ConveyLinx ERSC Complete Guide

Version 5.1 — Dec 09, 2021

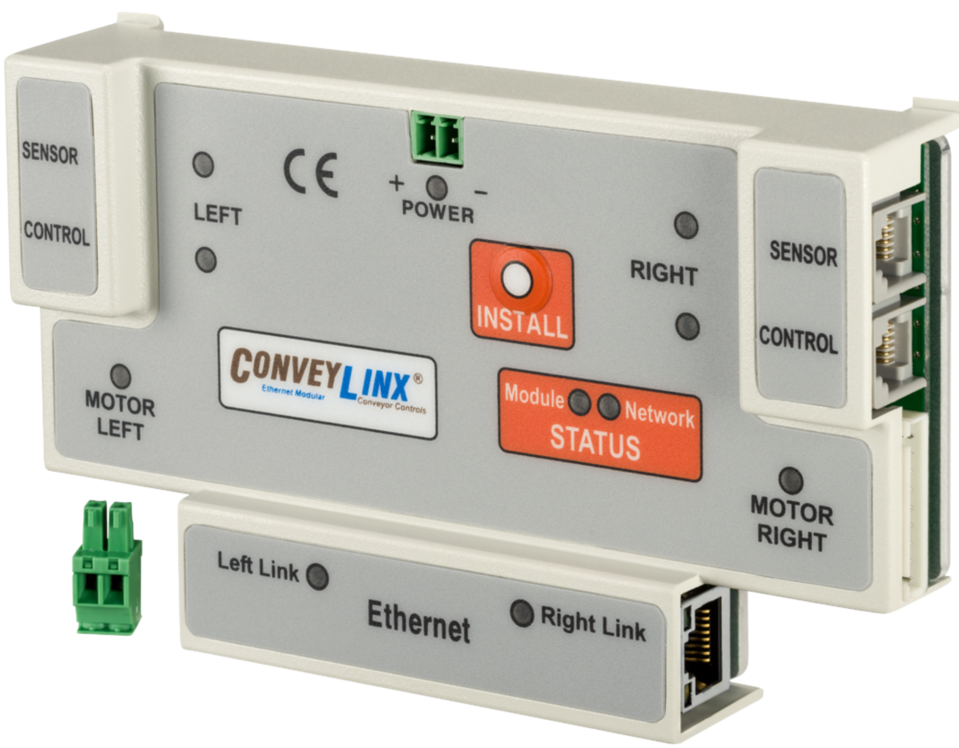


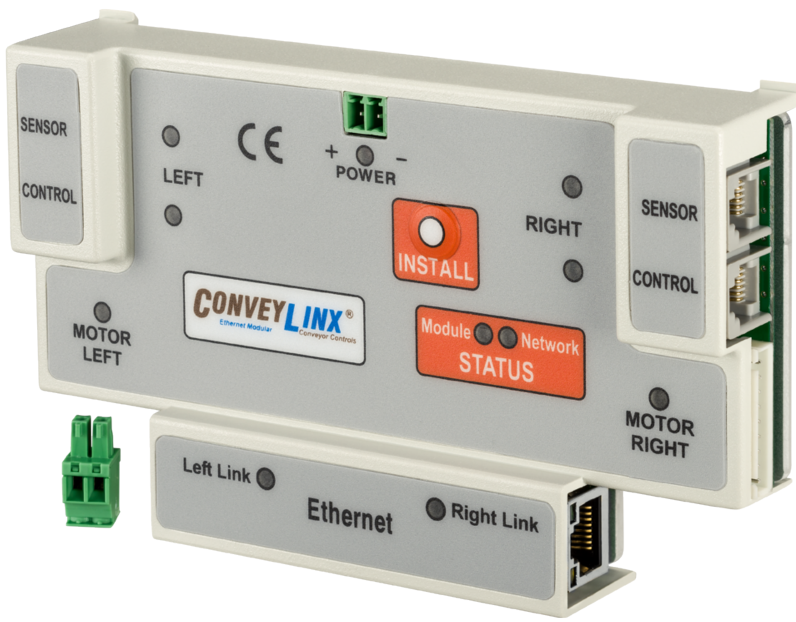
Table of Contents

1. About This Manual

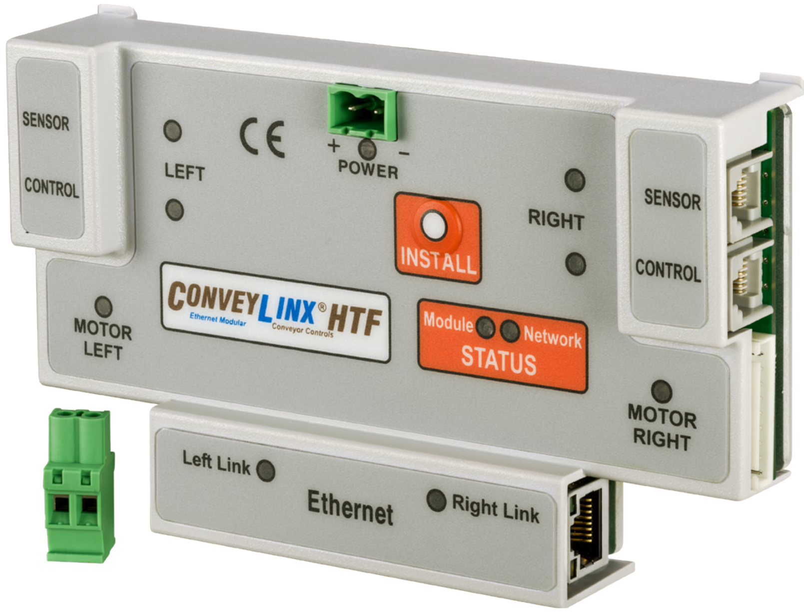
> ConveyLinx ERSC Complete Guide Version 5.1 includes functionality for Conveylinx-ERSC Firmware versions 4.27 / 5.07.
> For ConveyLinx-ERSC Firmware versions 4.25 / 5.02 please refer to ConveyLinx ERSC Complete Guide Manual Version 5.0

Products Covered in this Manual

ConveyLinx-ERSC
Article Number
3010-0000



ConveyLinx-HTF
Article Number
3010-3000



ConveyLinx-ERSC-E
Article Number
3110-0000

ConveyLinx-HTF-E
Article Number
3110-3000

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

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 Hardwired Interlock
How to Auto-Replace a module

Basic things you can do with 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

Some advanced things you can do with EasyRoll

4. Module Hardware

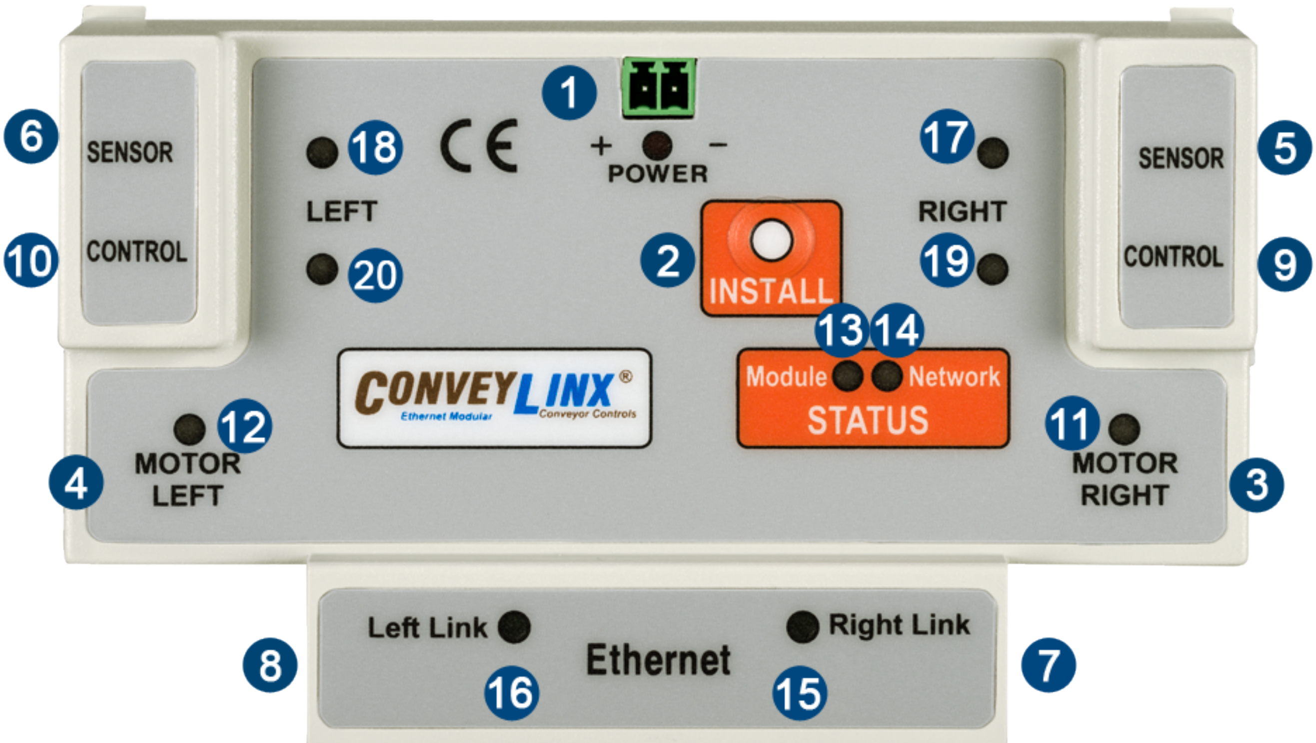
ConveyLinx Modules are designed to be installed and integrated into the conveyor's mechanical side frame assembly. The **ConveyLinx Module** is a controller for up to 2 Motorized Drive Roller (MDR) conveyor zones. Each **ConveyLinx Module** provides connection points for 2 MDR units with their corresponding 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 Module 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 Module Components
Mounting Dimensions
Motor Ports
Sensor & Control Ports
Ethernet Ports
ConveyLinx Module Internal Fusing
LED Status Indicators

4.1. Identifying Module Components

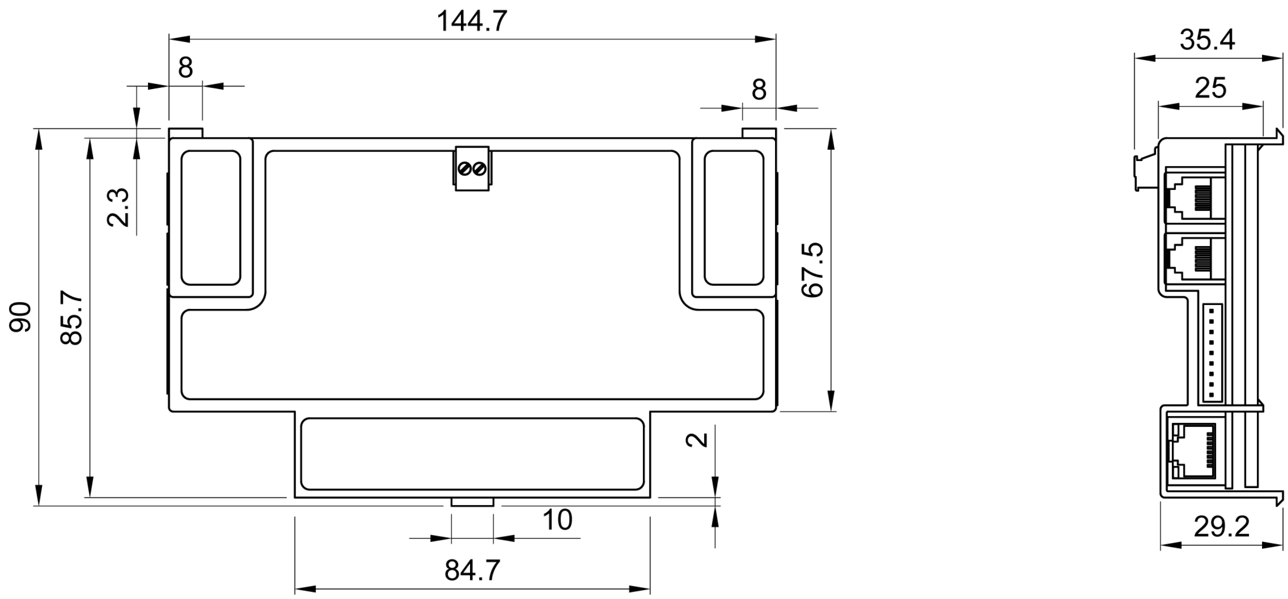


Item	Description
1	24VDC Power Connector
2	Install Button – Used for Auto-Configuration Procedure and Auto Module Replacement
3 & 4	Motor Left and Motor Right 9-pin JST style header for MDR/PGD connection
5 & 6	Left Sensor Port and Right Sensor Port RJ-12 style jack for zone photo-sensor connection
7 & 8	Link Left and Link Right RJ-45 style Ethernet network communication connection between modules
9 & 10	Left Control Port and Right Control Port RJ-12 style ports for discreet hard-wired signal connections for non-networked interface interlocks and zone control

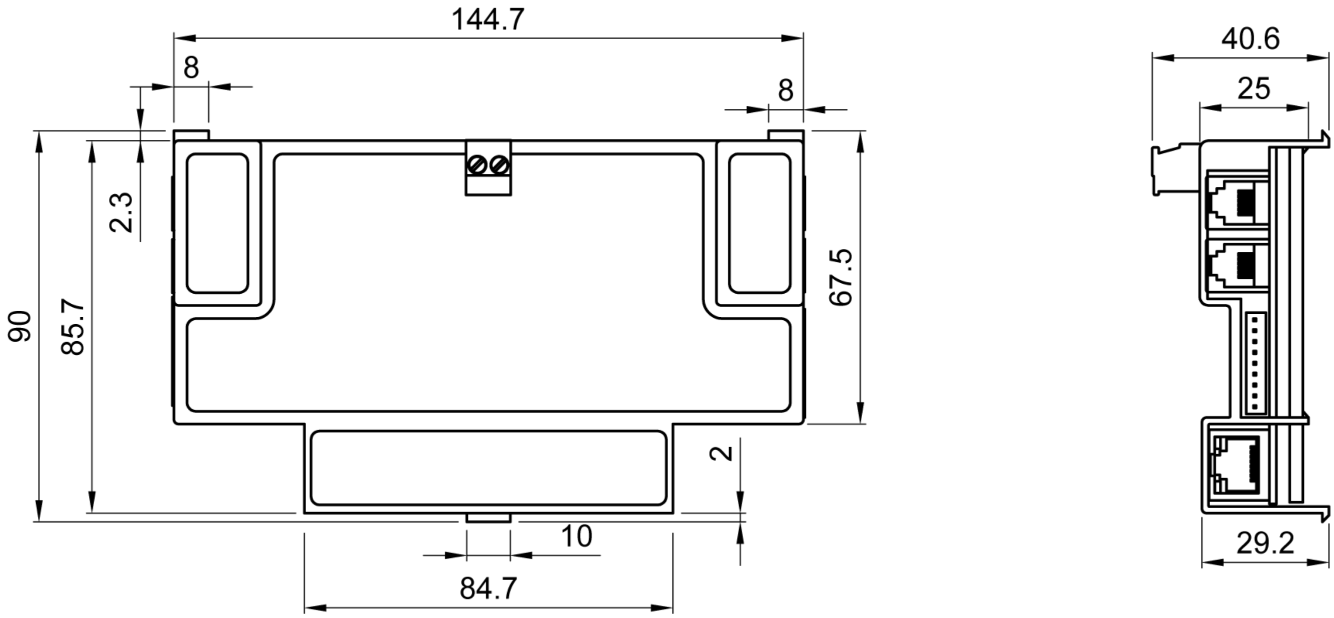
11 & 12	Motor Left LED & Motor Right LED – Motor status indicators
13	Module Status LED Indicator
14	Module Network Status LED Indicator
15 & 16	Left Link & Right Link Status LED Indicators
17 & 18	Left Sensor & Right Sensor Status LED Indicators
19 & 20	Control Port Left & Control Port Right Status LED Indicators

4.2. Mounting Dimensions

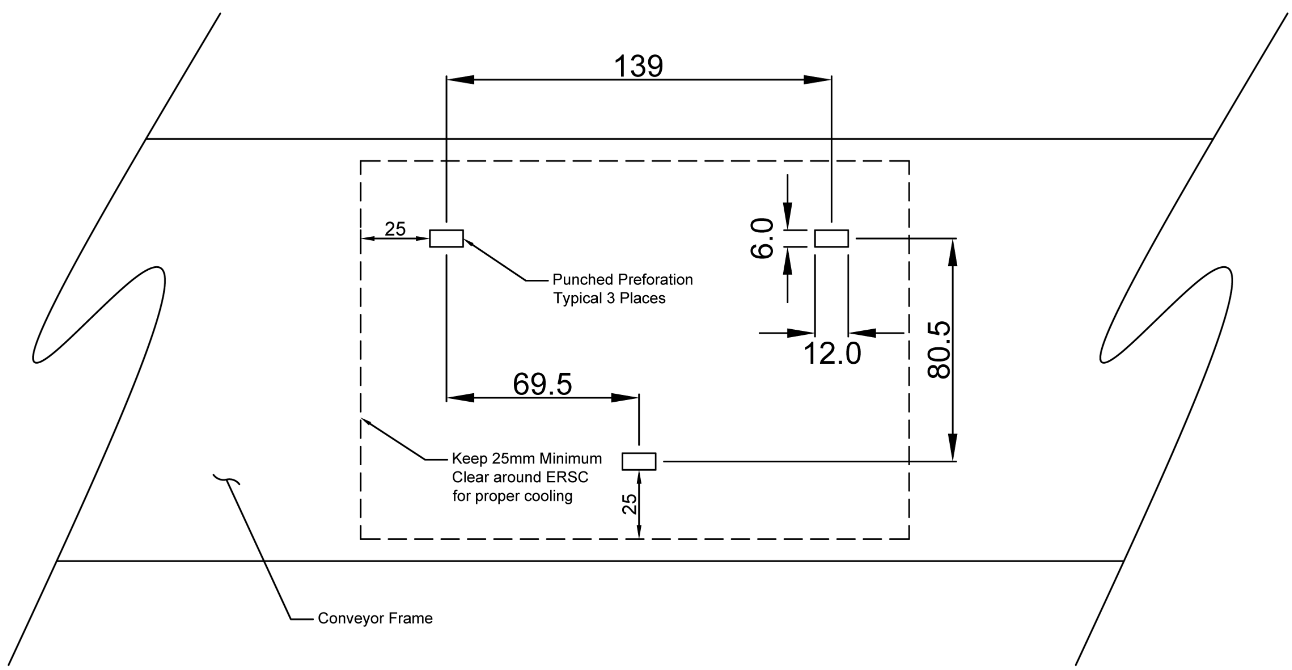
ConveyLinx (3010-0000) and ConveyLinxE (3110-0000)



ConveyLinx-HTF (3010-3000) and ConveyLinxE-HTF (3110-3000)

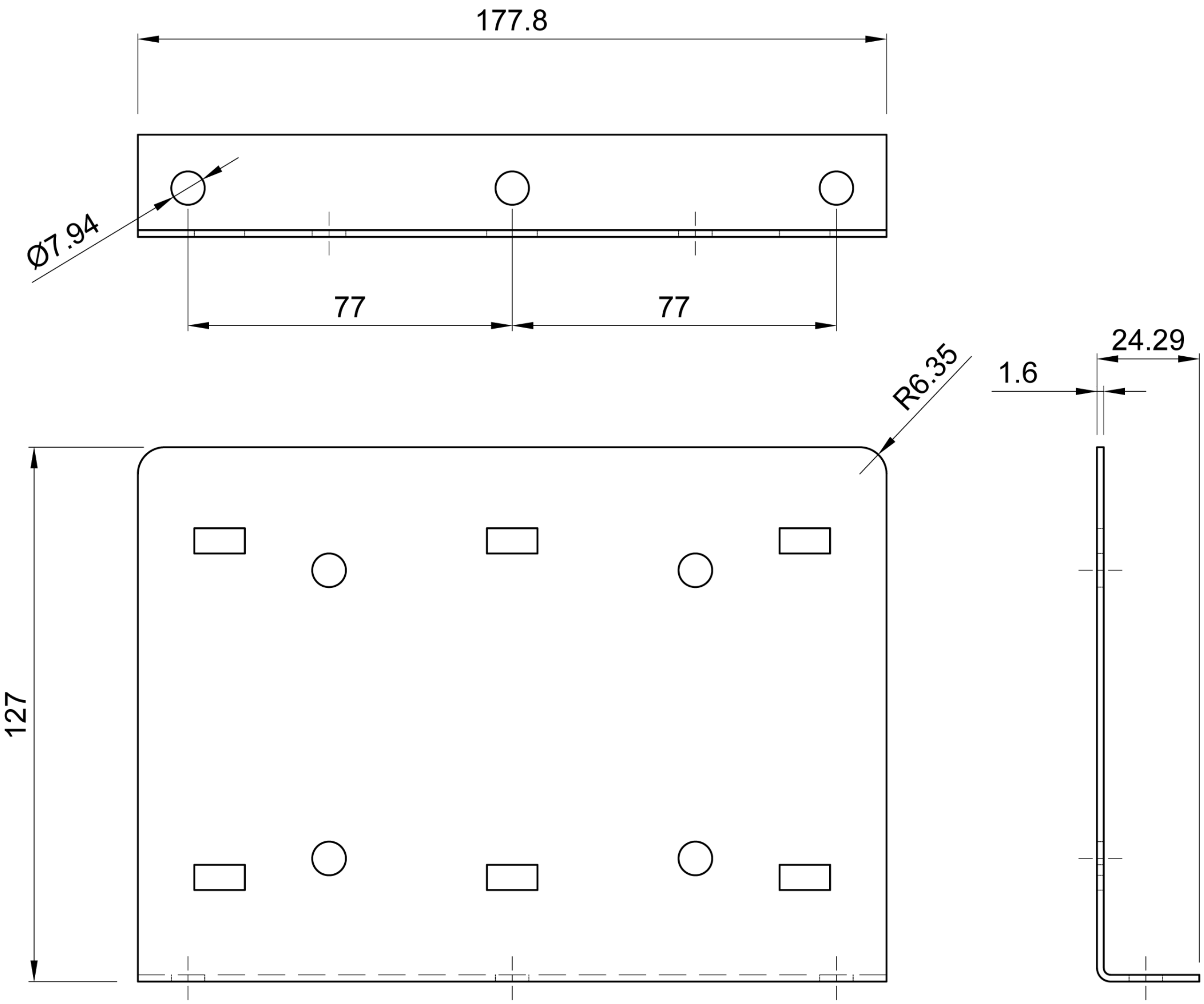


Conveyor Frame Preforations Dimensions



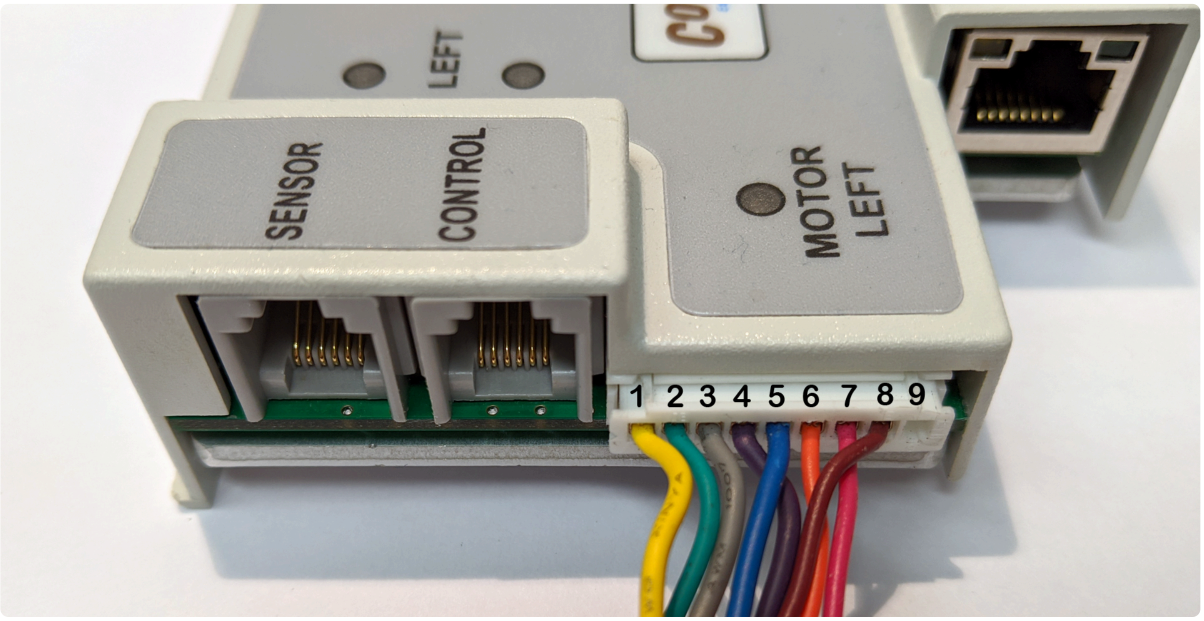
Optional Mounting Bracket

Pulseroller Order Type: **BRKT-UNIV** Order Code: **099-1002**



4.3. Motor Ports

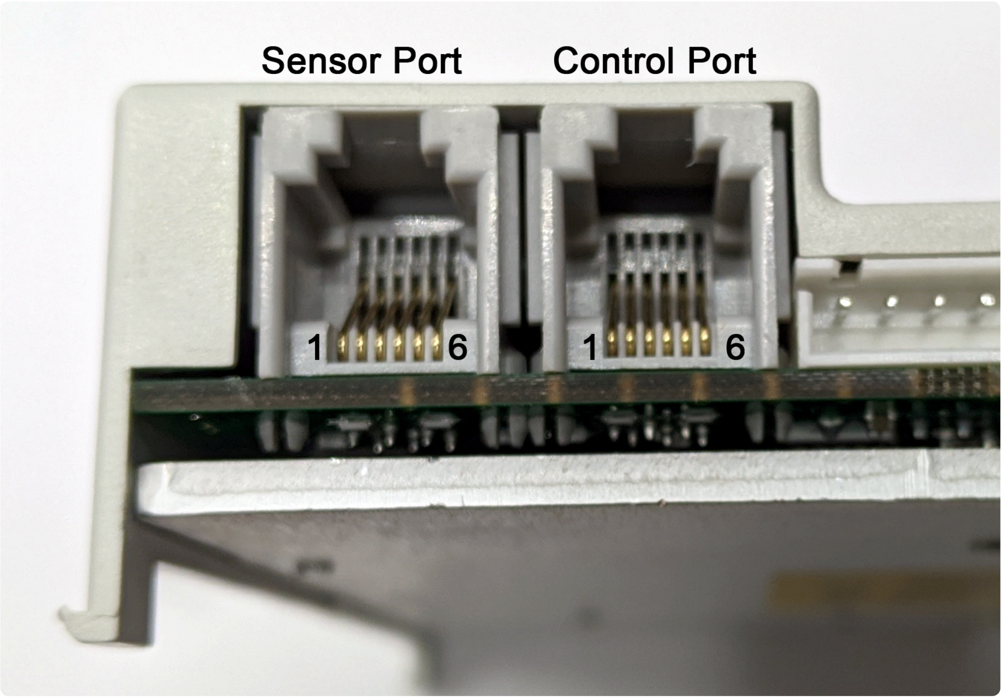
Both the Left and the Right Motor Ports utilize a 9-pin JST brand female receptacle. Each receptacle is mechanically keyed to assure proper orientation upon plugging in. The motor connector pin-outs are as shown



Pin	Description
1	GND
2	Vcc – Hall Effect Sensor Power
3	Motor Winding U
4	Motor Winding V
5	Motor Winding W
6	Hall Effect Sensor U
7	Hall Effect Sensor V
8	Hall Effect Sensor W
9	Optional – Mechanical Holding Brake Control

Please note the JST connector is keyed so you cannot plug it in upside down

4.4. Sensor & Control Ports



Each Sensor and Control Port is a standard RJ-12 style jack.

Sensor Port Pin Out

Pin	Signal	Description
1		Not Used
2	GND	Module DC Common
3	Sensor Error	Digital Input for Sensor's error output – Auto detect for NPN or PNP
4	Sensor State	Digital Input for Sensor's state output – Auto detect for NPN or PNP
5	Vcc	Module 24VDC Supply
6		Not Used

! Shorting Pins 2 & 5 (Vcc and Gnd) may damage the port's Sensor Detect circuit. If this circuit is damaged, the module will no longer properly detect a connected sensor and will not properly Auto-Configure.

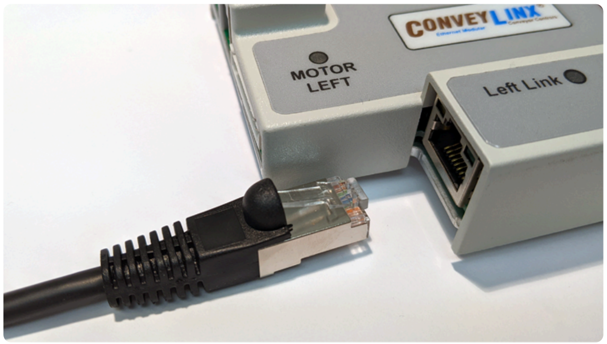
Control Port Pin Out

Pin	Signal	Description
1	Output E	Digital Output for Upstream/Downstream Interlock, SE Module recommended
2	GND	Module DC Common
3	P3 Input	Optional Local Accumulate Digital Input – Auto detect for NPN or PNP
4	P4 Input	Optional Interlock Digital Input – Auto detect for NPN or PNP
5	Vcc	Module 24VDC Supply
6	Output C	Digital Output for Upstream/Downstream Interlock, SE Module recommended

! Single output signal flows between *Pin 1* and *Pin 6*. This signal is very low power (@ 2mA). An SE Breakout Module is recommended when using of this signal.

4.5. Ethernet Ports

Both of these ports are standard RJ-45 jacks conforming to standard Ethernet connection pin-out.



RJ-45 Cable and Ethernet Port

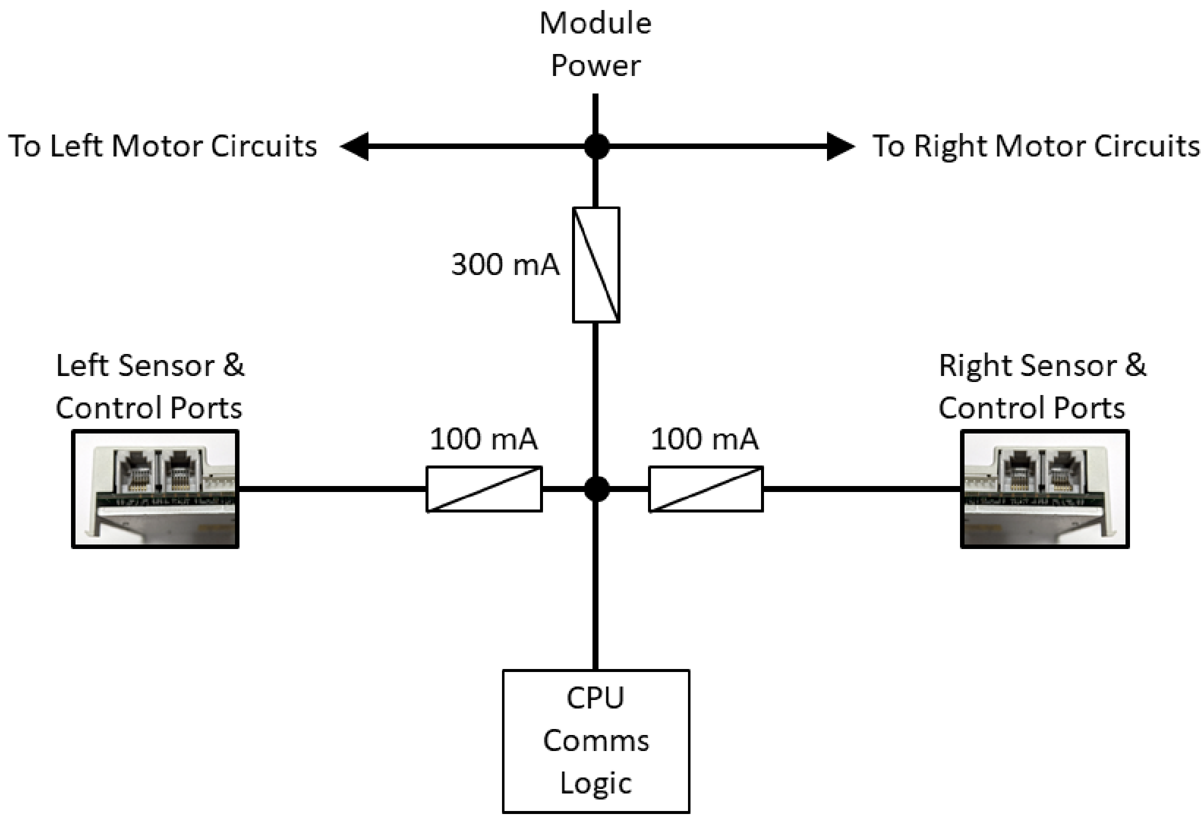
RJ-45 Cable Plugged in

Both Left & Right RJ-45 Cables Plugged In

> **!** All Hardware Revision 1 ConveyLinx Module must use Ethernet shielded crossover style cables. Hardware Revision 2 and later modules can use straight or crossover shielded Ethernet cables. Failure to use SHIELDED cables may result in data loss and unexpected results

4.6. Module Internal Fusing

Because the ConveyLinx Module utilizes a single external power connection for both control power and MDR power; the ConveyLinx Module includes internal re-settable fusing to protect the control power from the MDR power. The overall control power fuse is rated at 300 mA and this is the source for CPU, LED’s, sensor ports, control ports, etc. From this circuit, the Sensor and Control port pairs share their own separate 100 mA fuses.



