different way from the other DAPs for that device. The number of DAPs a device supports is vendor specific so every manufacturer decides how many access ways to provide for their device.

After you install the appropriate *GSDML* file into your programming environment based upon the ConveyLinx Family you are using, there will be two directories (or folders) of *DAPs*:

1. Contains the DAPs that are used when you want to use *Separate ConveyLinx Auto-Configuration* by either Install Button (ConveyLinx-ERSC Family) or by EasyRoll (ConveyLinx-Ai Family and ConveyLinx-ECO)
2. Contains DAPs that are used when you want to use *Integrated Configuration Direct from PLC Topology* to detect the topology of your ConveyLinx network and configure all

modules

14.1. Preparing Your Programming Environment

In order to use ConveyLinx with your PLC, you need to install the proper **configuration files** into your **TIA Portal** or **STEP 7** programming environment. The files required will be included in a downloadable **.zip archive from our site and will include an xml with the filename structure "GSDML-V2.xx-IndustrialSoftware-ConveyLinx-yyyymmdd.xml" and at least 1 bitmap file** (.bmp) with the graphic image of the particular module(s) for which the xml file specifies for your Siemens programming environment.

There are also optional (but highly recommended) **User Data Type (UDT)** files specific for each module type and **Mode of Operation** you wish to use.

> ✳ The files you will need are dependent on the specific Conveylinx module and module firmware version you are using.

14.2. Modes of Operation

When installed, ConveyLinx modules can be configured as one of the following 6 basic modes:

Full ZPA Mode

n this mode, ConveyLinx works as a **one or two zone ZPA controller**. When communicating with module in this mode, the S7-300 PLC can control mainly material handling properties such as accumulation, release, wake-up/full stop, read/write tracking, MDR parameters (speed, acceleration and deceleration), and ConveyStop. In this mode the PLC does not have access to directly control start/stop of MDRs because internal ZPA logic is operating the module. When communicating with ConveyLinx in ZPA mode, the PLC does not require fast reaction time as would be expected for a remote I/O device. Typically, 32 ms to 256 ms is the range of response time utilized between the PLC and a ConveyLinx module in ZPA mode. The data array instance lengths for input and output data mode are 64 bytes.

Reduced ZPA Mode

In this mode, the ConveyLinx modules operates as a **one or two zone ZPA controller** just like the Full ZPA Mode. However, the instance data presented to the S7-300 PLC is limited to basic material handling properties such as accumulation, release, wake-up/full stop, and MDR speed. In this mode, because the instances length is reduced to 30 bytes; the PLC may be able to accommodate more modules especially in applications where all ZPA features (such as tracking) are not required. Similarly to the Full ZPA Mode, the 32 ms to 256 ms response time is typical for this mode.

Merger Mode

In this mode, the module is running in **ZPA mode** and exchanging 64 bytes of data with the PLC. In addition to the ZPA logic, the module now runs the Merger task on one of their zones. Configuring the Merger zone, differs between the two configuration methods (see Configuration methods chapter). Similarly to the Full ZPA Mode, the 32 ms to 256 ms response time is typical for this mode. The exchanged data is the same as for the Full ZPA mode.

Full PLC Controlled Mode

In this mode, the ConveyLinx module is placed into **PLC I/O mode** with the EasyRoll software tool. When the module is in PLC I/O mode, **all internal ZPA logic is suspended** and the module requires an external PLC to read port inputs and run/stop MDRs connected to the module. The PLC contains any and all logic to process inputs from sensor/control ports, run stop MDRs, set direction/speed/accel/decel of MDRs, and more. Also, in this mode the PLC has access to registers in the PLC I/O module that can interface to neighboring ZPA modules for material

handling control and tracking data support. A ConveyLinx module in PLC I/O mode is also capable of **Servo functions** that are made available to the PLC. In this mode, the ConveyLinx module responds as would be expected for a remote I/O device. ConveyLinx will connect to the PLC with a 4 ms communication response time. The data array instance lengths for input and output data in this mode are 64 bytes.

Reduced PLC Controlled Mode

In this mode, the ConveyLinx module is configured and operates **without ZPA logic** the same as it does in Full PLC Controlled Mode. However in this mode, only port I/O and MDR run/stop/ speed functions are available to the PLC. In this mode, because the instances length is reduced; the PLC may be able to accommodate more modules especially in applications where all the features in Full PLC Controlled Mode are not required. ConveyLinx modules in this mode communicate at the same 4 ms response time as the Full PLC Controlled Mode. The data array instance lengths for input and output data in this mode are 16 bytes

PLC Controlled Mode with ConveyLogix Program

In this mode, the ConveyLinx module has been programmed with ConveyLogix and this program is running inside. ConveyLogix allows you to write your own programs and download it to the ConveyLinx module turning it into a small PLC. The ToPLCArray(WORD [16]) and FromPLCArray(WORD [16]) are exchanged with the PLC. What data is exchanged in those data arrays, is programmed by the engineer in the ConveyLogix program and in the PLC. The communication response time should be adjusted by the programmer, based on the size and complexity of the ConveyLogix program. The bigger it is, the higher update speeds should be selected.

14.3. Understanding the Two Configuration Methods

> ❗ Before going further, you will need to decide which of these two methods you are going to use!

In general, a network of ConveyLinx modules must be *Auto-Configured* before they can work together in ZPA mode or exchange any data with each other when in PLC I/O mode. The typical method is to utilize the *Install Button* (on ERSC family modules) or *EasyRoll* (for ConveyLinx-Ai Family and ConveyLinx-ECO). When you use Siemens S7 PLCs and TIA Portal or Step 7 Manager, you can have the ConveyLinx modules automatically **configured from the PLC based upon the topology** you create at design time in your Siemens PLC programming environment.

Separate ConveyLinx Auto-Configuration

When using this method, the configuration of a module or group of modules is initiated with the *Install Button* or *EasyRoll* software. All of the module parameters such as motor modes (ECO/BOOST etc.), module mode (PLC/ZPA), inter-module connections, Merger zone, IP addresses and all other general parameters are assigned during the *Auto-Configuration* procedure. The Profinet name is formed by the module during the Auto-Configuration procedure as well. In order write your program in the PLC, <u>you will need to know the exact Profinet name of each module you wish to connect and you do not have the flexibility to change these names.</u>

Advantages of Separate ConveyLinx Auto-Configuration

* Changing parameters like motor settings, ZPA functionality, etc. can be done for multiple modules all at once with EasyRoll
* Your PLC only has to connect to modules it needs to interface with and not every module in the network
* [ZPA Extension](#) function is able to be configured

Disadvantages to Separate ConveyLinx Auto-Configuraton:

* Profinet name of the module is set during Auto-Configuration and <u>cannot be modified</u>
* Your PLC must use the <u>exact syntax</u> of the module's Profinet name in order to connect
* You have to know in advance your exact network topology and I.P. addressing prior to programming your PLC

Integrated Configuration Direct from PLC Topology

> **!** This method is only available for ConveyLinx-ERSC Family for 4 series firmware 4.27 and higher and 5 series firmware 5.03 and higher

When using this method, you create your network topology within the Siemens PLC programming environment (TIA Portal or STEP 7). All pertinent module parameters become available within your environment and can be configured at design time. As long as your physical installed network topology of control modules matches your designed topology in your PLC programming environment; when you download your project to your PLC, the PLC takes care of all of the ConveyLinx control module configuration.

Advantages of Configuration from PLC Topology:

* No module naming restrictions
* No need to use separate EasyRoll software
* Module replacement is true plug and play because PLC handles the procedure
* All module parameters are integrated into the Siemens PLC programming environment

Disadvantages to Configuration from PLC Topology:

* You have to include all ConveyLinx modules in your topology regardless of whether you need to interact with them with your PLC logic for your system
* No built in ability to change parameters for multiple modules at the same time; you have to construct own PLC logic to do this
* ZPA Extension function is not available

14.4. Separate ConveyLinx Auto-Configuration

> **!** This section applies ONLY when you use Separate ConveyLinx Auto-Configuration (EasyRoll or ConveyLinx module Install button)

Adding ConveyLinx Modules to Your Project

Once you have installed the *GSDXML* file into your environment, ConveyLinx devices are available to be used in your project. You can find ConveyLinx devices in the *Hardware Catalog* window

Drill down the tree structure to get to the folder *Conveyor Control with EasyRoll configuration* to see the available DAPs
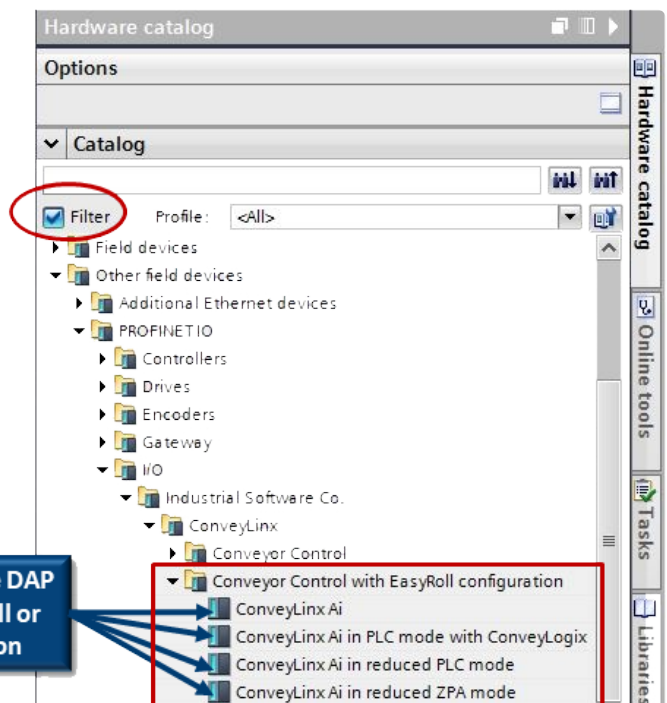
> **✱** The **Mode of Operation** of a given ConveyLinx module will determine which of the *DAPs* to drag and drop from the *Hardware Catalog* window into your project

14.4.1. Profinet Name

When ConveyLinx modules are *Auto-Configured* from either *EasyRoll* or by *Install button*, each module automatically generates its own specific **Profinet Names** each with an <u>exact syntax</u> that corresponds to the *Mode of Operation* (*ZPA* or *PLC I/O*) determined during Auto-Configuration. If you performed Auto-Configuration with an Install button, each module defaults to *ZPA Mode* and the *ZPA Profinet Names* will be automatically generated. If you use *EasyRoll* to change a module from ZPA to *PLC I/O Mode*, the **Profinet Names** are automatically updated for *PLC I/O Mode*

The following chart shows the exact syntax for Profinet Names:

Mode	Syntax	Remarks
Full ZPA	conveylinxzpa-xxx-yyy	xxx = 3rd Octet of I.P. Address
Reduced ZPA		
Full PLC Controlled	conveylinxplc-xxx-yyy	yyy = 4th Octet of I.P. Address
Reduced PLC Controlled		

> **!** With ConveyLinx Auto-Configured modules, you <u>cannot change the Profinet Name</u> in the module and you <u>MUST</u> use the <u>EXACT Profinet Name</u> in your Siemens programming environment in order to communicate with a given module.

> **✱** Please do not confuse the module's *General Name* or *Project Name* that you assign for a given device with its **Profinet Name**. The **Profinet Name** is required to be in the proper syntax or the module will not connect to the PLC. The module's *Project Name* is completely user's choice to identify the module in the your Siemens programming environment and is what is displayed in your project's Network View(s) and Project Tree.

14.4.2. Examples of Adding Modules

In this section we will provide an example that will add a single module of each Mode to a project. This will require a combination of selecting the proper *DAP* from the Hardware catalog coupled with using the correct *Profinet Name* to achieve our desired result. The following chart lists the 6 modules we want to add to our project and the proper Device and Profinet Name syntax needed.

| Module Mode	I.P. Address	DAP from Hardware Catalog	Profinet Name
Full ZPA	192.168.0.20	ConveyLinx-Ai	conveylinxzpa-0-20
Full PLC Controlled	192.168.101.21	ConveyLinx-Ai	*conveylinxplc-101-21*
Reduced ZPA	192.168.101.22	ConveyLinx-Ai in reduced ZPA mode	*conveylinxzpa-101-22*
Reduced PLC Controlled	192.168.20.123	ConveyLinx-Ai in reduced PLC mode	*conveylinxplc-20-123*
Full ZPA with Merger	192.168.0.21	ConveyLinx-Ai	*conveylinxzpa-0-21*
ConveyLogix Interface	192.168.0.22	ConveyLinx-Ai in PLC mode with ConveyLogix	*conveylogix-0-22*

✱ The following sections will show the details of adding each of these items to you TIA Portal environment
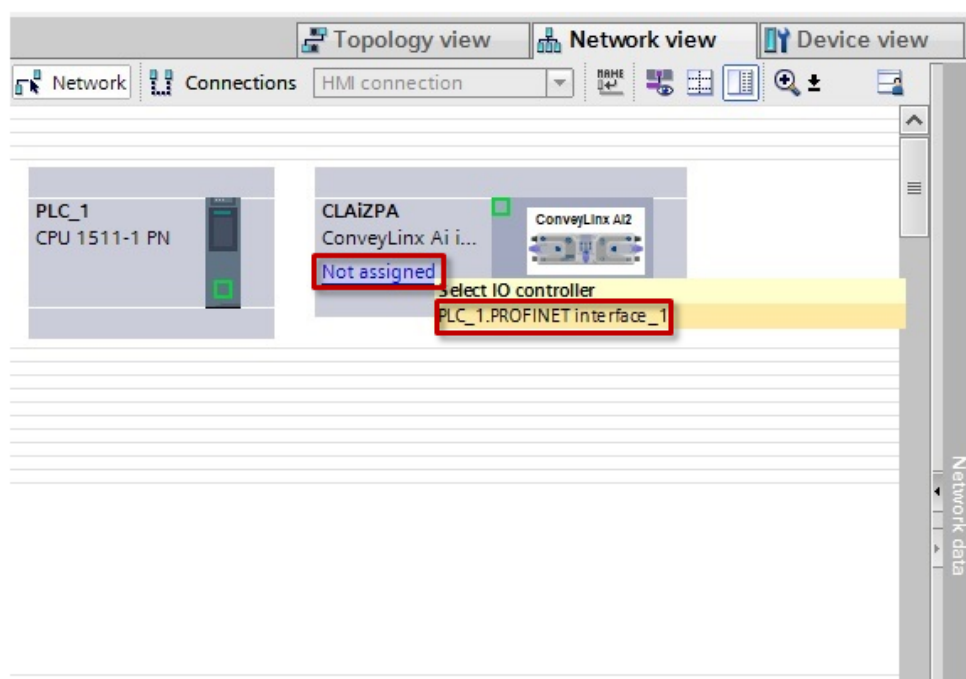
14.4.2.1. Full ZPA

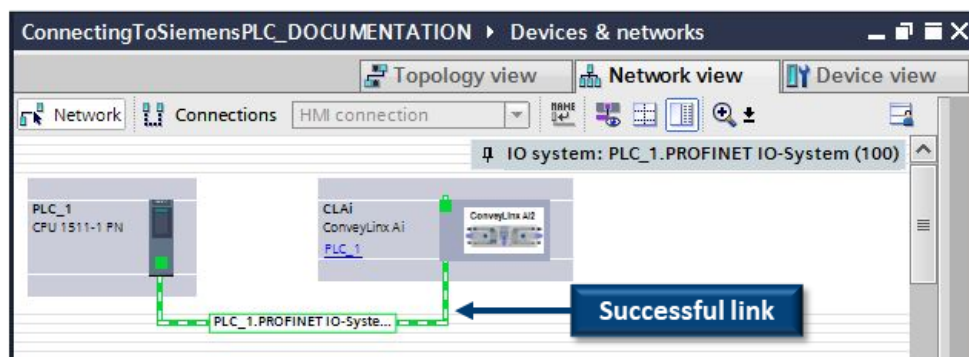## STEP 1: Add the Module to the PLC Network

From the *Catalog* window, drag and drop the *ConveyLinx-Ai* DAP into the *Network View* window
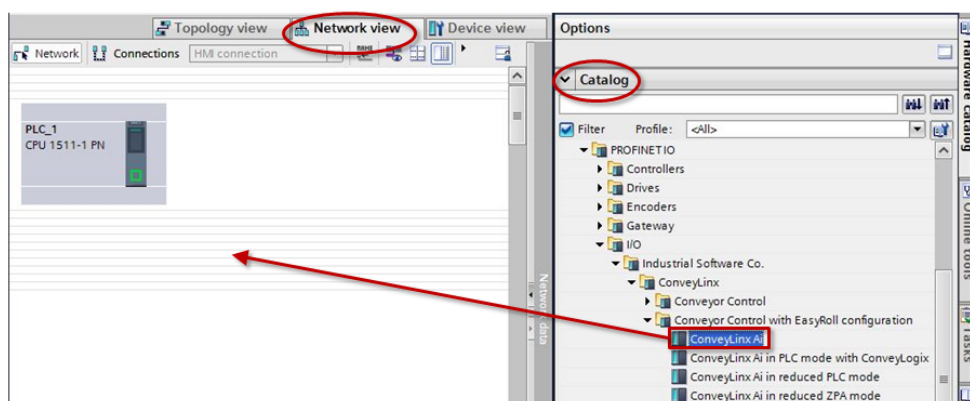
To attach the module to the PLC, click on the Not Assigned link and select the desired Profinet Interface of the PLC
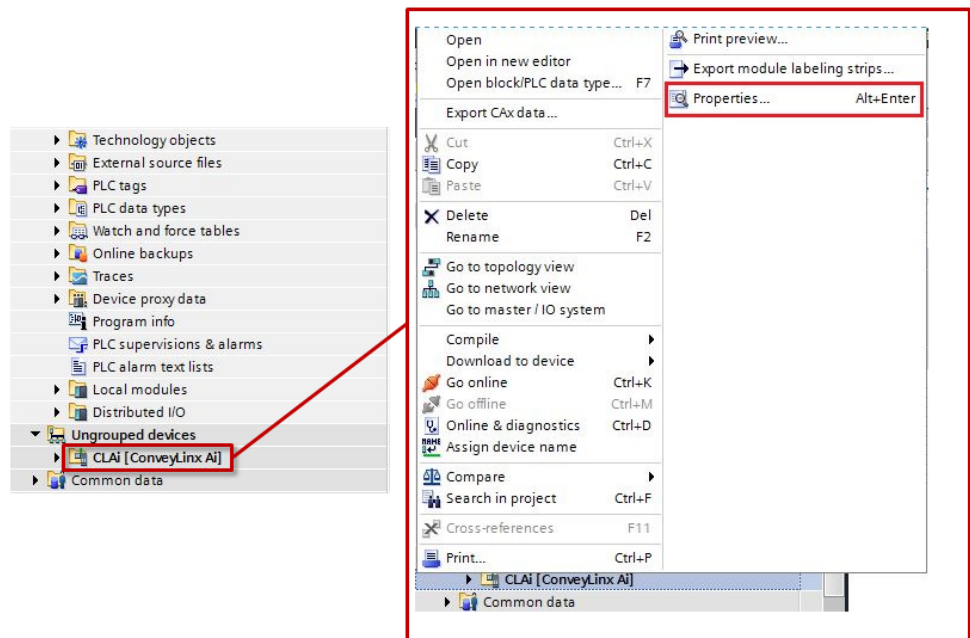