ConveyLinx-ERSC Internal Fusing Diagram

> ! User’s must take care in the types of sensors and devices that need to connect to a given ConveyLinx Module and assure that the current draw on any Sensor/ Control port pair does not exceed the fuse ratings. If any of the 100 mA fuses trips, there is no direct indication of this state. An indirect indication could be to see if a connected sensor is powered when plugged into either port.

> ! If the 300 mA fuse trips, upon the automatic reset of the fuse, the ConveyLinx Module should reboot on its own. However, the ConveyLinx Module may require the power to be cycled manually to fully restart

4.7. Over-Voltage Protection

There are two sources of potential excess voltage:

1. Over-voltage coming from the power supply
2. Over-voltage generated by an over-driven motor

Power Supply

The ConveyLinx Module utilizes an SMBJ30A “clamp down” chip on the power input to detect any voltage coming in that is greater than 33.3V. If this is detected, it shorts the +24V to GND to protect the remaining electronics on the module. This chip serves a dual function as it also helps filter ESD voltage spikes as well.

Over-Driven Motor

The ConveyLinx Module firmware monitors the motor voltage and if it rises above 30V, it automatically shunts the motor driver transistors together (the same as in normal braking) so that the energy is contained within the motor coils and away from the controller electronics. The inherent nature of motor coil geometry and construction allow for this driven state to be tolerated for several seconds or even minutes before there is any potential damage to the motor coil circuit. When the voltage drops below 29V, the ConveyLinx Module firmware allows the motor power transistors to switch back to their normal operation.

4.8. LED Status Indicators

ConveyLinx Module 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 Module

Communications

Indicator	Item	LED State	Description
Ethernet Left Link	16	OFF	No connection established
		Solid Green	Connection is established
		Blinking Green	When data transmission activity is occurring
Ethernet Right Link	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	11 & 12	OFF	Motor is not running and no faults detected
		Solid Green	Motor is running
		Flashing Green (intermittent)	Motor is being moved or rotated by external force
		Solid Red	Motor is not connected or motor is stalled
		Blinking Red	Motor is overloaded or over-heated
		Flashing Red	Motor Stopped: Short circuit detected between at least two of the phase windings

			Motor Running: Over-current condition

Network & Module Status

Indicator	Item	LED State	Description
Module Status	13	Solid Red	Module is booting up or during Auto-Replacement procedure, module is attempting to retrieve data from neighbor module(s)
		Blinking Red	Module is starting task processes
		Blinking Green	Module is ready
		Flashing Green & Blinking Red	Auto-replace procedure has been properly triggered
		Flashing Red & Blinking Green	Failsafe Mode
		Flashing Red	Auto Configure Mode is active
		Blinking Amber	Performing firmware upgrade check
		Solid Amber	Firmware upgrade in progress
Network Status	14	Solid Red	Starting Inter-module communications
		Blinking Red	Establishing inter-module connections
		Blinking Green	Inter-module communications established

Sensors

Indicator	Item	LED State	Description
Sensor Left & Right	17 & 18	Solid Green	Sensor is Blocked
		Solid Red	Sensor Error State (Pin 3) is Active
		Blinking Red	Arrival Jam or missing sensor
		Blinking Green/ Amber	Sensor Jam

		Flashing Green	When Sensor is blocked, indicates external device (PLC/PC controller or EasyRoll) has accumulated the zone and inhibiting release

Control Ports

Indicator	Item	LED State	Description
Control Port Left & Right	19 & 20	Solid Green	If acting as Upstream Port: Wake-up signal is logically enabled on Pin 4
			If acting as Downstream Port: Lane Full signal is logically enabled
			If Module is in PLC I/O Mode: Signal on Pin 4 is logically enabled
		Solid Red	Local Accumulate signal (Pin 3) is logically enabled
		Flashing Red	Module configuration error

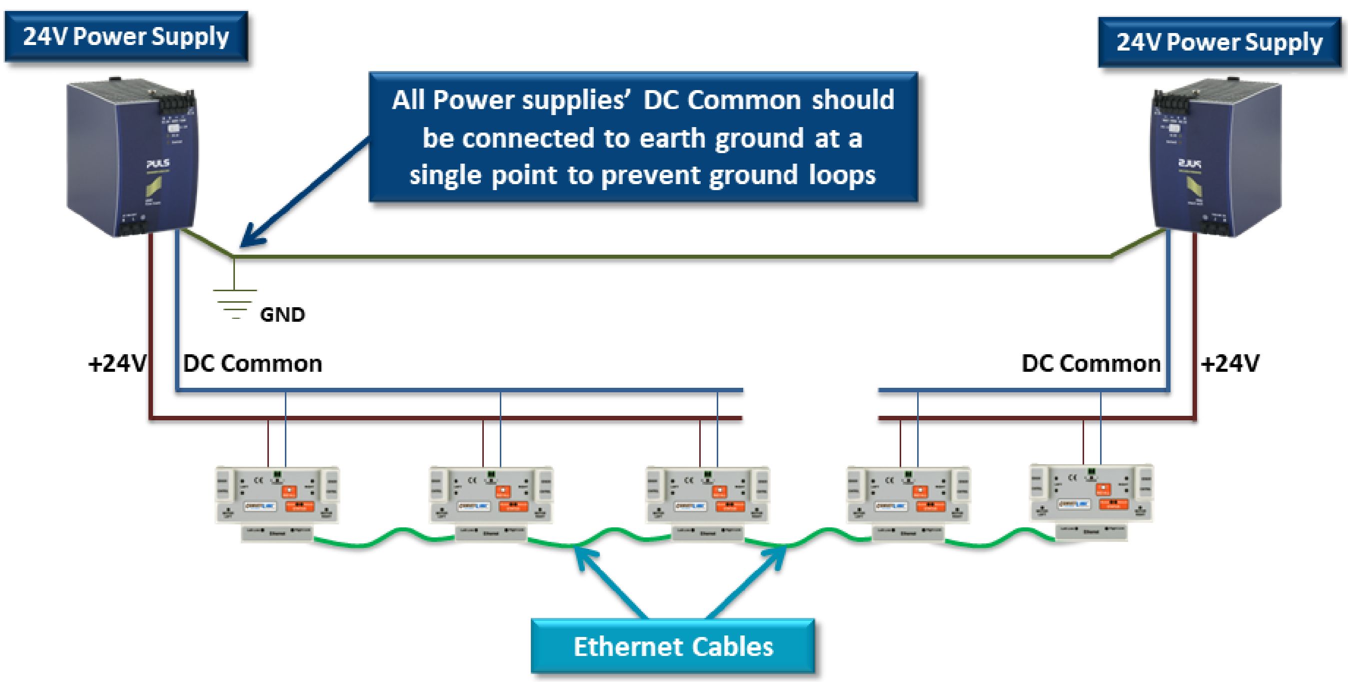
Special Cases

Indicator	Item	LED State	Description
All Sensor, Control Port & Motor	11 12 17 18 19 20	Flashing Red	Module in stopped state
Left Sensor, Left Control Port, & Left Motor	18 20 12	Flashing Green	ZPA zone on left side of module is in Maintenance Mode*
Right Sensor, Right Control Port, & Right Motor	17 19 11	Flashing Green	ZPA zone on right side of module is in Maintenance Mode*

*Maintenance mode only accessible via remote PLC.

4.9. Power and Network Connections

Once MDR's and photo-sensors have been connected based upon the desired conveyor arrangement (1 zone, 2 zone, dual motor zone, etc.); all the ConveyLinx Modules that make up the linear conveyor arrangement need to be interconnected with shielded Ethernet cables (Hardware Rev.1 modules must use shielded crossover Ethernet cables) and each module needs to receive a 24VDC power connection. The Power Plug connection is the power source for all logic, photo-sensors, and MDR's.
Multiple power supplies should have their DC Common terminals connected together as shown:



> ! This document 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. This document also assumes installation will follow proper equipment grounding practices. "DC common or -"on all power supplies should always be connected to ground. Improper power supply sizing and/or improper grounding practices will produce unexpected results.

Learn about power supply sizing

4.10. Power Supply Sizing

The current loading on the power supply for a group of ConveyLinx Modules 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	BOOST	BOOST 8
Power supply load **per Motor Port** at rated torque at maximum speed	2.5 A	3.5 A	3.5 A
Power supply load **per Motor Port** during motor starting period	3.0 A	5.0 A	8.0 A
Duration of motor starting period	5.0 sec	1.5 sec	3.0 sec

Please note that BOOST 8 is only available on ConveyLinx-ERSC-HTF module. You can select BOOST 8 in EasyRoll for a ConveyLinx-ERSC module, but it will not deliver the BOOST 8 current

Please note that the current values shown are per Motor Port, so if both Motor Ports are being used on a given ConveyLinx Module, the current load seen by the power supply for that module will be double the 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.

4.11. Technical Specifications

These specifications cover ConveyLinx ERSC Hardware Revisions 3 and above and Serial numbers 137101 and higher.

Power Connector

Power connector is included with the ConveyLinx Module when shipped from the factory

ERSC	Phoenix Contact PN: 1827127 MCVR 1,5/2-ST-3,81
ERSC-HTF	Phoenix Contact PN: 1912841 MVSTBW 2,5 HC/ 2-ST-5,08

Electrical Ratings

Power supply voltage	24.0V +/- 10%
Standby current consumption	< 120mA
Motor Starting Current	≤ 5.5A ERSC / ≤ 8A ERSC-HTF
Motor Rated Current	≤ 4A ERSC / ≤ 5A ERSC-HTF
Motor PWM Frequency	10 kHz +/- 0.1%

Maximum Ratings

! Operating outside these parameters may result in permanent ConveyLinx Module failure or unexpected device behavior

Minimum Operating Voltage	21V
Maximum Operating Voltage	30V
Storage temperature	-40ºC to 150º C (-40ºF to 300ºF)
Ambient Operating temperature (ERSC)	0ºC to 40ºC (32°F to 104°F)
Ambient Operating temperature (HTF)	-30ºC to 45ºC (-22ºF to 113°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	IP20
Maximum peak current	21.5A*
Maximum motor start current	12A

*This is the maximum current that will be allowed by the hardware over current protection circuitry. On board firmware limits the amount of current based on the quantity and motor types connected

Certifications & Standards

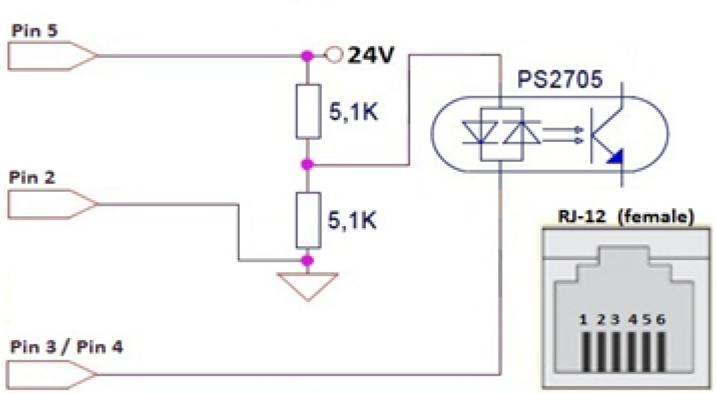
BDS EN 61131-2:2008	Programmable controllers — Part 2: Equipment requirements and tests
BDS EN 61000-6-2:2006	Electromagnetic compatibility (EMC) — Part 6-2: Generic standards – Immunity for industrial environments
BDS EN 61000-6-4:2007	Electromagnetic compatibility (EMC) — Part 6-4: Generic standards – Emission standard for industrial environments
BDS EN 55016-2-1+A1:2006	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	Electromagnetic compatibility – Requirements for household appliances, electric tools and similar apparatus — Part 1: Emission
BDS EN 61000-4-2+A1+A2:2004	Electromagnetic compatibility (EMC) Part 4-2: Electromagnetic discharge Immunity test
BDS EN 61000-4-3/A1:2008	Electromagnetic compatibility (EMC) Part 4-3 Radiated radio-frequency, electromagnetic field immunity test.
BDS EN 61000-4-4:2006	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 61000-4-6:2007	Electromagnetic compatibility (EMC) Part 4-6 Immunity to conducted disturbances, induced by radio-frequency field

BDS EN 61000-4-11:2006	Electromagnetic compatibility (EMC) Part 4-11 Voltage dips, short interruptions and voltage variations immunity tests

Sensor & Control Port I/O

Inputs

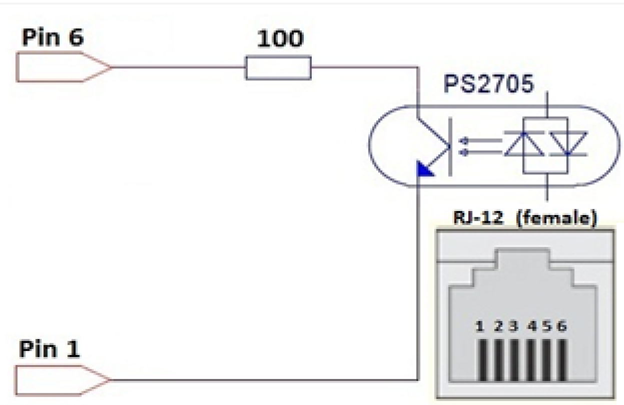
The Sensor and Control port both have 2 inputs each. Sensor and Control port inputs are auto-sensing for the connected circuit type. Input function as either PNP or NPN. Please note that both sourcing and sinking current will activate the input



Minimum ON current	1.5 mA
Maximum OFF current	0.4 mA

Outputs

The Control Port output is an NPN transistor whose Emitter (Pin 1) and Collector (Pin 6) are made available to be connected either as sourcing or sinking device in the user's input circuit



Minimum ON current	2 mA
Maximum ON current	8 mA
OFF current	< 100 nA

Sensor & Control Port Power Pins

Pin 5 of all RJ-12 ports provides 24V for powering up a photo-eye or for biasing the output transistor found on the control port. The current that those pins can supply is limited internally. Each side of the module is fused separately and each side’s control port and sensor port share a solid-state fuse rated at 100 mA. For example, If there is one photo-eye plugged into the left sensor port and one photo-eye plugged in the left control port, then the combined consumption of the two photo eyes must not exceed 100mA.

! 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

Supported motor types	3 phase BLDC motors with 3 Hall Effect sensors
PWM frequency	10 kHz +/- 0.1%
Maximum starting current	8A
Maximum rated current	5A
Motor Protection*	Coil-to-coil short, coil-to-Vcc short, overheating, over-voltage, under-voltage, stall sensing and protection
Brake output type	PNP (high side switch)
Brake output current	0.5A (1 A peak)

! *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 IO Mode as Outputs

Motor Coil Pins 3, 4, & 5

In certain modes of operation (PLC I/O and ConveyLogix PLC), Pins 3,4, and 5 can each be independently switched on and off as general purpose digital outputs. Any individual pin can sink up to 1A to ground in these modes, but the total for all 3 pins combined cannot exceed 1.5A. In general purpose I/O mode, these pins cannot source current.

Brake Output Pin 9

As of firmware version 4.19 and later, the brake output pin can be configured through remote PLC to operate as a general purpose output even if an MDR is connected to the port. This situation requires a special cable or break-out board to be used and the MDR in use cannot have and internal mechanical brake, as that mechanical brake requires a connection to pin 9 for proper operation. The brake-output pin 9 is a 24V high side switch (PNP) that can source up to 0.5A continuously and 1A peak.

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
- CC-Link IE Field Basic (FW 4.27 and later and ODVA FW 5.07 and later)

5. Auto-Configuration

The purpose of *Auto-Configuration* for networked ConveyLinx controls is to provide a simple and easy procedure for linear conveyor system commissioning that does not require a PC or PC based software to implement. The *Auto-Configuration* of Linear Conveyor feature of ConveyLinx requires only the proper interconnection of each module and the press of a button on the most upstream module to complete.

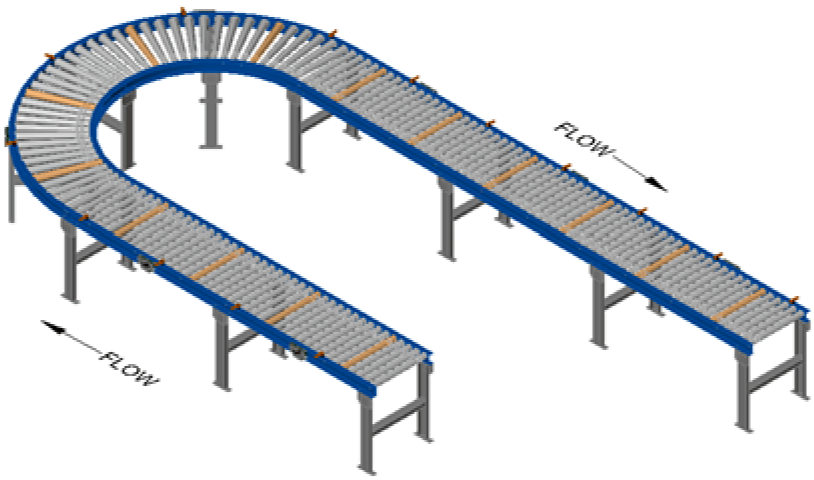
> A networked ConveyLinx solution is capable of controlling more complex conveyor paths that include diverting and merging equipment. However, this requires configuration with a PC and software.

Learn about:

Linear Conveyor
Auto-Configuration Examples
Auto-Configuration Procedure
Expected Auto-Configuration Results
What to do if things go wrong with Auto-Configuration

5.1. Linear Conveyor

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 or totes on the conveyor is continuous from in-feed zone to discharge zone.



Example of a Linear Conveyor

Learn about:

Device Connections to Modules
Examples that will generate errors
Motor rotation definition

5.1.1. Device Connections to Modules

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

In general, each ConveyLinx Module 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 Modules 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 Module can operate as a:

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

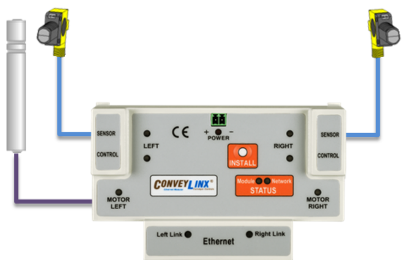
1 zone controller with 1 MDR and 1 photo-sensor on Left or Right Side

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

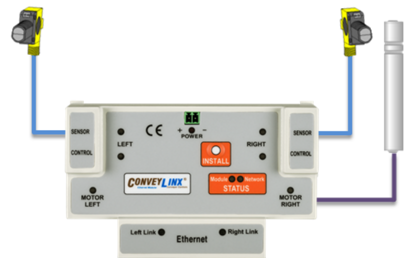
5.1.2. Examples that will generate errors

✻ 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 or cause interruption in the flow of items on the conveyor.

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

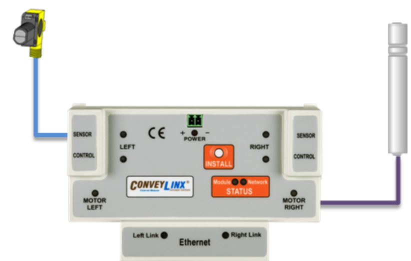


This module will configure as a two zone module but will generate a motor connection error for the missing motor on the Left side



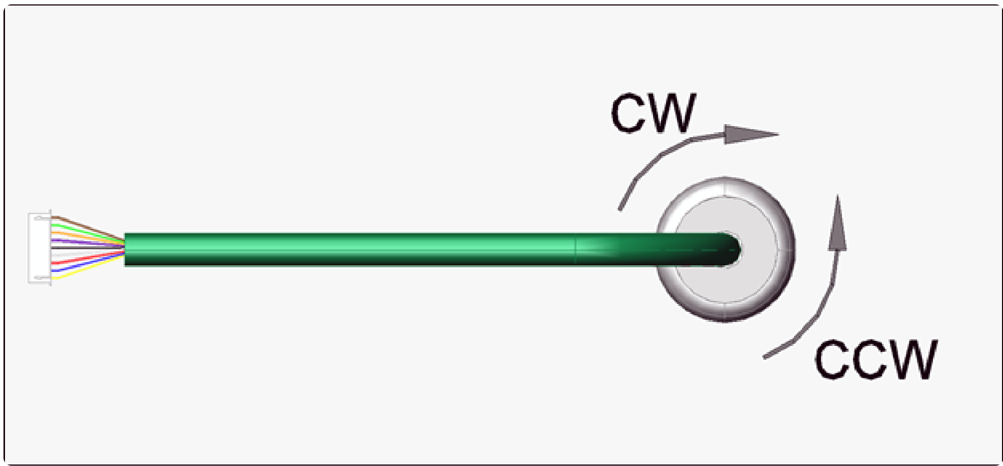
This module will configure as a single zone module on the Right side but will generate a motor connection error for the missing motor on the Right Side

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



5.1.3. Motor Rotation Definition

The ConveyLinx Module 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 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 Module. If the rotation direction is not correct after the *Auto-Configuration Procedure* completes, you can change it in *EasyRoll*. How to change the motor rotation direction

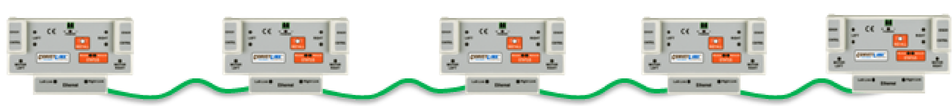
5.2. Procedure

The ConveyLinx *Auto-Configuration Procedure* routine will detect which photo-sensors are connected in order to configure a given ConveyLinx Module as a 1 or 2 zone controller. The physical order of module connections; from upstream to downstream, dictate the MDR direction and product flow logic.

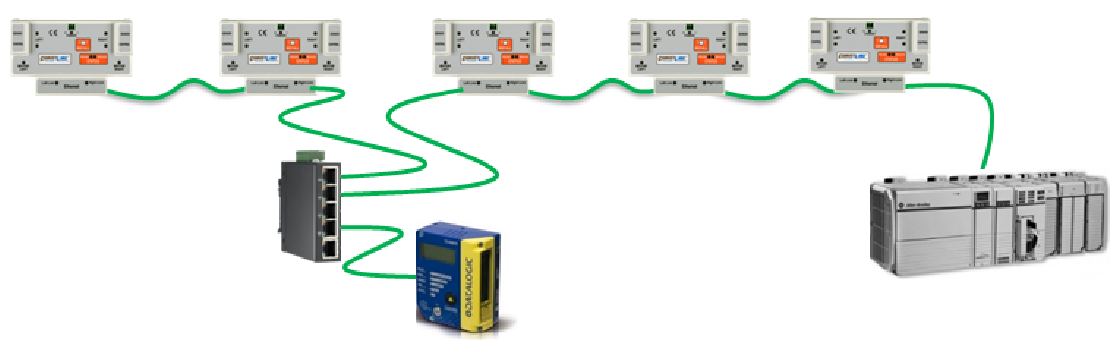
Before You Begin

- All sensors should be clear. *Auto-Configuration* assumes takes for granted that the state of each sensor is the “zone empty” state so that the module can auto-learn the block/ clear states for both retro-reflective and diffuse type sensors
- Make sure you have no more than 221 ConveyLinx Modules connected. *Auto-Configuration* will not configure past 221 ConveyLinx Modules in one network
- Make sure that there are no other Ethernet devices or switches connected in your string of ConveyLinx Modules. After *Auto-Configuration* is complete and successful, you can interrupt the Ethernet string with other devices and/or switches as needed.

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



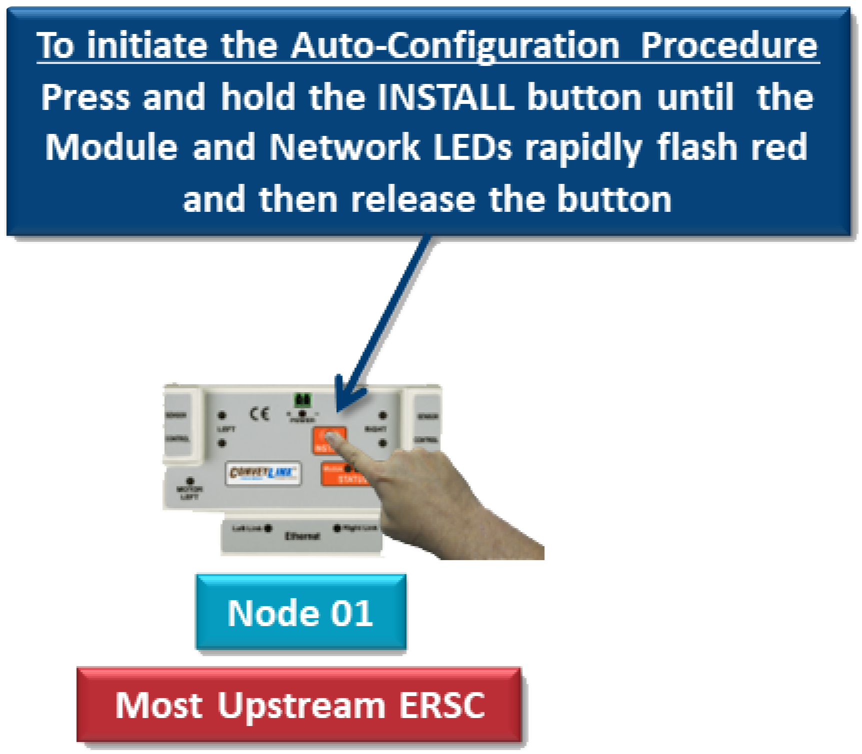
Once Auto-Configuration is complete – you can insert Switches and connect devices as needed



The direction of flow of the conveyor dictates where to begin the *Auto-Configuration Procedure*. The ConveyLinx Module 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 Module will automatically connect to all downstream ConveyLinx Modules and set their I.P. address for communication. Then the routine automatically sets the direction of flow.

Auto-Configuration Procedure



1. Press and hold INSTALL button on the most upstream/in-feed end ConveyLinx Module and keep it held for 5 seconds
2. After 5 seconds the Module Status LED will flash RED
3. Once Module Status LED begins flashing RED, the INSTALL button must be released within 2 seconds. If held for longer than 2 seconds; the procedure is cancelled
4. Once the INSTALL button has been released within the 2 second time window, the ConveyLinx Module will be initiated as most upstream module and the Auto-Configuration Procedure routine will begin.
5. Observe that all of the LED's (with the exception of the Left Link and Right Link LEDs) will flash on and off RED on all the ConveyLinx Modules in the network string

https://www.youtube.com/embed/XDJkC8jpZIM?rel=0

! In order 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.

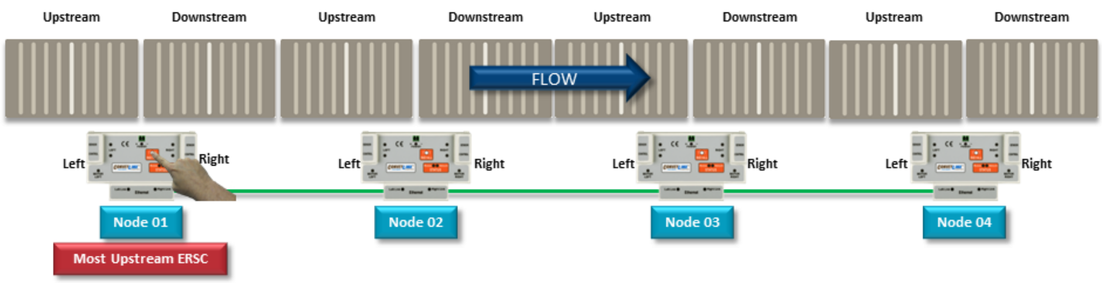
Please note that once a network has been configured; pressing and holding the Install button on any module that has neighboring ConveyLinx Modules on both sides of it will not initiate a new Auto-Configuration Procedure. The module will detect that it is not the most upstream unit and abort the procedure. However, the ConveyLinx Module will perform its local re-booting procedure. This procedure will take a few seconds to complete

! Please note that pressing and holding the Install button on the most **DOWNSTREAM** ConveyLinx Module **WILL** initiate a new Auto-Configuration Procedure and attempt to establish the flow in the opposite direction

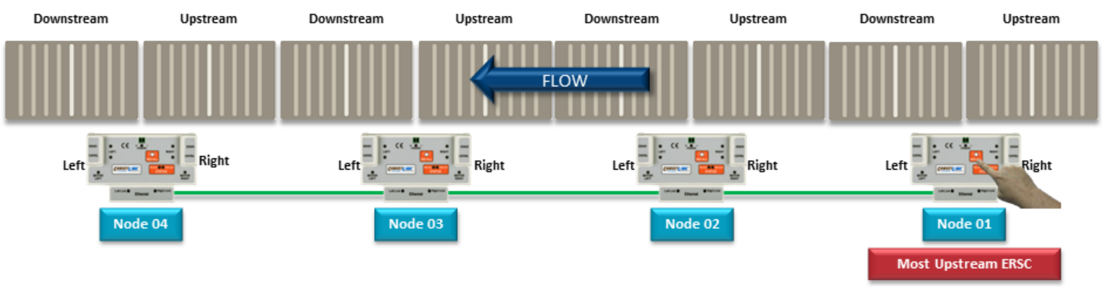
5.3. Auto-Configuration Examples

> 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*

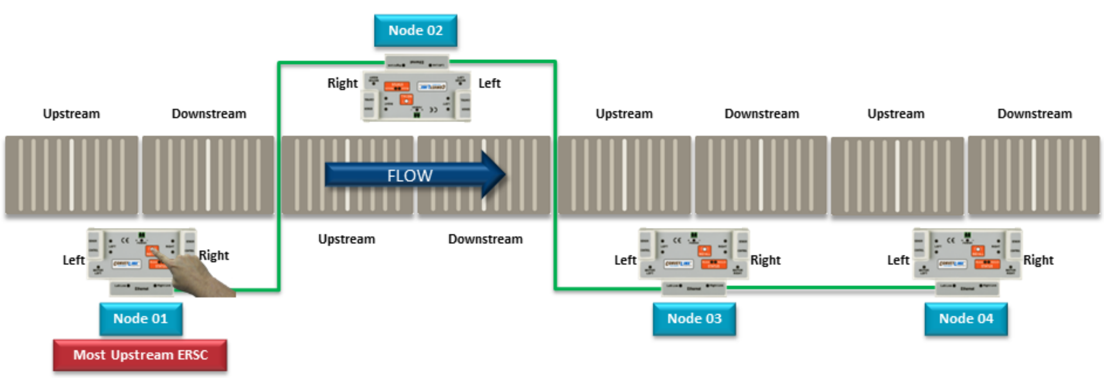
Left to Right Flow Example



Right to Left Flow Example



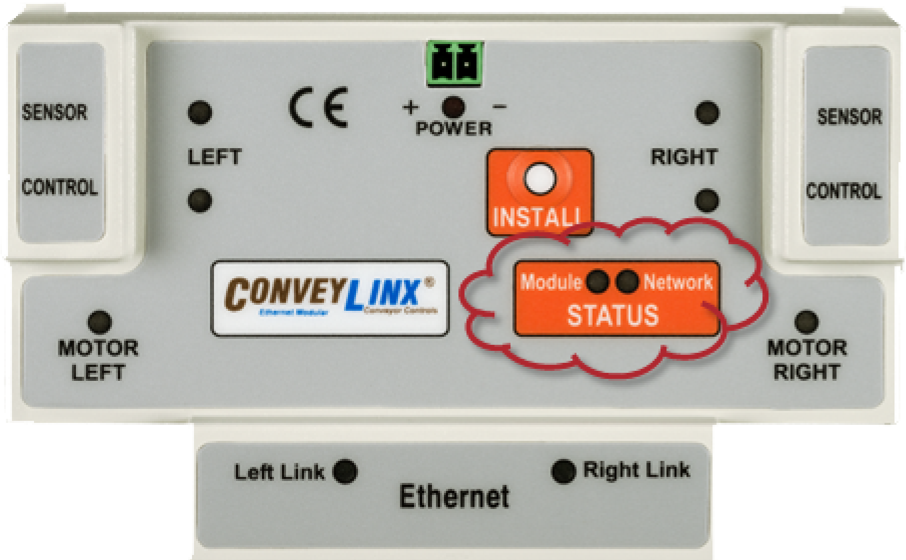
Left to Right Flow with Opposite Side Module Example



5.4. Expected Results

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

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



Please note that the time to complete the *Auto-Configuration Procedure* is dependent on the number of ConveyLinx Modules 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 Module network and perform a *Discover* function. The *Discover* function will display all modules that it finds and from the list your most upstream ConveyLinx Module should have the 4th octet of its IP Address as .20 and you should see each module you configured in the list.