When attached, you will see the green graphic linking the PLC and the module together

# STEP 2: Assign the EXACT Profinet Name to the Module

In the *Project Tree* locate your newly created module under the *Ungrouped devices* folder. Right Click on your module to show a pop-up menu and select *Properties*

A new window will open. Select *General* from the menu on the left. In the *Name* field enter the required name for the device. In our example it is *conveylinxzpa-0-20* from the Example Table
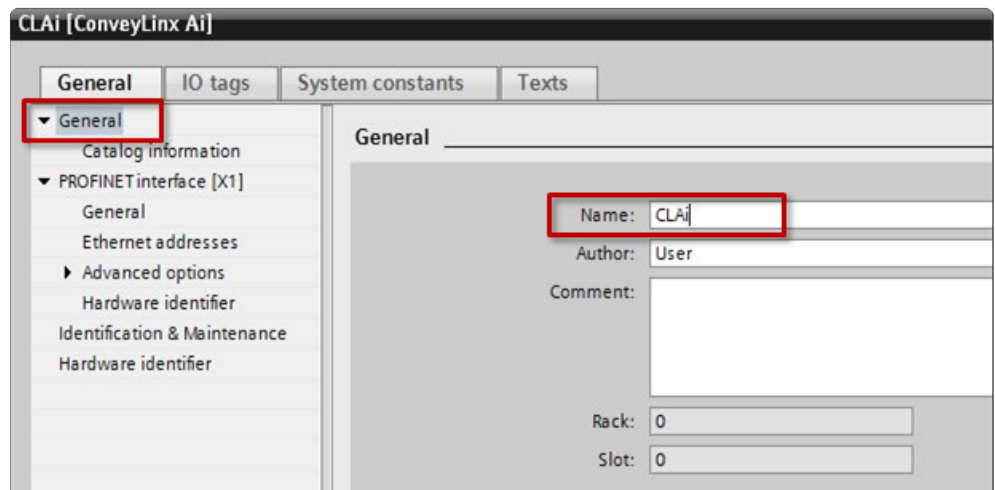
Type in the **EXACT** *Profinet Name* and press enter to make the change
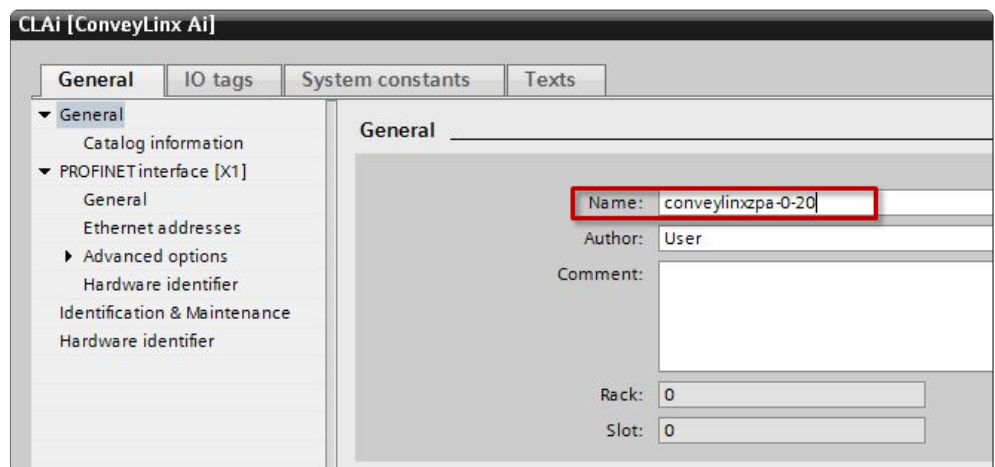
STEP 3: Enter the I.P. Address

In the same properties area that you changed the *Profinet Name*, click on *Ethernet addresses* and enter the I.P. address. In our example it is *192.168.0.20* from the Example Table

STEP 4: Select Update Time Interval

In the same properties area that you changed the *Profinet Name*, click on *Advanced options* and select a proper Update Time. For a ConveyLinx module in ZPA mode, the recommended Update time should be between 32ms and 512ms.

14.4.2.2. Full PLC Controlled

STEP 1: Add the Module to the PLC Network

From the *Catalog* window, drag and drop the *ConveyLinx-Ai* DAP into the *Network View* window
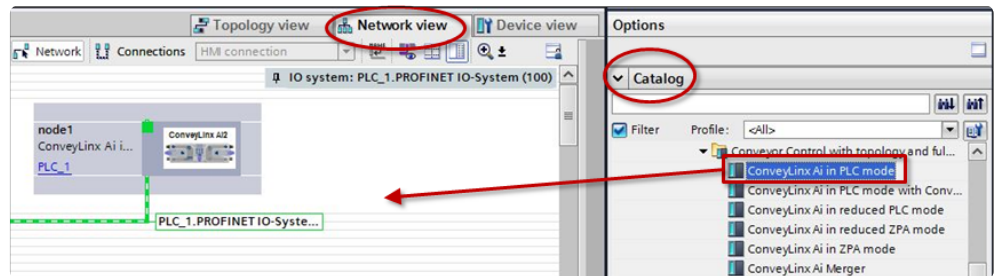
To attach the module to the PLC, click on the Not Assigned link and select the desired Profinet Interface of the PLC
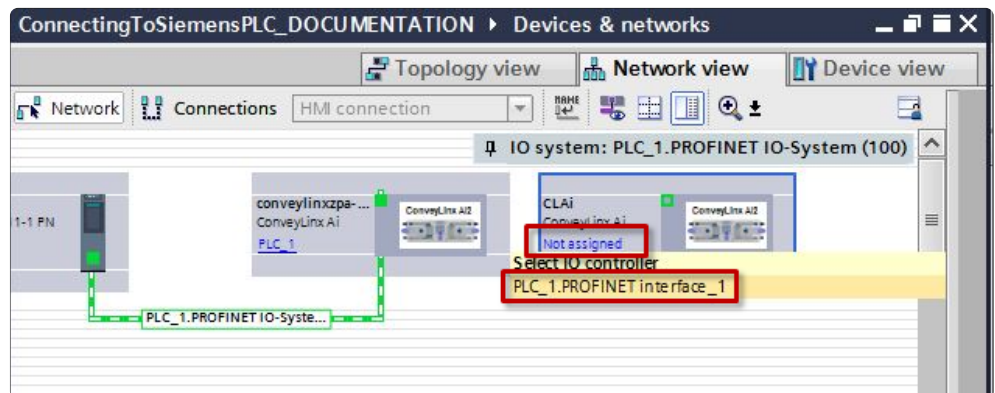
When attached, you will see the green graphic linking the PLC and the module together
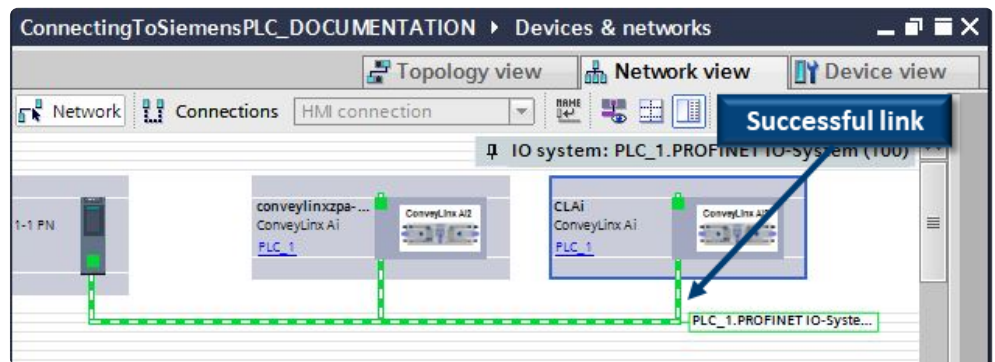
STEP 2: Assign the EXACT Profinet Name to the Module

Follow the same STEP 2 as in Full ZPA Example. For this example the correct *Profinet Name* per our Example Table is *conveylinxplc-101-21*

STEP 3: Enter the I.P. Address

Follow the same STEP 3 as in Full ZPA Example. For this example the correct I.P. Address per our Example Table is *192.168.101.21*

STEP 4: Select Update Time Interval

Follow the same [STEP 4 as in Full ZPA Example](#). Because this module's I/O and motors are under complete PLC control, the fastest Update time of 4ms is recommended.

14.4.2.3. Reduced ZPA

STEP 1: Add the Module to the PLC Network

From the *Catalog* window, drag and drop the *ConveyLinx-Ai in reduced ZPA mode* DAP into the *Network View* window
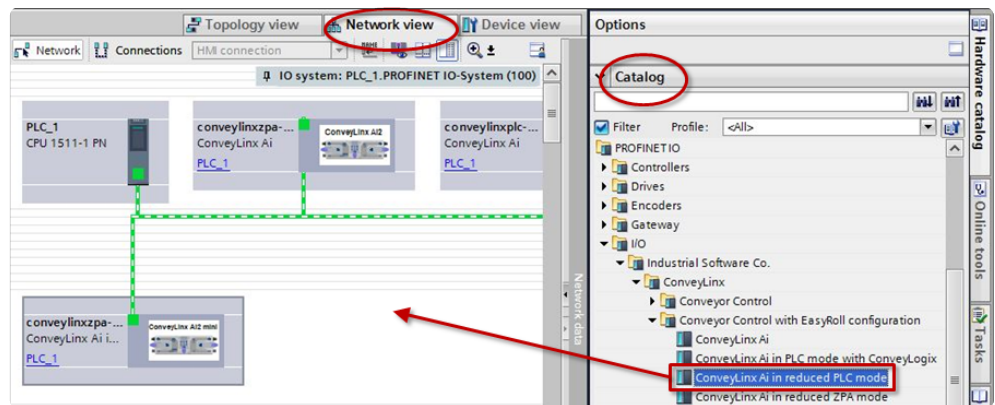
To attach the module to the PLC, click on the Not Assigned link and select the desired Profinet Interface of the PLC
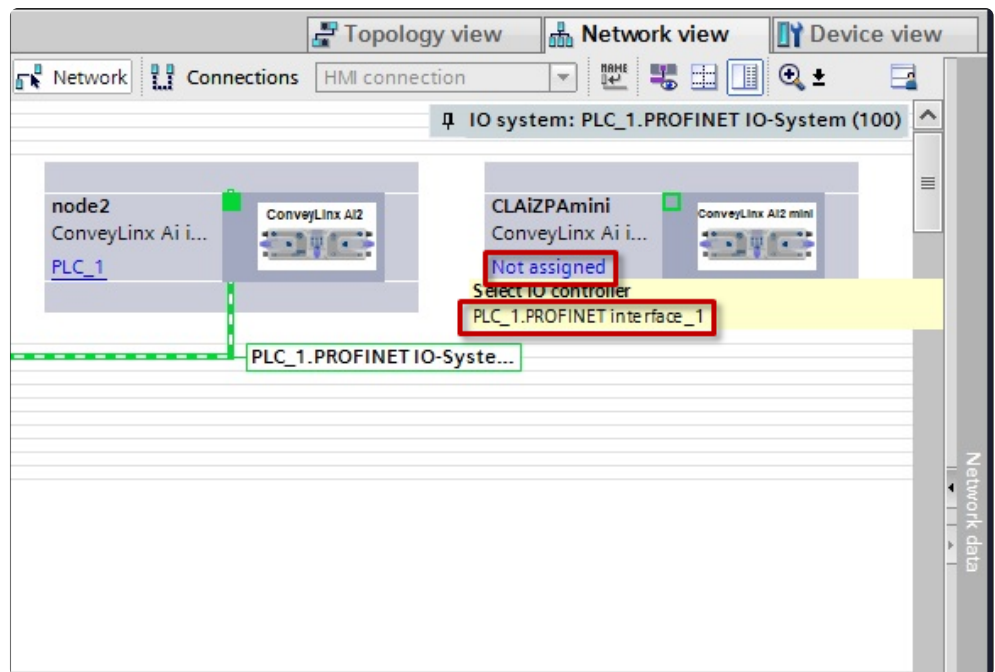
When attached, you will see the green graphic linking the PLC and the module together
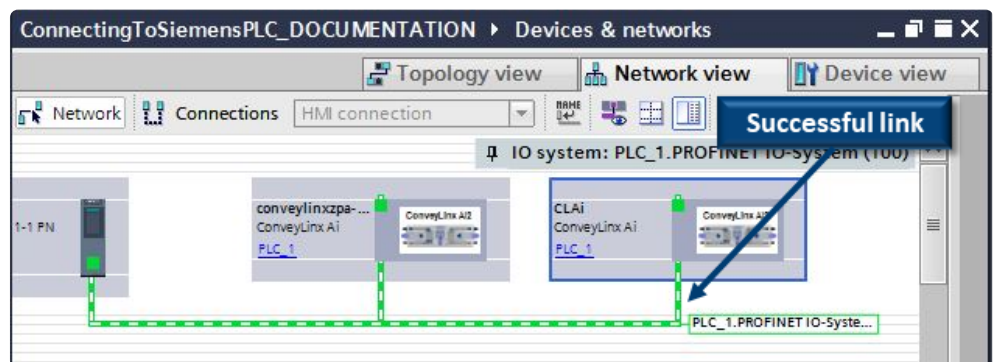
STEP 2: Assign the EXACT Profinet Name to the Module

Follow the same STEP 2 as in Full ZPA Example. For this example the correct *Profinet Name* per our Example Table is *conveylinxzpa-101-22*

STEP 3: Enter the I.P. Address

Follow the same STEP 3 as in Full ZPA Example. For this example the correct I.P. Address per our Example Table is *192.168.101.22*

STEP 4: Select Update Time Interval

Follow the same STEP 4 as in Full ZPA Example. Because this module is in ZPA mode, the recommended Update time to be between 32ms and 512ms.

14.4.2.4. Reduced PLC

STEP 1: Add the Module to the PLC Network

From the *Catalog* window, drag and drop the *ConveyLinx-Ai in reduced ZPA mode* DAP into the *Network View* window

To attach the module to the PLC, click on the Not Assigned link and select the desired Profinet Interface of the PLC

When attached, you will see the green graphic linking the PLC and the module together

STEP 2: Assign the EXACT Profinet Name to the Module

Follow the same STEP 2 as in Full ZPA Example. For this example the correct *Profinet Name* per our Example Table is *conveylinxplc-20-123*

STEP 3: Enter the I.P. Address

Follow the same STEP 3 as in Full ZPA Example. For this example the correct I.P. Address per our Example Table is *192.168.20.123*

STEP 4: Select Update Time Interval

Follow the same STEP 4 as in Full ZPA Example. Because this module's I/O and motors are under complete PLC control, the fastest Update time of 4ms is recommended.

14.4.2.5. Merger Mode

The *Merge* function is configured in *EasyRoll*. Follow the exact same steps as for a [Full ZPA](#) module to add a module that will eventually be configured as merging zone in *EasyRoll*

14.4.2.6. PLC Controlled with ConveyLogix Interface

STEP 1: Add the Module to the PLC Network

From the *Catalog* window, drag and drop the *ConveyLinx-Ai in PLC mode with ConveyLogix* DAP into the *Network View* window
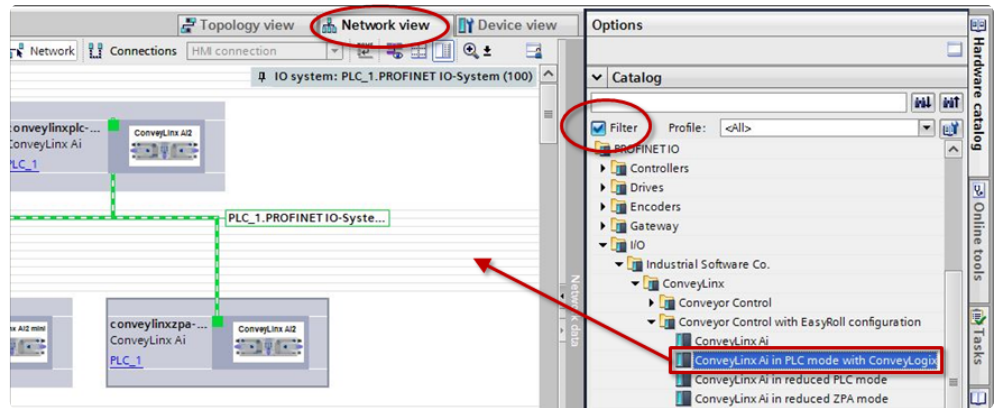
To attach the module to the PLC, click on the Not Assigned link and select the desired Profinet Interface of the PLC
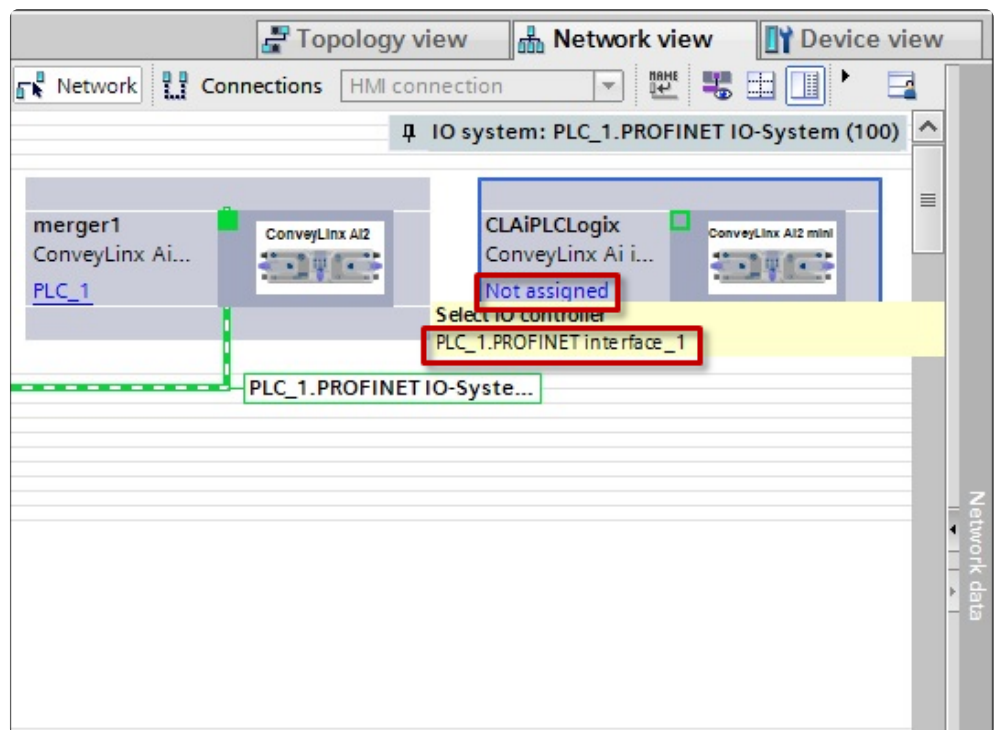
When attached, you will see the green graphic linking the PLC and the module together
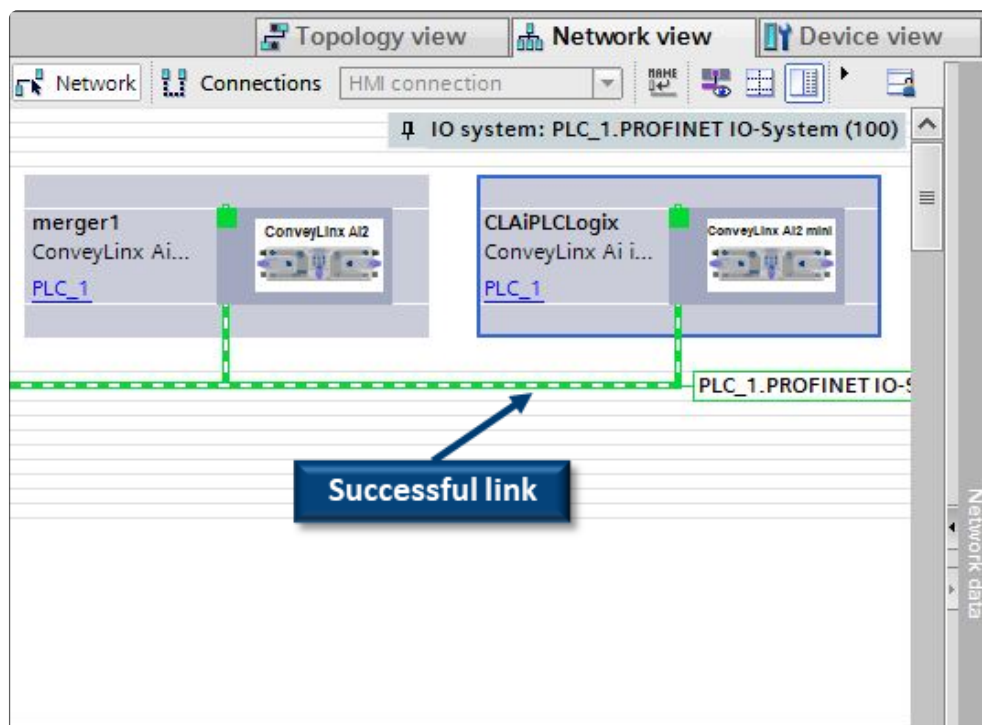
STEP 2: Assign the EXACT Profinet Name to the Module

Follow the same STEP 2 as in Full ZPA Example. For this example the correct *Profinet Name* per our Example Table is *conveylogix-0-22*

STEP 3: Enter the I.P. Address

Follow the same STEP 3 as in Full ZPA Example. For this example the correct I.P. Address per our Example Table is *192.168.0.22*
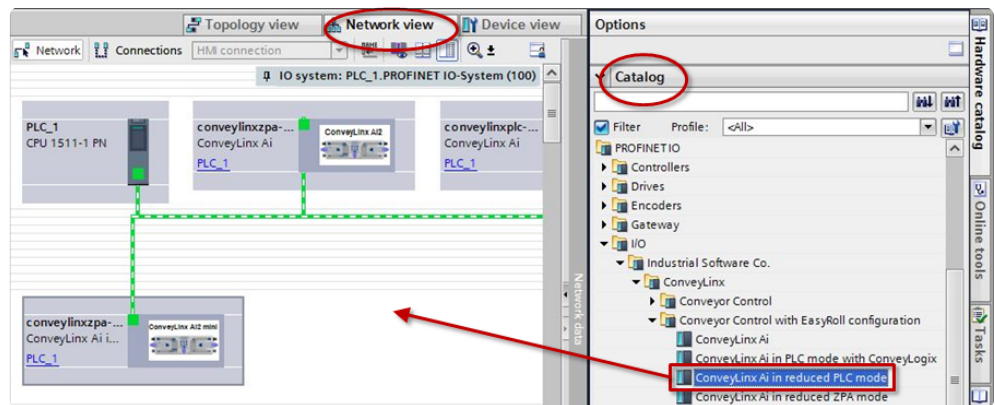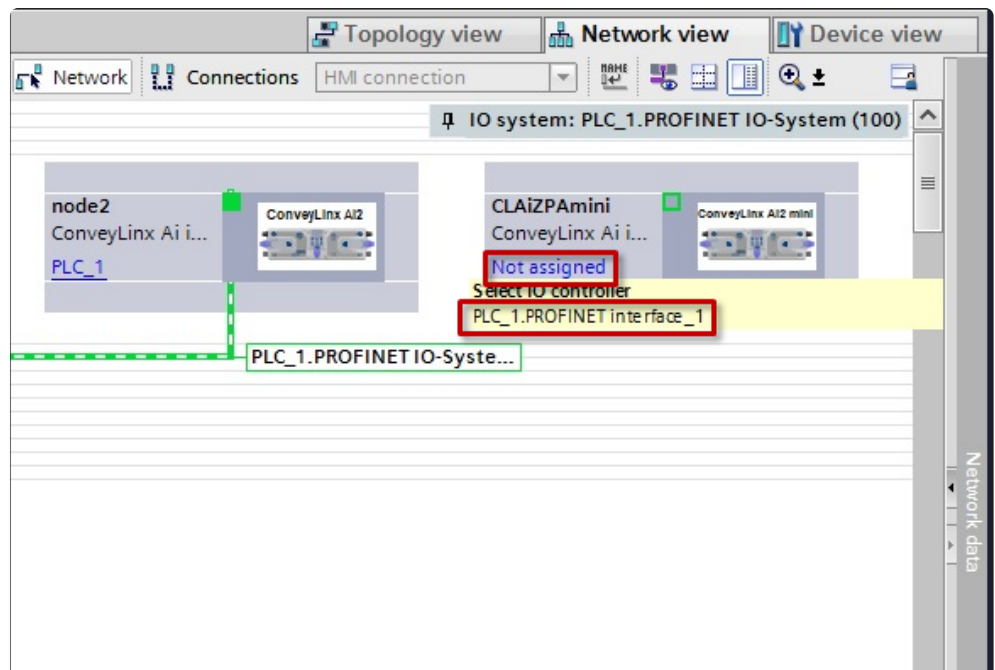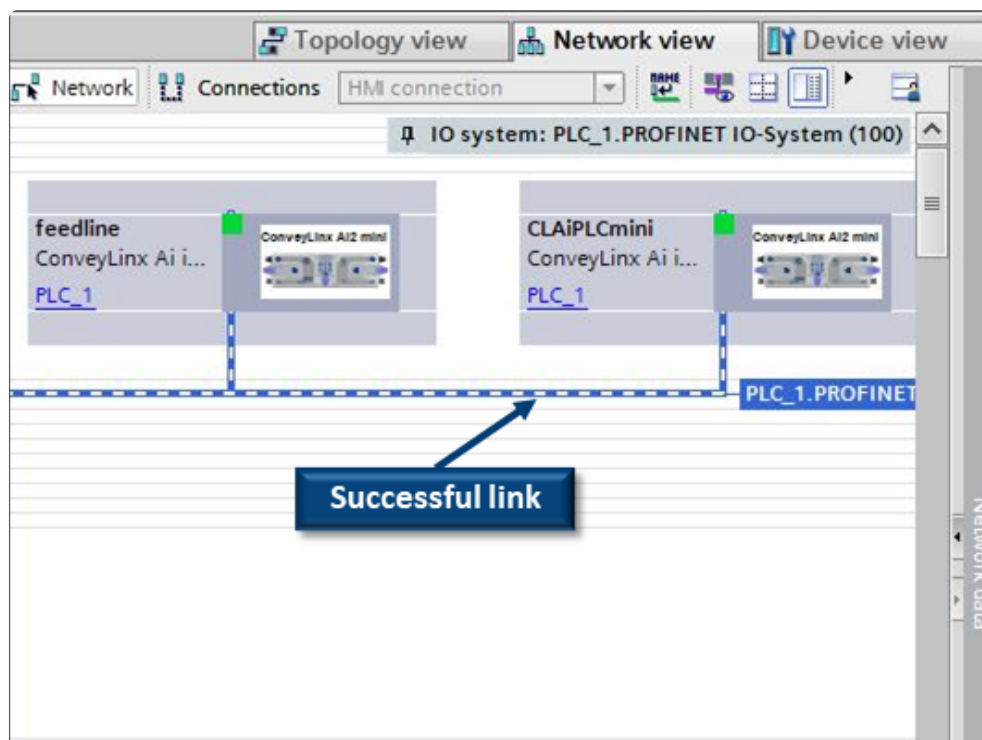
STEP 4: Select Update Time Interval

Follow the same STEP 4 as in Full ZPA Example. Because this module is running a ConveyLogix program, the Update time interval is recommended to be set based upon the CPU usage the ConveyLogix program is causing on the module.

> ✳ The CPU usage on a ConveyLinx module can be viewed when you go on-line with the ConveyLogix programming software

CPU Usage and Recommended Update Times

The following table provides the recommended Update times based upon CPU usage of your ConveyLogix program.

CPU Usage	Minimum Recommended Update Time
< 50%	8ms
50% – 60%	16ms
60% – 70%	32ms
70% – 80%	64ms
80% – 90%	128ms

! Using a faster than recommended Update Time may produce unexpected results.

! It is recommended that you do not operate ConveyLinx modules with ConveyLogix programs that utilize more that 90% CPU usage. ConveyLinx module CPU usages above 90% can result in compromised module performance.

14.5. Integrated PLC Topology Configuration

> **!** This section applies ONLY when you use TIA Portal or STEP 7 Manager Topology to configure your ConveyLinx modules

> **!** ConveyLinx-ERSC Family of modules require firmware 5.07 or later for PLC Topology configuration

Adding ConveyLinx Modules to Your Project

Once you have installed the *GSDXML* file into your environment, ConveyLinx devices are available to be used in your project. You can find ConveyLinx devices in the *Hardware Catalog* window

Drill down the tree structure to get to the folder *Conveyor Control with topology and full PLC Configuration* to see the available DAPs

These are the DAPs to use when you are utilizing PLC Topology for module configuration
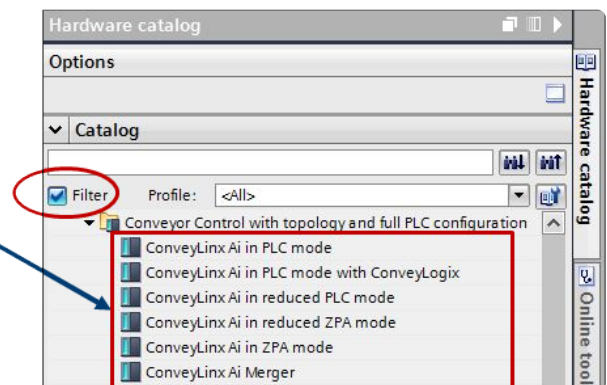
> **✱** The <u>Mode of Operation</u> of a given ConveyLinx module will determine which of the *DAPs* to drag and drop from the *Hardware Catalog* window into your project

14.5.1. Profinet Name

There are no module-specific name conventions when you use *Integrated PLC Topology Configuration* from the PLC. The name of the device can be whatever you want. TIA Portal will auto-correct if the name is outside of Profinet limitations.

✳ For users new to ConveyLinx and are familiar with the Siemens platform should find the Integrated PLC Topology Configuration quicker and easier to implement.

❗ Please note that Step7 Manager will not auto-correct if you enter a non-compliant name. A frequent mistake is using upper case letters which are not allowed according to the Profinet standard. TIA Portal will autocorrect upper case, but Step7 Manager will not auto-correct and will eventually result in program error.

14.5.2. Adding Modules

You add modules using the exact same procedures as depicted in the Separate ConveyLinx Auto Configuration section. The main difference is that you must use the DAPs created for Integrated PLC Topology Configuration.

Update Time

The DAPs for Integrated PLC Topology Configuration are already pre-configured with the lowest recommended Update time for the given functionality. This means that you can skip STEP 4 when adding your modules.

The Update time will default to the *Calculate update time automatically* setting. You can always manually enter a longer time if needed.
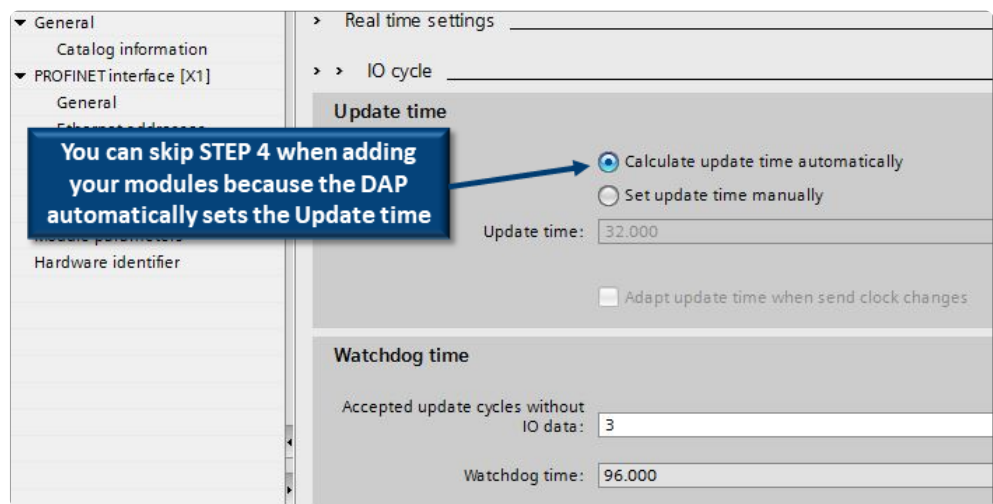
14.5.3. Topology Example

An example of a conveyor layout is a great way to illustrate the Integrated PLC Topology Configuration method.
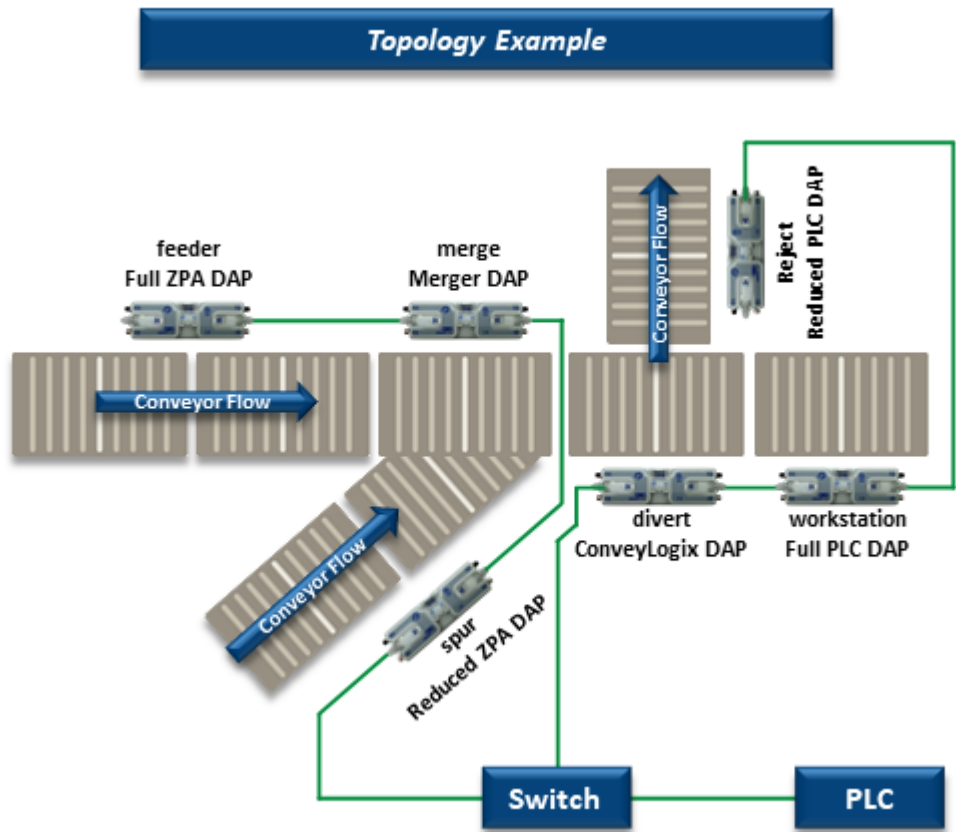
Module Mode	I.P. Address	DAP from Hardware Catalog	Profinet Name
Full ZPA	192.168.0.20	ConveyLinx-Ai in ZPA mode	*feeder*
Full ZPA with Merger	192.168.0.21	ConveyLinx-Ai merger	*merge*
ConveyLogix Interface	192.168.0.22	ConveyLinx-Ai in PLC mode with ConveyLogix	*divert*
Full PLC Controlled	192.168.0.23	ConveyLinx-Ai in PLC	*workstation*

		mode	
Reduced ZPA	192.168.10.20	ConveyLinx-Ai in reduced ZPA mode	*spur*
Reduced PLC Controlled	192.168.15.20	ConveyLinx-Ai in reduced PLC mode	*reject*

For this example, add these to your network in the same fashion as described earlier in the [Examples of Adding Modules](#) section. Because our example is using a switch, you need to select a Profinet switch from the Catalog. In our example we selected a SCLANCE XF204 four port switch. Here is a view showing all the items added to our topology including the Profinet switch and ConveyLinx modules.
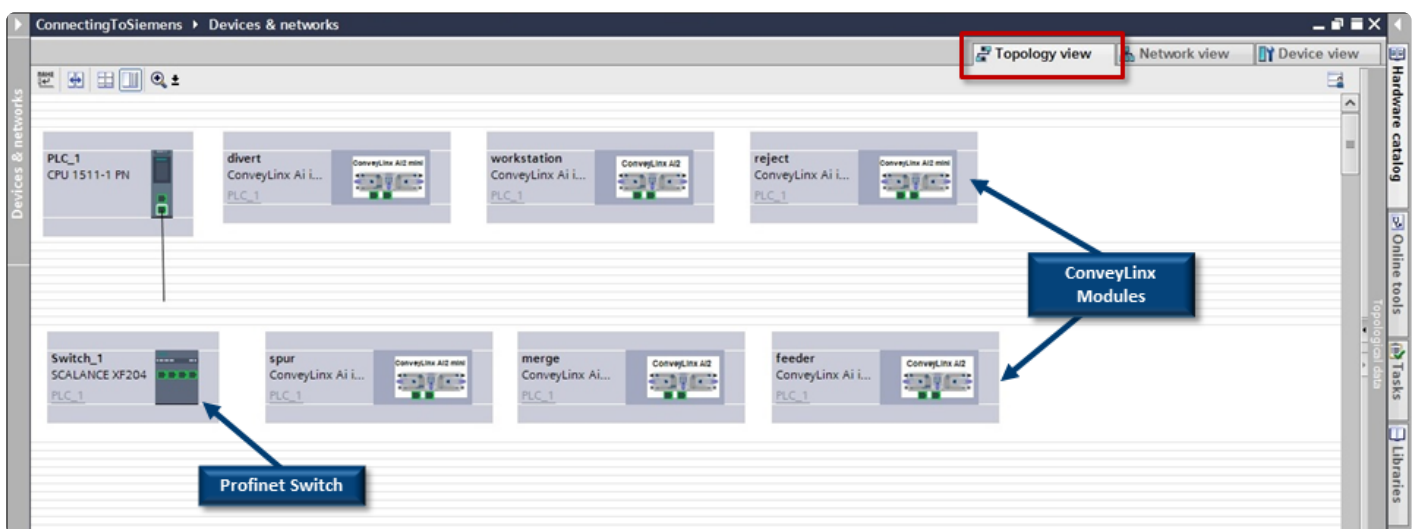
Connecting PLC and Switch