5.5. What to do if things go wrong

Here are some troubleshooting tips:

Unexpected Motor Behavior

- If ConveyLinx Module was previously configured and its motor settings were changed in EasyRoll including rotation direction, these setting are not changed to default when a new Auto-Configuration procedure is performed.
- Make sure you have the correct motor type setting in EasyRoll. How to change Motor Type
- Make sure that there are no motor errors. How to jog motor and look for errors

Network and 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 Module 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 are invalid

Either or both Network and Module Status LEDs are 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 Module up to that point should be configured properly but the remaining modules will not.
- When removing a ConveyLinx Module 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
Hard-Wired Interlocks
Automatic Module Replacement
How to Reset to Factory Defaults

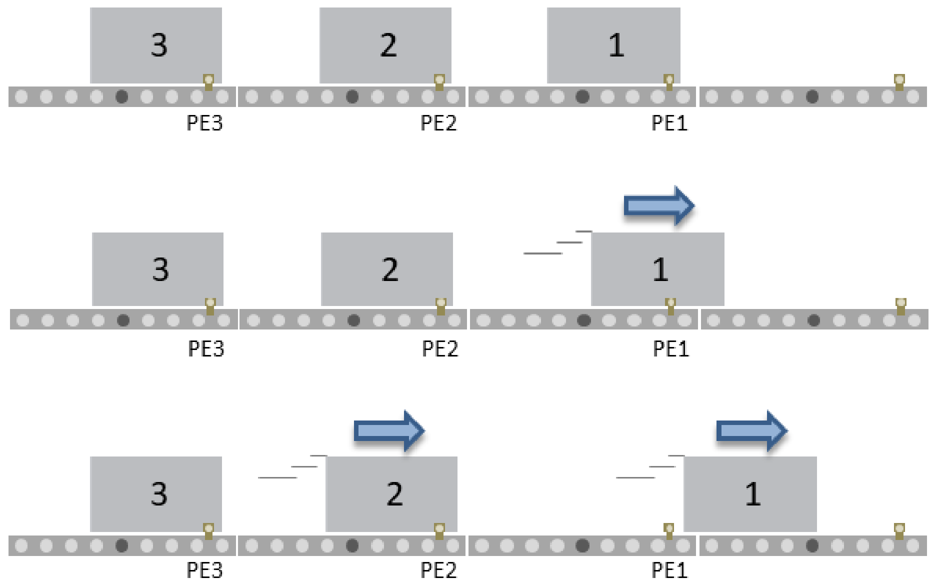
6.1. Singulation Release Mode

Singulation Release is the mode where the ConveyLinx Module 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



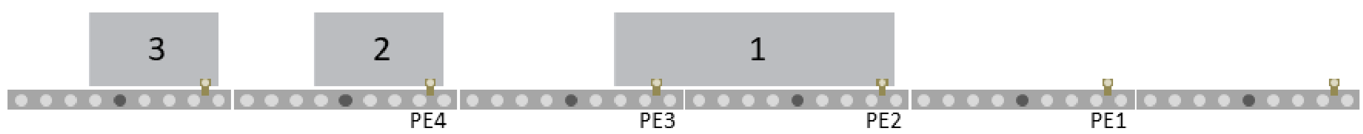
✻ View other ZPA Release Modes and how to change them

6.2. Flex Zone Recognition

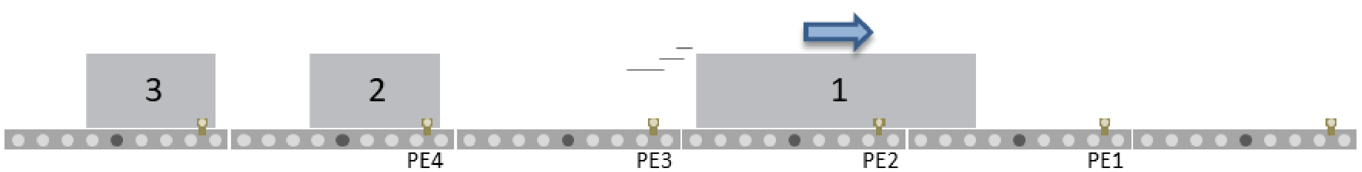
ConveyLinx Modules will automatically detect that a given carton is longer than one zone length and automatically adjust accumulation control so that the longer carton occupies two logical zones and will keep the next upstream carton from conveying into the longer carton.

> ! Flex Zone mode only functions in singulation release mode

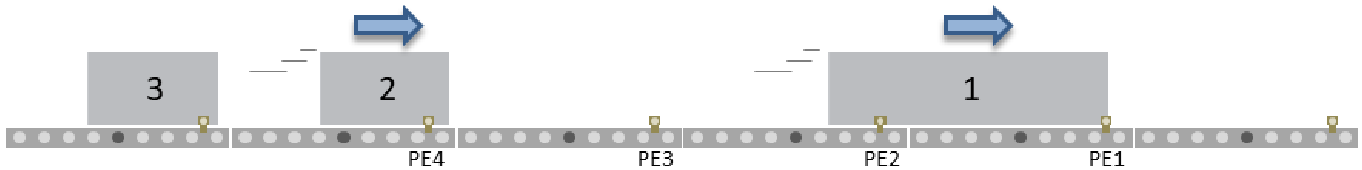
All 3 Cartons are accumulated and the Zone at PE3 has Flex Zone enabled



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 Module:

Sensor Jam

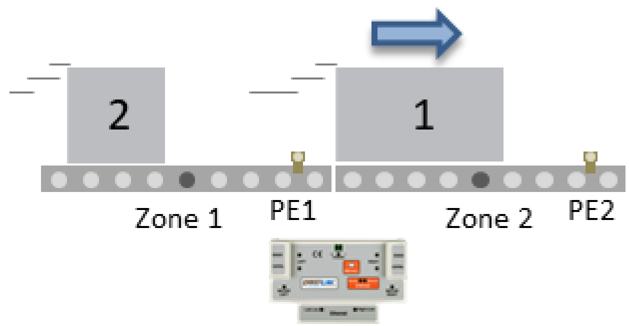
Arrival Jam

> Both of these Jam conditions utilize a single 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

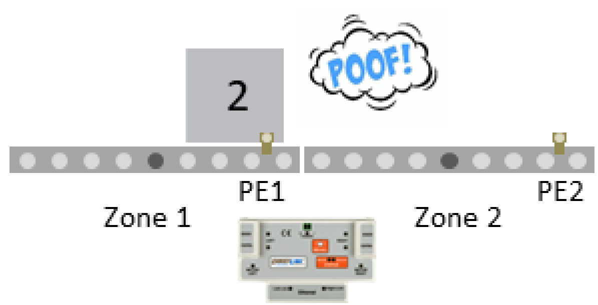
6.3.1. Arrival Jam

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 Module will produce an *Arrival Jam* fault. Once an Arrival Jam occurs, the ConveyLinx Module 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]

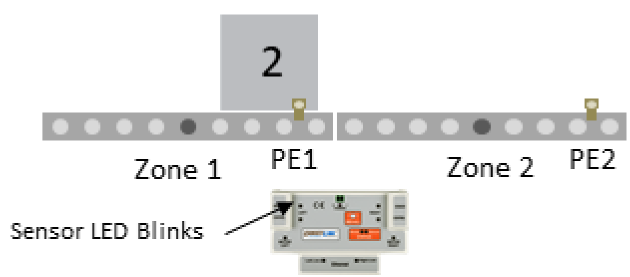
Carton 1 has left PE1 and is on its way to PE2 and because PE1 is clear, Carton 2 is entering Zone 1



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



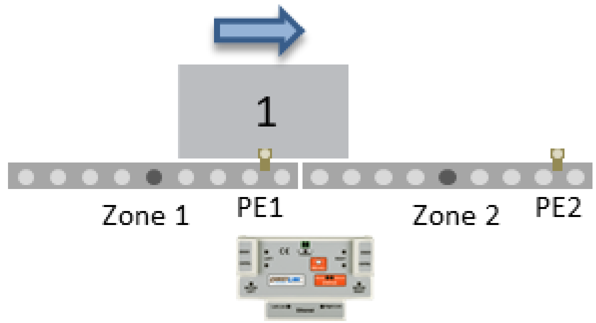
✻ How to Disable Arrival Jam

✻ How to change the Jam and Auto Clear Timers

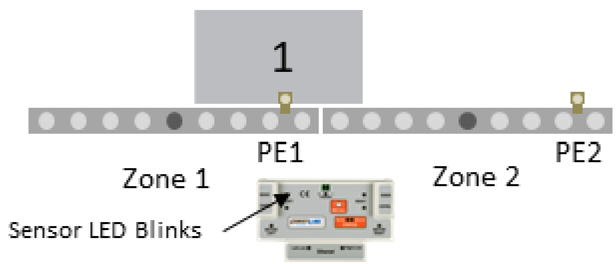
6.3.2. Sensor Jam

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 Module will detect a *Sensor Jam*. This will be indicated on the corresponding Sensor LED on the module as shown here.

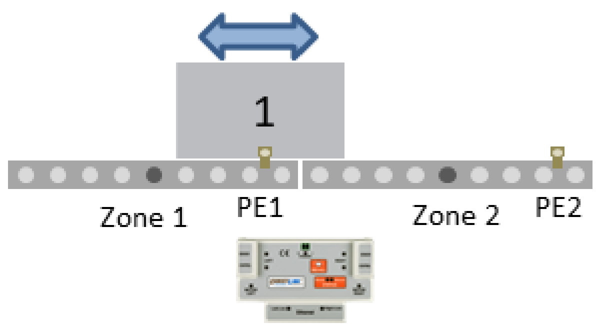
Carton 1 is leaving Zone 1 but has got stuck. Zone 1 and Zone 2 motors are running but PE1 is remaining blocked



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. Hard-Wired Interlocks

Once your linear conveyor has been Auto-Configured and running in ZPA mode; the ConveyLinx Modules will attempt to convey all items from upstream to downstream in the direction of flow. The next logical step that you will want to do is Interlock with the most upstream and/or downstream zones and perhaps interrupt flow in a zone somewhere in the middle.

Control Ports Provide Interlock Function

By default after configuration, each control port provides:

- An input for an Interlock Signal (*Pin 4*)
- An input for Local Zone Accumulate (*Pin 3*)
- An output that indicates if zone is occupied (Pins 1/6 Circuit)

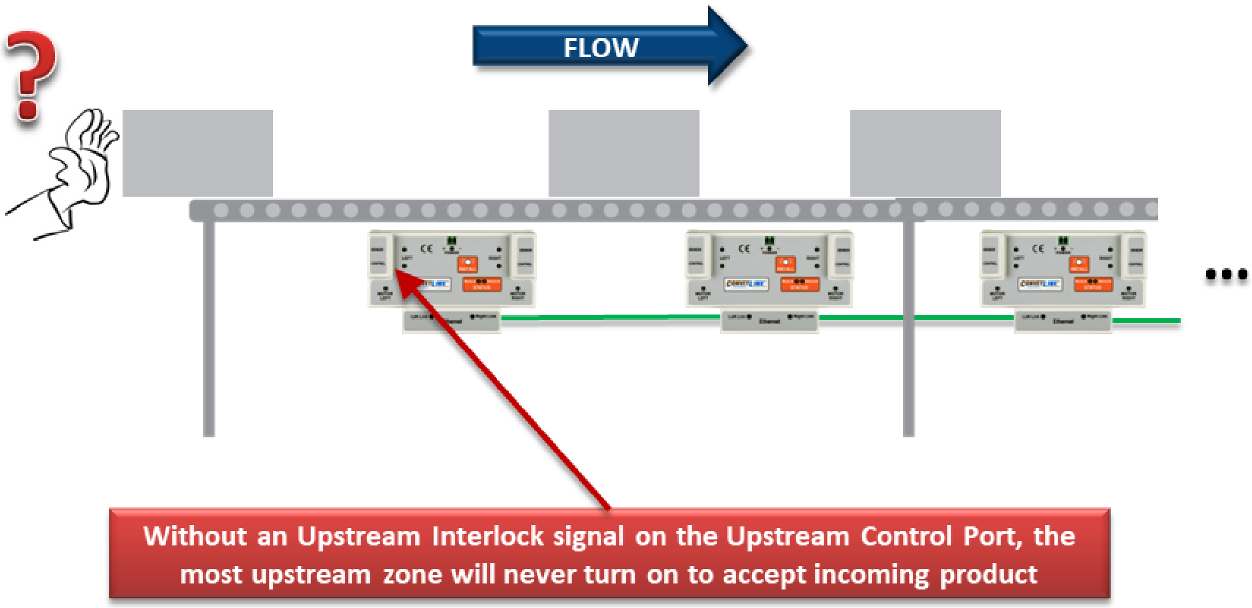
The Interlock Input (*Pin 4*) will act as either a ***Wake Up*** or a ***Lane Full*** depending on whether the Control Port is configured on the Upstream or Downstream side of the module. When each ConveyLinx Module has been Auto-Configured, either its Left or Right side will be configured as the Upstream side and the other will be the Downstream side depending on the flow established during Auto-Configuration.

Wake Up interlock should only be used on the **MOST Upstream** zone and the *Lane Full* interlock should only be used on then **MOST Downstream** zone of any given subnet.

Learn About:

Upstream Interlock (Wake Up)
Downstream Interlock (Lane Full)
Local Zone Accumulate
Interlocks with a Single Zone Module
Interlock Input versus Local Accumulate Input
Using a Photoeye for Wake Up
Using a Photoeye for Lane Full
SE-4 Breakout Module

6.4.1. Upstream Interlock (Wake Up)



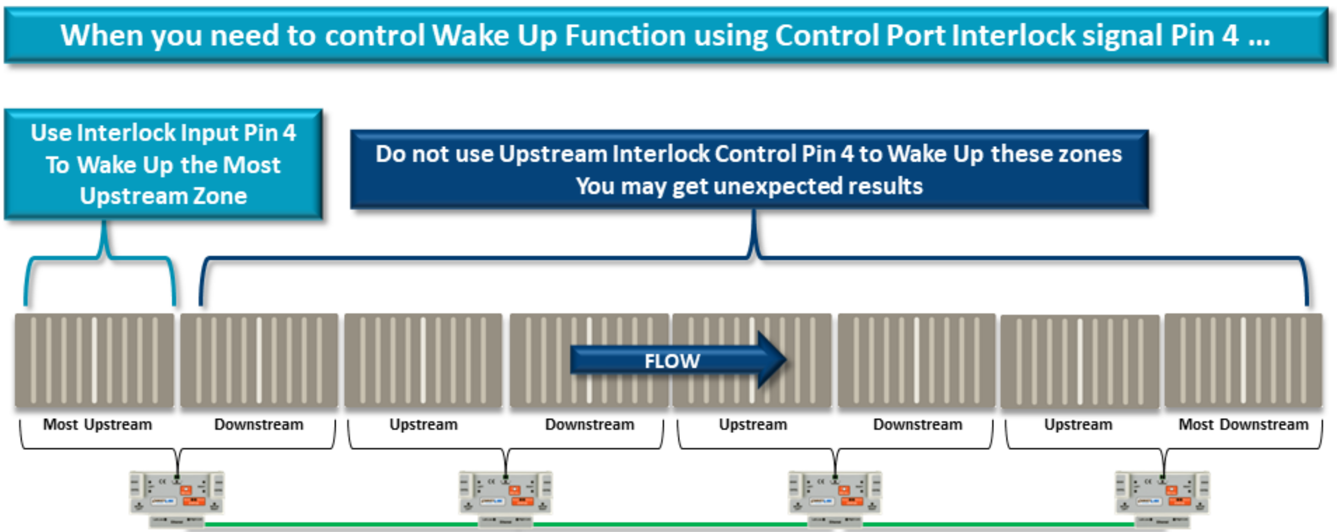
Interlock with the most upstream zone to get an input to see if is clear and then turn on an output (*Wake Up*) to tell it to run to accept a new item

Whichever Control Port (Left or Right) is on the *Upstream Side* of the module will accept a *Wake Up* signal input on its interlock input *Pin 4*. If the zone is unoccupied, when a digital signal is applied to *Pin 4*, the zone will run its motor to accept an item.

> **!** For an Upstream Side Control Port the motor will continue to run as long as the signal is being applied to *Pin 4* and the zone sensor remains unblocked. If the signal is removed, the zone will continue to run for Jam Timer setting (5 seconds by default) and then stop

When NOT to use Pin 4 as a *Wake Up*

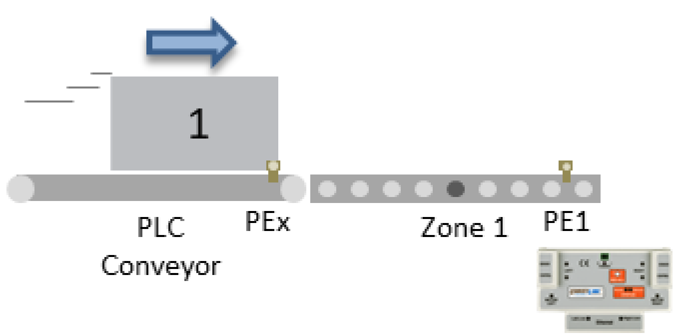
Only use the Interlock *Pin 4* signal for *Wake Up* on the most upstream zone of a network. Do not use the *Pin 4* signal on any of the other zones on modules with an upstream connection to another ConveyLinx Module – You may get unexpected results because signal may or may not over-ride the status coming from upstream over the network.



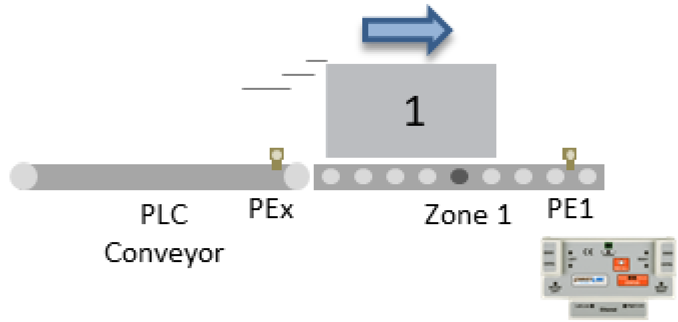
Wake Up Interlock with Handshake

The external controls providing the *Wake Up* signal to *Pin 4* will sometimes also require an output signal from the ConveyLinx Module that will indicate when this upstream zone is occupied in order to implement a handshake. This zone occupied signal is provided by the Control Port's output signal.

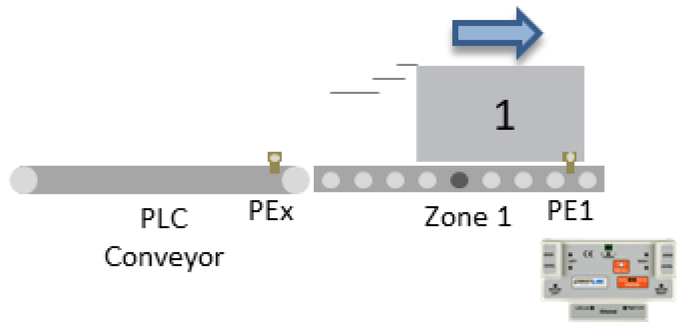
Upstream Control Port Output is OFF and PLC signal to Upstream Control Port Pin 4 is OFF. When Carton 1 reaches PLC connected PEx, the PLC energizes its output signal to Upstream Control Port Pin 4 and Zone 1 Motor starts to run



When Carton 1 clears PLC connected PEx, the PLC turns off its output signal to Upstream Control Port Pin 4



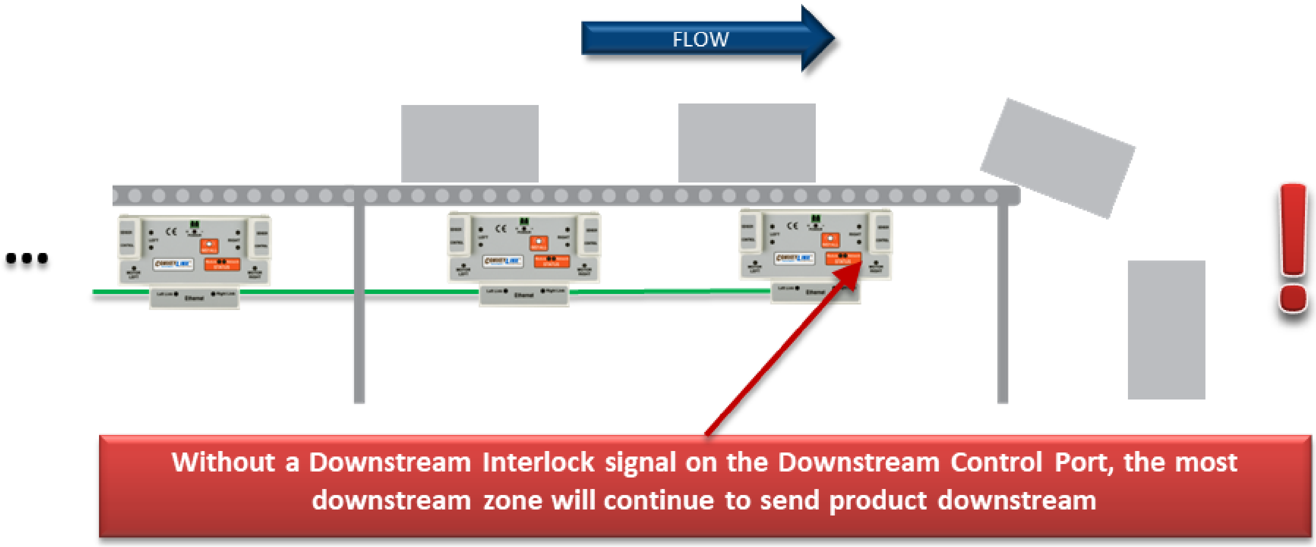
When Carton arrives at Zone 1 PE1, the Upstream Control Port Output is ON to indicate that Zone 1 is occupied. The PLC can connect this signal to a digital input to use in its logic to stop its conveyor if another carton arrives at PEx



[How to hook up PLC I/O to Control Port]

[How to Wake Up the Upstream zone with a single sensor]

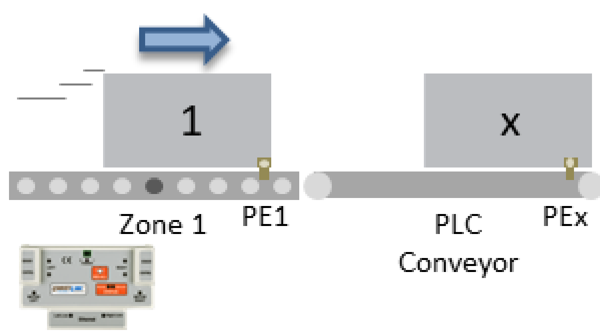
6.4.2. Downstream Interlock (Lane Full)



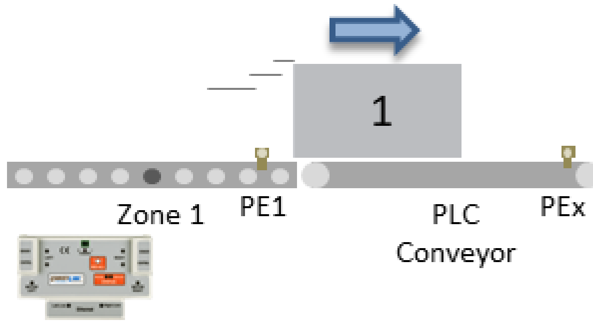
Interlock with the most downstream zone to turn on an output tell it to stop when something shows up (*Lane Full*) and then get an input to see if the last downstream zone is occupied

On a newly Auto-Configured linear conveyor without any intervention from external signals, if an item is on the conveyor, it will convey to the last downstream zone and try to continue on. Whichever Control Port (Left or Right) is on the *Downstream Side* of the ConveyLinx Module will accept a *Lane Full* signal input on its interlock input *Pin 4*. When a digital signal is applied to *Pin 4*, the zone will still run its motor to accept an item but it will stop and accumulate when the item reaches the sensor.

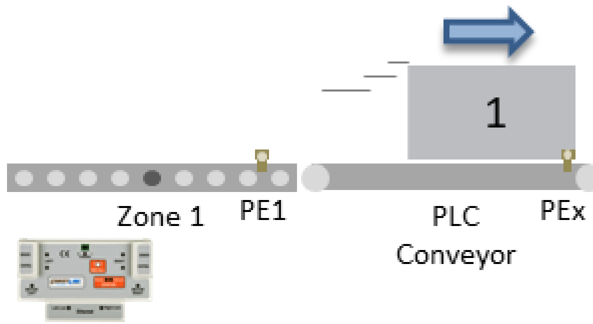
PLC output signal to Downstream Control Port *Pin 4* is ON because PEx is blocked. Because *Pin 4* is ON, when Carton 1 arrives at PE1 it will stop and accumulate. Downstream Control Port *Output* is ON when Carton 1 is blocking PE1



When PLC output signal to *Pin 4* is OFF, Zone 1 motor runs and Carton 1 discharges. Module is waiting for *Pin 4* to turn ON in order to avoid an Arrival Jam. When PE1 is clear, Downstream Control Port *Output* is OFF



When Carton 1 arrives at PEx, the PLC turns ON its output signal to *Pin 4* indicating to the module that Carton 1 arrived so Zone 1 will not generate an Arrival Jam. If another carton conveys into Zone 1, it will accumulate as long as *Pin 4* signal is ON



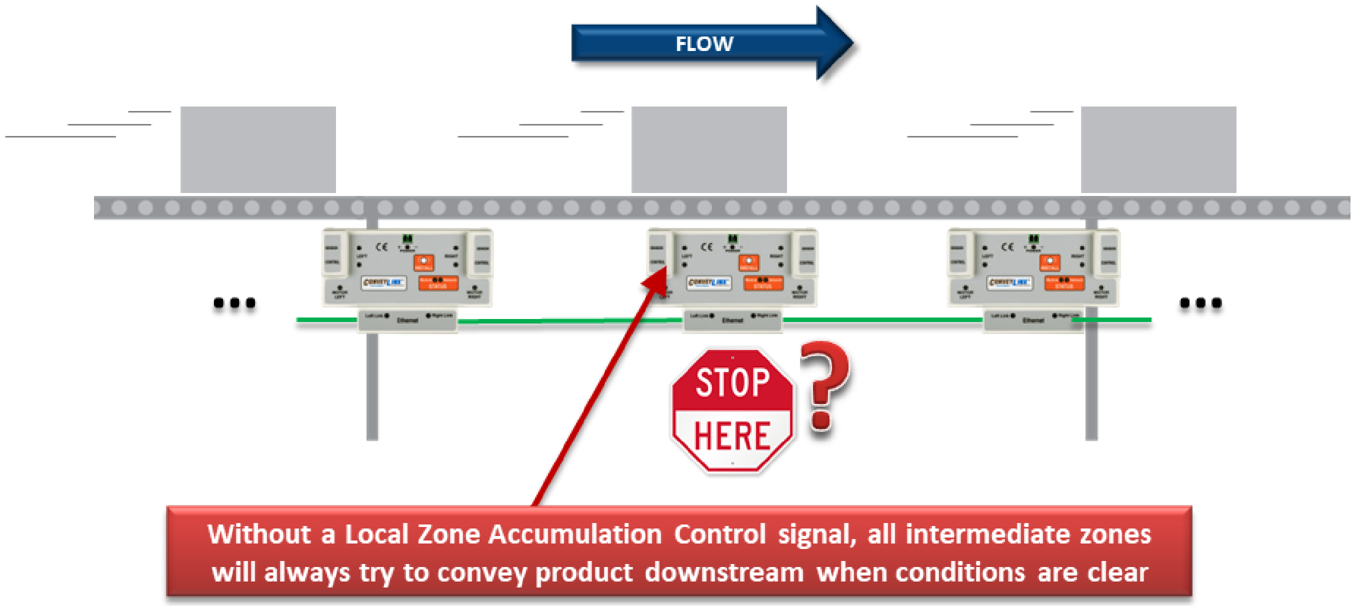
When *Pin 4* is used on a Downstream Control Port the cycling off and on of this signal generates an “Arrival” in the module logic and thus prevents an Arrival Jam

How to hook up PLC I/O to Control Port

How to add Lane Full to the most Downstream zone with a single sensor

6.4.3. Local Zone Accumulate

On a newly Auto-Configured linear conveyor without any intervention from external signals, items will not stop on any zones in the middle unless the next downstream zone is occupied.



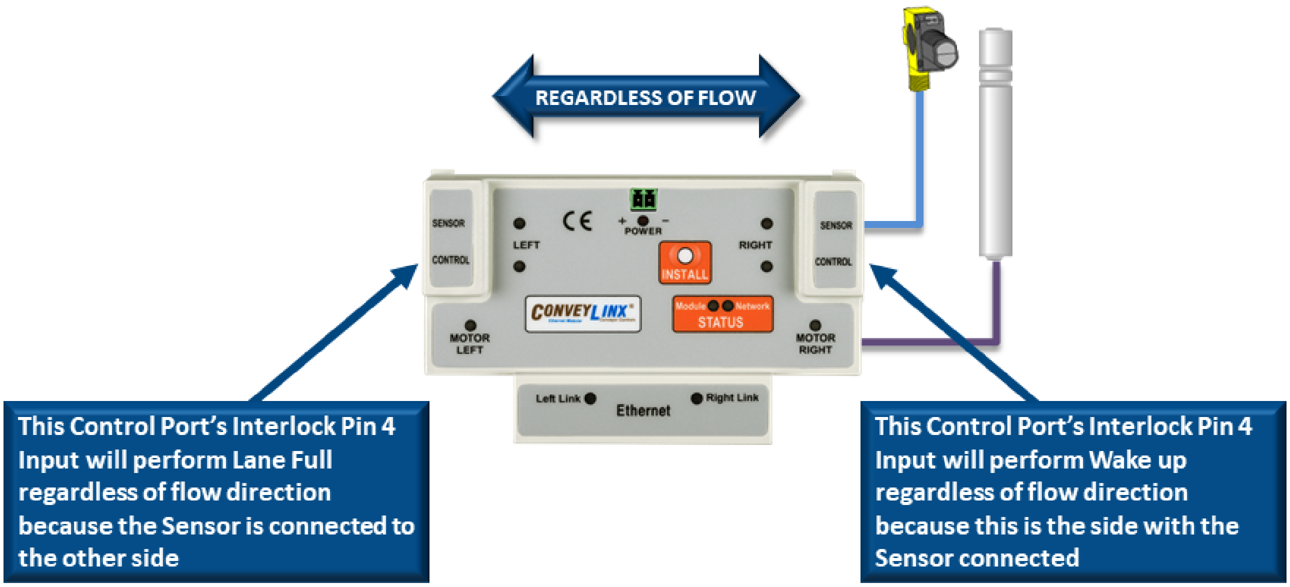
To cause either the upstream or downstream zone on a ConveyLinx Module to accumulate an item when it arrives in a zone, you energize the *Local Zone Accumulate* signal on *Pin 3* of the Control Port for the zone.

> ! If you re-energize the *Pin 3* signal while the item is in transit and still blocking the zone sensor, the item will stop and if left like this, the zone will eventually generate an Arrival Jam. You can modify the function of the *Pin 3* operation by making it active only upon leading edge of the zone sensor

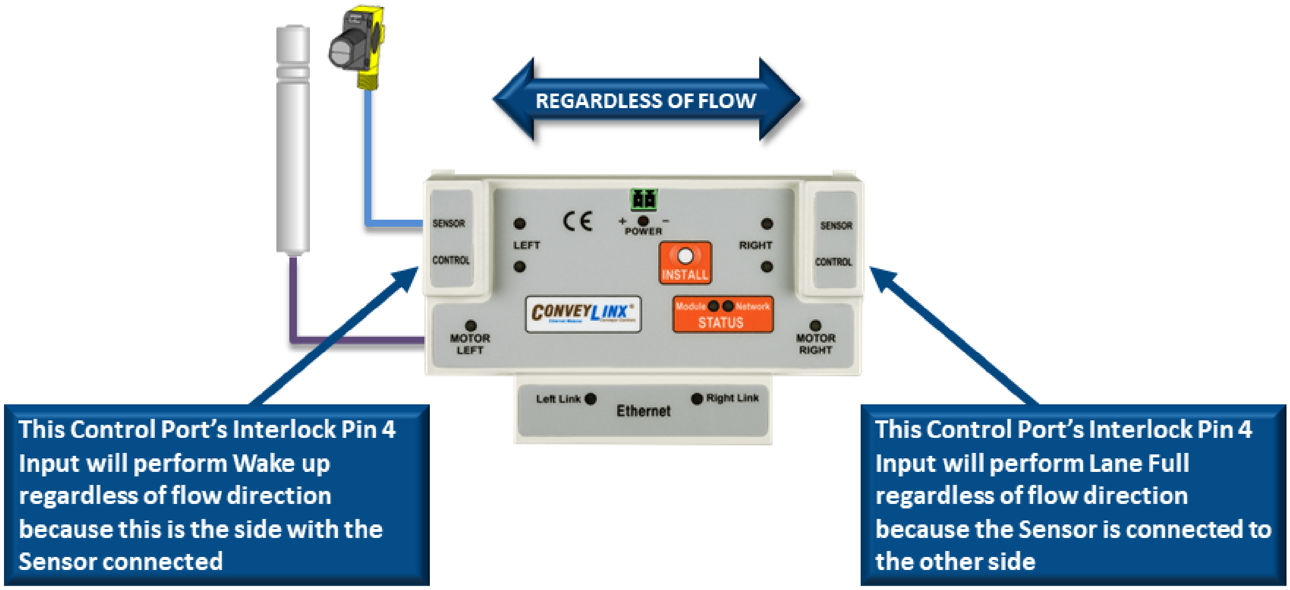
6.4.4. Interlocks with Single Zone Module

When you have a *single zone ConveyLinx Module* (one sensor and one or two motors); whichever side of the ConveyLinx Module (Left or Right) that has the *Sensor* attached will always be the Upstream Zone of the ConveyLinx Module and that same side's Control Port *Pin 4* will provide the *Wake Up* function.