To make a topology connection, click and drag from the small connection square on the PLC to one of the ports on the switch.
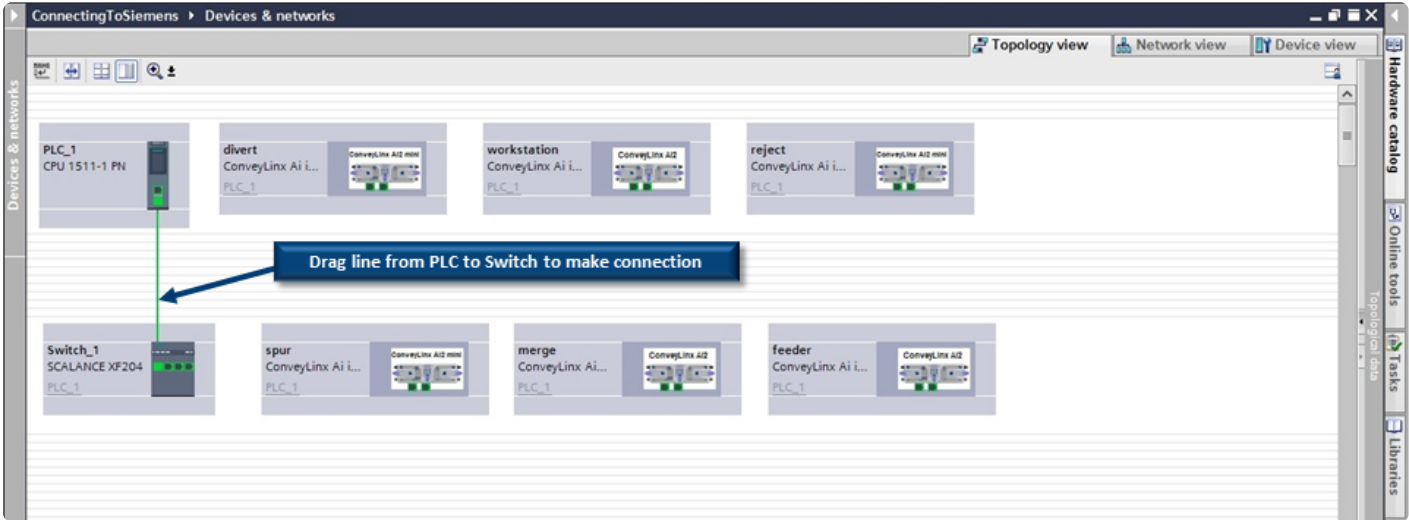
14.5.3.1. Connecting 1st Subnet of Modules

Click and drag from one of the Profinet switch ports to the *divert* ConveyLinx module's left port. Repeat the process to connect the *divert* module to the *workstation* module, and then the *workstation* module to the *reject* module.
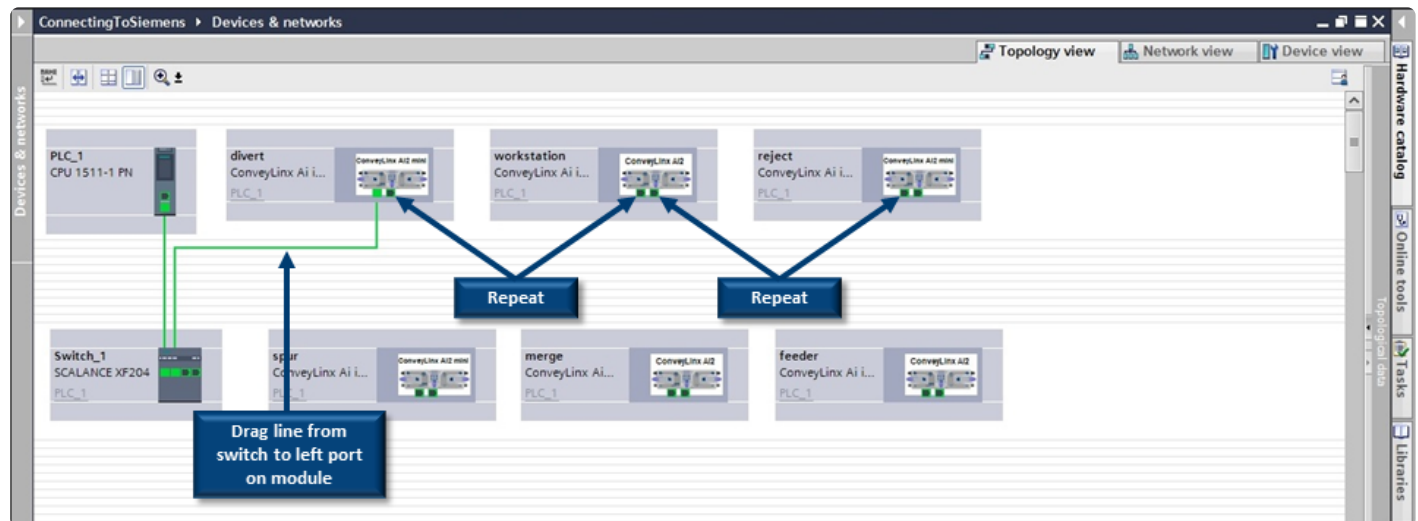
When you are done with the 1st subnet of modules, it should look like the following:
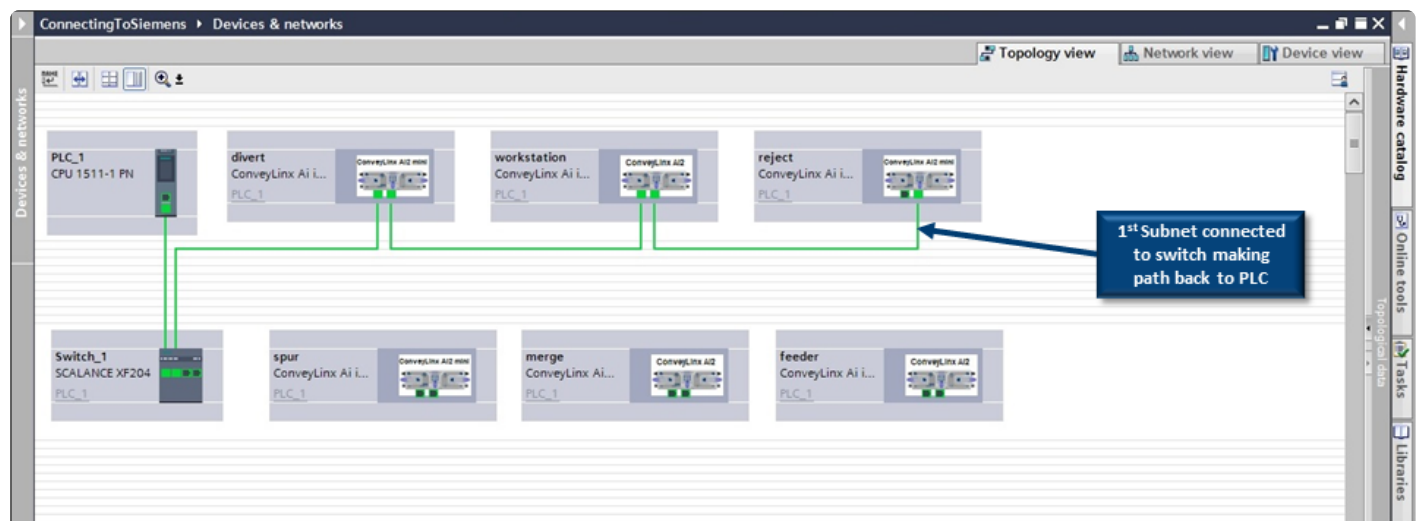
14.5.3.2. Connecting 2nd Subnet of Modules

Same as for the 1st Subnet, click and drag connection from switch to *spur* module and *spur* module to the right port on the *merge* module. Connect the left port of the *merge* module to the left port of the *feeder* module.

> ✳ Note which port on the *merge* module is connected to which specific port on its neighbor modules.
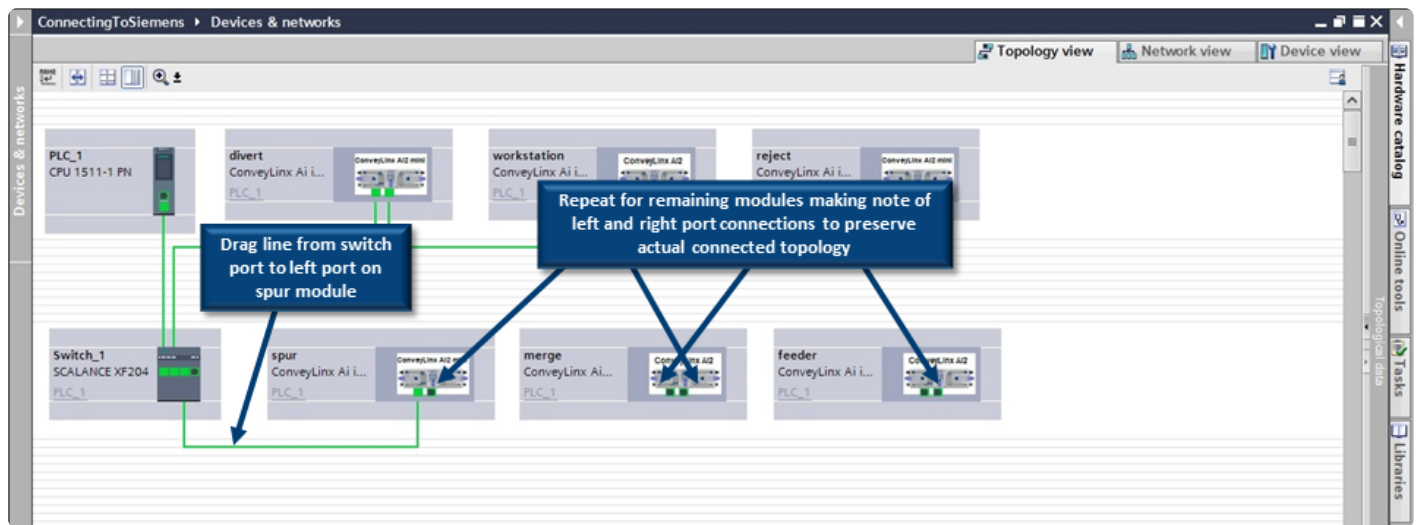
When you are done with the 2nd subnet of modules, it should look like the following:
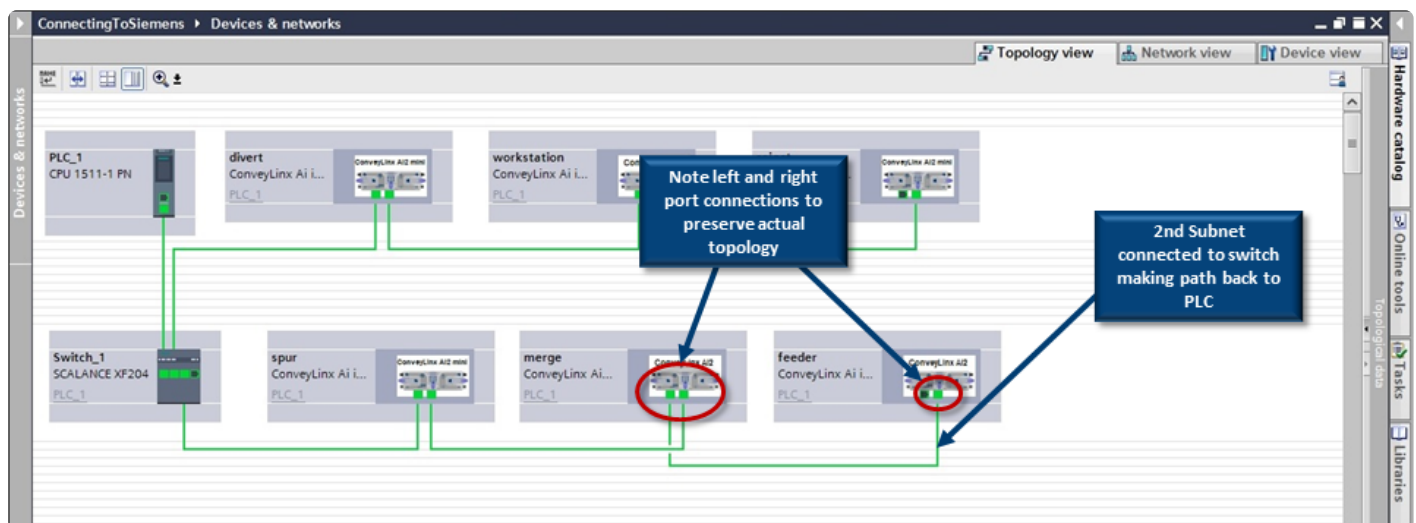
14.5.4. Module Configuration

With *Integrated PLC Topology Configuration*, you do not need *EasyRoll* for initial module configuration or changing ConveyLinx module parameters. The PLC programming environment provides the ability to configure most all the parameters and settings you can access with EasyRoll.
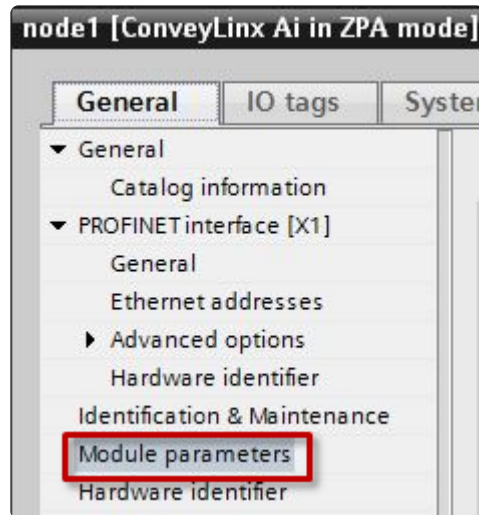
The following sections show the various parameters you have access to for each of the DAPs along with links to the corresponding EasyRoll explanations of the particular parameter:

[ZPA/Reduced ZPA Modes](#)
[PLC/Reduced PLC Modes](#)
[Merger Mode](#)
[ConveyLogix Mode](#)

14.5.4.1. ZPA/Reduced ZPA Modes

General

There is no equivalent setting in *EasyRoll* because these are determined by the *Auto-Configuration* procedure

Set up for master/slave motor rollers in a belted zone. There is no equivalent setting in *EasyRoll*

Sensor Debounce setting

Enable/Disable FlexZone operation

Change *Upstream* ConveyLinx Connection. If you want to perform the *EasyRoll* equivalent of selecting "None" for the connection, select the drop down option "Connect the the module with the below IP address" and enter "0" for each IP address octet.

6   Change *Downstream* ConveyLinx Connection. If you want to perform the *EasyRoll* equivalent of selecting "None" for the connection, select the drop down option "Connect the the module with the below IP address" and enter "0" for each IP address octet.

14.5.4.1.1. Upstream/Downstream Zones

1. [Set zone's release mode of operation](#)

2. Timer value when using [GAP Train Release](#)

3. Change the logical polarity of the [Sensor](#)

4. Configure Sensor port to accept Push Pull type sensor

5. Change motor's [Default Rotation Direction](#)

6. Enable/Disable various [zone options](#)

7. Set zone's [Motor Type](#)

8. Enter desired [Motor Speed](#)

9. Change motor's [Brake Method](#)

A. Enter desired motor [Accel/Decel](#) values

> ✻ These settings are identical for the Downstream Zone

14.5.4.1.2. Upstream/Downstream Zone Timing

1. Change zone's <u>Run After Timer</u> from its default value

2. Change zone's <u>Induct Forward and Reverse Timer</u> values

3. Change zone's <u>Jam Timer</u> from its default value

4. Configure zone's <u>Look Ahead Slowdown</u> feature operation

✽ These settings are identical for the Downstream Zone

14.5.4.1.3. Sensor Port Pin 2 Usage
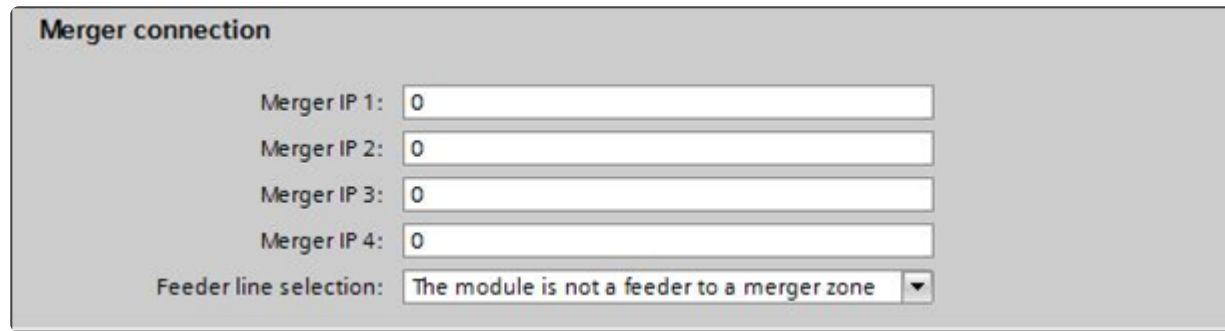
Select the operation and configuration of each Sensor Port's Pin 2

14.5.4.1.4. Connection to Merger Module

Merger connection

Merger IP 1:	0
Merger IP 2:	0
Merger IP 3:	0
Merger IP 4:	0
Feeder line selection:	The module is not a feeder to a merger zone ▼

These settings are only used if this given module is discharging onto another ConveyLinx module that is configured as a *Merger*. In *EasyRoll* this operation is performed when connected to the Merger module's *Merge Zone*. The difference here is that you enter the I.P. address of the *Merger* module where the *Merge Zone* is located. In the *EasyRoll* example you are connected to the *Merger* module's *Merge Zone* and you enter the I.P. address of the ZPA module that is feeding the *Merge Zone*. This function is depicted in an example in the *ConveyMerge* section.

14.5.4.2. PLC/Reduced PLC Mode

General

1	There is no equivalent setting in *EasyRoll* because these are determined by the *Auto-Configuration* procedure
2	Change *Upstream* ConveyLinx Connection. If you want to perform the *EasyRoll* equivalent of selecting "None" for the connection, select the drop down option "Connect the the module with the below IP address" and enter "0" for each IP address octet.
3	Change *Downstream* ConveyLinx Connection. If you want to perform the *EasyRoll* equivalent of selecting "None" for the connection, select the drop down option "Connect the the module with the below IP address" and enter "0" for each IP address octet.

Left / Right Side Configuration

For PLC mode, this function is not set in *EasyRoll* but with populating data in the [Set Sensor Port Input Mask](#) register from the PLC. When you use *Integrated PLC Topology Configuration*, you set this function here and you do not need to write data from your PLC logic

①

② There is not an equivalent function in EasyRoll

③ Change motor's [Default Rotation Direction](#)

④ Set zone's [Motor Type](#)

⑤ Enter desired [Motor Speed](#)

⑥ Change motor's [Brake Method](#)

⑦ Enter desired motor [Accel/Decel](#) values

✳ These settings are identical for the Right side

14.5.4.3. Merger Mode

The *Merger DAP* is essentially a *Full ZPA* mode module with an added parameter to configure the *merger*. So the [General](#), [Upstream/Downstream Zones](#), [Upstream/Downstream Zone Timing](#), and [Sensor Port Pin 2 Usage](#) parameters are the same as previously described.

Merger Configuration

Merger		
Merger type:	Merger is disabled	▼
Merger mode:	First come first serve	▼
Left IP 1:	0	
Left IP 2:	0	
Left IP 3:	0	
Left IP 4:	0	
	☐ Enable T-Merge	
	☐ Enable T-Merge Move-To-Sensor	
Right IP 1:	0	
Right IP 2:	0	
Right IP 3:	0	
Right IP 4:	0	
	☐ Enable T-Merge	
	☐ Enable T-Merge Move-To-Sensor	
T-Merger Accept timer:	0	
T-Merger Sending timer:	0	

An example of using these parameters in _*EasyRoll* is shown in section [Conventional Spur Merge](#) from the ConveyMerge topic.

> ✱ For complete information about setting up and using *ConveyMerge*, please refer to the [ConveyMerge section](#)

14.5.4.4. ConveyLogix Mode

The *ConveyLogix DAP* is essentially the same as a Full PLC mode module, so the previous topic for PLC/Reduced PLC Mode is identical for the ConveyLogix DAP.

14.6. Accessing Data from ConveyLinx Modules

There are 3 methods to access data from your modules:

1. Raw Unmapped Direct from the Module

This method allows you to access the raw data to/from the array of I/O data defined in the DAP directly from within one of you program blocks. You then initiate the communications using GETIO/SETIO functions within your block. This method is typically only used by advanced programmers or programmers who do not want to use tags or UDTs. This method is not recommended for new users.

> ✻ This method is not recommended for new users or applications where multiple programmers have to edit and understand the same program

2. Module Data Elements Mapped to Tags

With this method, you create separate program tags for the particular word/integer array element you want to read or write to on a given module. This method would maybe be desirable if you know you only need to access a small number of items from a given module and not all available data. This method also requires that you have detailed knowledge of the structure of the given module's Input/Output Assembly Instances and operational mode as defined in the Developer's Guide. Also, this method does not allow for simple access to sub-element items such as Boolean bits within a Word element.

> ✻ This method is not recommended in applications where you need to access more than one or two data elements from a module because you have to manually create individual tags for each element you want to access and it is not very sub-element friendly.

3. Module Data Instances Mapped to User Defined Types (UDTs)

With this method, you import a UDT file provided by Pulseroller that has pre-defined tags and structures that match up with each operational mode available for the module. You then use these intuitively named tags directly in your PLC program. This method allows you to have access to all of the data that is made available to/from the module based upon its assigned DAP and operational mode.

✱ Using the UDTs is the recommended method of data access with ConveyLinx modules because these have been tested and verified by Pulseroller.

14.6.1. Raw Unmapped Data Direct from Module

> ✳ This method is not recommended for new users or applications where multiple programmers have to edit and understand the same program

With this method of data access, you are copying data to/from the module's I/O memory directly to your function block's local variables by using the GETIO/SETIO instructions. The I/O memory addresses for you module were assigned when you added the module to your project. The memory locations are dependent upon the order you added them to you project and the amount of memory used as defined in the DAP.

Example

You are creating a function block called *Block_1* and you want to read the input data from and write output data to the ConveyLinx module we created called *workstation* from our Topology Example. For your local static variables in your *Block_1*, you need to set the Data types for your *in* and *out* variables to be *Array[0..31] of Word*. To read data to the local in variable you need to use the *GETIO* function and to write data to the local out variable you need to use the *SETIO* function. Both the GETIO and SETIO functions require an *ID* parameter which corresponds to the specific module's input and output memory.

Create your *Block_1* and set the *Data type* for the *in* and *out* variables. We know the *workstation* module is in *Full PLC Mode* so it requires 64 Bytes (32 Words) of input and output data
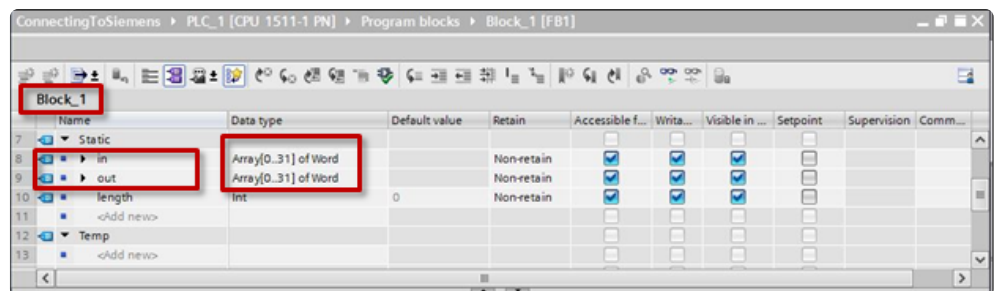
We need to find the *Hardware Identifier* for the inputs for the *workstation* module. Go to *Ungrouped devices* in the project tree, select the *workstation* module, select *virtual input module*, and select *Properties*
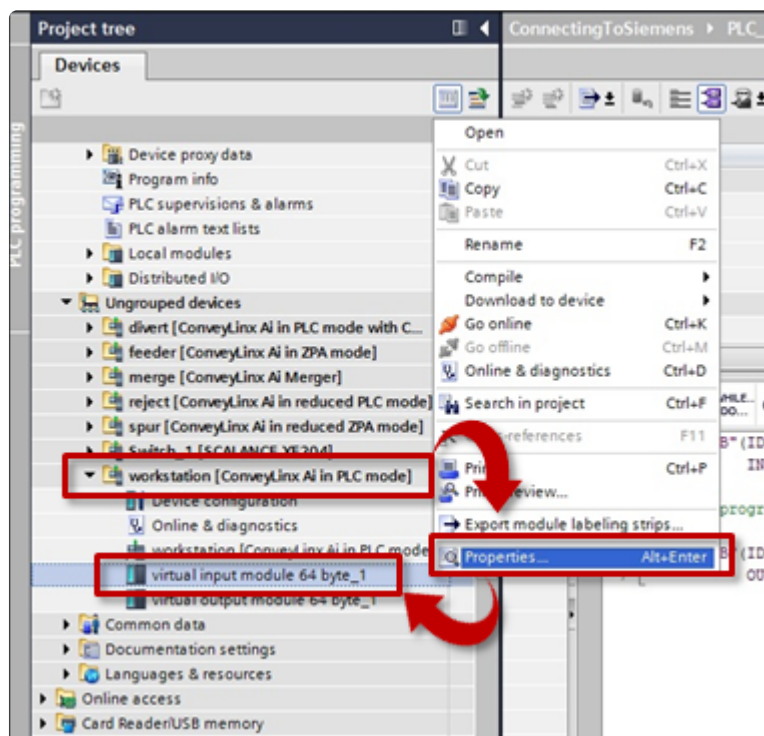
From here you can see that the *Hardware identifier* for the *inputs* for the *workstation* module is *274*. Note this value because you will need it later when you configure your *GETIO* function.
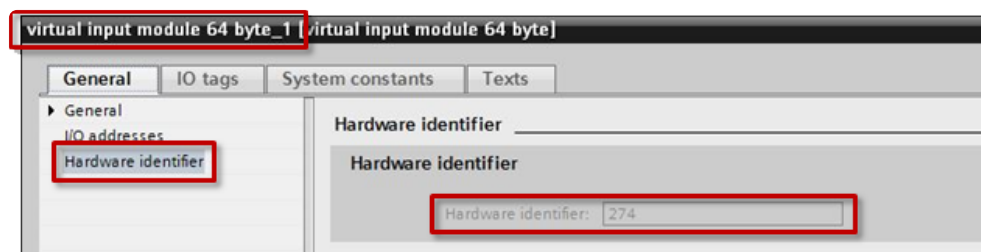
Follow the same procedure from the project tree to select the *virtual output module*. From here you can see that the *Hardware identifier* for the *outputs* for the *workstation* module is *275*. Note this value because you will need it later when you configure your *SETIO* function.
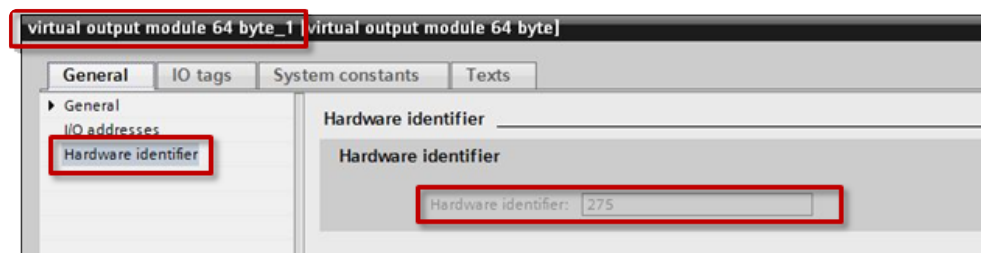
Here is the *Block_1* program code showing the *GETIO* function utilizing the *Hardware identifier* value of *274* and putting the data to the *in* local tag. There is also code showing the *SETIO* function utilizing the *Hardware identifier* value of *275* and using the *out* local tag.
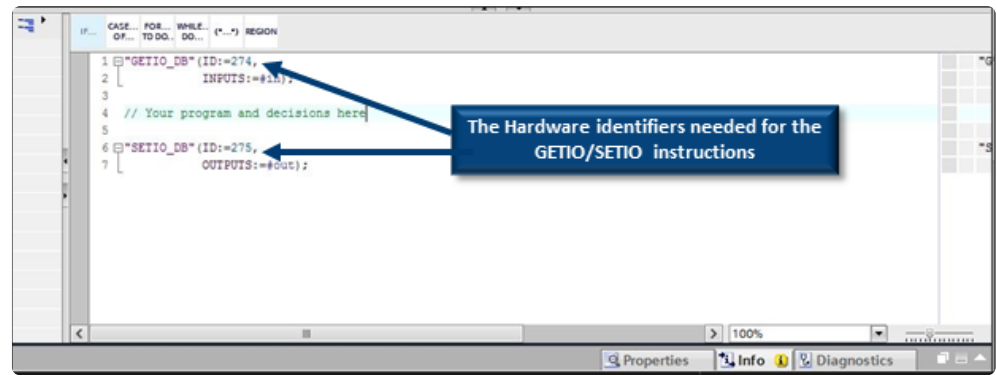
14.6.2. Module Data Elements Mapped to Tags

> ✳ This method is not recommended in applications where you need to access more than one or two data elements from a module because you have to manually create individual tags for each element you want to access

This method allows you to create your own *global tags* with meaningful names and map them to a specific *Word* address within a given module. To use this method you will have to be very familiar with the Instance Assembly structure for the ConveyLinx-Ai Family Controllers

Example

In this example we want to read the sensor port inputs and be able to send a clear jam command to the upstream zone on the ConveyLinx module we named *feeder* in our Topology Example. We also want to read the sensor port inputs and control the left motor on the ConveyLinx module we named *workstation* in our Topology Example.

Create Tags for our *feeder* Module

From *Network View*, locate the module *feeder*, right click and select *Device configuration* and from *Device view* expand the window from the right to display the *Device overview*. This will show you the I and Q address byte locations in memory that were assigned when you added the module to you project. As you can see, the *feeder* module's *I* and *Q* are located in bytes *0 thru 63* respectively.
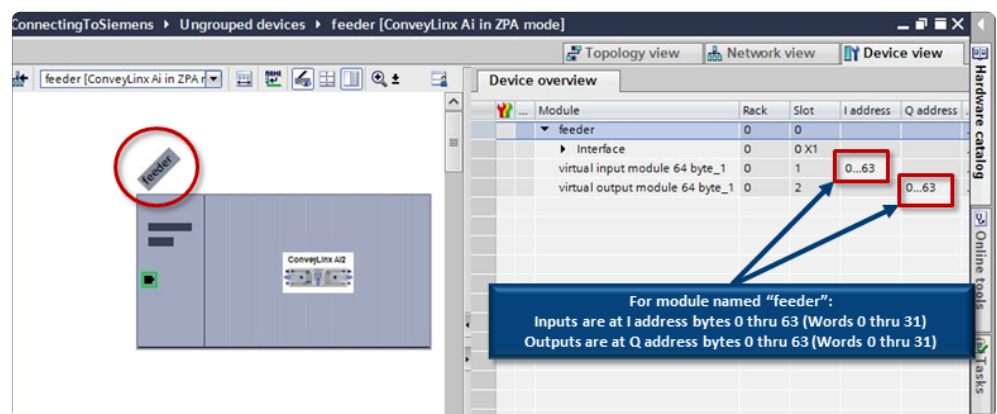
We want to read the *Sensor Port Inputs* register from the *feeder* module. As shown in the Port Inputs and ConveyStop Status description from the PLC Developer's Guide section, we can see that the *Sensor Port Inputs* data is at *word register offset 01* within the ZPA Mode assembly instance. To point our new tag to the correct starting byte, we take our word address x 2 to get

the starting byte offset address. In our example, this means that the Sensor Port Inputs word register data will be located at I address bytes 36 and 37. When we declare our tag with the W syntax, it will know that the tag requires the 2 bytes beginning at I address 36.

From our *Default tag table*, create a new tag and let's call it *feeder_SensorPortInputs*. Select *Word* as the *Data type* and enter *%IW36* for the *Address*.
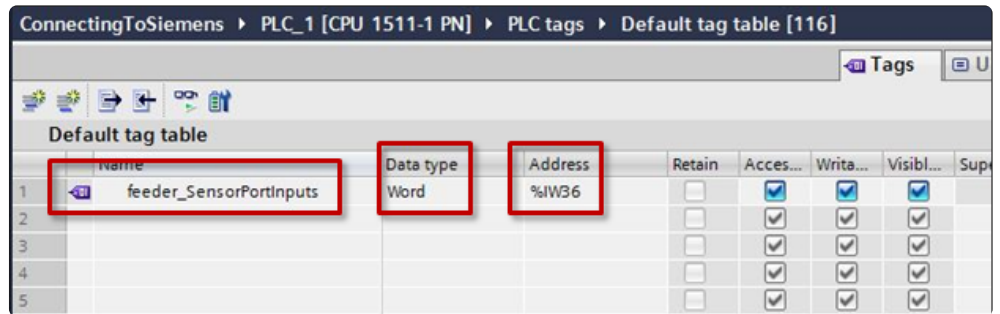
We want to write a command to the *Clear Jam for Local Upstream Zone* register to the *feeder* module. As shown in the ConveyStop and Clear Jams description from the PLC Developer's Guide section, we can see that the *Clear Jam for Local Upstream Zone* data is at *word register offset 20* within the ZPA Mode assembly instance. To point our new tag to the correct starting byte, we take our word address x 2 to get the starting byte offset address. In our example, this means that the Sensor Port Inputs word register data will be located at Q address bytes 40 and 41. When we declare our tag with the W syntax, it will know that the tag requires the 2 bytes beginning at Q address 40.

From our *Default tag table*, create a new tag and let's call it *feeder_ClearJamUpstreamZone*. Select *Word* as the *Data type* and enter *%QW40* for the *Address*.
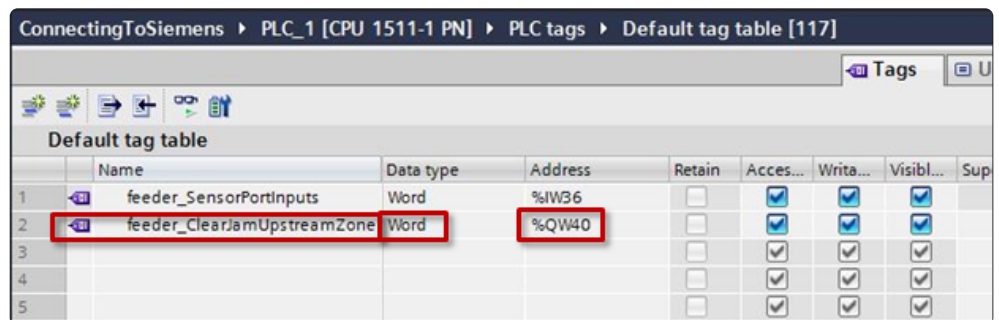
Create Tags for our *workstation* Module
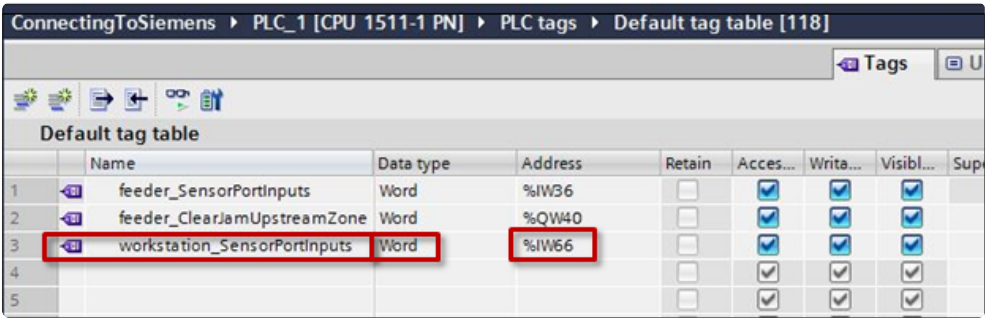
From *Network View*, locate the module *workstation*, right click and select *Device configuration* and from *Device view* expand the window from the right to display the *Device overview*. This will show you the I and Q address byte locations in memory that were assigned when
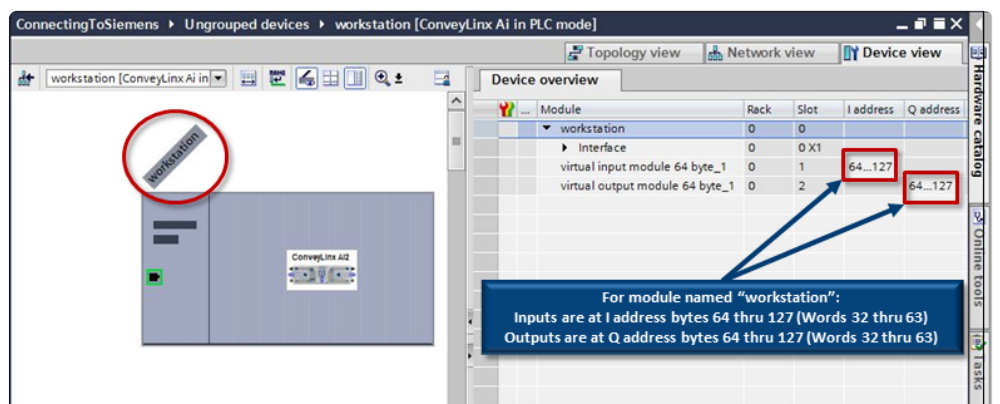
you added the *workstation* module to you project. As you can see, the *workstation* module's *I* and *Q* are located in bytes *64 thru 127* respectively.