Single Zone ConveyLinx Module with Sensor on RIGHT Side



Single Zone ConveyLinx Module with Sensor on LEFT Side

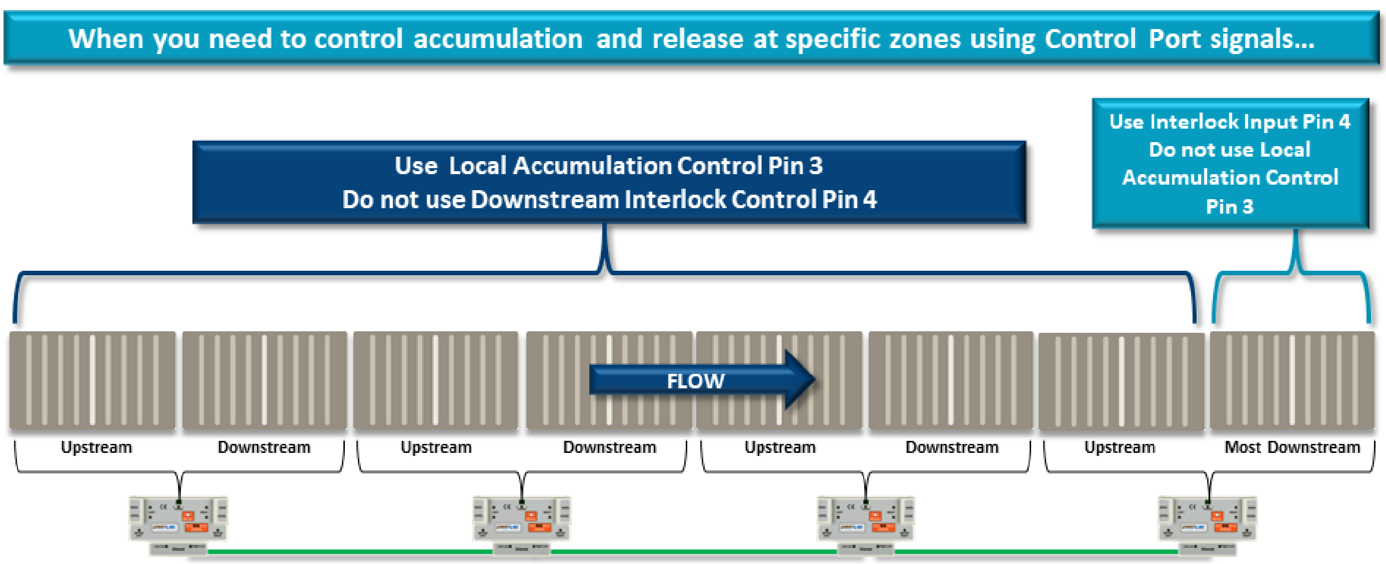

REGARDLESS OF FLOW
SENSOR
CONTROL
LEFT
POWER
RIGHT
INSTALL
CONVEYLINX
Module
Network
STATUS
MOTOR LEFT
MOTOR RIGHT
Left Link
Ethernet
Right Link
This Control Port's Interlock Pin 4 Input will perform Wake up regardless of flow direction because this is the side with the Sensor connected
This Control Port's Interlock Pin 4 Input will perform Lane Full regardless of flow direction because the Sensor is connected to the other side

6.4.5. Interlock Input versus Local Accumulate Input

By default, energizing the Interlock Signal *Pin 4* on a Control Port on the Downstream side of a ConveyLinx Module will cause that zone to accumulate an item when it arrives on the zone sensor. You get the same result if you energize the Local Accumulate Signal on *Pin 3*. So if all you want to do is accumulate a zone, which signal should you use and does it make any difference? The answer is yes it can make a difference

> To accumulate the most downstream zone, use the Interlock Input Pin 4

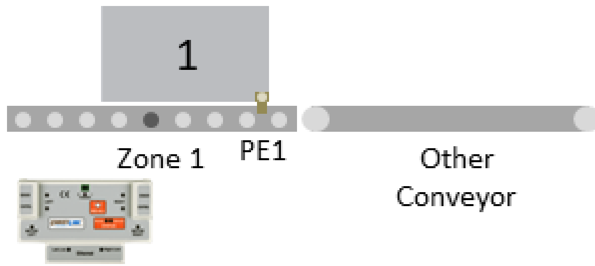
> To accumulate any other zone, use the Local Accumulate Input Pin 3



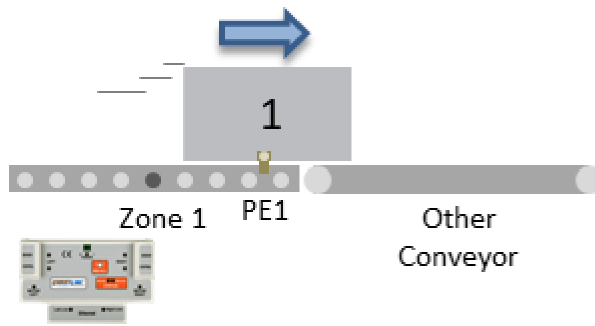
What happens if you use Pin 3 when you should use Pin 4?

Suppose you want to control the release of multiple items by toggling the *Pin 3* signal. After you release the first one, you will get an Arrival Jam because the module is looking for the *Pin 4* signal to be energized in order to "confirm the downstream arrival" of the item. So, by using *Pin 3* in this case, you will have to wait the Jam Time plus the Auto Clear Time between release of each item.

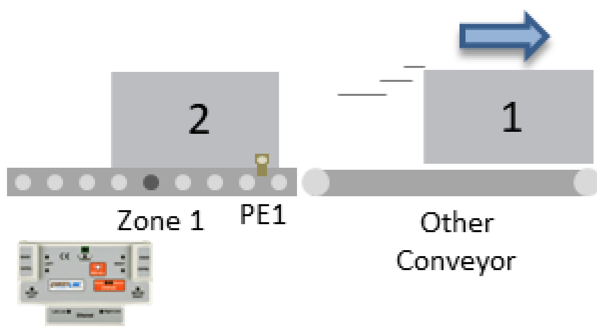
Downstream Control Port *Pin 3* is ON and Carton 1 accumulates when it arrives



When *Pin 3* is turned OFF, Zone 1 will run to convey Carton 1 on downstream

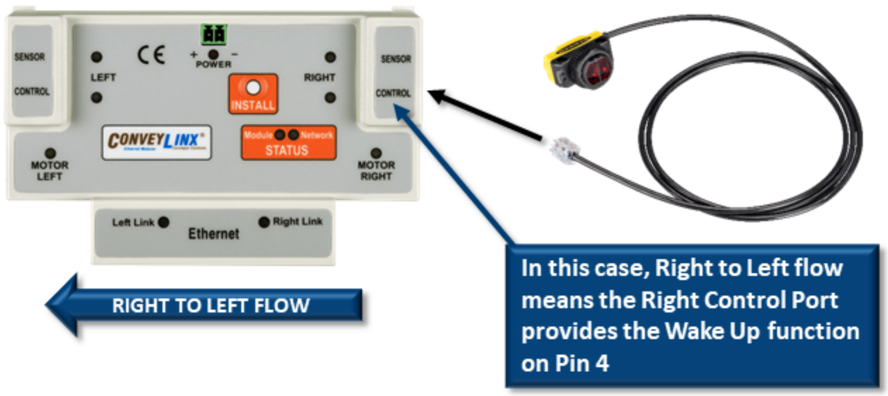


When Carton 1 clears PE1, *Pin 3* signal is turned back ON. Because *Pin 3* is ON, Carton 2 will stop upon arrival at PE1 *however because there is no signal to Pin 4 to indicate arrival downstream, <u>Zone 1 will generate an Arrival Jam !!!</u>*

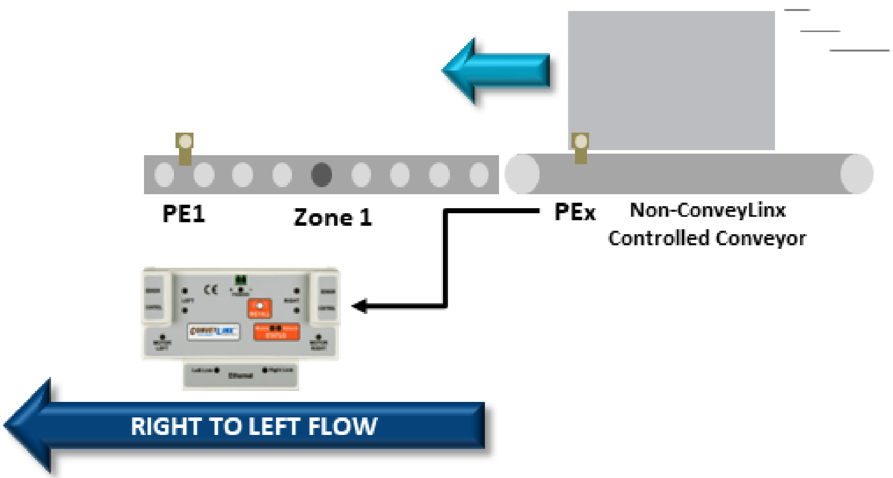


6.4.6. Using a Photoeye for Wake-Up

Because standard RJ-11/12 Sensors put their blocked/ clear output on Pin 4; this type of Sensor can be plugged into the Upstream Control Port and provide the Pin 4 Wake Up signal



When PEx is plugged into the Right Control Port and its output is ON, Zone 1 will Wake Up if it is unoccupied. If Zone 1 is occupied, Zone 1 will not wake up and run



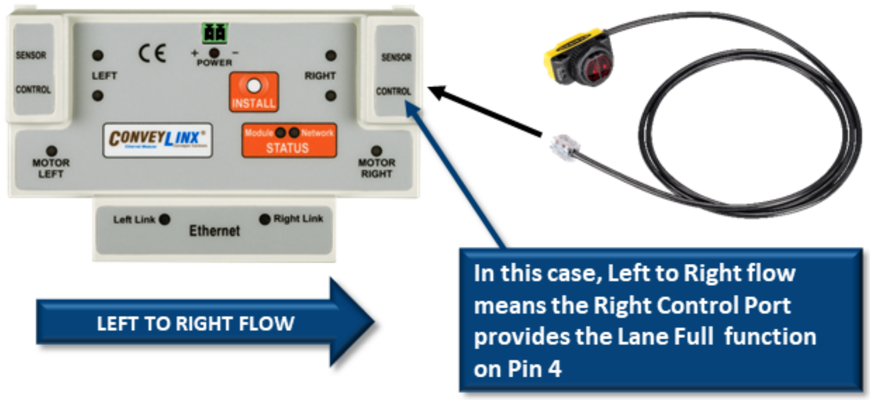
✻ If you do not have a Sensor with RJ-11/12 connector; you can use an SE-4 module to wire you sensor to the Control Port

✻ By default, your Sensor has to energize its output when it is blocked. You can change *Pin 4* to accommodate a Sensor that de-energizes its output when it is blocked

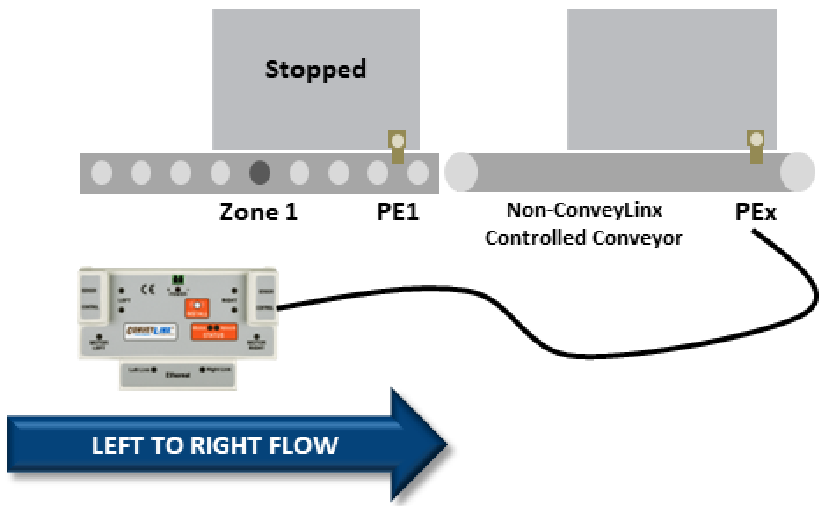
! If your sensor has an Error or Health Output signal, you need to cut the signal wire for this on the RJ-12 connector in order to keep this signal from inadvertently energizing Pin 3

6.4.7. Using a Photoeye for Lane Full

Because standard RJ-11/12 Sensors put their blocked/ clear output on Pin 4; this type of Sensor can be plugged into the Downstream Control Port and provide the Pin 4 Lane Full signal.



When PEx is plugged into the Right Control Port and its output is ON, Zone 1 will see a Lane Full condition and stop any item that arrives at PE1. If PEx output is OFF, Zone 1 will continue to run and convey item downstream



* If you do not have a Sensor with RJ-11/12 connector; you can use an SE module to wire you sensor to the Control Port

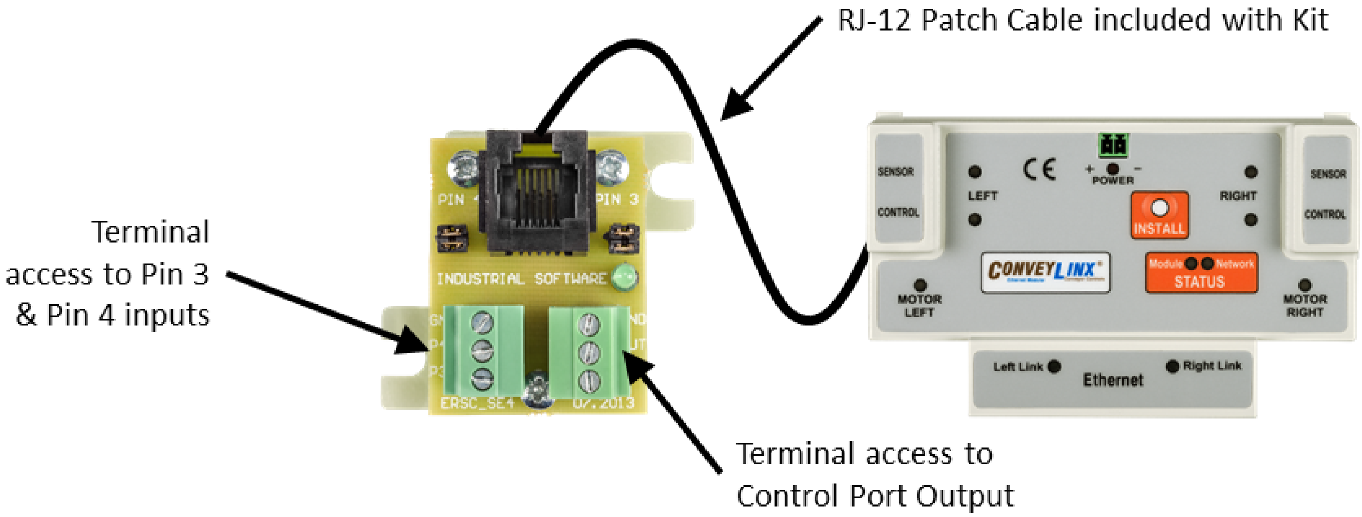
* Block and Clear Timer function for the Lane Full Sensor

* By default, your Sensor has to energize its output when it is blocked to make *Lane Full* operate. You can change *Pin 4* to accommodate a Sensor that de-energizes its output when it is blocked

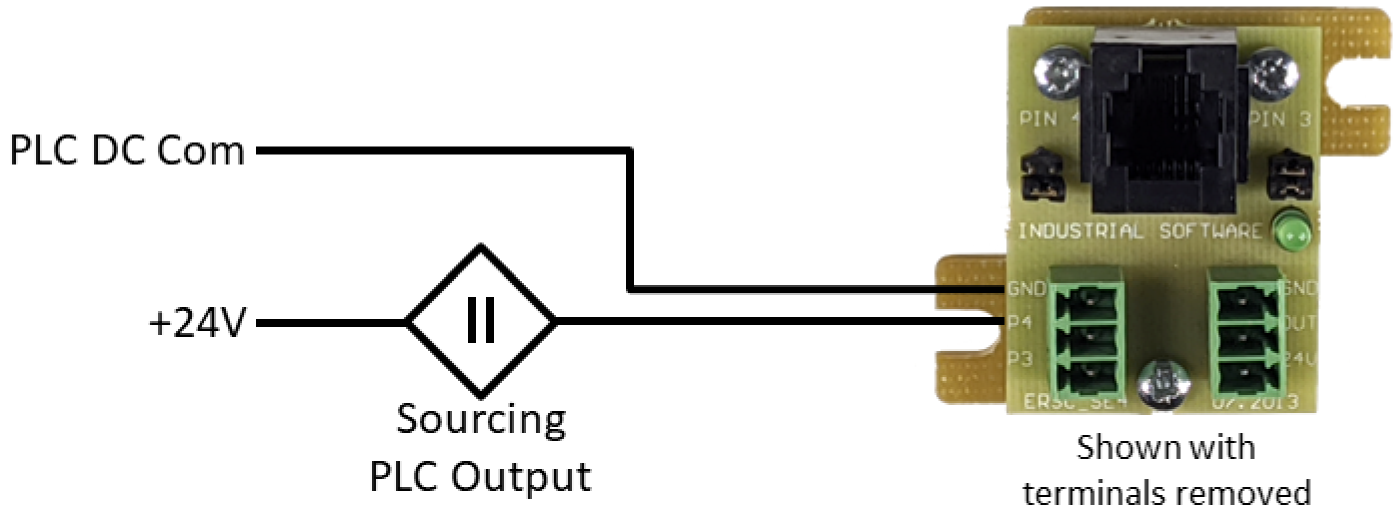
! If your sensor has an Error or Health Output signal, you need to cut the signal wire for this on the RJ-12 connector in order to keep this signal from inadvertently energizing Pin 3

6.4.8. SE Breakout Module

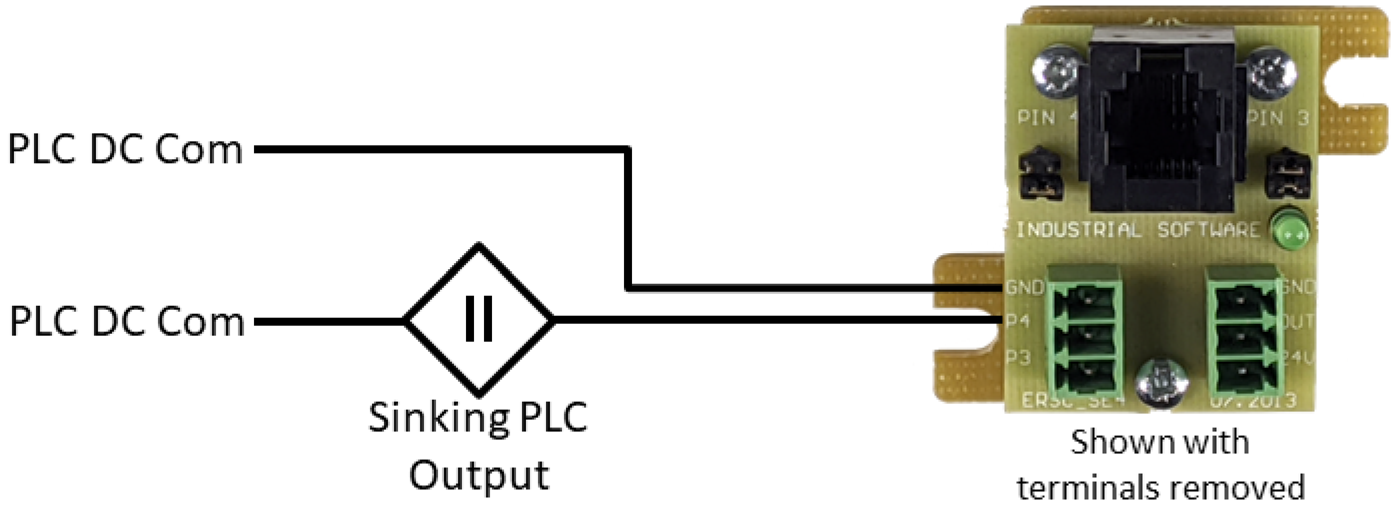
The SE breakout module provides convenient wiring terminals for access to *Pin 3* and *Pin 4* inputs as well as the Control Port's output. SE Inputs (Pin 3 and Pin 4) are PNP-NPN auto-sensing so you can use sourcing or sinking PLC output modules. The terminals *P3* and *P4* on the SE-4 correspond to *Pin 3* and *Pin 4* respectively



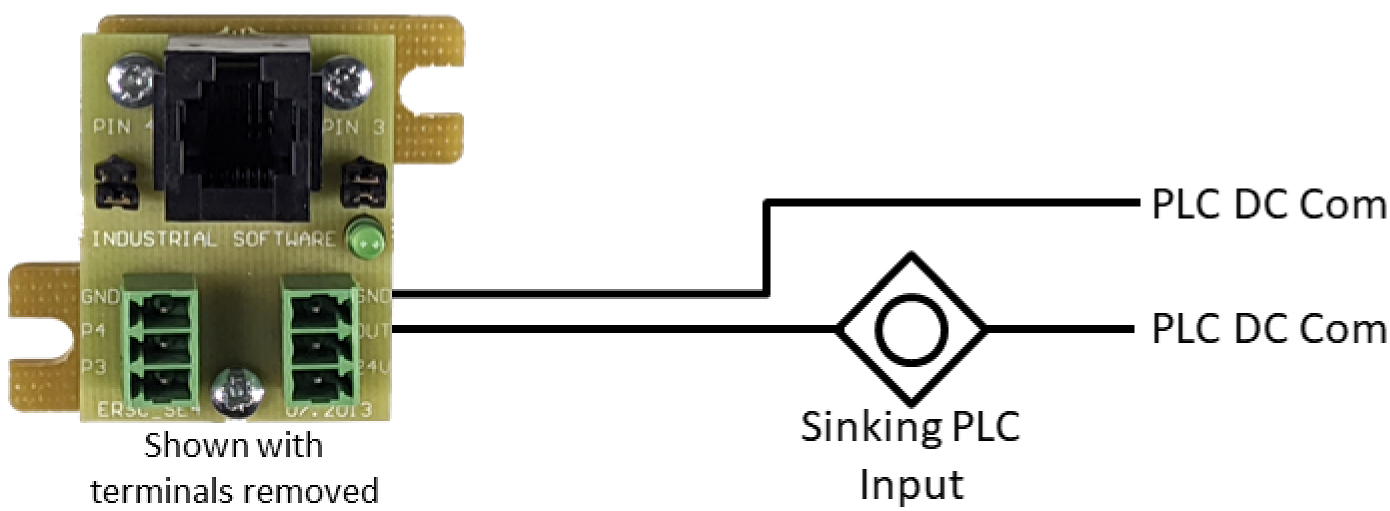
PLC Sourcing Output to P3 or P4



PLC Sinking Output to P3 or P4

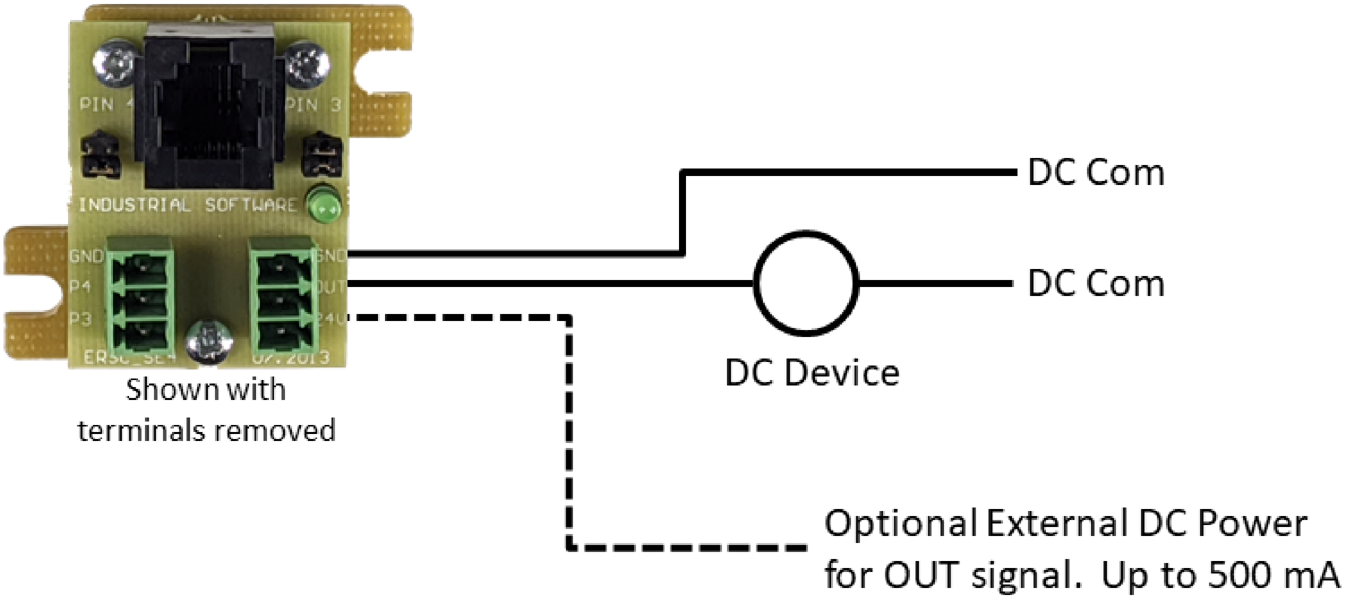


SE OUT to PLC Sinking Input



! Please note that the Control Port output (*OUT*) signal on the SE is always PNP and is subject to the power limitations of the module's internal fusing

SE OUT with External Power



6.5. Automatic Module Replacement

Once a linear conveyor has been commissioned by *Auto-Configuration Procedure*, the ConveyLinx Module 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 Module 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 Module.

! For Automatic Module Replacement to work:

- 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

Module Auto Replacement Procedure

1. Disconnect existing module’s motor(s), network, photo-sensor(s), hardware, and power connections. The order of disconnection does not matter.
2. Connect new module’s motor(s), sensor(s), hardware, and network connections only. DO NOT CONNECT POWER YET.
3. Press and hold INSTALL button. While holding the INSTALL button pressed; connect module power.
4. Observe the Module Status LED. About 1 second after applying power; the Module Status LED will start blinking red. DO NOT RELEASE THE INSTALL BUTTON YET. After another second, you will see that the Module Status LED will start to blink both RED and GREEN. This is the indication that the Auto Replace Procedure is starting and you MUST release the Install Button NOW. Releasing the Install Button before you see the Module Status LED flashing both red and green will result in cancelling the Auto Replace Procedure.
5. To tell if the *Auto Replace Procedure* has been properly initiated, the replaced module will turn its Sensor and Control Port LEDs on solid red. Also, all modules on the network will briefly flash their sensor and control port LEDs. This is normal and they will return to normal operation within a few seconds.

6. Wait for at least 3 minutes before using the system or cycling power or plugging or unplugging any Ethernet connections. The module being replaced will automatically upgrade or downgrade its firmware if needed and restore its configuration settings. This will involve multiple automatic restarts of the module. Do NOT cycle power to the module during this 3-minute period.

7. When both the Module Status and Network LED’s are blinking green, then the module replacement has been a success.

! Once the Auto Replacement procedure has started; if power is lost on the replacing module, it may become inoperable. In this case, you may need to reset it back to factory default settings before attempting to use it as a replacement module. If it does not return to factory default settings, you may need to return it to PULSEROLLER to be repaired. How to reset a module back to factory default settings

6.6. Reset to Factory Default Settings

There may be instances when you want to return a ConveyLinx Module 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 Install Button until the Module LED begins to flash
3. When the Module Status and Network Status LEDs both blink green, the procedure is complete

! If the module you wish to reset was installed in a subnet that was subsequently Locked from EasyRoll; then this module must first be Unlocked using EasyRoll before you can perform the procedure to reset to factory default setting.

How to unlock a module with EasyRoll

Factory Default Settings

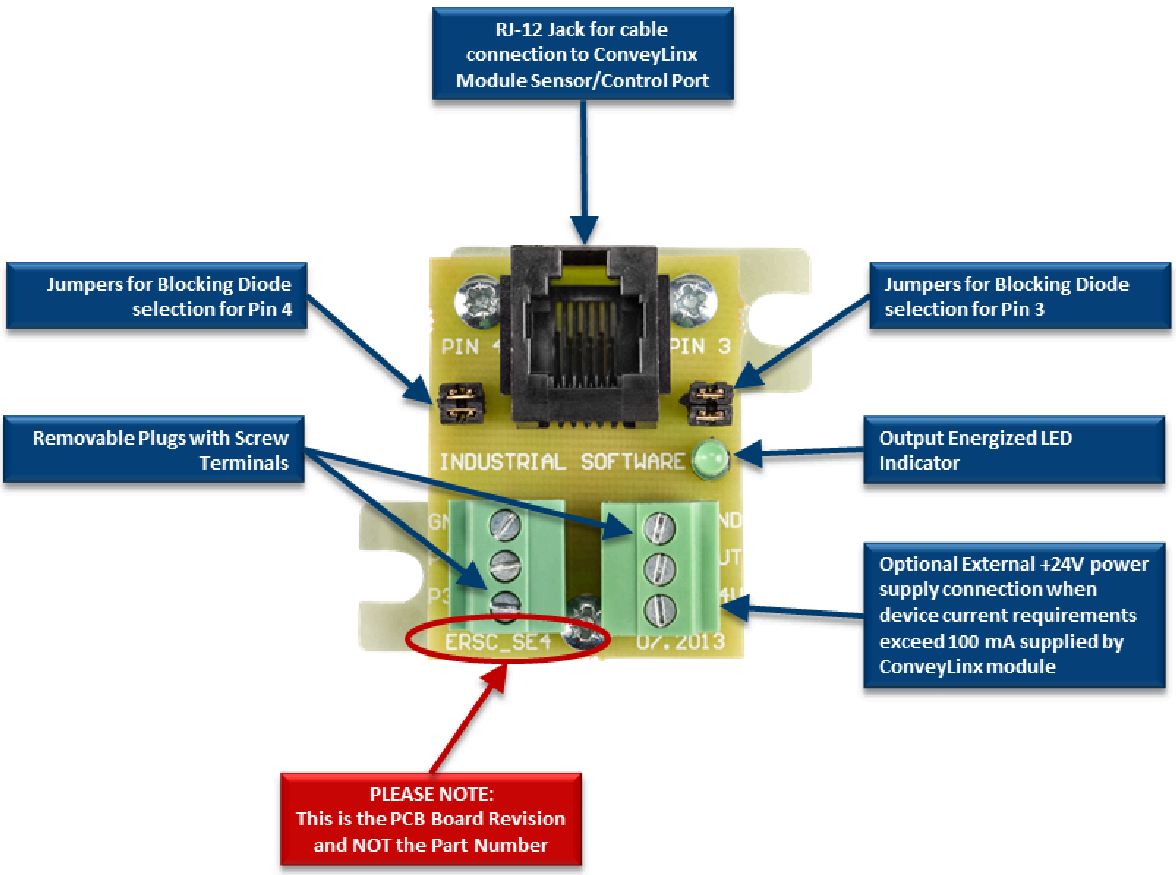
Item	Value or Setting
IP Address Settings	IP: 192.168.202.20
	Subnet Mask: 255.255.128.0
	Default Gateway: 192.168.202.1
Left & Right Motor Settings	Senergy ECO
	Normal Braking
	Closed Loop
	100% Speed
	CW Direction
	100 Pulses Acceleration
	50 Pulses Deceleration
Module Mode	ZPA – 2 Zone – Left to Right Flow
	Singulation Release Mode
	All Settings Options Unchecked

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
Control Ports	All Options Unchecked
Sensors	Sensors are “ON is Blocked”
	Sensor Health are “ON is Error”
Connections	All are Cleared

7. SE Breakout Module

The *SE Breakout Module* is used to provide convenient screw terminal wiring access to the RJ-12 ports (Sensor and Control) on the ConveyLinx Module. The *SE Breakout Module* utilizes an RJ-12 to RJ-12 interface cable to connect to the ConveyLinx Module.

This module is typically sold as the *SE Breakout Module Kit* (Order Code 8087-0017). This kit includes the modules and an 8 inch RJ-12 cable to connect the SE to the ConveyLinx Module



Additional Functionality

Blocking Diodes

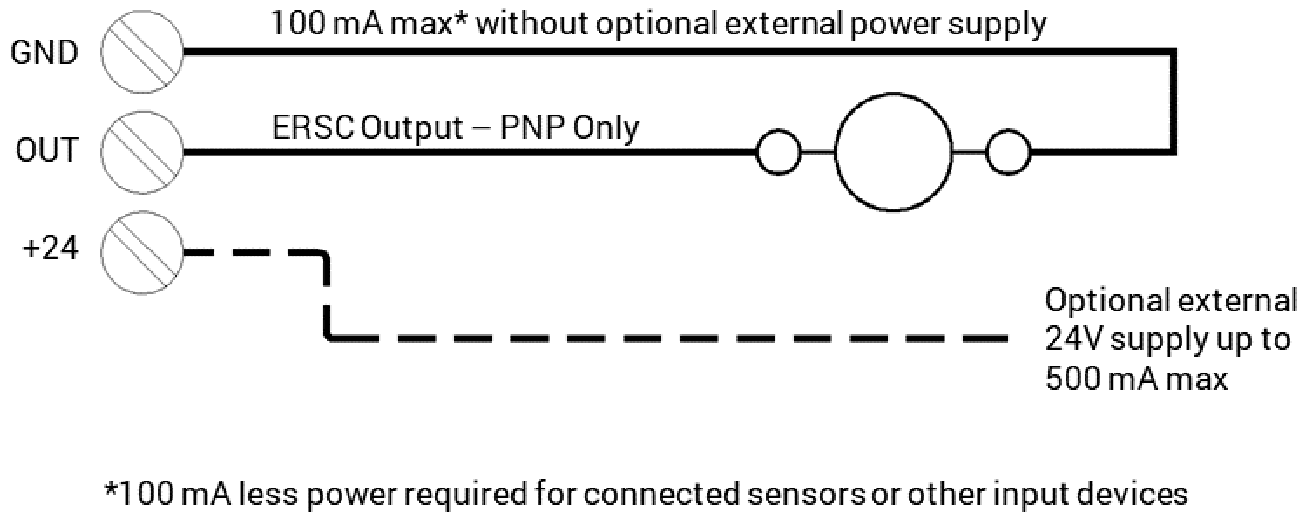
In addition to providing simple wiring connectivity to the ConveyLinx Module RJ-12 pins, the SE

Breakout Module also provides blocking diodes for both the Local Zone Accumulate input signal (Pin 3) and the Interlock input signal (Pin 4). The SE module’s diodes are equipped with user settable jumpers to allow configurations for both sourcing and sinking solid state PLC output modules.

Output Amplifier with Optional External Power Source

The SE Breakout Module also includes an amplifier for the Control Port’s output circuit (Pin 1 and Pin 6) to allow up to 100mA of current load. This feature will allow the Control Port output to drive a small inductive load such as a relay coil as well as a PLC input. Because of current restrictions on each pair of Sensor and Control Ports, you can optionally connect an external +24V power source to the “+24V” terminal on the SE Breakout Module to provide power to connected devices. The SE Module’s output can accommodate up to 500 mA with this external power. The SE Module also contains circuitry to prevent a connected external power source from back-feeding into the ConveyLinx Module’s power bus.

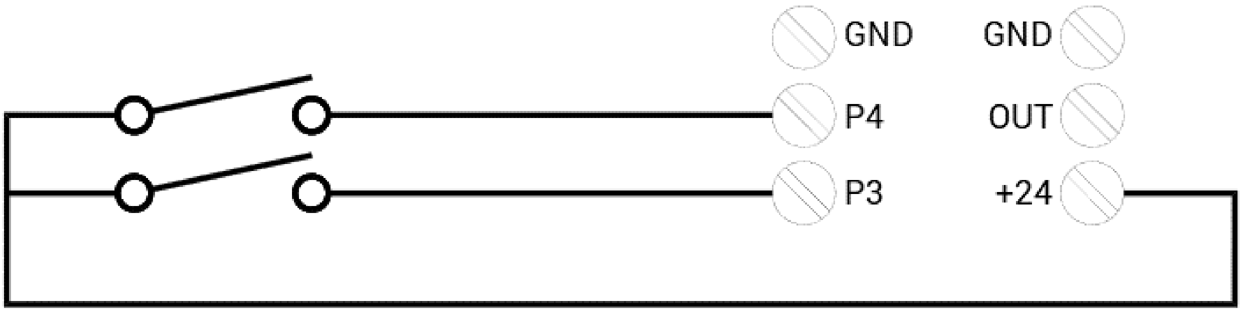
7.1. Typical Output Connection



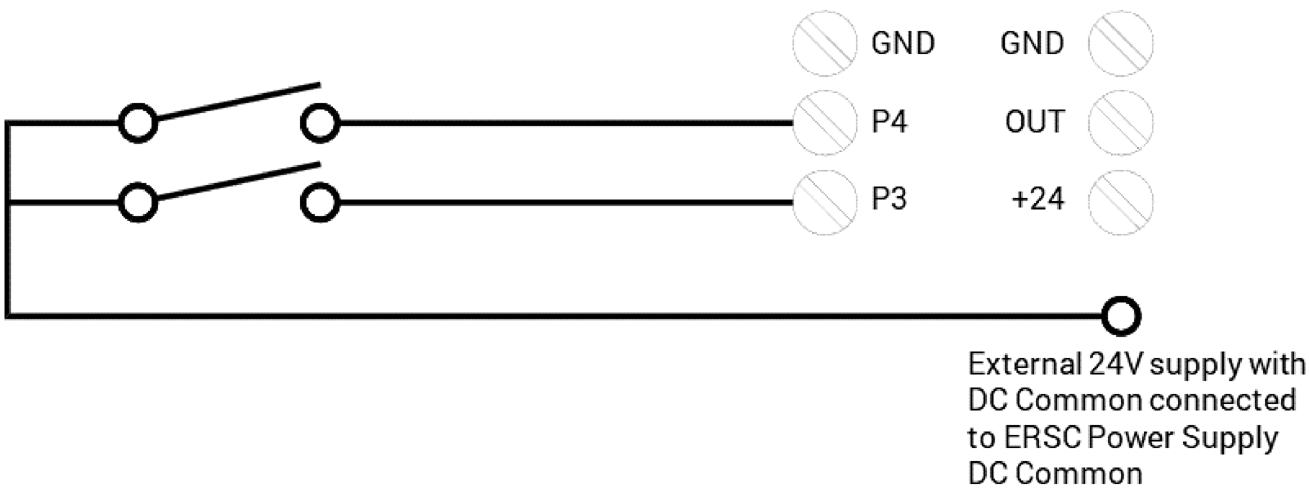
> ! If using optional external power supply, this external supply's DC common MUST be connected to ERSC power supply DC common

7.2. Typical Input Connections

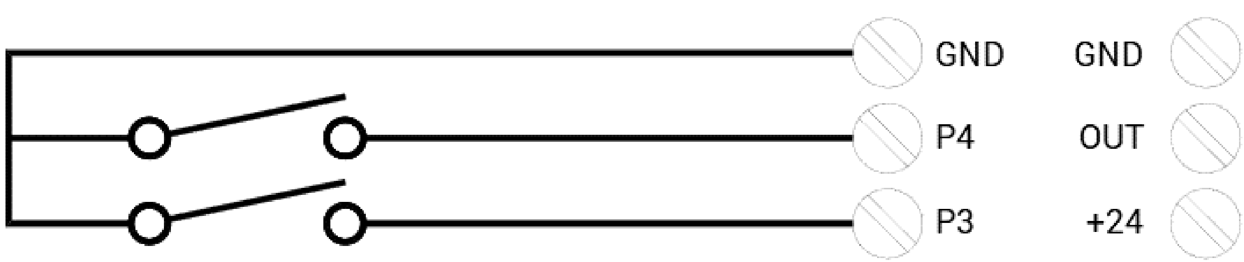
PNP Inputs – powered from ERSC



PNP Inputs – powered from external supply



NPN Inputs



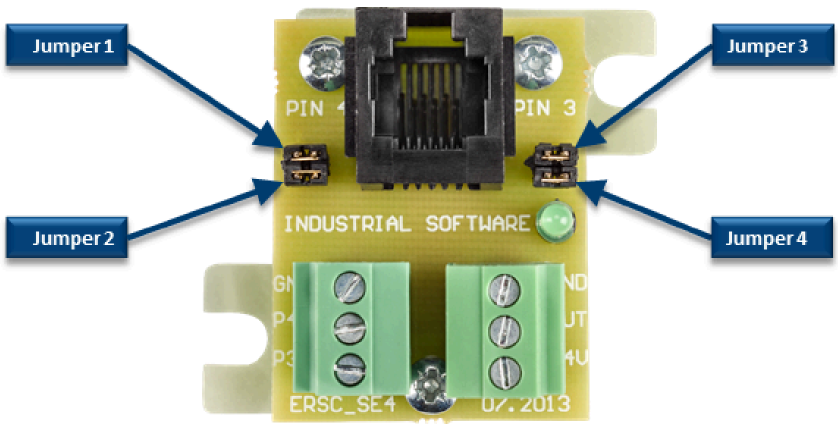
7.3. Input Circuit Blocking Diode Jumpers

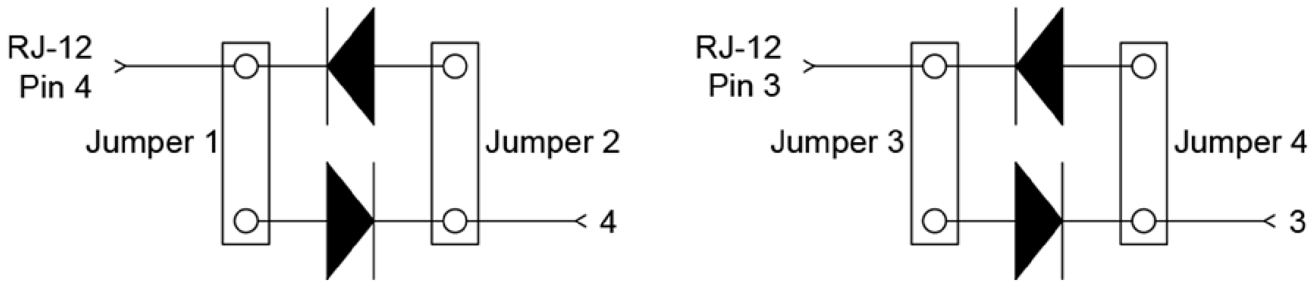
A typical situation for using an SE breakout Module is to connect a PLC’s digital output to one of the ERSC’s input signals for interlock control. Some solid state digital output circuits for some PLC models can provide false inputs to the ERSC module because of the module’s PNP/NPN auto-detection circuitry. Some PLC solid state digital outputs can provide an electrical path to ground when powered off. This can result in the ERSC module’s NPN auto-detection circuit to interpret this condition as a positive NPN signal and thus energize its input circuit. The SE Breakout module provides jumper selectable diodes in order to block this opposite flow in order to keep from false triggering an ERSC input.

Please note that NOT ALL PLC SOLID STATE OUTPUTS present this behavior.

! For relay or dry contact signals to the P3/P4 inputs, leave both jumpers installed

Both the P3 and P4 inputs on the SE Breakout Module has a pair of user removable jumpers to allow configuration to block a ground path when connecting a PNP input signal and to block a voltage when connecting an NPN input signal.





Please note that if both jumpers remain installed, the ERSC Module will auto detect PNP/NPN operation and there will be no protection from reverse current path from Solid State devices

! At least one jumper MUST remain installed. If BOTH jumpers are removed then there will be NO SIGNAL (PNP nor NPN) connected to the ERSC

Examples

For a solid state PLC output module SOURCING 24V:

For P4 – remove Jumper 1 and leave Jumper 2 installed
For P3 – remove Jumper 3 and leave Jumper 4 installed

For a solid state PLC Output module SINKING 0V:

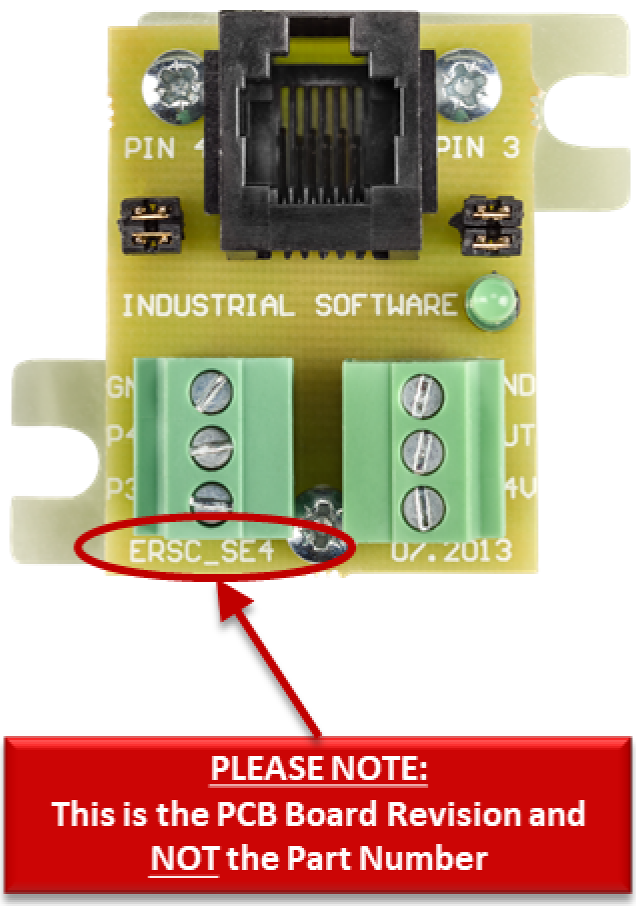
For P4 – remove Jumper 2 and leave Jumper 1 installed
For P3 – remove Jumper 4 and leave Jumper 3 installed

7.4. Notes on SE Module Revisions

Because there have been a handful of revisions to the SE breakout module since its introduction, this section documents the changes and updates made for each revision.

Part Number and Circuit Board Revision

Customer have been confused (and rightly so) because initially Pulseroller changed the Part Number for the SE module upon each revision because these revisions changed the functionality of the module. It was often interpreted by customers that the markings on the Printed Circuit Board (PCB) were the part number. Actually, the markings on the PCB are the PCB's revision level. For part numbers SE-2, SE-3, and SE-4; the PCB revision matched the part number because there were functional changes with each revision. *However since the SE-4 PCB revision there have been no functional changes to the SE breakout module and no part number change.*



✻ PCB revisions greater than SE-4 are functionally equivalent to the SE-4 part number. As of the date of this publication the current PCB revision is SE-6. This

PCB revision is functionally equivalent to SE-4.

Revision History

In order to support previous revisions to the SE Breakout Module, the following items describe the functionality and possible limitations of each revision.

Part Number ERSC-SE2

Original design providing bias diodes for the Pin 3 and Pin 4 inputs to allow solid state PLC output modules to operate without inadvertently triggering the PNP/NPN auto detect feature of the ERSC. This design also provided the Control Port Output amplifier to allow higher current devices to be connected to this output. The OUT signal is PNP only.

> ! Because the Control Port Output is in an indeterminate state upon ERSC boot up; the OUT signal could possibly momentarily energize upon initial power up or re-boot of the ERSC. Users must take proper care when using the OUT signal in their control system logic.

> ! The 24V terminal on the ERSC-SE2 is directly connected to the ERSC's Control port power. If users connect an external power source to the 24V terminal on the SE2; this will back-feed power into the ERSC. If this is done, when the main power to the ERSC is disconnected; the ERSC could still power up. User must take care to not connect external power source to the 24V terminal.

Part Number ERSC-SE3

This version contains a modification to block the OUT signal from energizing during the ERSC's initial power-up or re-boot sequence. Thus, users no longer have to worry about an inadvertent OUT signal being energized during initialization or reboot of the ERSC. For convenience; an LED has been added that illuminates when the OUT signal is being energized.

> ! The 24V terminal on the ERSC-SE3 is directly connected to the ERSC's Control port power. If users connect an external power source to the 24V terminal on the SE3; this will back-feed power into the ERSC. If this is done, when the main power to the ERSC is disconnected; the ERSC could still power up. User must take care to not connect external power source to the 24V terminal.

Part Number ERSC-SE4

This version contains a diode circuit that blocks an external power source from back-feeding into the ERSC’s Control Port power. Users can now provide external power for devices connected to the Control Port in cases where these device’s power requirements exceed the maximum limits of the port.

8. EasyRoll Software

The EasyRoll Software Configuration Tool is a PC based application that provides the ability to change each module's default parameters that are otherwise not accessible from the module's built-in Auto-Configuration routine.

Learn About:

[How to Install EasyRoll]
[Getting to know the Main Screen]
[ZPA Upstream/Downstream Settings]
[Motor Settings]
[Diagnostic Window]
[Advanced Dialog]

8.1. Installing EasyRoll on your PC

The files for *EasyRoll* can be download 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"

8.2. ConveyLinx Ethernet Definition

All ConveyLinx Modules communicate over Ethernet network and use TCP/IP based protocols for normal function. All TCP/IP protocols require that each device on a network have a unique I.P. address assigned to it in order to function properly.
An I.P. address is in the format of: AAA.BBB.CCC.DDD where AAA, BBB, CCC, and DDD are numerical values between 0 and 255.

> * For the purposes of ConveyLinx; the AAA.BBB.CCC portion of the I.P. address taken together is defined as the Subnet. The DDD value of the address minus 19 is defined as the Node.

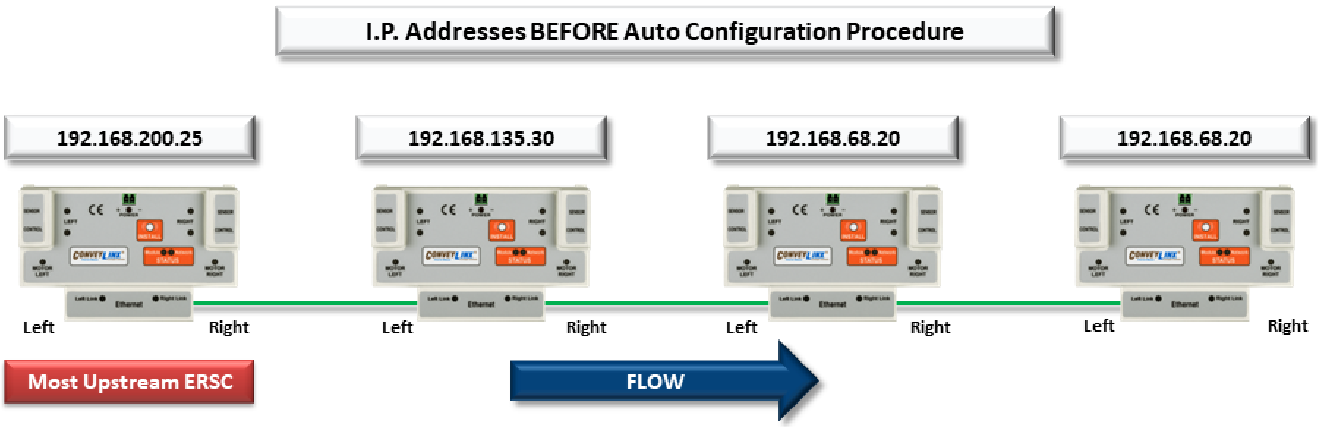
For example; if a module has an I.P. address of "192.168.25.20" then its *Subnet* address is "192.168.25" and its *Node* is 1 (i.e. 20-19 = 1)

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.

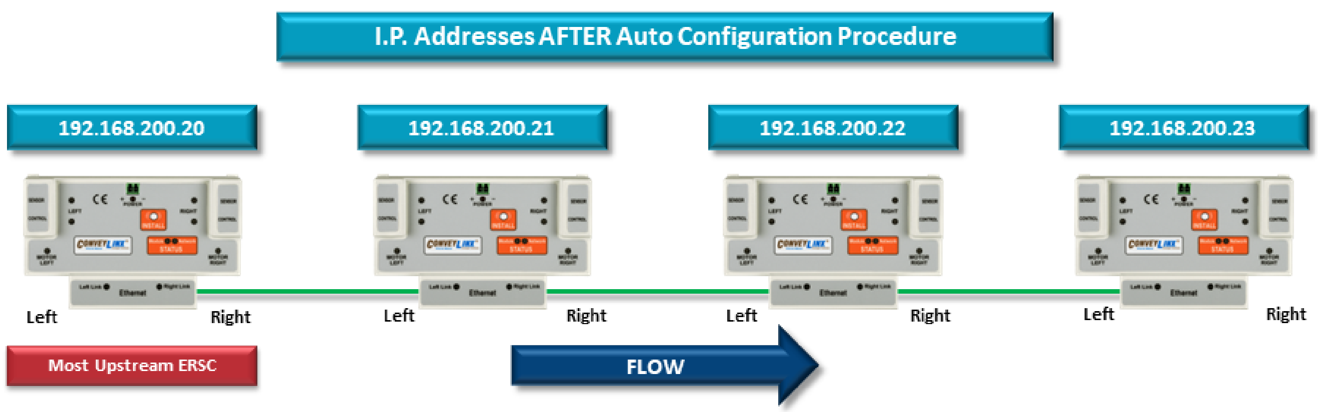
When the *Auto Configuration Procedure* is initiated; one of the many things that occur is that each module is automatically assigned a new I.P. address. This I.P. address for all modules is determined by the *Subnet* of the I.P. address already stored inside the most upstream module. Even if all downstream 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.



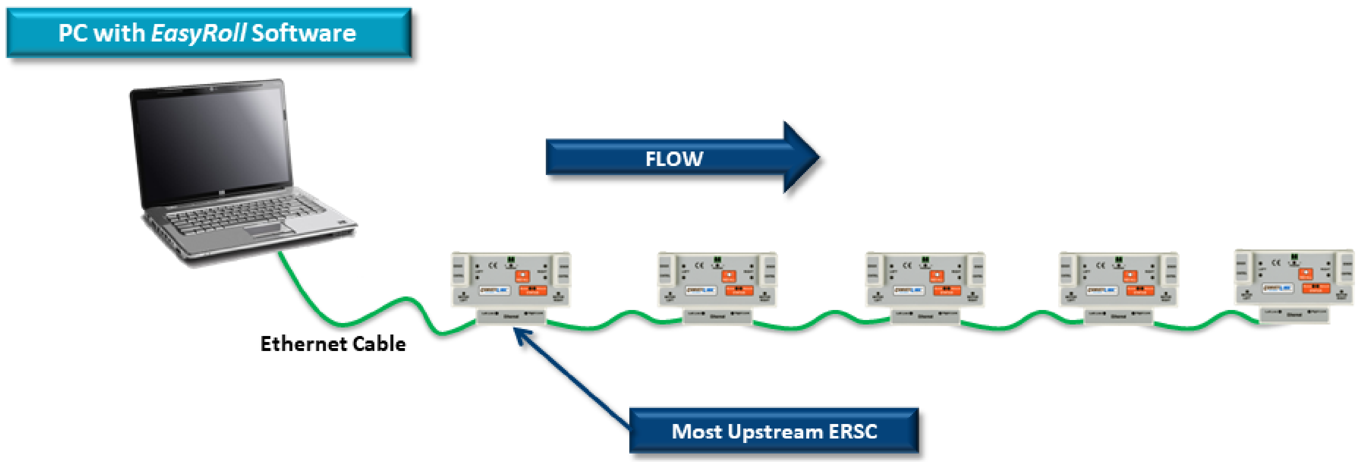
Once the *Most Upstream Module* is identified and the *Auto-Configuration Procedure* is performed; all 4 modules will have their I.P. address configured as shown.



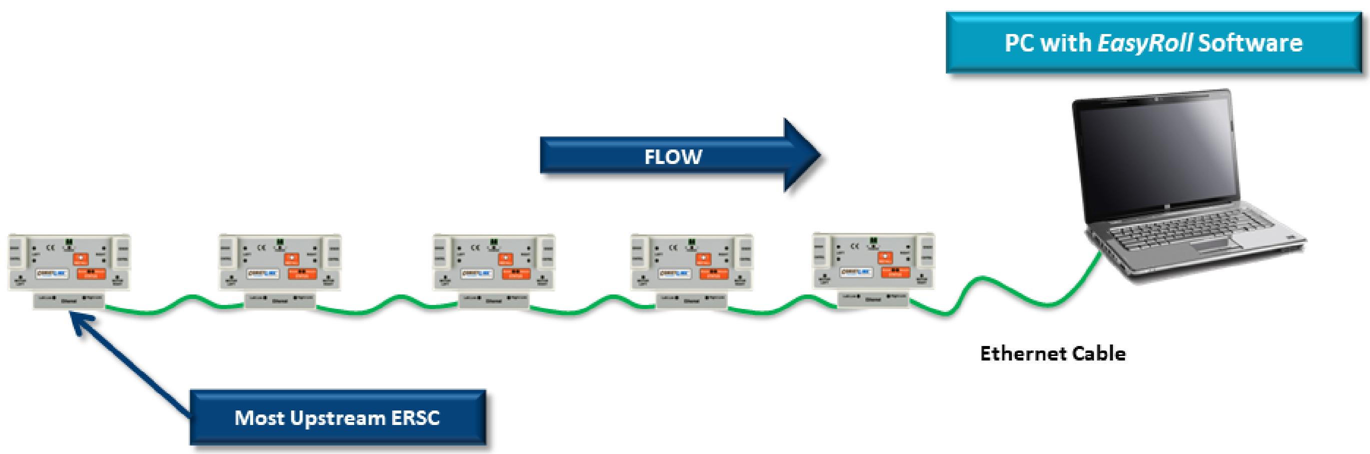
The *Auto-Configuration Procedure* will assign **Nodes** up to and including **DDD** octet 240. Therefore, each **Subnet** is limited to **221** module **Nodes**

8.3. Connecting your PC to ConveyLinx

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



OR



What do you want to do next?

The Subnet has already been Auto-Configured, my PC gets IP address form DHCP server, and I want to connect with EasyRoll and change default settings

- If you know the ***Subnet*** address of you modules, go to the Main Screen and enter in the *Subnet* address at the top and click *Refresh*.

- If you do not know the *Subnet* address of your modules, go to the Advanced Dialog and use the *Discover* function to view all connected modules to see which *Subnet* address you want to connect to and then go to the Main Screen and enter the *Subnet* address at the top and click *Refresh*

The Subnet has already been Auto-Configured, I need to manually set my IP address on my PC, and I want to connect with EasyRoll and change default settings

If you know the *Subnet* address of you modules, set your PC's IP Address withing the same subnet and subnet mask as your modules. Then go to the Main Screen and enter in the *Subnet* address at the top and click *Refresh*. If you do not know the *Subnet* address of your modules, go to the Advanced Dialog and use the *Discover* function to view all connected modules to see which *Subnet* address you want to connect and set you PC's IP Address within the same subnet and subnet mask. Then go to the Main Screen and enter the *Subnet* address at the top and click *Refresh*

I want to change the IP address of my Auto-Configuration master module and then perform an Auto-Configuration Procedure

Go to the Advanced Dialog and use the *Discover* function to view all connected modules. You will need to locate the specific module whose address you want to change. Double click this module from the list and its information will appear to the right side of the dialog. Enter in the IP Address and Subnet Mask you want to use and then click the *Set* button

8.4. Using Discover Function

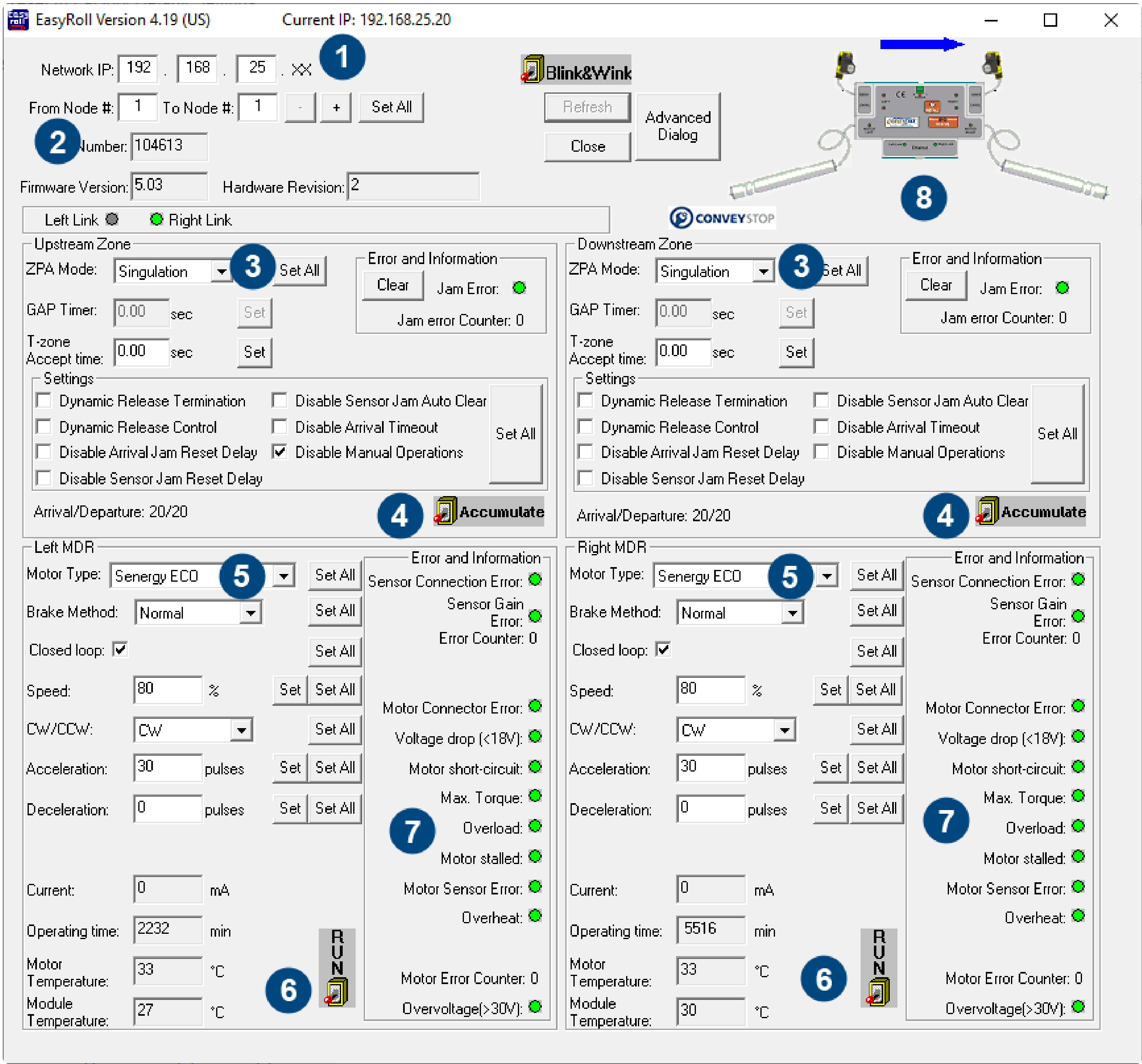
The *Discover* function is often one of the first activities you will need or want to do when you connect to your *ConveyLinx* network with EasyRoll. The *Discover* function tells *EasyRoll* to look out on the network and report back a list of all *ConveyLinx* modules it finds, regardless of what subnet they are in or their IP address.

> Because the Discover function does not use TCP/IP connections to find ConveyLinx modules, your PC does not have to have its IP address and subnet mask set to match any ConveyLinx module IP addresses. However, in order to eventually connect to any module and make any changes with EasyRoll, you will need to make sure your PC and the ConveyLinx network you want to connect to are in the same IP address subnet.

What you can do with the Discover Function

- See the IP address, serial number, and firmware version of each connected module
- Change the IP address and Subnet Mask for any module
- Enable/Disable DHCP server for Auto Configuration Master module

8.5. 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, speed control, 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

8.5.1. 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, it can define a range of Nodes you want to work with.

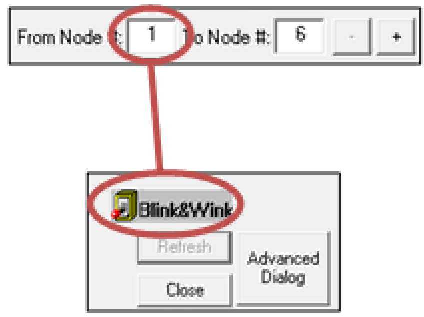
Clicking the + and – buttons will increment / decrement the Node value in the *From Node #* box and display the module data for the new Node selected. Please note that if you increment past the value of the last physical Node installed, you will receive an error message

> ✻ 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.

8.5.2. Node Identification

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

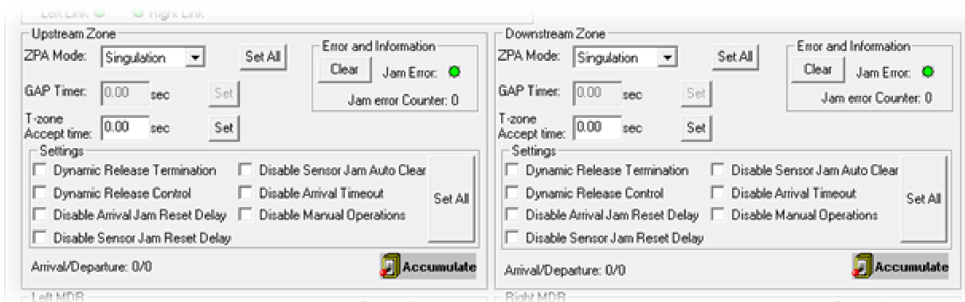
If a valid Node is selected in the ***From Node #*** text box and its information is displayed on the main screen; clicking the *Blink & Wink* switch icon will signal the selected module to blink on and off all of its LED indicators. Click the *Blink & Wink* switch again to turn this off



Please note that when Blink & Wink is active, the module stops all motors

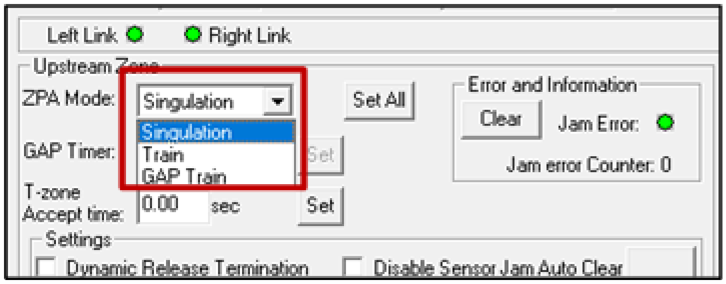
8.5.3. ZPA Upstream/Downstream Zone Settings

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.