We want to read the *Sensor Port Inputs* register from the *workstation* module. As shown in the Sensor Ports description from the PLC Developer's Guide section, we can see that the *Sensor Port Inputs* data is at *word register offset 1* within the PLC I/O Mode assembly instance. To point our new tag to the correct starting byte, we take our word address x 2 to get the starting byte offset address. In our example, this means that the Sensor Port Inputs word register data will be located at I address bytes 2 and 3 *within the workstation module*. We also need to account for where the workstation module is located within the entire I/O memory. We know that the workstation module starts at %I64 / %Q64, so we need to add our offset of 2 to the *starting point of the workstation modules memory*. This results in an offset of 66 (64 + 2). When we declare our tag with the W syntax, it will know that the tag requires the 2 bytes beginning at I address 66.
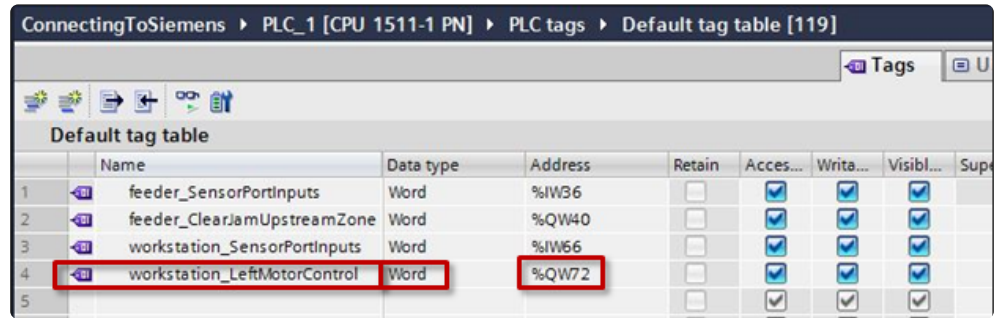
From our *Default tag table*, create a new tag and let's call it *workstation_SensorPortInputs*. Select *Word* as the *Data type* and enter *%IW66* for the *Address*.

We want to write a command to the *Left Motor Control* register to the *workstation* module. As shown in the Left Motor Control description from the PLC Developer's Guide section, we can see that the *Left Motor Run / Reverse* data is at *word register offset 4* within the PLC I/O Mode assembly instance. To point our new tag to the correct starting byte, we take our word address x 2 to get the starting byte offset address. In our example, this means that the Left Motor Run / Reverse word register data will be located at Q address bytes 8 and 9 *within the workstation module*. We also need to account for where the workstation module is located within the entire I/O memory. We know that the *workstation* module starts at %I64 / %Q64, so we need to add our offset of 2 to the *starting point of the workstation modules memory*. This results in an offset of 72 (64 + 8). When we declare our tag with the W syntax, it will know that the tag requires the 2 bytes beginning at Q address 72.

From our *Default tag table*, create a new tag and let's call it *workstation_LeftMotorControl*. Select *Word* as the *Data type* and enter *%QW72* for the *Address*.

14.6.3. Module Data Instances Mapped to User Defined Types (UDTs)

✱ Using the UDTs is the recommended method of data access with ConveyLinx modules because these have been tested and verified by Pulseroller.

User Data Types (*UDTs*) are used to generate structured data type blocks within the PLC programming environment for each *DAP's* inputs and outputs. When you create tags to use in your program, you can then assign the appropriate *data types* for the given ConveyLinx module's *inputs* and *outputs*.

Each given *DAP* consumes some number of input memory bytes and output memory bytes. When you use *UDTs* and create tags with the UDT data types; the *PLC's I/O memory* is properly allocated and the memory boundaries between modules are easily established.

14.6.3.1. Installing UDTs into Programming Environment

In your *Project tree*, expand *External source files* and select *Add new external file*. Browse to the location on your PC where you saved the UDT file you downloaded from Pulseroller.com. In this example we are using ConveyLinx-Ai Family modules so we select the corresponding file.
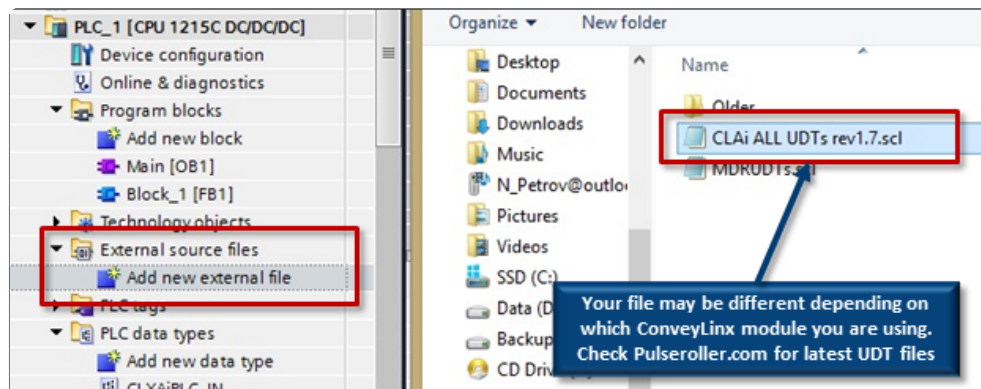
Once you select the file, it will appear as an item in the *External source files* folder. Select the file, right click, and select *Generate blocks from source*.
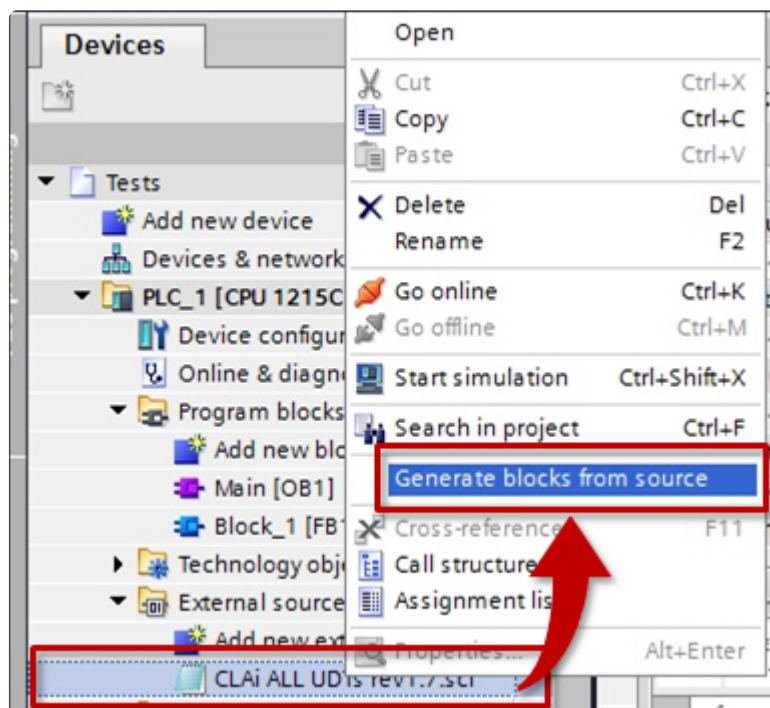
Once the blocks have been generated, you can go to the *PLC data types* folder in the *Project tree* and expand the folder to see the *UDTs* generated. In this example, all of the UDTs required for the ConveyLinx-Ai Family have been generated.
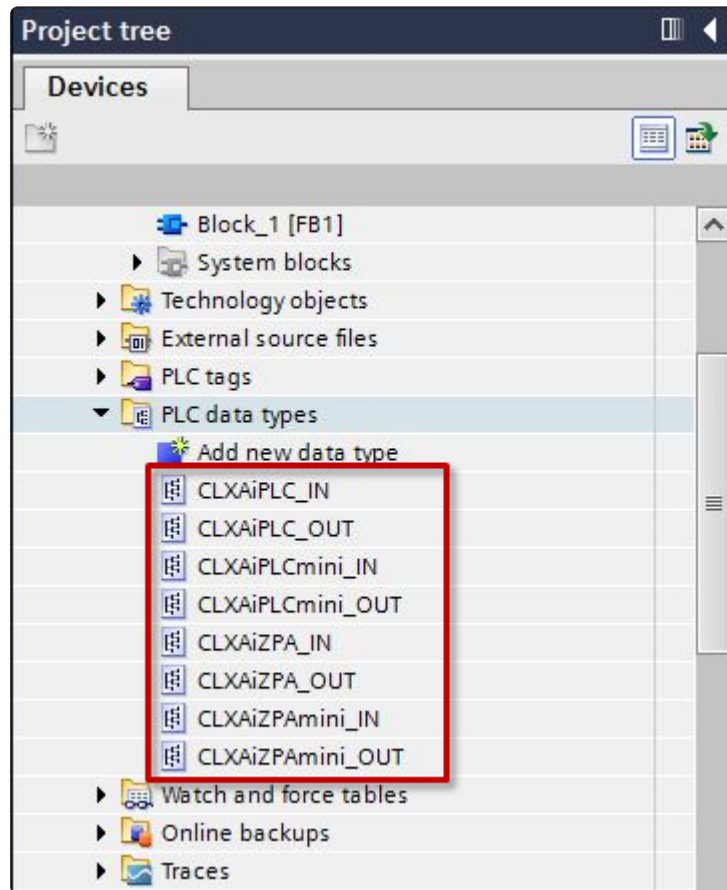
14.6.3.2. Selecting the correct UDT for the Module's Assigned DAP

The UDT's are structured based upon the I/O memory being utilized by the ConveyLinx module's mode of operation. The DAP selected for you module is dependent upon what you want to do with it and which of the two configuration methods you are going to use. These sections indicate the combinations of functionality, the DAP to select, it memory requirements, and the UDT to use.

Ai Family DAPs when using Separate ConveyLinx Auto-Configuration

Module Mode	Module Mode	Input Bytes	Input UDT	Output Bytes	Output UDT
Full ZPA	*ConveyLinx Ai*	64	CLXAiZPA_IN	64	CLXAiZPA_OUT
Full ZPA with Merger	*ConveyLinx Ai*	64	CLXAiZPA_IN	64	CLXAiZPA_OUT
Reduced ZPA	*ConveyLinx Ai in reduced ZPA mode*	30	CLXAiZPAmini_IN	30	CLXAiZPAmini_OUT
Full PLC	*ConveyLinx Ai*	64	CLXAiPLC_IN	64	CLXAiPLC_OUT
Reduced PLC	*ConveyLinx Ai in reduced PLC mode*	16	CLXAiPLCmini_IN	16	CLXAiPLCmini_OUT
PLC with ConveyLogix	*ConveyLinx Ai in PLC mode with ConveyLogix*	32	N/A	32	N/A

Ai Family DAPs when using Topology Configuration

Module Mode	Module Mode	Input Bytes	Input UDT	Output Bytes	Output UDT

Full ZPA	*ConveyLinx Ai in ZPA mode*	64	CLXAiZPA_IN	64	CLXAiZPA_OUT
Full ZPA with Merger	*ConveyLinx Ai Merger*	64	CLXAiZPA_IN	64	CLXAiZPA_OUT
Reduced ZPA	*ConveyLinx Ai in reduced ZPA mode*	30	CLXAiZPAmini_IN	30	CLXAiZPAmini_OUT
Full PLC	*ConveyLinx Ai in PLC mode*	64	CLXAiPLC_IN	64	CLXAiPLC_OUT
Reduced PLC	*ConveyLinx Ai in reduced PLC mode*	16	CLXAiPLCmini_IN	16	CLXAiPLCmini_OUT
PLC with ConveyLogix	*ConveyLinx Ai in PLC mode with ConveyLogix*	32	N/A	32	N/A

ERSC Family DAPs when using Separate ConveyLinx Auto-Configuration

Module Mode	Module Mode	Input Bytes	Input UDT	Output Bytes	Output UDT
Full ZPA	*ConveyLinx in ZPA mode*	64	CLXERSCZPA_IN	64	CLXERSCZPA_OUT
Full ZPA with Merger	*ConveyLinx in ZPA mode*	64	CLXERSCZPA_IN	64	CLXERSCZPA_OUT
Reduced ZPA	*ConveyLinx in reduced ZPA mode*	30	CLXERSCZPAmini_IN	30	CLXERSCZPAmini_OUT
Full PLC	*ConveyLinx in PLC mode*	64	CLXERSCPLC_IN	64	CLXERSCPLC_OUT
Reduced	*ConveyLinx*	16	CLXERSCPLCmini_IN	16	CLXERSCPLCmini_OUT

PLC	*in reduced PLC mode*				
PLC with ConveyLogix	*ConveyLinx in PLC mode with ConveyLogix*	32	N/A	32	N/A

ERSC Family DAPs when using Topology Configuration

Module Mode	Module Mode	Input Bytes	Input UDT	Output Bytes	Output UDT
Full ZPA	*ConveyLinx in ZPA mode*	64	CLXERSCZPA_IN	64	CLXERSCZPA_OUT
Full ZPA with Merger	*ConveyLinx Merger*	64	CLXERSCZPA_IN	64	CLXERSCZPA_OUT
Reduced ZPA	*ConveyLinx in reduced ZPA mode*	30	CLXERSCZPAmini_IN	30	CLXERSCZPAmini_OUT
Full PLC	*ConveyLinx in PLC mode*	64	CLXERSCPLC_IN	64	CLXERSCPLC_OUT
Reduced PLC	*ConveyLinx in reduced PLC mode*	16	CLXERSCPLCmini_IN	16	CLXERSCPLCmini_OUT
PLC with ConveyLogix	*ConveyLinx in PLC mode with ConveyLogix*	32	N/A	32	N/A

14.6.3.3. UDT Assignment Example

To use UDTs, you create your own tag in the *Default tag table* as shown in the previous section and instead of using one of the elementary *Data types*, you select one of the *UDTs* we generated. So for the *feeder* module, we know from the previous section that its starting address for both Inputs and outputs is byte 0. So, we are going to create a tag to store all of the *feeder* module's inputs and a tag to write to all of the *feeder* module's outputs.

Example Modules and UDT Assignment

Using the modules we defined in the Topology Example section, we are going to assign each module's input and output data to the UDTs we generated in the previous section.

Profinet Name	Module Mode	DAP from Hardware Catalog	Input UDT Name	Output UDT Name
feeder	Full ZPA	ConveyLinx-Ai in ZPA mode	*CLXAiZPA_IN*	*CLXAiZPA_OUT*
merge	Full ZPA with Merger	ConveyLinx-Ai merger	*CLXAiZPA_IN*	*CLXAiZPA_OUT*
divert	ConveyLogix Interface	ConveyLinx-Ai in PLC mode with ConveyLogix	N/A	N/A
workstation	Full PLC Controlled	ConveyLinx-Ai in PLC mode	*CLXAiPLC_IN*	*CLXAiPLC_OUT*
spur	Reduced ZPA	ConveyLinx-Ai in reduced ZPA mode	*CLXAiZPAmini_IN*	*CLXAiZPAmini_OUT*
reject	Reduced PLC Controlled	ConveyLinx-Ai in reduced PLC mode	*CLXAiPLCmini_IN*	*CLXAiPLCmini_OUT*

Adding *feeder* Module
Adding *workstation* Module

[Adding Remaining Modules](#)

What about ConveyLogix Interface?

There is no UDT required for the *ConveyLogix* interface because the input and output data have no pre-defined meanings or operations. The I/O data for a module using *ConveyLogix* is a "blank" block of 32 input and 32 output bytes (16 input and 16 output Words) that is available for the programmer to use as needed based upon the application.

For this example, you can access the *divert* module by either the [Raw Unmapped Data Direct from Module](#) or [Module Data Elements Mapped to Tags](#) method previously described.

14.6.3.3.1. Add feeder Module

From our *Default tag table*, create a new tag and let's call it *feeder_Inputs*. Select *CLXAiZPA_IN* as the *Data type*.
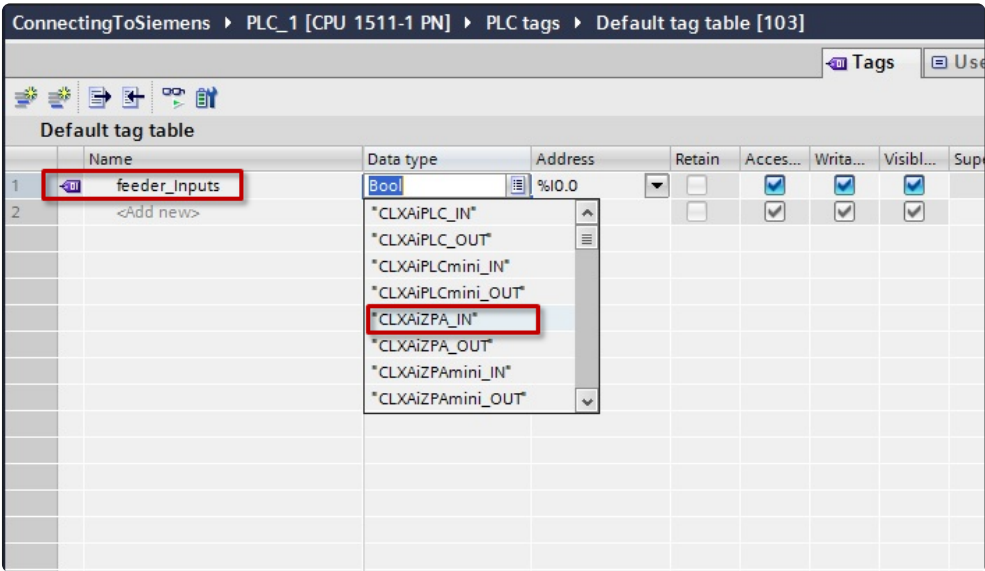
Because we know from [Module Data Elements Mapped to Tags section example](#) that the *feeder* module starts at Input byte offset 0, we know what to enter for the address.
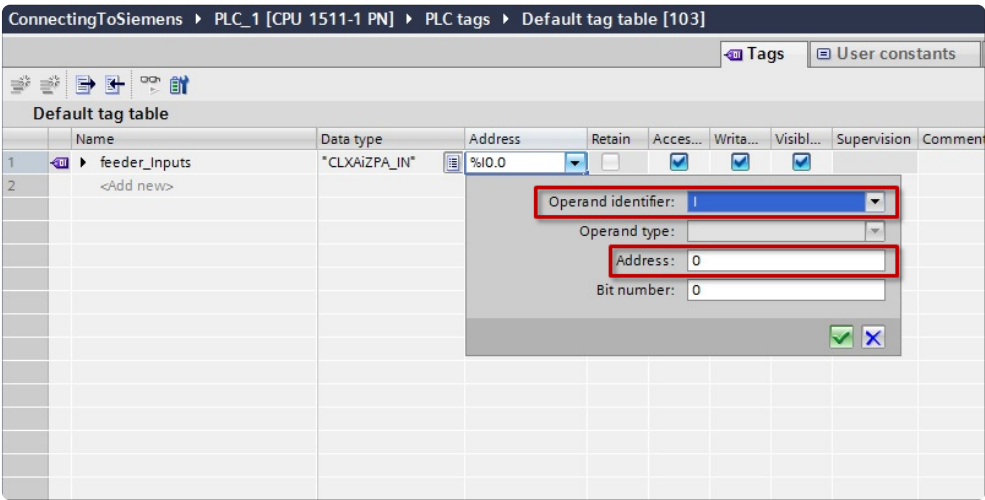
From our *Default tag table*, create a new tag for the outputs and let's call it *feeder_Inputs*. Select *CLXAiZPA_OUT* as the *Data type*.
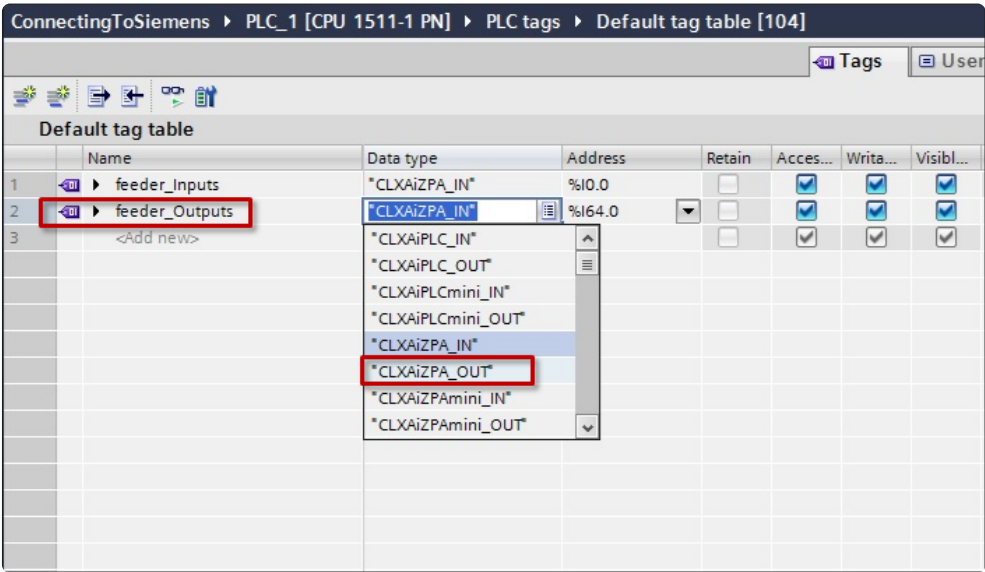
Because we know from [Module Data Elements Mapped to Tags section example](#) that the outputs for the *feeder* module starts at Output byte offset 0, we know what to enter for the address.
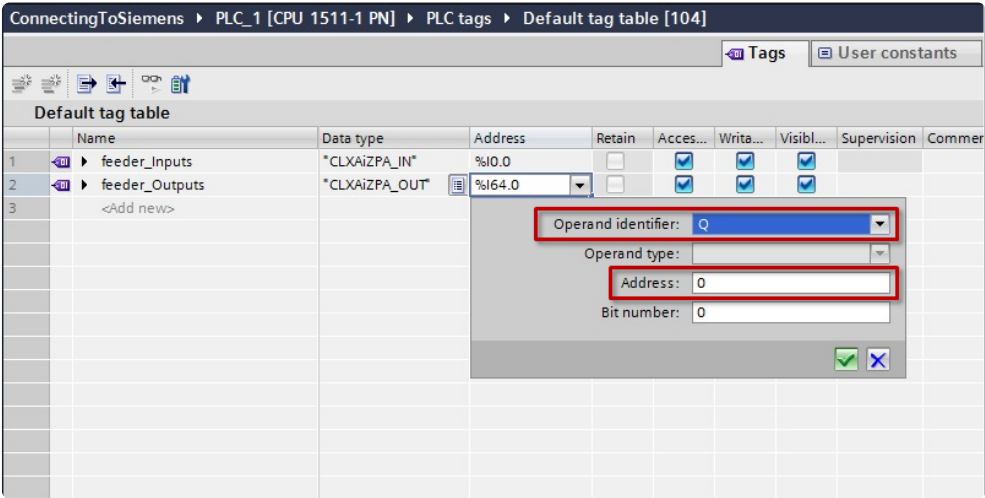
To continue and add the *workstation* module, from our *Default tag table*, create a new tag and let's call it *workstation_Inputs*. Select *CLXAiPLC_IN* as the *Data type*.
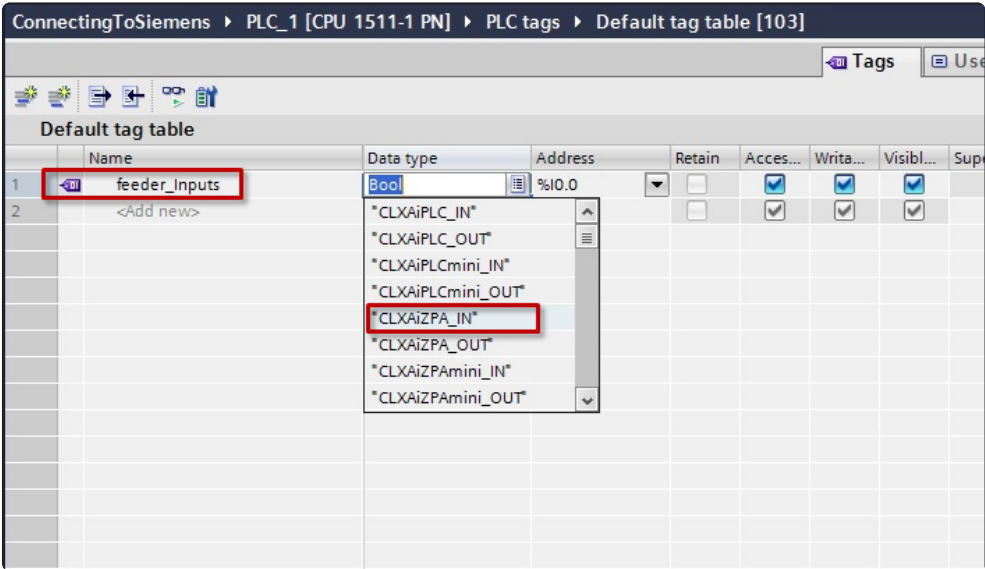
14.6.3.3.2. Add workstation Module

From our *Default tag table*, create a new tag and let's call it *workstation_Inputs*. From the chart in the [UDT Assignment Example topic](#), we know that the *workstation* module is in Full PLC I/O mode. Select *CLXAiPLC_IN* as the *Data type*.
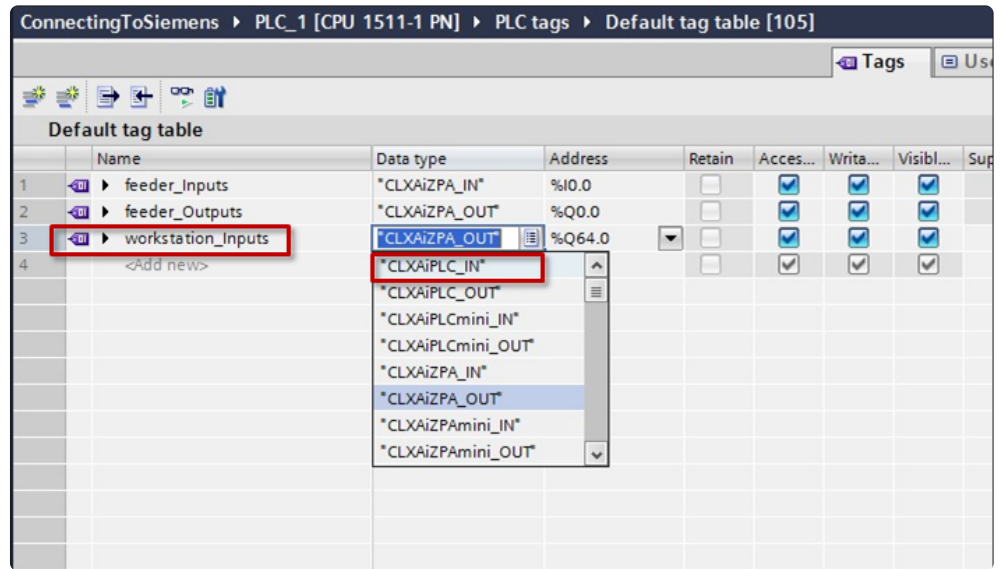
Because we know from [Module Data Elements Mapped to Tags section example](#) that the *workstation* module starts at Input byte offset 64, we know what to enter for the address.
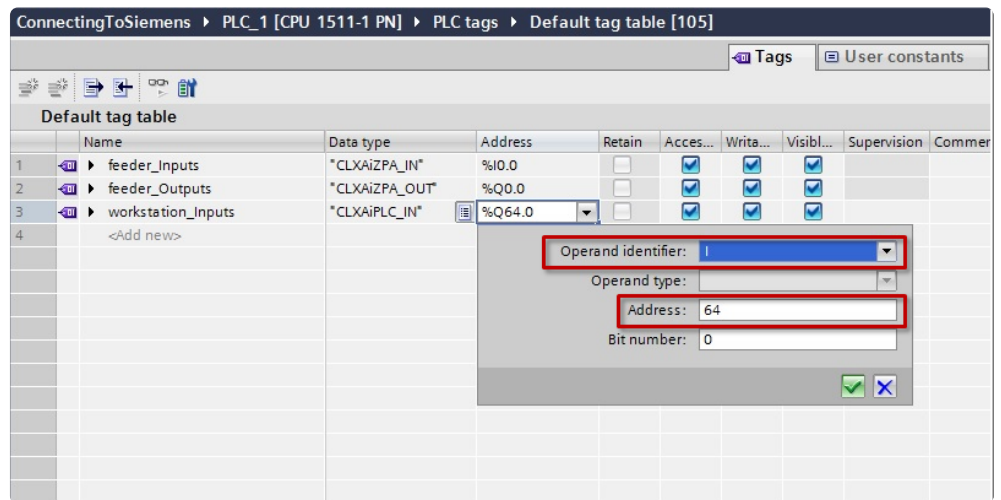
From our *Default tag table*, create a new tag for the outputs and let's call it *workstation_Outputs*. Because we know that the *workstation* module is in Full PLC I/O mode, select *CLXAiPLC_OUT* as the *Data type*.
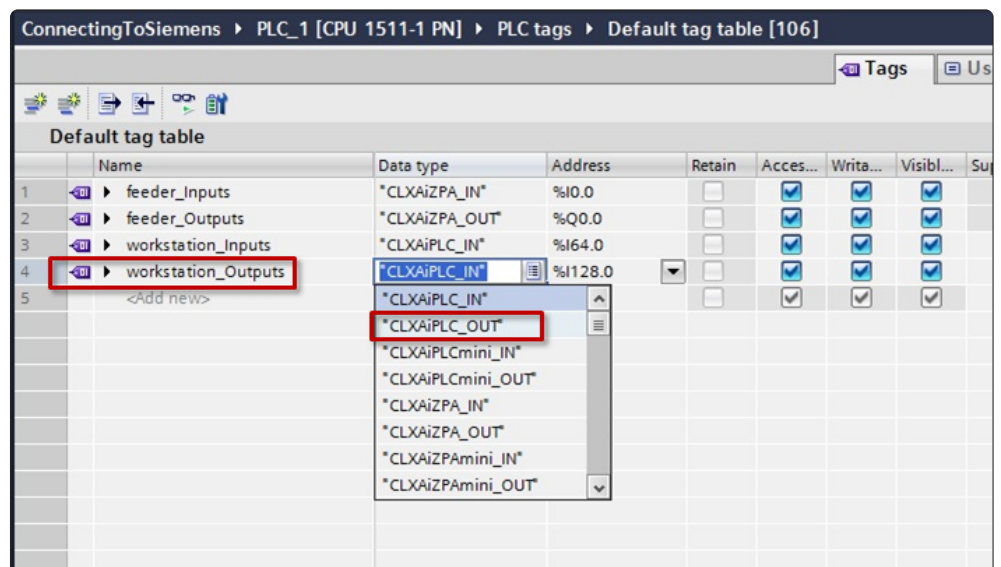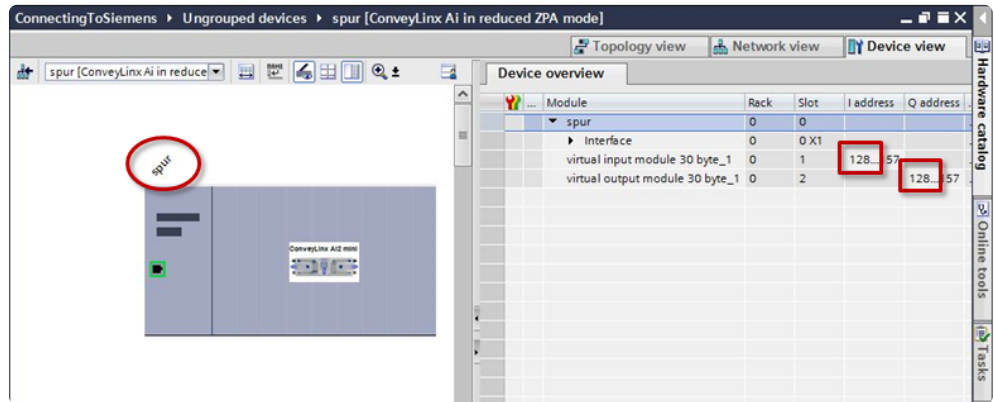
Because we know from [Module Data Elements Mapped to Tags section example](#) that the outputs for the *workstation* module starts at Output byte offset 64, we know what to enter for the address.

		Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervision	Commer
		ConnectingToSiemens ▶ PLC_1 [CPU 1511-1 PN] ▶ PLC tags ▶ Default tag table [106]								
								◁ Tags	⊞ User constants	
		Default tag table								
1	▶	feeder_Inputs	"CLXAiZPA_IN"	%I0.0	☐	☑	☑	☑		
2	▶	feeder_Outputs	"CLXAiZPA_OUT"	%Q0.0	☐	☑	☑	☑		
3	▶	workstation_Inputs	"CLXAiPLC_IN"	%I64.0	☐	☑	☑	☑		
4	▶	workstation_Outputs	"CLXAiPLC_OUT"	%I128.0	☐	☑	☑	☑		
5		<Add new>								

Operand identifier: Q
Operand type:
Address: 64
Bit number: 0
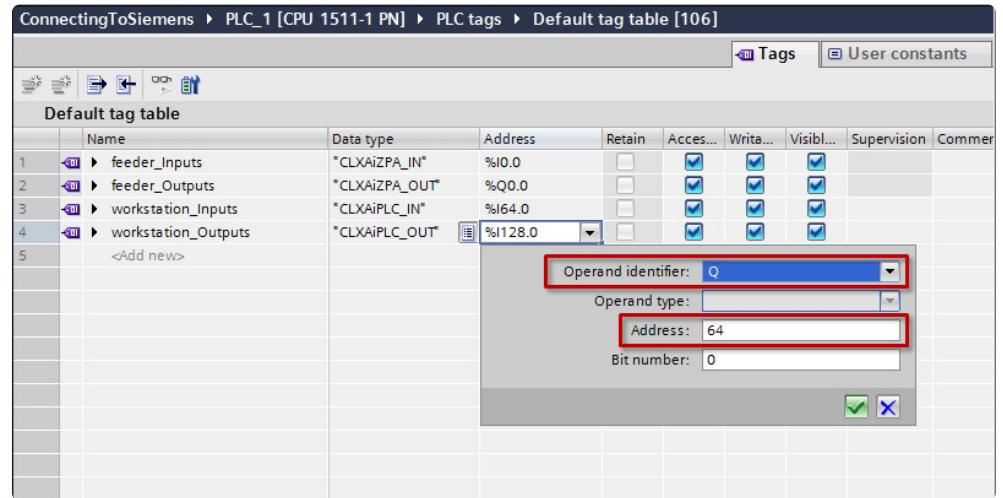
14.6.3.3. Add remaining Modules
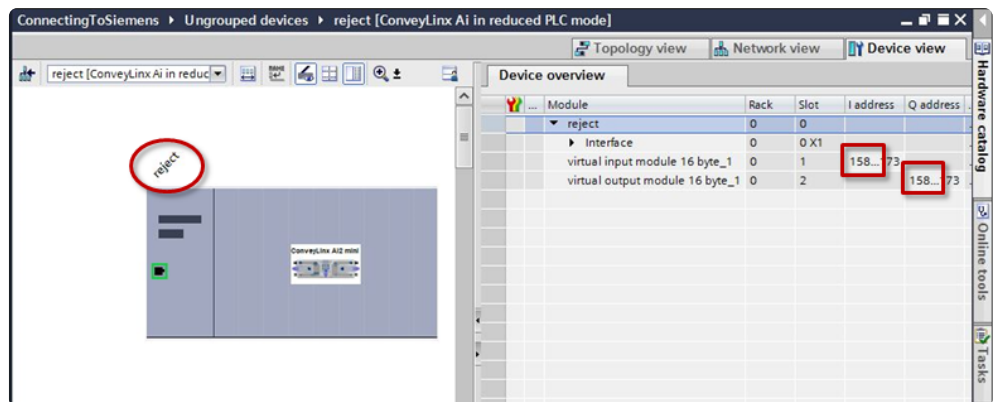
The remaining modules are added in similar fashion. We go to the *Device view* for each of the modules to see the starting byte addresses for their respective input and output data and use this when we create our tags for these modules.

For the *spur* module, when we look at its *Device view* we see that its starting input and output byte offset is **128**. From the chart in the UDT Assignment Example topic, we know that the *spur* module is in Reduced ZPA mode.
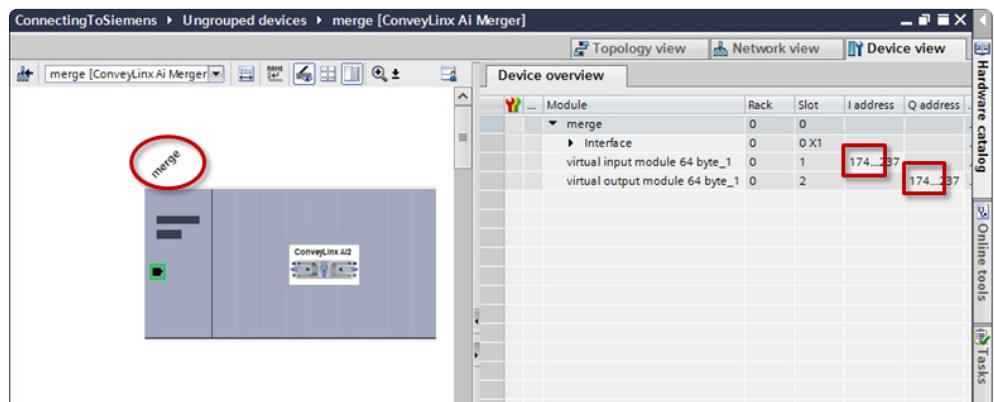
For the *reject* module, when we look at its *Device view* we see that its starting input and output byte offset is **158**. From the chart in the UDT Assignment Example topic, we know that the *spur* module is in Reduced PLC mode.

For the *reject* module, when we look at its *Device view* we see that its starting input and output byte offset is **174**. From the chart in the UDT Assignment Example topic, we know that the *spur* module is in Reduced PLC mode.

With all starting byte offsets known and following the naming convention we used for the *feeder* and *workstation* modules, we can complete creating our tags mapped to UDTs

		Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervision	Co
1		feeder_Inputs	"CLXAiZPA_IN"	%I0.0	☐	☑	☑	☑		
2		feeder_Outputs	"CLXAiZPA_OUT"	%Q0.0	☐	☑	☑	☑		
3		workstation_Inputs	"CLXAiPLC_IN"	%I64.0	☐	☑	☑	☑		
4		workstation_Outputs	"CLXAiPLC_OUT"	%Q64.0	☐	☑	☑	☑		
5		spur_Inputs	"CLXAiZPAmini_IN"	%I128.0	☐	☑	☑	☑		
6		spur_Outputs	"CLXAiZPAmini_OUT"	%Q128.0	☐	☑	☑	☑		
7		reject_Inputs	"CLXAiPLCmini_IN"	%I158.0	☐	☑	☑	☑		
8		reject_Outputs	"CLXAiPLCmini_OUT"	%Q158.0	☐	☑	☑	☑		
9		merge_Inputs	"CLXAiZPA_IN"	%I174.0	☐	☑	☑	☑		
10		merge_Outputs	"CLXAiZPA_OUT"	%Q174.0	☐	☑	☑	☑		
11		<Add new>			☐	☑	☑	☑		

Tags | User constants

Default tag table

14.7. User Data Types (UDTs)

User Data Types (*UDTs*) are used to generate structured data type blocks within the PLC programming environment for each *DAP's* inputs and outputs. When you create tags to use in your program, you can then assign the appropriate *data types* for the given ConveyLinx module's *inputs* and *outputs*.