8.5.3.1. ZPA 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

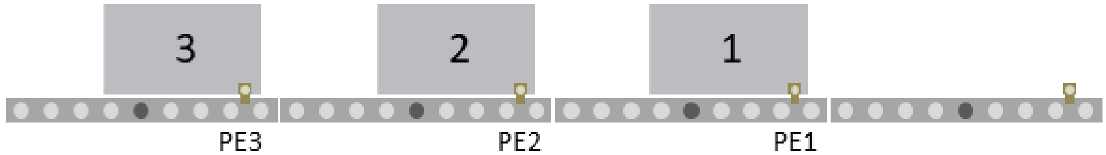
* Please note that Singulation and Train Release Modes are configurable per Zone and can be mixed on the same network

8.5.3.1.1. Singulation Release

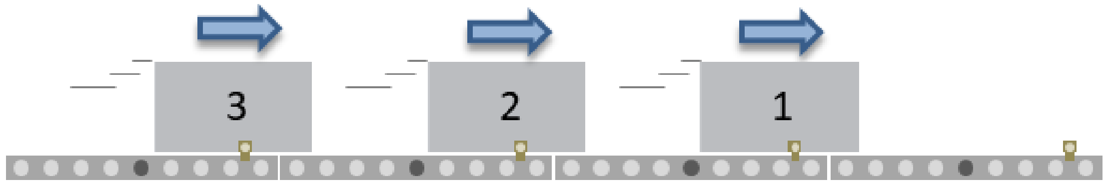
✻ Singulation Mode is the default setting and it is described in the Default Settings and Operation section

8.5.3.1.2. Train Release

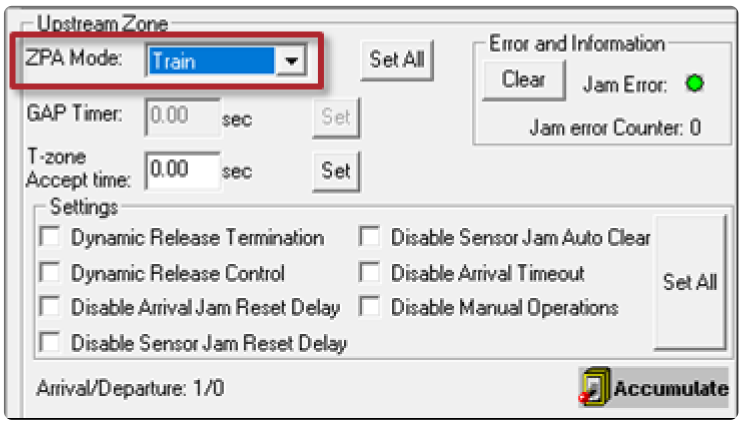
All 3 Cartons are Accumulated



When Carton 1 releases – Cartons 2 and 3 also release at the same time



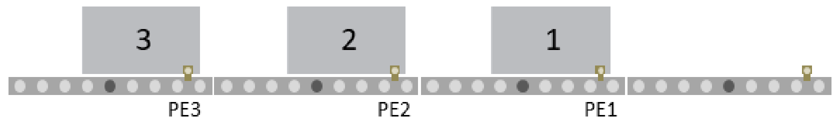
Select *Train* from the pull-down box for Train Mode



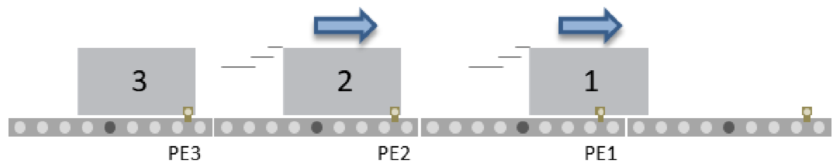
! Please note that starting many zones in Train Release mode particularly with heavy loads can cause voltage drops on your power supplies. Be sure to take care in sizing your power supply needs. You should consider GAP Train Release Mode if power supply sizing is a concern

8.5.3.1.3. GAP Train Release

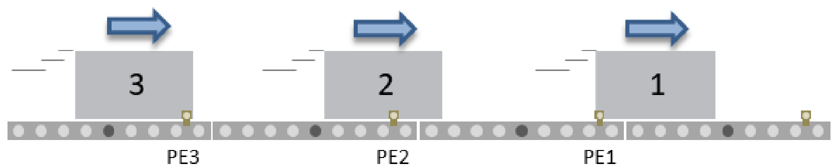
All 3 Cartons are Accumulated



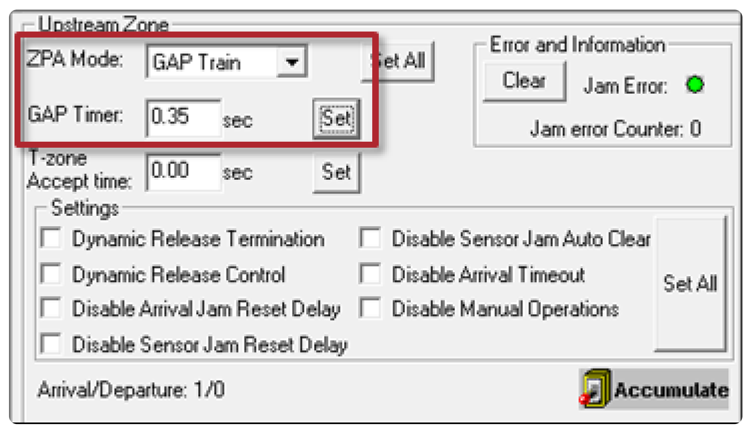
When Carton 1 releases and Zone 2's GAP time has expired, Carton 2 will begin to release. Carton 3 remains accumulated



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.

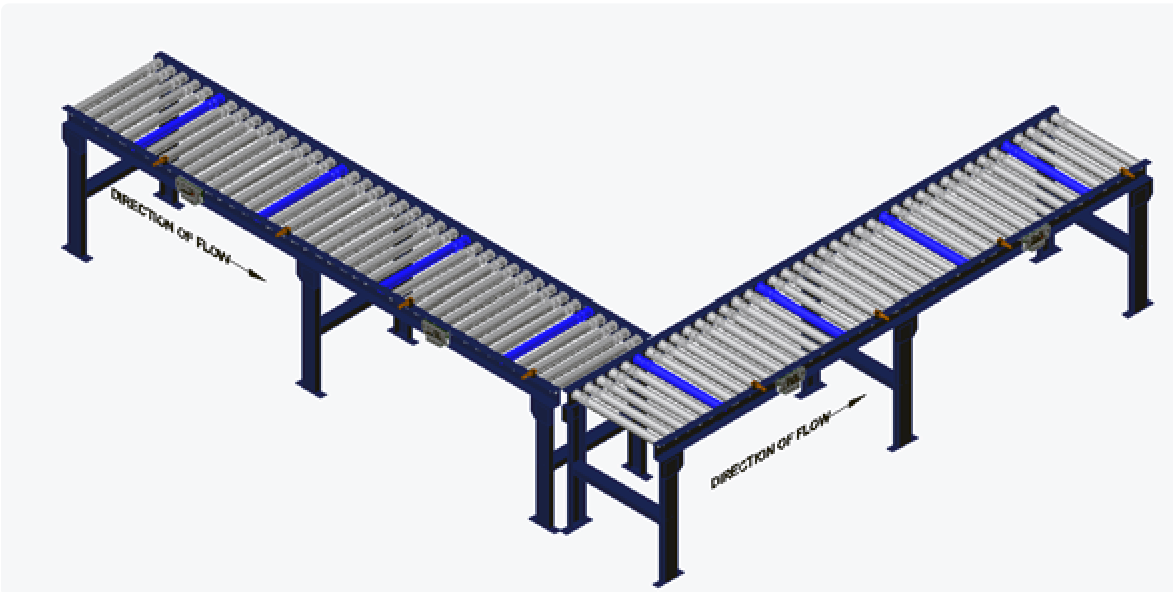


! Please note the GAP time does not create a gap or cause any delay when items are in transportation. The gap timer is only activated when a given zone accumulates an item and then it is allowed to release

* 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

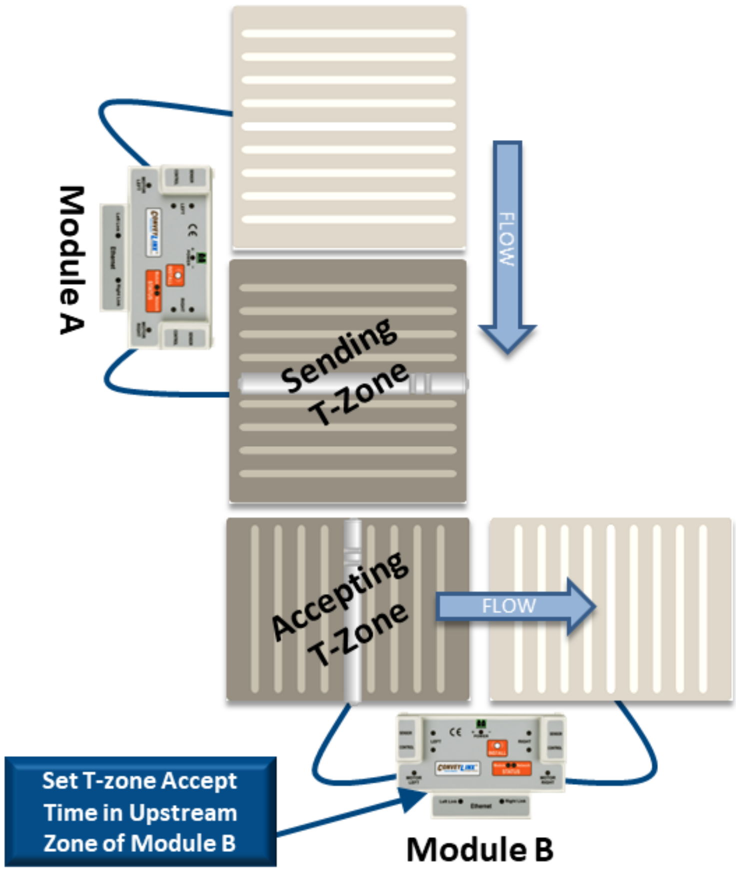
8.5.3.2. T-Zone Settings

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.



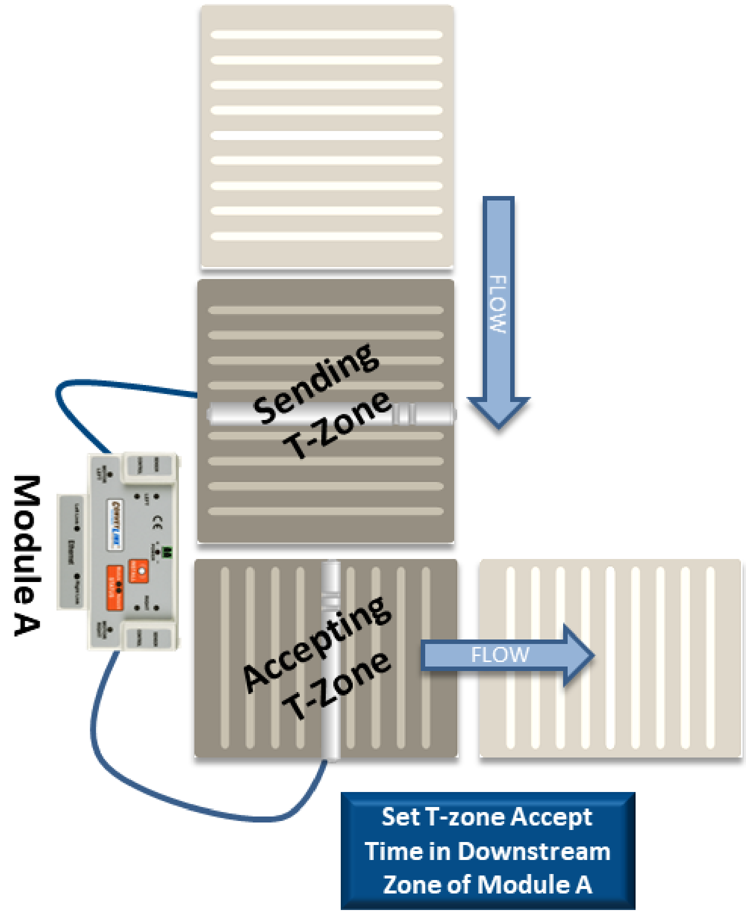
> ! Material handling considerations such as discharge conveyor speed and load weight have to be analysed prior to implementing a T-Zone configuration. Be sure to verify your mechanical design and carton characteristics before utilizing a T-Zone arrangement.

T-Zone Between 2 Separate Modules

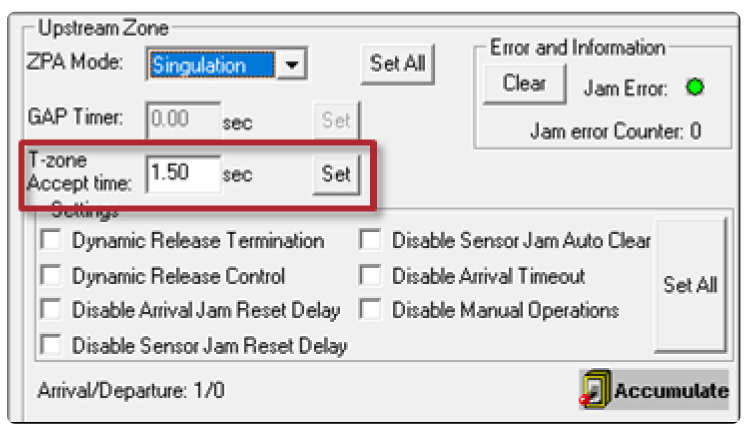


> **!** This example assumes both Module A and Module B are in the same *Subnet*. Otherwise you will need to use EasyRoll to establish connections between Module A and Module B.

T-Zone on a Single Module



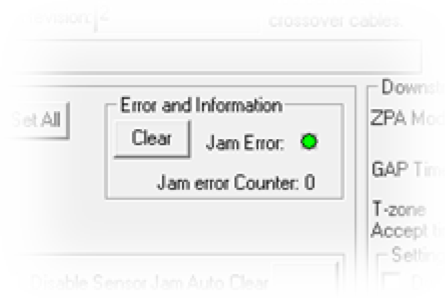
Enter the value, for example 0.200 for 200 milliseconds and click the Set button. Whether to change the Upstream Zone or Downstream Zone value on the main screen is dependent upon which zone is the *accepting zone*. In our example we entered 1.5 seconds



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

8.5.3.3. 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.

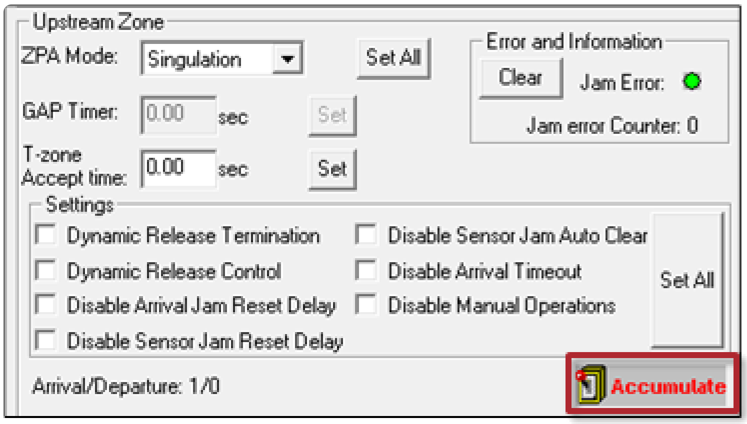


> The *Clear* button is a convenience so you do not have to go the zone in question and remove or otherwise clear the item from the zone sensor. If Sensor Jam Auto Clear is enabled, clicking the Clear button will re-initiate this procedure.

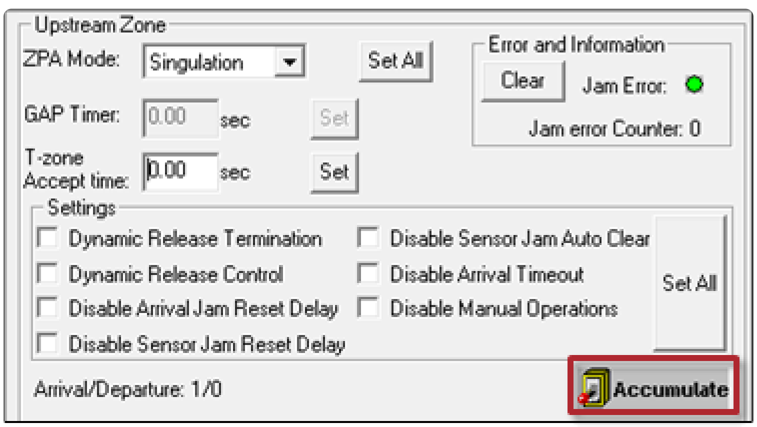
8.5.3.4. Accumulate Control from Main Screen

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.

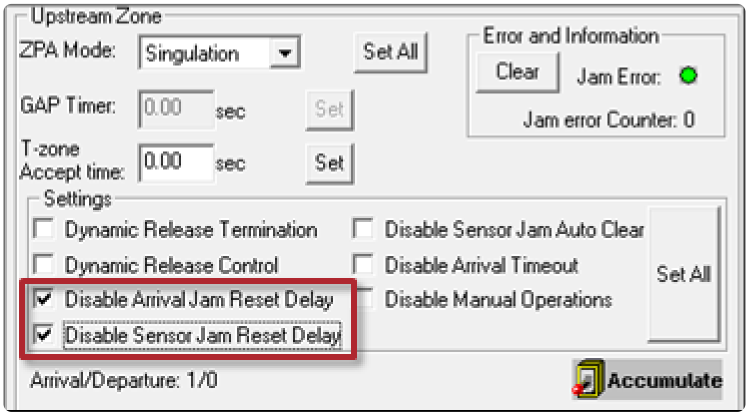
8.5.3.5. Settings Checkboxes

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.

8.5.3.5.1. 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

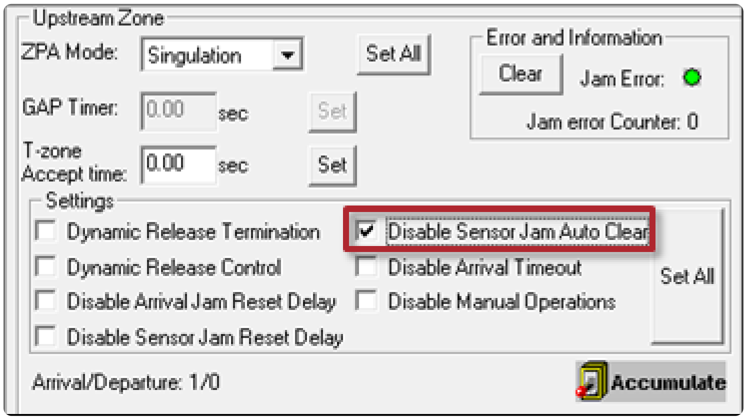


! Selecting either of these options will not eliminate the detection of the particular jam condition; it simply eliminates the default Auto Clear Time delay the logic utilizes before automatically clearing the condition.

8.5.3.5.2. 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

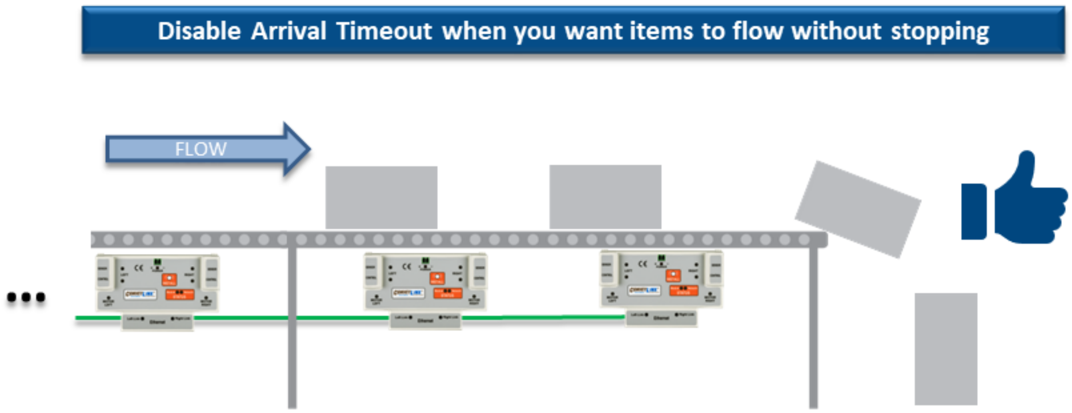
Clicking the *Disable Sensor Jam Auto Clear* check box will disable the 3 attempts to clear the *Sensor Jam* and will cause the zone to remain in the jam state until the sensor is cleared manually



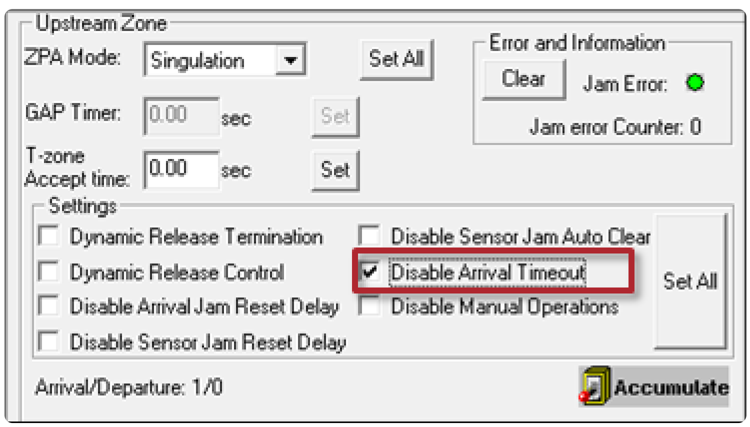
✻ You can also attempt to clear a Sensor Jam using the Clear button for the Zone

8.5.3.5.3. 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



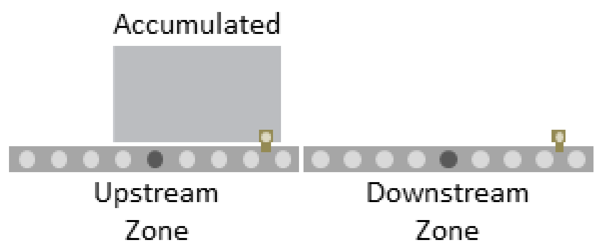
Clicking the *Disable Arrival Timeout* check box will prevent the discharging zone from registering an *Arrival Jam* condition. Upstream cartons will not wait to enter the discharging zone once a carton has left the discharging zone



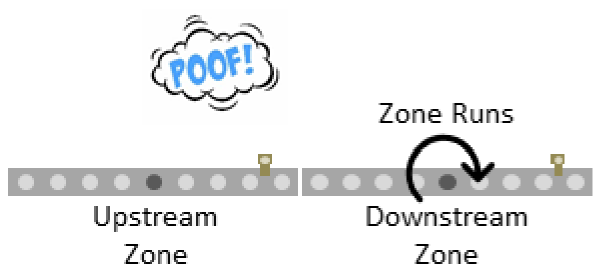
8.5.3.5.4. Disable Manual Operation

In normal ZPA operation, if a zone is in Accumulation (either by external device or Local Zone Accumulation control by Pin 3 on the Control Port) 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”

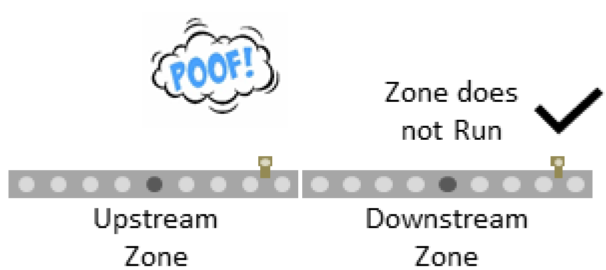
Carton accumulated in Upstream Zone by Local Accumulate signal on Pin 3 or external network signal



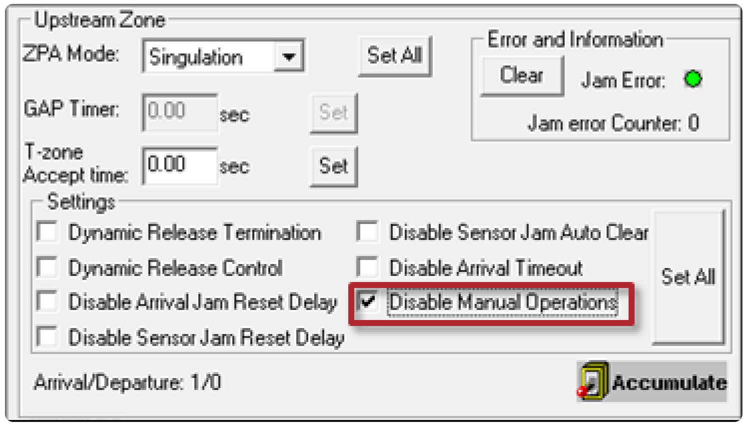
If carton is removed from Upstream Zone, the Downstream Zone will run to “find the missing carton”. Run time is the duration of the Jam Timer setting (5 sec. by default)



If you check the *Disable Manual Operations* check-box for the Downstream Zone, then it will not run when upstream carton is removed



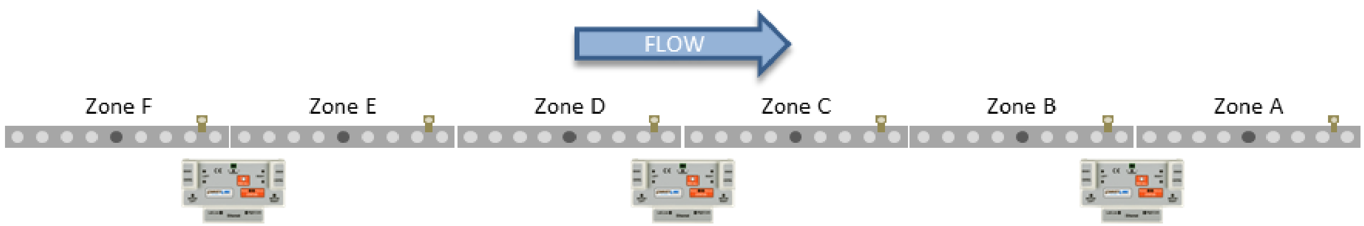
You first navigate the main screen to the *zone you want to prevent from running* when it's upstream neighboring zone has its carton removed. Clicking the *Disable Manual Operations* check box will prevent this zone from running when its upstream neighbor has an accumulated carton manually removed



8.5.3.5.5. Dynamic Release

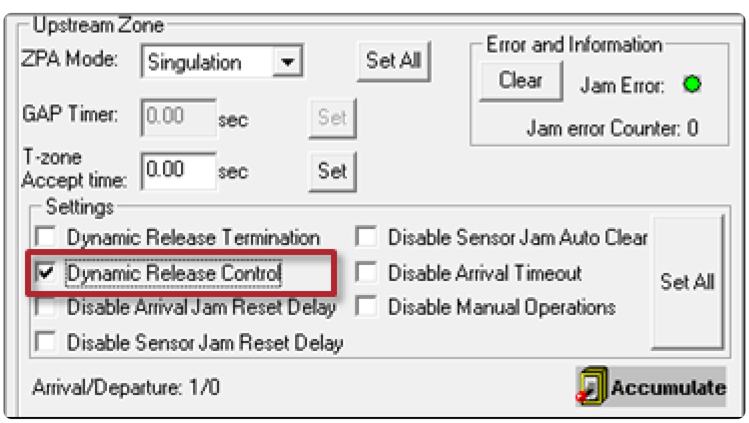
Some conveyor applications may require that you be able to switch a defined quantity of zones between singulation and train release modes depending on operational conditions. *Dynamic Release* allows you to utilize *Control Port* signals to remotely perform this switching. There are two zones that require configuration in order to use *Dynamic Release*. The most downstream zone in the range of zones you wish to control is the *Dynamic Release Control* zone and the most upstream zone in the range of zones you wish to control is the *Dynamic Release Termination* zone. The *Dynamic Release Control* zone requires you to energize its *Control Port Pin 3* signal to toggle between release modes. If the default release mode for the range is Singulation, then energizing *Pin 3* will switch the range to Train. If the default release mode is Train, then energizing *Pin 3* will switch the range to Singulation.

5 Zone Dynamic Release Example

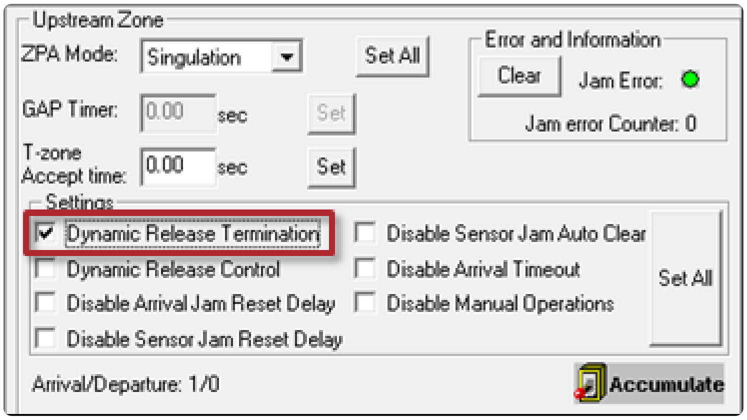


For our example, we want to dynamically switch Zones A through E between *Singulation Release* and *Train Release* modes. All Zone's default setting in our example is Singulation Release mode. Zone A will be the *Dynamic Release Control* zone and Zone E will be the *Dynamic Release Termination* zone. We will wire our signal to Zone A Control Port Pin 3 to to make this work. When we energize the signal, Zones A thru E will operate in Train Release Mode and when the signal is de-energized, Zones A thru E will operate in Singulation Release mode.

You first navigate the main screen to the most downstream zone of the *Dynamic Release* range (Zone A). Clicking the *Dynamic Release Control* check-box will set this zone's Control Port to look for *Pin 3* energized to switch the release mode



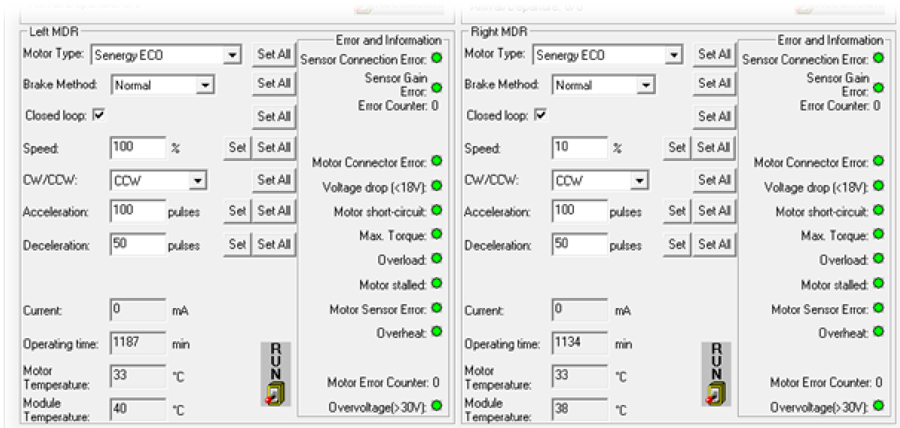
Then you navigate the main screen to the upstream zone of the Dynamic Release range (Zone E). Clicking the *Dynamic Release Termination* check-box will set this zone as the "termination" of the Dynamic Release range



8.5.4. Motor Settings

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

- Change Motor Type
- Change Brake Method
- Change Speed Control Method
- Change Speed Setting
- Change Rotation Direction
- Change Acceleration/ Deceleration

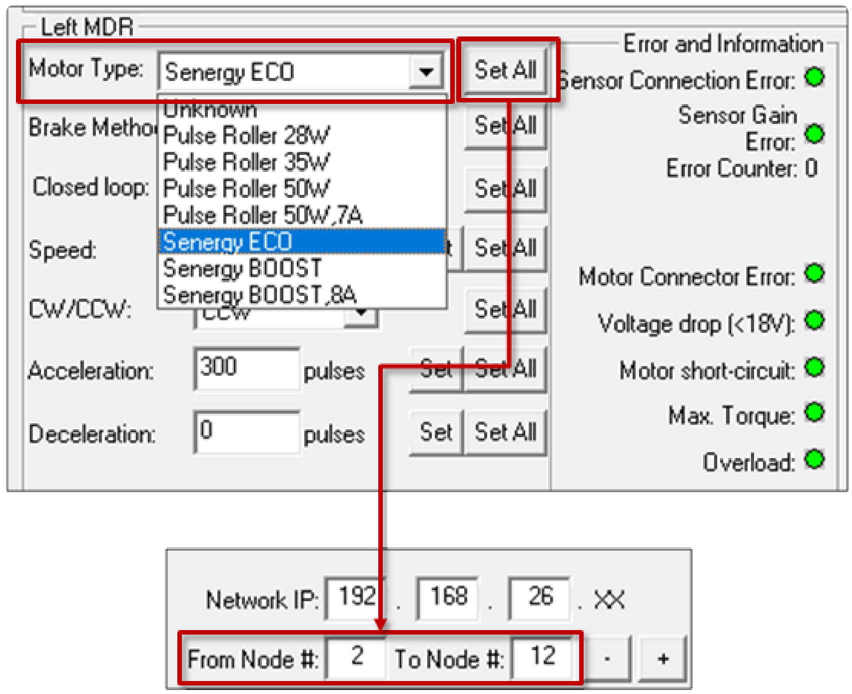


Please note that the Motor Settings grouped by motor plugged into the physical Left and Right sides of the module and are not based upon Upstream or Downstream product flow.

8.5.4.1. Motor Type

The *Motor Type* drop-down box lists all motor brand and types whose profiles are available for the module. *Senergy 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.

Selecting a *Motor Type* from the drop down box immediately makes the change for the current Node. 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. In our example, the *From Node # / To Node #* boxes have the values of 2 and 6. Changing the selection in the *Motor Type* drop down will change Node 2 and then by clicking *Set All* it will change Nodes 3 thru 12 to the same setting as Node 2



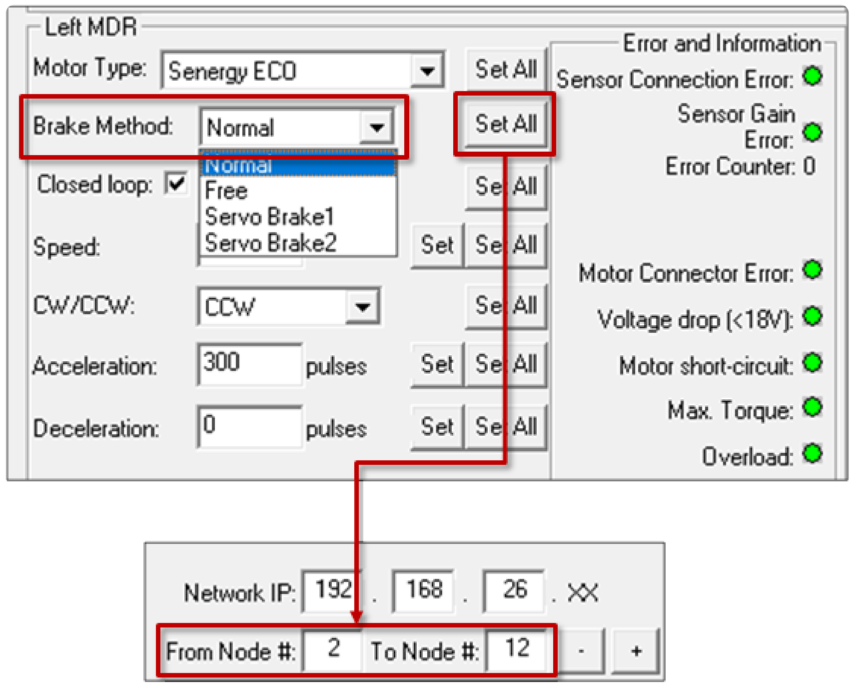
✱ Please note that if you change the *Motor Type* and this module goes through another *Auto Configuration* procedure, the *Motor Type* setting **will not** reset back to default. It will remain unchanged at the last selected setting.

! Please consult your particular MDR's documentation and review your application if you are unsure as to which motor-type setting to use

8.5.4.2. Brake Method

The *Brake Method* drop-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.