Each given *DAP* consumes some number of input memory bytes and output memory bytes. When you use *UDTs* and create tags with the UDT data types; the *PLC's I/O memory* is properly allocated and the memory boundaries between modules are easily established.

14.7.1. UDTs for Ai2 Family

Conveyor Control DAPs for Separate ConveyLinx Auto-Configuration

ConveyLinx Mode	DAP from Catalog	Input Bytes	Output Bytes
Full ZPA	*ConveyLinx Ai*	64	64
Full PLC	*ConveyLinx Ai*	64	64
Full ZPA with Merger	*ConveyLinx Ai*	64	64
PLC with ConveyLogix	*ConveyLinx Ai in PLC mode with ConveyLogix*	32	32
Reduced PLC	*ConveyLinx Ai in reduced PLC mode*	16	16
Reduced ZPA	*ConveyLinx Ai in reduced ZPA mode*	30	30

Conveyor Control DAPs for Integrated PLC Topology Configuration

ConveyLinx Mode	DAP from Catalog	Input Bytes	Output Bytes
Full PLC	*ConveyLinx Ai in PLC Mode*	64	64
PLC with ConveyLogix	*ConveyLinx Ai in PLC mode with ConveyLogix*	32	32
Reduced PLC	*ConveyLinx Ai in reduced PLC mode*	16	16
Reduced	*ConveyLinx Ai in*	30	30

ZPA	reduced ZPA mode		
Full ZPA	ConveyLinx Ai in ZPA mode	64	64
ZPA with Merge	ConveyLinx Ai Merger	64	64

✱ UDT Details in the following sections are based upon file *CLAi_All_UDTs_rev1.9.udt*

14.7.1.1. Full ZPA Mode Inputs

CLXAiZPA_IN

Element Name	Sub Element	Data Type	Link
LocalUpstreamZoneStatus_Rev		Byte	More Info
LocalUpstreamZoneStatus_Fwd		Byte	
LocalDownstreamZoneStatus_Rev		Byte	
LocalDownstreamZoneStatus_Fwd		Byte	
ArrivalCountUpstreamZone		Int	More Info
DepartureCountUpstreamZone		Int	
ArrivalCountDownstreamZone		Int	
DepartureCountDownstreamZone		Int	
Diagnostic	LeftMotorOverheat	Bool	More Info
	LeftMotorMaxTorque	Bool	
	LeftMotorShort	Bool	
	LeftMotorNotConn	Bool	
	LeftMotorOverload	Bool	
	LeftMotorStalled	Bool	
	LeftMotorBadHall	Bool	
	LeftMotorNotUsed	Bool	
	ModuleResetFlag	Bool	
	Index06Bit01	Bool	
	OverVoltage	Bool	
	LeftMotorAnyErr	Bool	
	ConnectionsNotOK	Bool	
	UpstreamJamErr	Bool	
	LeftSensLowGain	Bool	
	LowVoltage	Bool	

	RightMotorOverheat	Bool	
	RightMotorMaxTorque	Bool	
	RightMotorShort	Bool	
	RightMotorNotConn	Bool	
	RightMotorOverload	Bool	
	RightMotorStalled	Bool	
	RightMotorBadHall	Bool	
	RightMotorNotUsed	Bool	
	Index07Bit00	Bool	
	Index07Bit01	Bool	
	OverVoltage1	Bool	
	RightMotorAnyErr	Bool	
	Index07Bit04	Bool	
	DownstreamJamErr	Bool	
	RightSensLowGain	Bool	
	LowVoltage1	Bool	
TrackingUpstreamZone		DWord	More Info
TrackingDownstreamZone		DWord	
ReleaseCounterUpstreamZone		Int	
ReleaseCounterDownstreamZone		Int	
GetDischargeTracking_Fwd		DWord	More Info
GetDischargeTracking_Rev		DWord	
AllSensorPortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	

	Bit14	Bool	
	Heartbeat	Bool	
	LeftPin2	Bool	
	Bit01	Bool	
	RightPin2	Bool	
	Bit03	Bool	
	LeftSensor	Bool	
	Bit05	Bool	
	RightSensor	Bool	
	Bit07	Bool	
Index19		Word	
ConveyStopStatus	Bit08	Bool	More Info
	Bit09	Bool	
	StopActiveCommandPLC	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bit00	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	StopActiveOtherModule	Bool	
	StopActiveLostConn	Bool	
	StopActiveLostPLC	Bool	
Future		Array[21..31]	

		of Word	

14.7.1.2. Full ZPA Mode Outputs

CLXAiZPA_OUT

Element Name	Sub Element	Data Type	Link
InductTrackingOnUpstreamZone		DWord	More Info
InductTrackingOnDownstreamZone		DWord	
AccumulateControlUpstream	AccumUpstreamToThisZone	Bool	More Info
	SetArrival	Bool	
	JogDefaultDir	Bool	
	JogOppDir	Bool	
	WakeUp	Bool	
	MaintMode	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Accumulate	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
AccumulateControlDownstream	AccumUpstreamToThisZone	Bool	More Info
	SetArrival	Bool	
	JogDefaultDir	Bool	
	JogOppDir	Bool	
	WakeUp	Bool	
	MaintMode	Bool	

	Bit14	Bool	
	Bit15	Bool	
	Accumulate	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
SetSpeedLeftMotor		Int	More Info
SetSpeedRightMotor		Int	
SetReleaseCountUpstream		Int	More Info
SetReleaseCountDownstream		Int	
SetInductStatus		Word	More Info
SetDishargeStatus		Word	
SetInductTrackingFwd		DWord	
SetInductTrackingRev		DWord	
ClearMotorError		Word	More Info
Index17		Word	
Index18		Word	
ConveyStopControl		Word	More Info
ClearJamUpstream		Word	More Info
ClearJamDownstream		Word	
GlobalDirAccModeUpstr		Word	More Info
GlobalDirAccModeDnstr		Word	
Future		Array[24..31]	

		of Word	

14.7.1.3. Reduced ZPA Mode Inputs

CLXAiZPAmini_IN

Element Name	Sub Element	Data Type	Link
LocalUpstreamZoneStatus_Rev		Byte	
LocalUpstreamZoneStatus_Fwd		Byte	More Info
LocalDownstreamZoneStatus_Rev		Byte	
LocalDownstreamZoneStatus_Fwd		Byte	
ArrivalCountUpstreamZone		Int	
DepartureCountUpstreamZone		Int	More Info
ArrivalCountDownstreamZone		Int	
DepartureCountDownstreamZone		Int	
Diagnostic	LeftMotorOverheat	Bool	
	LeftMotorMaxTorque	Bool	
	LeftMotorShort	Bool	
	LeftMotorNotConn	Bool	
	LeftMotorOverload	Bool	
	LeftMotorStalled	Bool	
	LeftMotorBadHall	Bool	
	LeftMotorNotUsed	Bool	More Info
	ModuleResetFlag	Bool	
	Index06Bit01	Bool	
	OverVoltage	Bool	
	LeftMotorAnyErr	Bool	
	ConnectionsNotOK	Bool	
	UpstreamJamErr	Bool	
	LeftSensLowGain	Bool	
	LowVoltage	Bool	

	RightMotorOverheat	Bool	
	RightMotorMaxTorque	Bool	
	RightMotorShort	Bool	
	RightMotorNotConn	Bool	
	RightMotorOverload	Bool	
	RightMotorStalled	Bool	
	RightMotorBadHall	Bool	
	RightMotorNotUsed	Bool	
	Index07Bit00	Bool	
	Index07Bit01	Bool	
	OverVoltage1	Bool	
	RightMotorAnyErr	Bool	
	Index07Bit04	Bool	
	DownstreamJamErr	Bool	
	RightSensLowGain	Bool	
	LowVoltage1	Bool	
ReleaseCounterUpstreamZone		Int	More Info
ReleaseCounterDownstreamZone		Int	
AllSensorPortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Heartbeat	Bool	
	LeftPin2	Bool	
	Bit01	Bool	

	RightPin2	Bool	
	Bit03	Bool	
	LeftSensor	Bool	
	Bit05	Bool	
	RightSensor	Bool	
	Bit07	Bool	
Future		Array[11..14] of Word	

14.7.1.4. Reduced ZPA Mode Outputs

CLXAiZPAmini_OUT

Element Name	Sub Element	Data Type	Link
AccumulateControlUpstream	AccumUpstreamToThisZone	Bool	More Info
	SetArrival	Bool	
	JogDefaultDir	Bool	
	JogOppDir	Bool	
	WakeUp	Bool	
	MaintMode	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Accumulate	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
AccumulateControlDownstream	AccumUpstreamToThisZone	Bool	More Info
	SetArrival	Bool	
	JogDefaultDir	Bool	
	JogOppDir	Bool	
	WakeUp	Bool	
	MaintMode	Bool	
	Bit14	Bool	

	Bit15	Bool	
	Accumulate	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
SetSpeedLeftMotor		Int	More Info
SetSpeedRightMotor		Int	
SetReleaseCountUpstream		Int	More Info
SetReleaseCountDownstream		Int	
SetInductStatus		Word	More Info
SetDishargeStatus		Word	
ClearMotorError		Word	More Info
Index09		Word	
Index10		Word	
ClearJamUpstream		Word	More Info
ClearJamDownstream		Word	
GlobalDirAccModeUpstr		Word	More Info
GlobalDirAccModeDnstr		Word	

14.7.1.5. Full PLC Mode Inputs

CLXAiPLC_IN

Element Name	Sub Element	Data Type	Link
ConveyStopStatus	Bit08	Bool	More Info
	Bit09	Bool	
	StopActiveCommandPLC	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bit00	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	StopActiveOtherModule	Bool	
	StopActiveLostConn	Bool	
	StopActiveLostPLC	Bool	
AllSensorPortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Heartbeat	Bool	

	LeftPin2	Bool	
	Bit01	Bool	
	RightPin2	Bool	
	Bit03	Bool	
	LeftSensor	Bool	
	Bit05	Bool	
	RightSensor	Bool	
	Bit07	Bool	
SensorDetect	Bits08_15	SInt	More Info
	RightSensorDetect	Bool	
	LeftSensorDetect	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
VoltageMotors		Int	More Info
LeftMotorCurrent		Int	More Info
LeftMotorFreq		Int	
LeftMotorCalcTemp		Byte	
ModuleTemp		Byte	
LeftMotorDiagnostic	Overheat	Bool	More Info
	MaxTorque	Bool	
	ShortCircuit	Bool	
	MotorNotConnected	Bool	
	Overload	Bool	

	Stalled	Bool	
	BadHall	Bool	
	MotorNotUsed	Bool	
	MotorStatus1	Bool	
	MotorStatus2	Bool	
	MotorPortinDOmode	Bool	
	ReservedBit03	Bool	
	ReservedBit04	Bool	
	BoardOverheat	Bool	
	OverVoltage	Bool	
	LowVoltage	Bool	
RightMotorCurrent		Int	
RightMotorFreq		Int	More Info
RightMotorCalcTemp		Byte	
ModuleTemp1		Byte	
RightMotorDiagnostic	Overheat	Bool	
	MaxTorque	Bool	
	ShortCircuit	Bool	
	MotorNotConnected	Bool	
	Overload	Bool	
	Stalled	Bool	
	BadHall	Bool	More Info
	MotorNotUsed	Bool	
	MotorStatus1	Bool	
	MotorStatus2	Bool	
	MotorPortinDOmode	Bool	
	Bit03	Bool	
	Bit04	Bool	

	Bit05	Bool	
	OverVoltage	Bool	
	LowVoltage	Bool	
LeftMotorDIOstatus	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	ShortCircuit	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bits00_07	SInt	
RightMotorDIOstatus	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	ShortCircuit	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bits00_07	SInt	
UpstreamModuleStatus		Word	More Info
DownstreamModuleStatus		Word	
TrackingFromUpstream		DWord	
Index18		Word	
ServoLocationLeft		Int	More Info
ServoLocationRight		Int	
ServoStatusLeft		Word	

ServoStatusRight		Word	
LeftMotorActualSpeed		Word	More Info
RightMotorActualSpeed		Word	
Future		Array[25..31] of Word	

14.7.1.6. Full PLC Mode Outputs

CLXAiPLC_OUT

Element Name	Sub Element	Data Type	Link
ConveyStopControl		Word	More Info
LeftMotorAsDIO	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	EnableDIO	Bool	
	Bit00	Bool	
	EnergizeM8Pin4	Bool	
	EnergizeM8Pin3	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
RightMotorAsDIO	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	

	Bit14	Bool	
	EnableDIO	Bool	
	EnergizeM8Pin2	Bool	
	Bit01	Bool	
	EnergizeM8Pin3	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
SensorPortsDO	Bit08	Bool	
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	More Info
	EnergizeLeftPortPin2	Bool	
	EnergizeRightPortPin2	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	EnableLeftPortPin2	Bool	
	EnableRightPortPin2	Bool	
	Bit07	Bool	
LeftMotorControl	MotorDirection	Bool	More Info
	Bit09	Bool	

	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Run	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
LeftSetMotorBrakeMode		Word	More Info
LeftSetMotorSlaveMode		Word	
RightMotorControl	MotorDirection	Bool	
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	More Info
	Bit14	Bool	
	Bit15	Bool	
	Run	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	

	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
RightSetMotorBrakeMode		Word	More Info
RightSetMotorSlaveMode		Word	
LeftSetMotorSpeed		Int	More Info
RightSetMotorSpeed		Int	More Info
LeftSetMotorAccel		Int	More Info
LeftSetMotorDeccel		Int	
RightSetMotorAccel		Int	More Info
RightSetMotorDeccel		Int	
ClearMotorError		Word	More Info
SendStatusToDownstream		Word	More Info
SendStatusToUpstream		Word	
SensorPolarity	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Left_Pin2	Bool	
	Bit01	Bool	
	Right_Pin2	Bool	

	Bit03	Bool	
	Left_Pin4	Bool	
	Bit05	Bool	
	Right_Pin4	Bool	
	Bit07	Bool	
TrackingToDownstream		DWord	More Info
Index22		Word	
ServoControlDistanceLeft		Int	More Info
ServoControlCommandLeft		Word	
ServoControlDistanceRight		Int	
ServoControlCommandRight		Word	
Future		Array[27..31] of Word	

14.7.1.7. Reduced PLC Mode Inputs

CLXAiPLCmini_IN

Element Name	Sub Element	Data Type	Link
AllSensorPortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Heartbeat	Bool	
	LeftPin2	Bool	
	Bit01	Bool	
	RightPin2	Bool	
	Bit03	Bool	
	LeftSensor	Bool	
	Bit05	Bool	
	RightSensor	Bool	
	Bit07	Bool	
SensorDetect	Bits08_15	SInt	More Info
	RightSensorDetect	Bool	
	LeftSensorDetect	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	

	Bit06	Bool	
	Bit07	Bool	
LeftMotorCalcTemp		Byte	More Info
ModuleTemp		Byte	
LeftMotorDiagnostic	Overheat	Bool	More Info
	MaxTorque	Bool	
	ShortCircuit	Bool	
	MotorNotConnected	Bool	
	Overload	Bool	
	Stalled	Bool	
	BadHall	Bool	
	MotorNotUsed	Bool	
	MotorStatus1	Bool	
	MotorStatus2	Bool	
	MotorPortinDOmode	Bool	
	Bit03	Bool	
	Bit04	Bool	
	BoardOverheat	Bool	
	OverVoltage	Bool	
	LowVoltage	Bool	
RightMotorCalcTemp		Byte	More Info
ModuleTemp_1		Byte	
RightMotorDiagnostic	Overheat	Bool	More Info
	MaxTorque	Bool	
	ShortCircuit	Bool	
	MotorNotConnected	Bool	
	Overload	Bool	
	Stalled	Bool	

	BadHall	Bool	
	MotorNotUsed	Bool	
	MotorStatus1	Bool	
	MotorStatus2	Bool	
	MotorPortinDOmode	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	OverVoltage	Bool	
	LowVoltage	Bool	
LeftMotorDIOstatus	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	ShortCircuit	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bits00_07	SInt	
RightMotorDIOstatus	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	ShortCircuit	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bits00_07	SInt	

14.7.1.8. Reduced PLC Mode Outputs

CLXAiPLCmini_OUT

Element Name	Sub Element	Data Type	Link
LeftMotorAsDIO	Bit08	Bool	
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	EnableDIO	Bool	More Info
	Bit00	Bool	
	EnergizeM8Pin4	Bool	
	EnergizeM8Pin3	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
RightMotorAsDIO	Bit08	Bool	
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	More Info
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	

	EnableDIO	Bool	
	EnergizeM8Pin2	Bool	
	Bit01	Bool	
	EnergizeM8Pin3	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
SensorPortsDO	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	EnergizeLeftPortPin2	Bool	
	EnergizeRightPortPin2	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	EnableLeftPortPin2	Bool	
	EnableRightPortPin2	Bool	
	Bit07	Bool	
LeftMotorControl	MotorDirection	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	

	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Run	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
RightMotorControl	MotorDirection	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Run	Bool	
	Bit01	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	

	Bit07	Bool	
LeftSetMotorSpeed		Int	More Info
RightSetMotorSpeed		Int	
ClearMotorError		Word	More Info

14.8. Troubleshooting

Issue	Possible Cause	Action
PLC cannot communicate with a ConveyLinx module	Incorrect Profinet name syntax	Verify that the Profinet Name follows the syntax as previously described. If online, you can check module connections to your PLC with the *Accessible Nodes* function in the TIA Portal
	Incorrect I.P. address	Make sure module's IP address is correct as previously described in Adding Modules
	Subnet mask mismatch	Make sure PLC's subnet mask matches the subnet mask in all ConveyLinx modules
	Subnet mask mismatch	For Full ZPA, Full PLC Controlled, and Reduced PLC Controlled modes – ConveyLinx-ERSC firmware 4.3 and higher For Reduced ZPA Mode – ConveyLinx firmware 4.24 and higher
Intermittent Communication drop out between PLC and module	Improper Module Update time	Verify that the Update time is properly set based upon the DAP you have chosen. Here is an example for ZPA mode
	PLC's connection limit or I/O memory usage limit has been reached	Verify quantity of connections and memory limitations for the PLC processor being used and upgrade processor if needed. Reduce number of connected modules if connection limit has been reached If PLC I/O memory limit has been exceeded, change Full ZPA or Full PLC mode modules to their equivalent "reduced" DAP to reduce I/O memory required
	Faulty cabling	Make sure all cables are the recommended type for the installation environment and that all connections are good
	Incorrect switch type	Verify that the switch being used is Profinet rated and is selected from one of the units available in the Catalog in the PLC programming environment