Selecting a *Brake Method* from the drop down box immediately makes the change for the current Node. 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. In our example, the *From Node # / To Node #* boxes have the values of 2 and 6. Changing the selection in the *Brake Method* drop down will change Node 2 and then by clicking *Set All* it will change Nodes 3 thru 12 to the same setting as Node 2



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 1	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 2 of the 3 motor coils in the proper sequence to move the rotor back to its original stop position
Servo Brake 2	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 all 3 of the motor coils in the proper sequence to move the rotor back to its original stop position

Please note that if you change the *Brake Method* and this module goes through another *Auto Configuration* procedure, the *Brake Method* setting **will not** reset back to default. It will remain unchanged at the last selected setting.

Servo Brake 1 and 2 are functionally equivalent. Servo Brake 2 utilizes more power and provides more holding torque. Consequently, because Servo 2 uses more current, the potential for heat build-up is present depending on your application. If Servo Brake 1 provides enough holding torque for the application, it is recommended using it in lieu of Servo Brake 2. Servo Brake 2 should only be used when Servo Brake 1 does not provide enough holding torque for the application

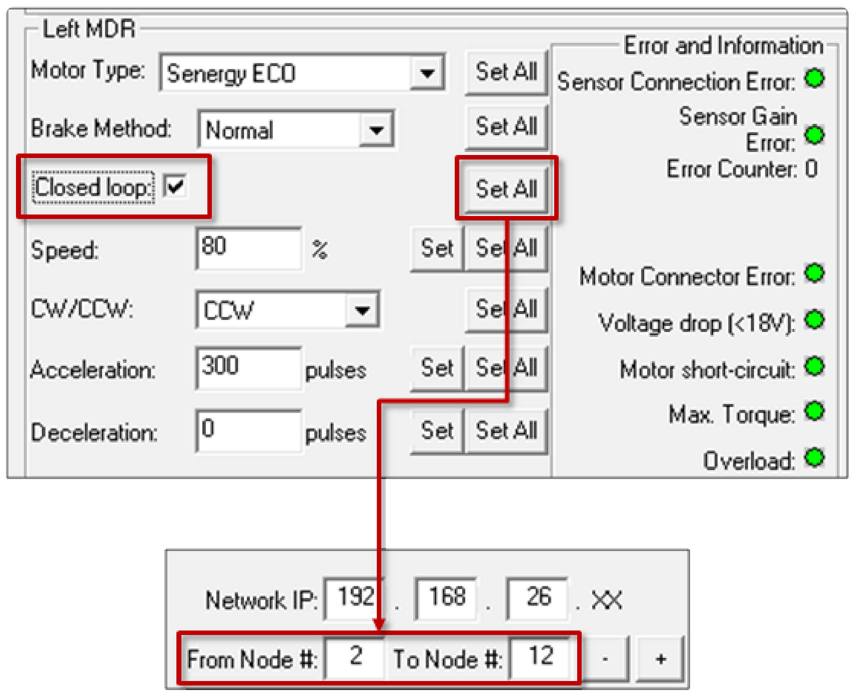
8.5.4.3. Speed Control Method

You can change the Speed Control Method between **Open Loop** and **Closed Loop**

Closed Loop

Closed Loop speed control utilizes a proportional – integral (PI) algorithm to regulate MDR speed. When enabled, the module motor control processor accepts the input from the *Hall Effect* sensors to measure rotor speed and will adjust the motor output accordingly to attempt to regulate speed. The value entered is in percent of the motor's rated speed. ***Closed Loop*** speed regulation can be used for applications where you need to maintain constant speed with varying carton weights. The default setting is for ***Closed Loop*** to be enabled.

Checking the ***Closed Loop*** check box enables **Closed Loop** speed control method. You can also use the ***Set All*** button to set the selected speed control method for each motor in the *From Node # / To Node #* boxes. In this example the *Left MDRs* from nodes 2 thru 12 will each get the same *Speed Control Method* speed setting.



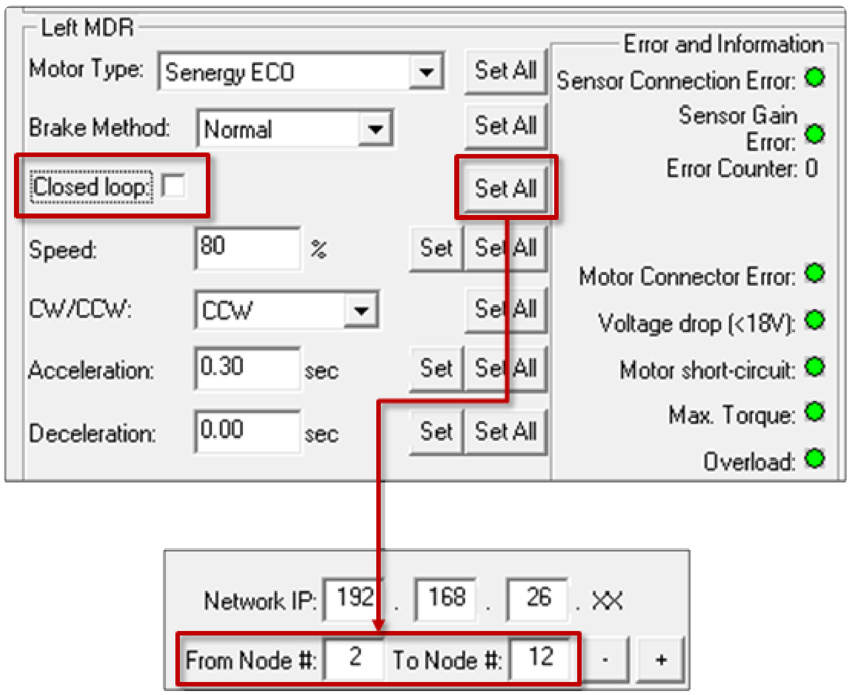
Also note that when Closed Loop is enabled; the units for Acceleration and Deceleration change from time-based (seconds) to distance-based (pulses). Motor pulses can be converted to linear distance based upon knowing the particular speed code of your MDR and its tube diameter. Please refer to Motor Pulse to Distance Calculation for details and an example of calculating this distance.

Closed Loop speed control will provide PWM voltage to the MDR up to the limit of the particular MDR's selected profile and/or the current limits allowed by the module's built in protection algorithms and circuitry. To avoid unexpected results, please review your particular mechanical application along with the published MDR performance data prior to implementing Closed Loop speed control

Open Loop

Open Loop speed control does not utilize any such feedback to regulate speed. The value entered is the percent *PWM* waveform (or "throttle") that is output to the motor and the actual rotor speed will fluctuate depending on the mechanical loading on the motor

Unchecking the ***Closed Loop*** check box disables Closed Loop speed control method and enables ***Open Loop*** speed control method. You can also use the ***Set All*** button to set the selected speed control method to each motor in the ***From Node # / To Node #*** boxes. In this example, all of the *Left MDRs* from nodes 2 thru 12 will each get the same *Speed Control Method* speed setting.



Please note that if you change the ***Speed Control Method*** and this module goes through another ***Auto Configuration*** procedure, the ***Speed Control Method*** setting **will not** reset back to default. It will remain unchanged at the last selected setting.

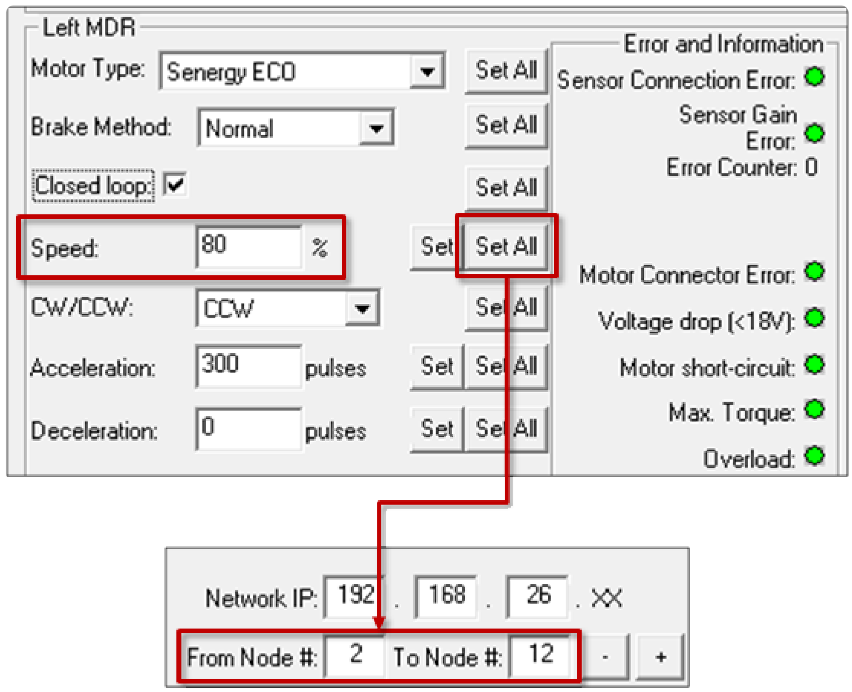
8.5.4.4. Speed

The definition of the *Speed* setting value depends upon the *Speed Control Method* and *Motor Type* selected:

Closed Loop

Closed Loop is the default *Speed Control Method* and the value entered is in percent of the motor rated speed based upon the *Motor Type* selected. The ConveyLinx module will adjust the motor output based upon fluctuating motor loading in order to maintain this set speed.

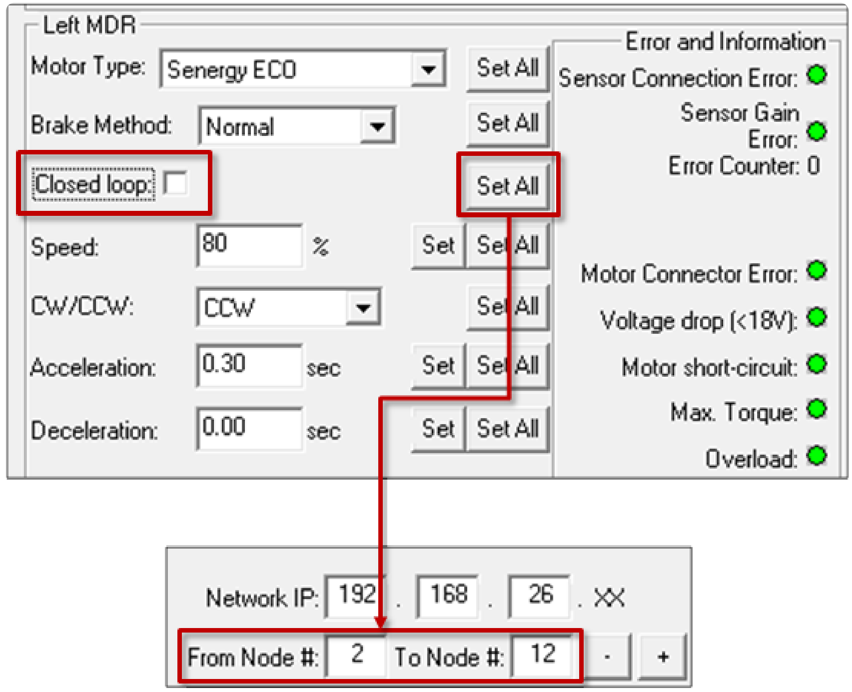
Type in the desired speed as a percentage. Valid values are from 1 to 100. You can also use the *Set All* button to set the entered speed to each motor in the *From Node # / To Node #* boxes. In this example we set the speed to be 80% for Node 2 and when we click the *Set All* button, the *Left MDRs* from nodes 2 thru 12 will each get the same speed setting



Open Loop

Open Loop value entered is in % of the selected *Motor Type's* rated Pulse Width Modulation (PWM) voltage. With this *Speed Control Method*, the PWM voltage being fed to the motor remains constant and the actual measured motor speed will fluctuate depending upon motor load.

Type in the desired speed as a percentage. Valid values are from 1 to 100. 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 we set the speed to be 80% for Node 2 and when we click the *Set All* button, the *Left MDRs* from nodes 2 thru 12 will each get the same speed setting



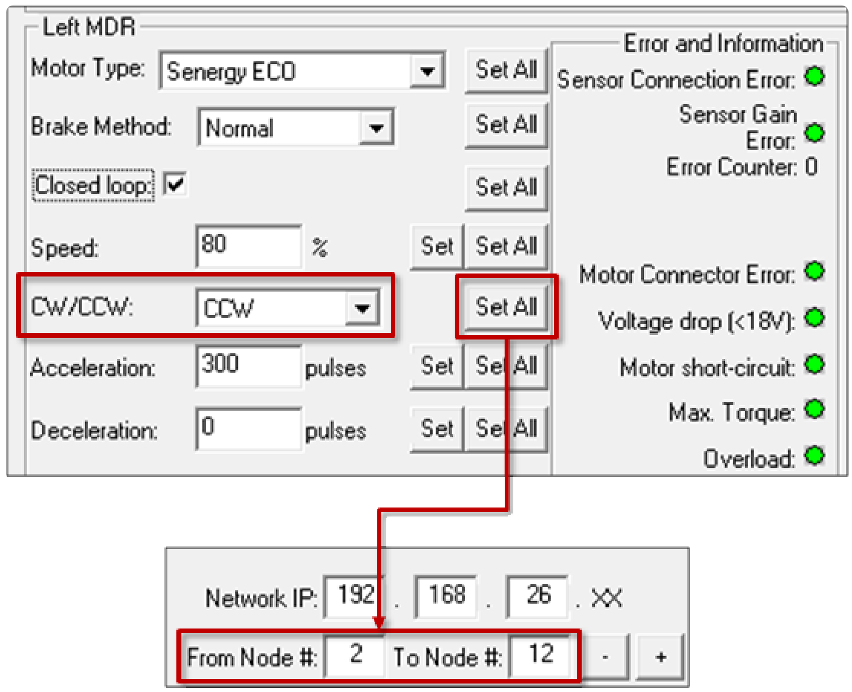
Please note that if you change the *Speed* and this module goes through another *Auto Configuration* procedure, the *Speed* setting **will not** reset back to default. It will remain unchanged at the last selected setting.

Please consult your particular motor brand and model's documentation for determining the actual speed of the MDR's for your application. The actual mechanical speed of the MDR is determined by the mechanical gearing attached to the motor. All Senergy MDR's have identical motors but will run at very different speeds and have very different torque characteristics depending on the mechanical gearing used.

8.5.4.5. 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 2 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.

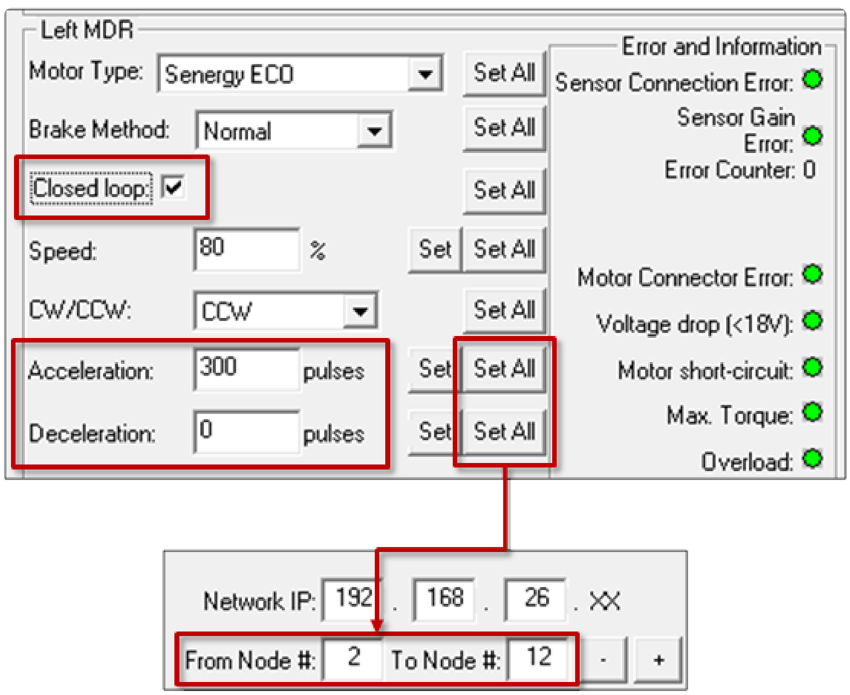
8.5.4.6. Acceleration/Deceleration

Acceleration and *Deceleration settings* are fields you enter values similar to Speed. The units of the value you enter are dependent upon which *Speed Control Method* you have selected.

When in Closed Loop

In **Closed Loop**, the units shown are **Motor Pulses** which means the control will accelerate and decelerate in terms of motor rotor distance based upon Hall Effect sensor state change and the controller’s current limits and the mechanical limits of the connected drive train. Because the module does not have any data regarding the gear reduction ratio of the installed MDR, you will need to work with the Motor Pulse to Distance Calculation to find out the amount of linear distance the particular acceleration or deceleration will take

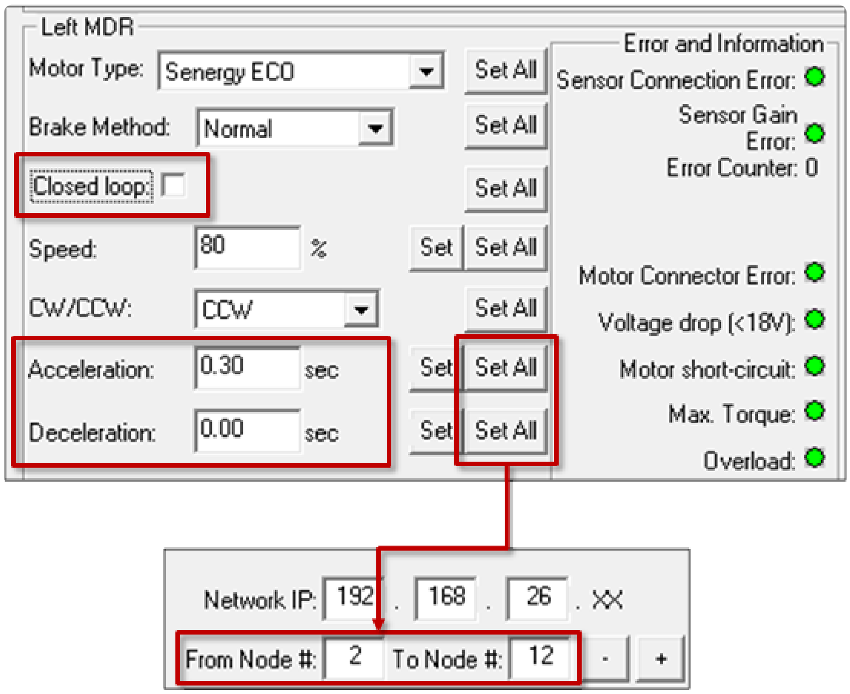
Type in the desired accel and decel values in **motor pulses**. You can also use the *Set All* button to set the entered values to each motor in the *From Node # / To Node #* boxes. In this example we set the accel to 300 and the decel to 0 for Node 2 and when we click the *Set All* button, the *Left MDRs* from nodes 2 thru 12 will each get the same speed setting



When in Open Loop

Units shown are seconds which means that the control will accelerate and decelerate for the time specified within the controller’s current limits and the mechanical limits of the connected drive train

Type in the desired accel and decel values in **seconds**. You can also use the *Set All* button to set the entered values to each motor in the *From Node # / To Node #* boxes. In this example we set the accel to 0.30 and the decel to 0 for Node 2 and when we click the *Set All* button, the *Left MDRs* from nodes 3 thru 12 will each get the same speed setting



✻ Please note that if you change either of the *Accel/Decel* values and this module goes through another *Auto Configuration* procedure, the *Accel/Decel* settings **will not** reset back to default. They will remain unchanged at their last selected settings.

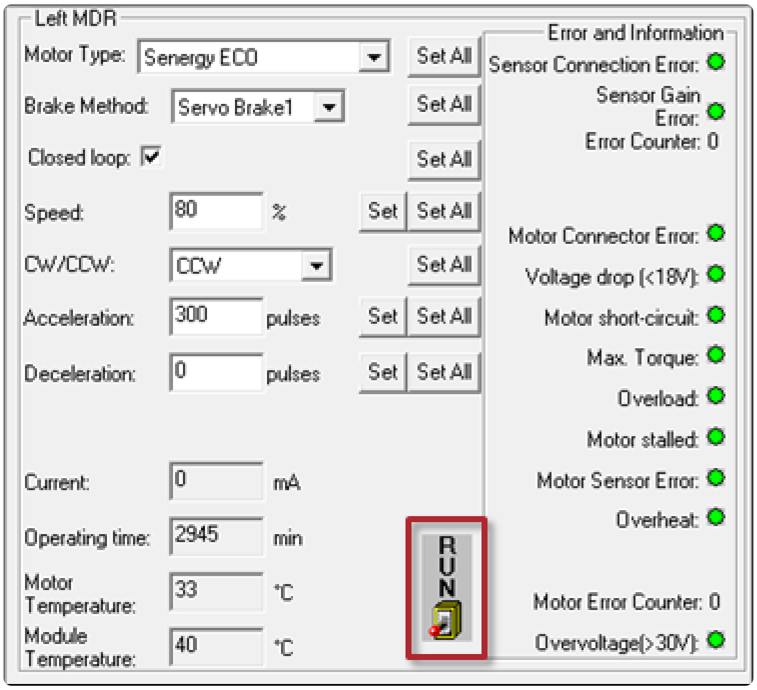
! Please note that very small deceleration values can cause the module to indicate a Short Circuit Error. This can happen at speeds greater than 75% and deceleration values less than 0.1 seconds (in Open Loop) or 50 pulses (in Closed Loop) with Servo Brake 1 or Servo Brake 2 enabled.

8.5.4.7. Motor Jog and Error Indicators

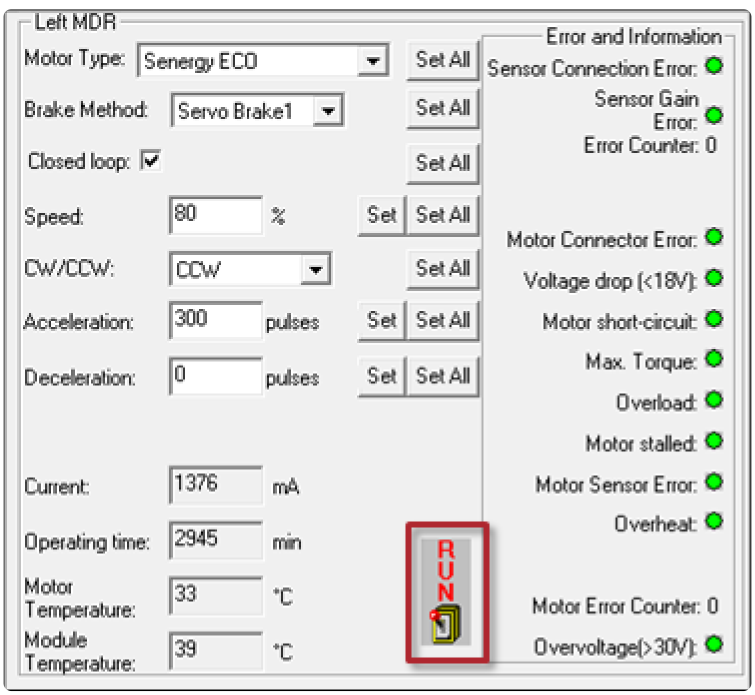
Motor Jog (Run) Function

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

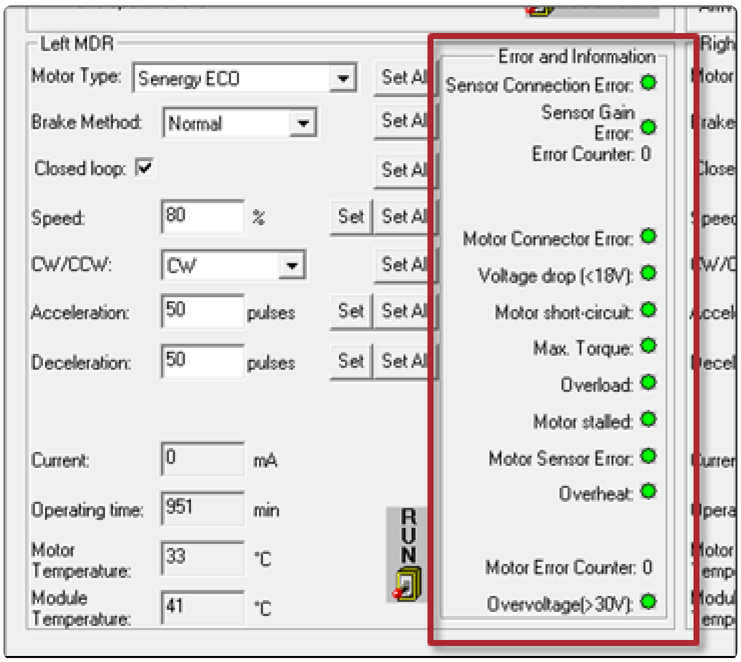


> Please note the **Motor Jog (Run) Function** rotates the motor in the direction indicated by the Rotation Direction selection which may or may not be the direction of flow established by the **Auto Configuration** procedure.

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 are also status indicators for the Sensors connected to the Left and Right sides of the modules.

Indicators are Green when the condition does not exist and will be Red when the condition does exist.



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 3 of the Sensor port changes state from its Auto-Configuration state
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 not rotating sufficiently for at least 20 seconds
Motor Stalled	Occurs when the motor is commanded to run and the rotor is not turning
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 calculated motor temperature has exceeded 105°C or the module temperature has exceeded 90°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

8.5.4.8. Motor Pulse to Distance Calculation

For the Senergy motor there are 30 pulses per revolution of the motor shaft. The formula to calculate the linear distance the roller would turn in one pulse is:

mm/pulse = (π × *Roller Diameter*) / (*30* × *Gear Ratio*)

> Roller Diameter and Gear Ratio values for specific roller diameters and speed codes are found in our Pulseroller Global Motor Roller Technologies Product Catalog available at www.pulseroller.com

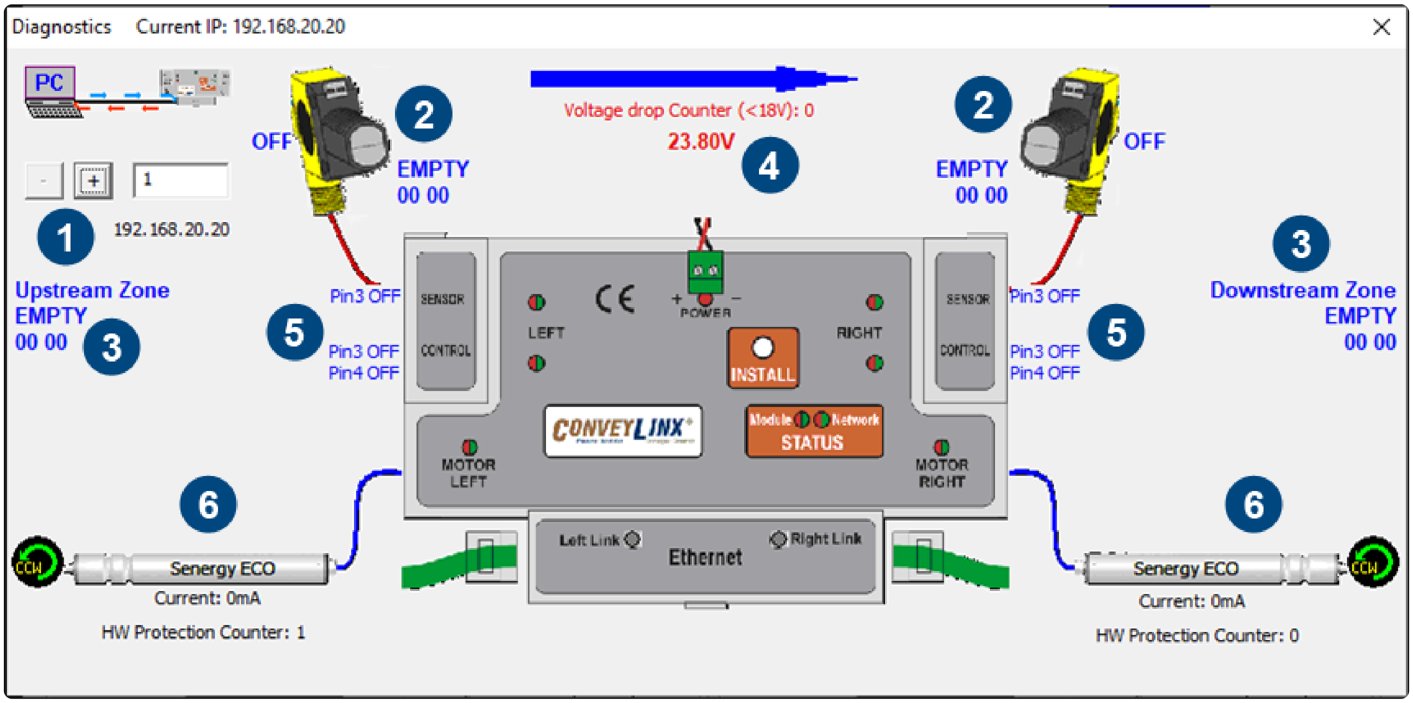
For example, we have a standard Senergy roller with a diameter of 48.6 mm (standard 1.9" tube) and our speed code is 60. From the chart on page 10 (2019 edition of the Catalog) indicates that the Gear ratio for a 60-speed code is 11. From this information we plug in the values into our formula:

(π × 48.6) / (30 × 11) = 0.463 *mm per pulse*

For this 60-speed code Senergy roller, when you set the Deceleration in Closed Loop mode to 50 pulses; the roller will decelerate for a distance of:

0.463 _ mm/pulse_ × 50 *pulses* ≅ 23 *mm*

8.5.5. Diagnostic Window

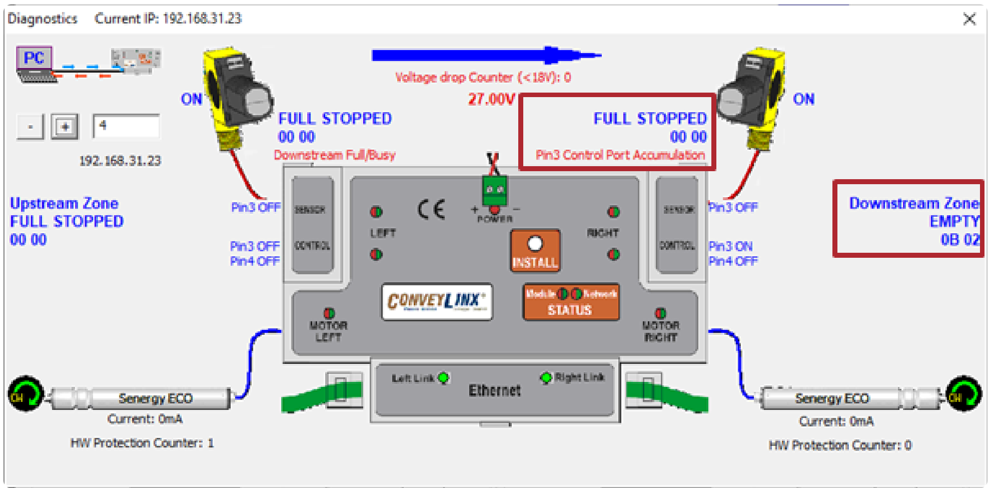


Indicator	Description
1	You can navigate to the next or previous module's Diagnostic Window by clicking the + and – buttons
2	Displays the current ZPA status of the zone and the state of the sensor
3	Displays the status of the upstream and downstream zones to the selected module. Note that the values depicted below the text ("00 00" in the example) indicate the contents of the Tracking Registers. Please refer to ConveyLinx ERSC PLC Developers Guide for details on how to use 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 the current state of each Sensor Port and Control Port input signals
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 downstream zone on the selected Module, it shows "Pin3 Control Port Accumulation" as the reason a carton is stopped at this zone even though the zone that is downstream of this module is empty

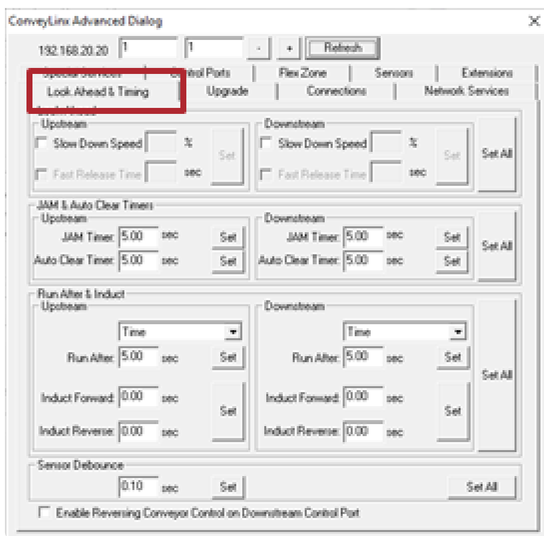


8.6. Advanced Dialog

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

8.6.1. Look Ahead and Timing Tab

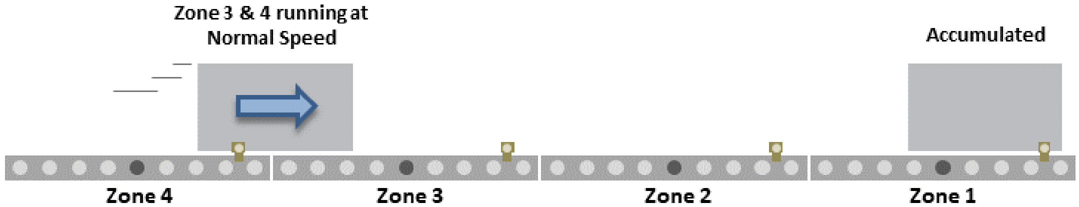
The Look Ahead and Timing Tab contains 4 sections of settings pertaining to ZPA functionality:

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

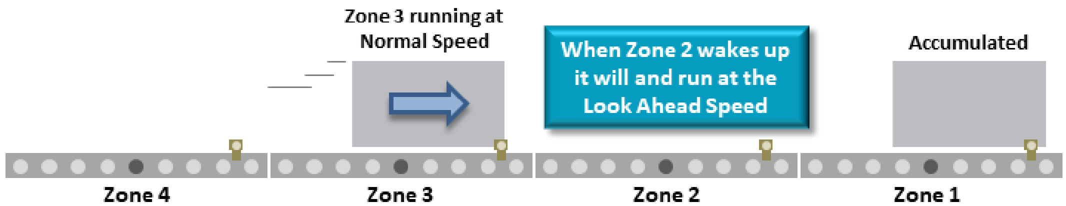
8.6.1.1. Look Ahead Slowdown Feature

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

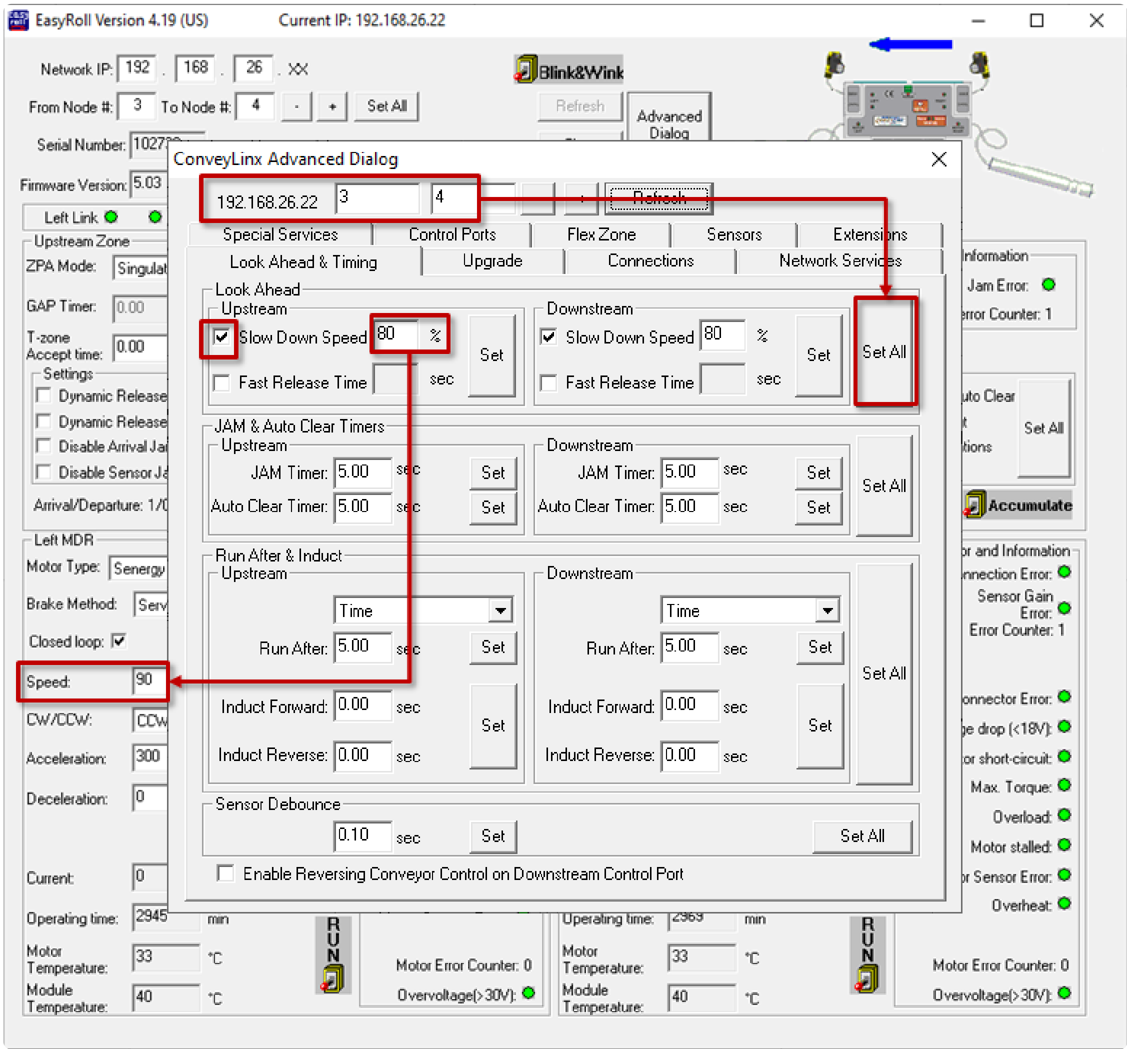
A carton is accumulated and stopped at Zone 1 and another carton is conveying at normal set speed



When carton reaches end of Zone 3, Zone 2 will wake up and run at the Look Ahead Speed



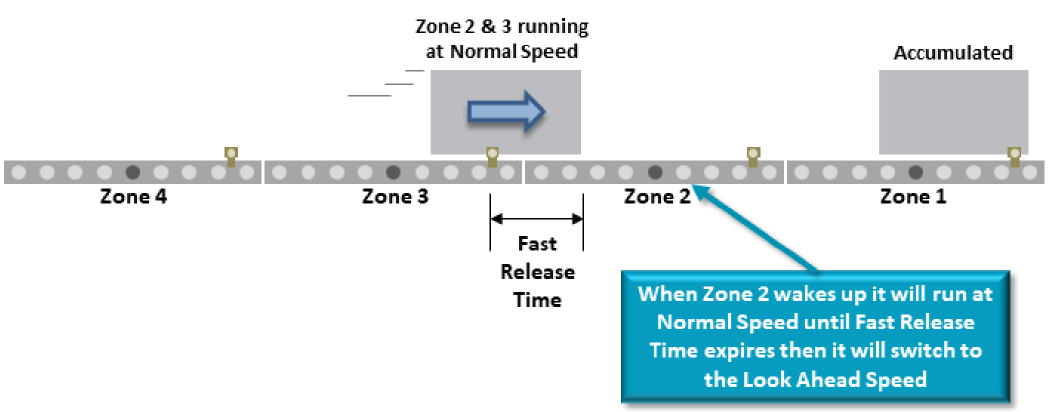
Click the checkbox to enable the *Look Ahead* feature for the selected Node. Clicking the *Set* buttons will download the setting to the respective zone on the selected Node. The value entered for the slowdown speed is in percent of the Node's normal speed as set on the main screen. In this example, the *Look Ahead* speed will be proportional to 80% of 90% PWM current. If the PWM speed on the main screen was set to 70% PWM, then the slowdown speed would be "80% of 70% PWM". Also note there is a *Set All* button that will apply the settings to the range of Nodes entered



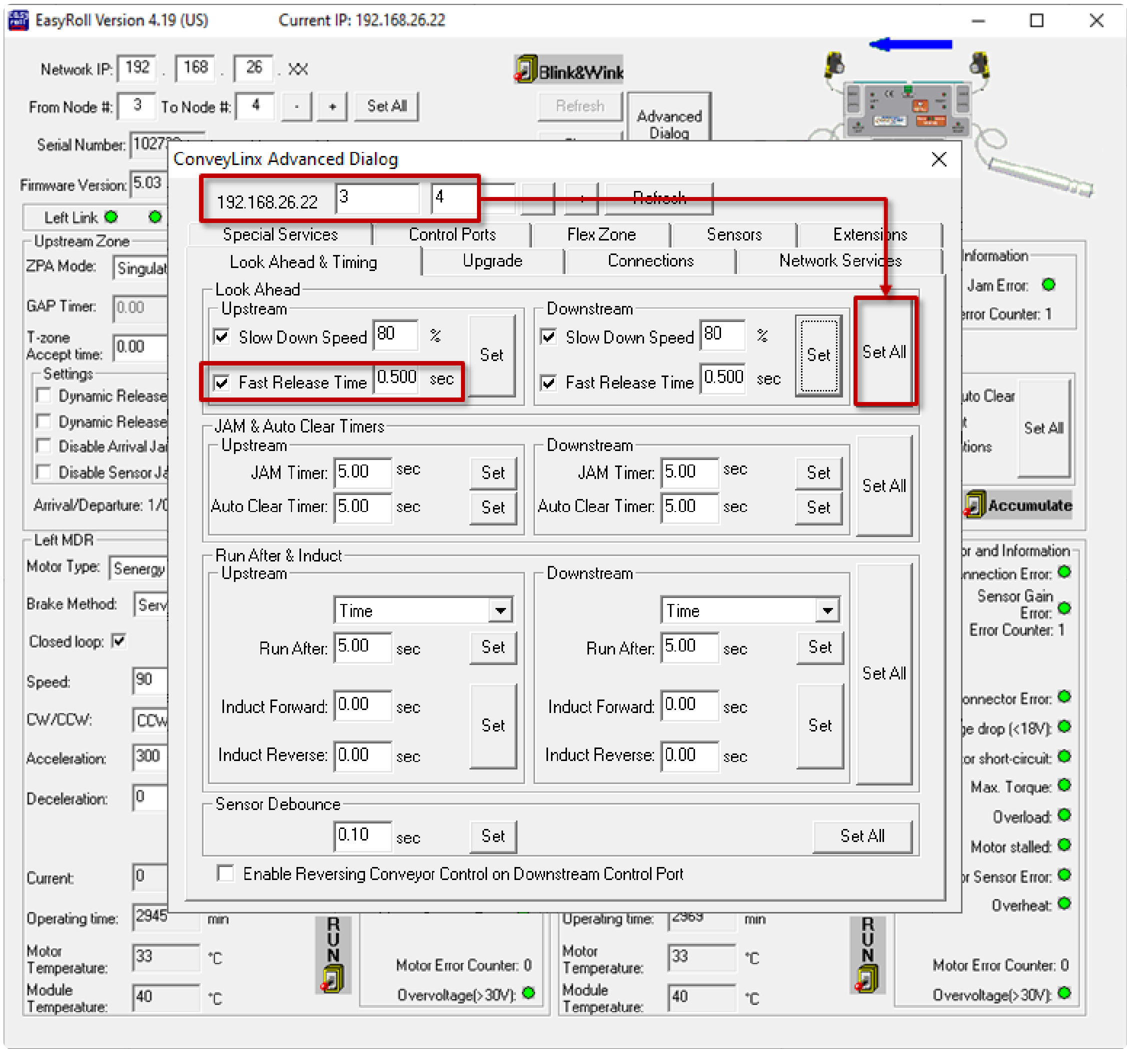
Fast Release Time

The *Fast Release Time* option allows you to set a delay before the *Look Ahead Slowdown* 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 Look Ahead Slowdown speed



To use the *Fast Response Time* option, click the *Fast Release Time checkbox* and enter a time value (in seconds) into the value 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

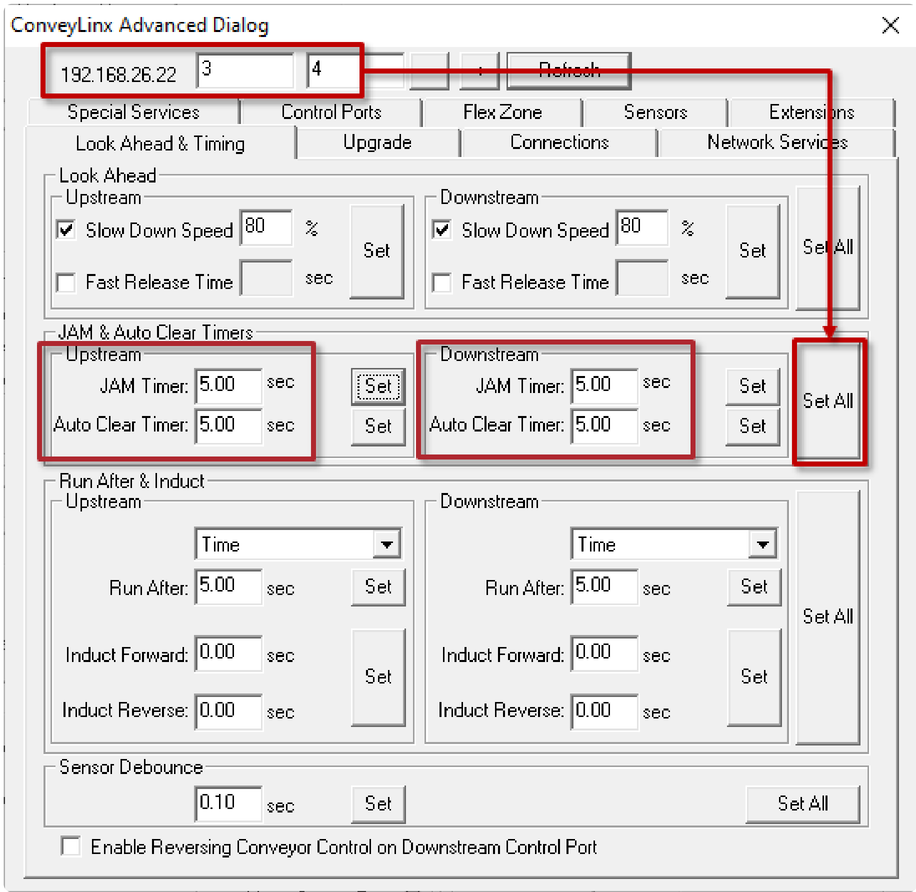


8.6.1.2. Jam Auto Clear Timers

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 times 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

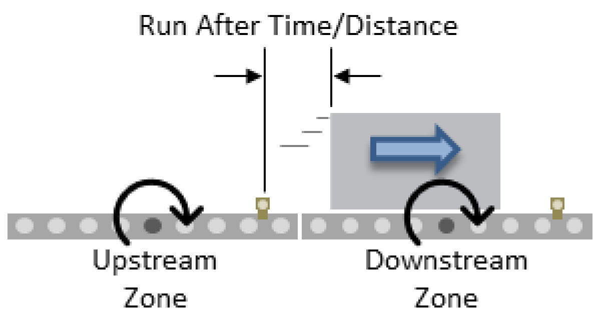
Enter new values for either or both the *Jam Timer* and/or *Auto Clear Timer* and click the corresponding *Set* button for each. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog



8.6.1.3. 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.

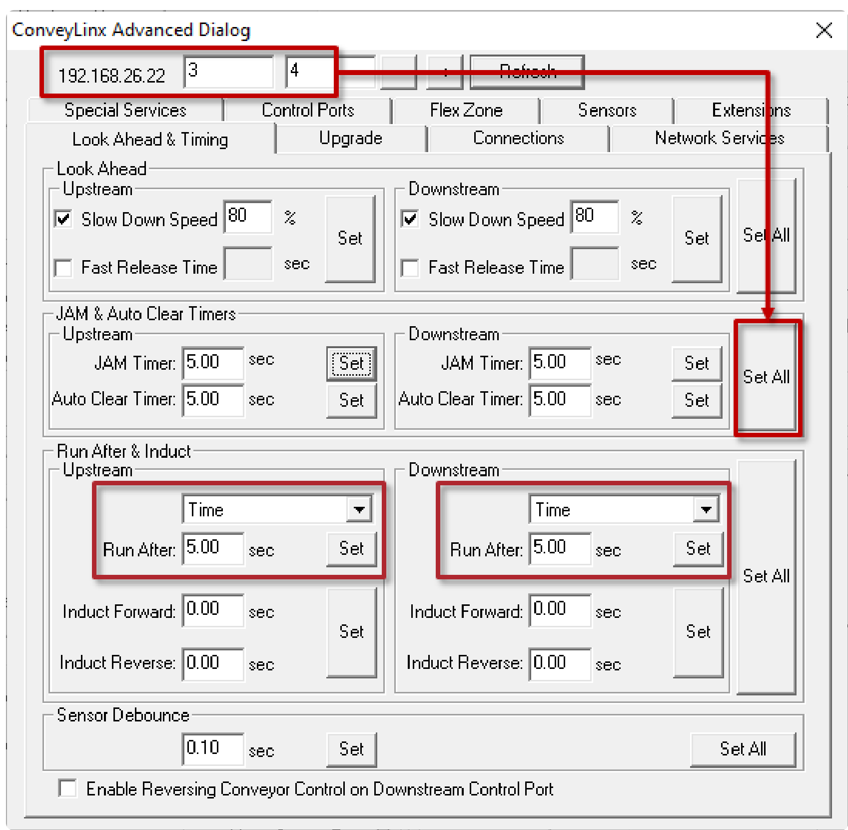
This extra run time allows the zone to run so that the trailing edge of the carton can completely pass the photo-sensor and fully enter the next zone. This value is adjustable to compensate for special conditions where a zone photo-sensor is required to be placed farther upstream or downstream



✻ Please note that Run After does not affect throughput rate. A longer Run After time will not prevent an upstream item from entering the zone.

Run After Time

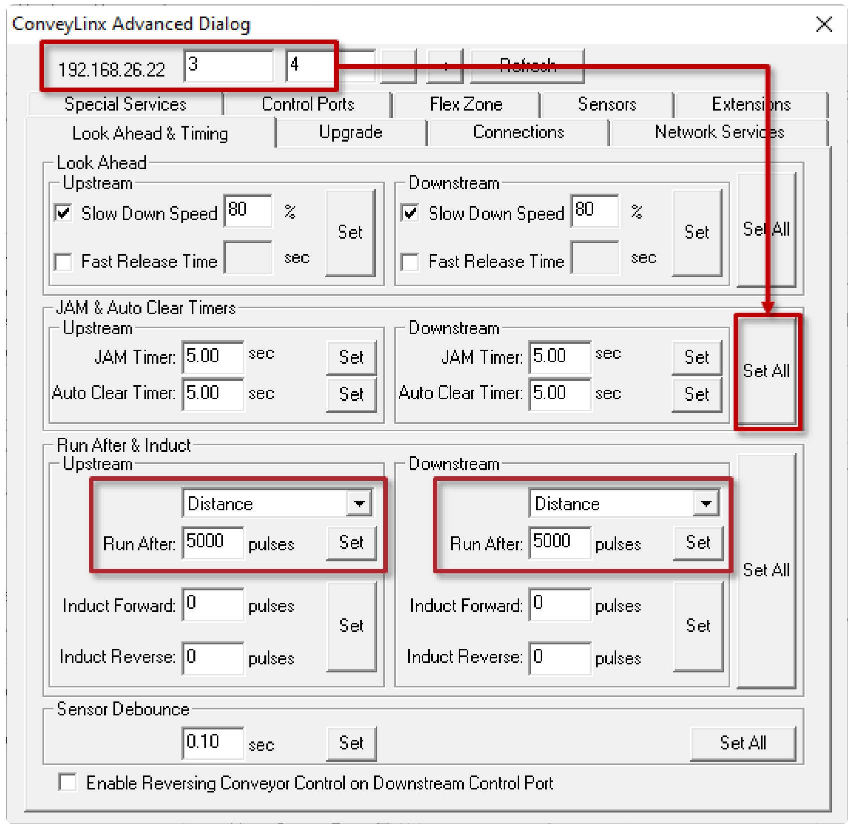
Enter new values for upstream and/or downstream *Run After* time and click the corresponding *Set* button. The default is 5 seconds and the valid range is 0 to 65 seconds. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog



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 motor pulses instead of seconds

Select *Distance* from the drop-down box and enter new values for upstream and/or downstream *Run After* pulses and click the corresponding *Set* button. The default is 5000 pulses 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

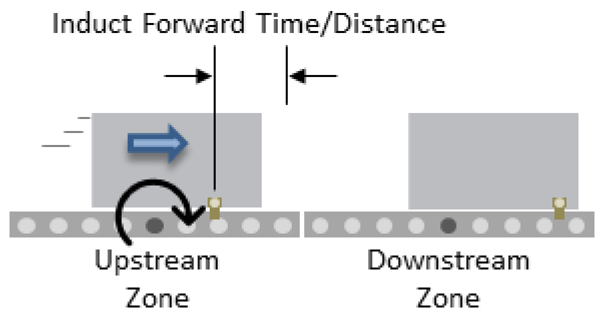


See Motor Pulse to Distance Calculation for details on figuring out the distance traveled

8.6.1.4. Induct Forward Time/Distance

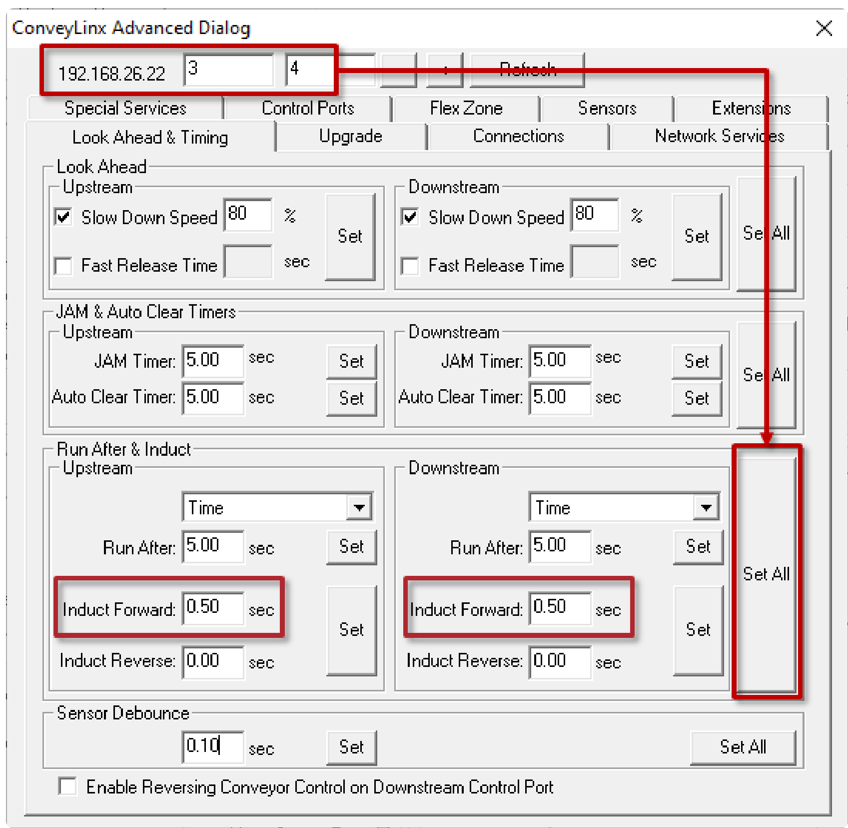
Induct Forward value is used to cause the MDR to continue to run after the zone's photo-sensor has been blocked when receiving a carton from upstream.

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



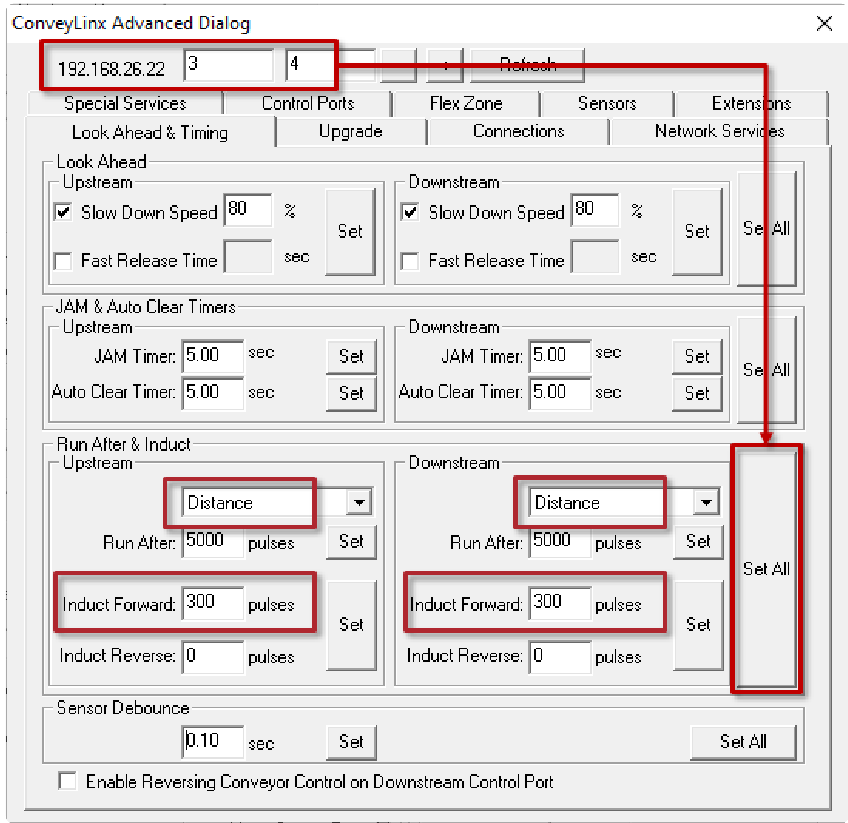
Induct Forward Time

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. In opur example we entered 0.5 seconds. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog



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 pulses and the valid range is 0 to 65,535 pulses. In our example we entered 300 pulses. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog



See <u>Motor Pulse to Distance Calculation</u> for details on figuring out the distance traveled

8.6.1.5. Induct Reverse Time/Distance

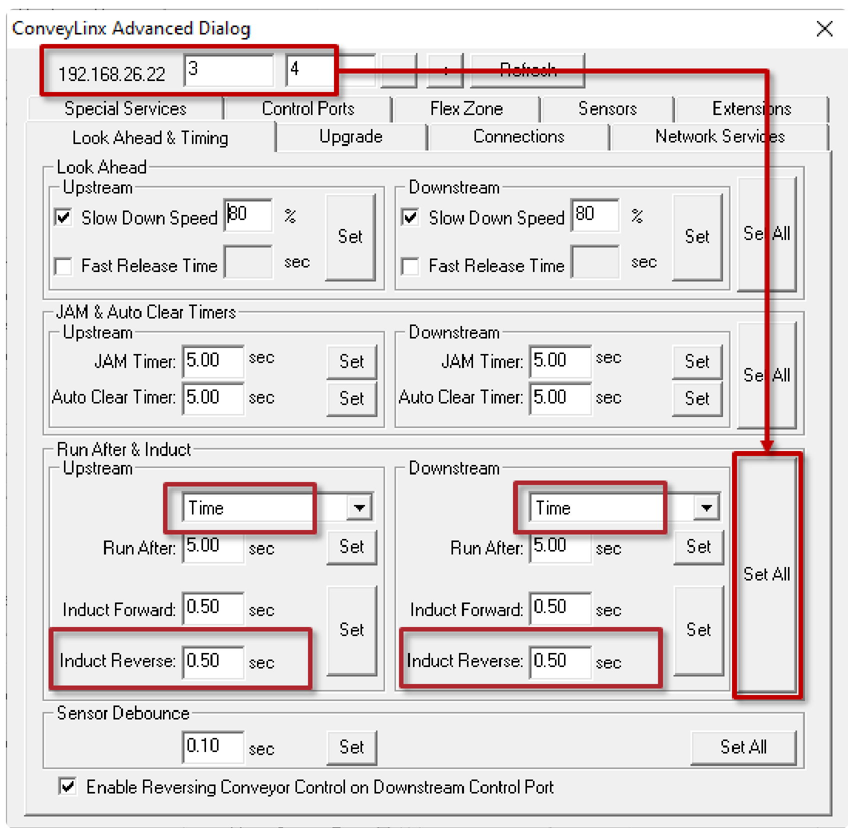
Induct Reverse is only applicable for Reversing Control in ZPA Mode

Induct Reverse value is used in conjunction with Induct Forward when you are setting up Reversing Conveyor in ZPA Mode to cause the MDR to continue to run after the zone's photo-sensor has been blocked when receiving a carton from upstream with reference to the "reverse" direction.

Typically for *Reversing Control*, the Sensor is mounted in the center of the zone and the *Induct Reverse Time/Distance* is equal to the *Induct Forward Time/Distance*. These are adjusted separately so you can compensate for situations when you cannot put the Sensor in the center of the zone

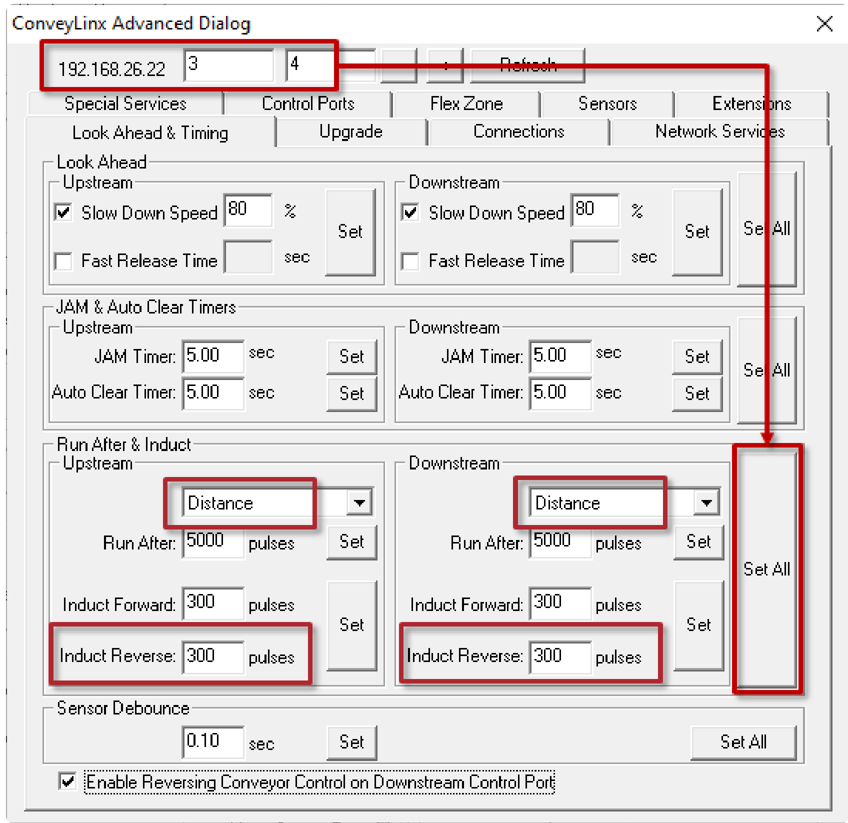
Induct Reverse Time

Enter new values for upstream and/or downstream *Induct Reverse* time and click the corresponding *Set* button. The default is 0 seconds and the valid range is 0 to 65 seconds. In our example we entered 0.5 seconds. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog



Induct Reverse Distance

Select Distance from the drop down box and enter new values for upstream and/or downstream *Induct Reverse* pulses and click the corresponding *Set* button. The default is 0 pulses and the valid range is 0 to 65,535 pulses. In our example we entered 300 pulses. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog

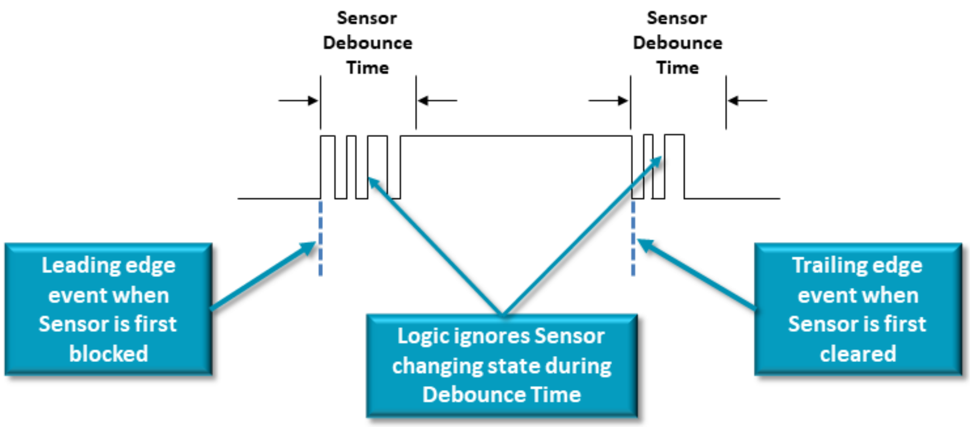


✻ See Motor Pulse to Distance Calculation for details on figuring out the distance traveled

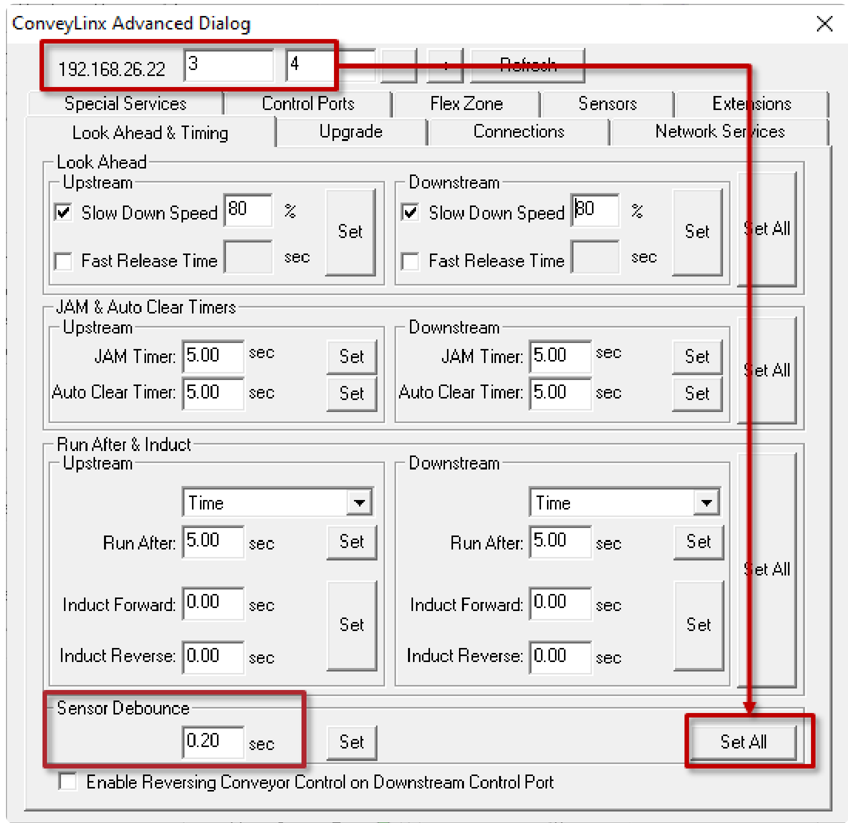
8.6.1.6. Sensor Debounce

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 **not a delay prior to detecting a carton** 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



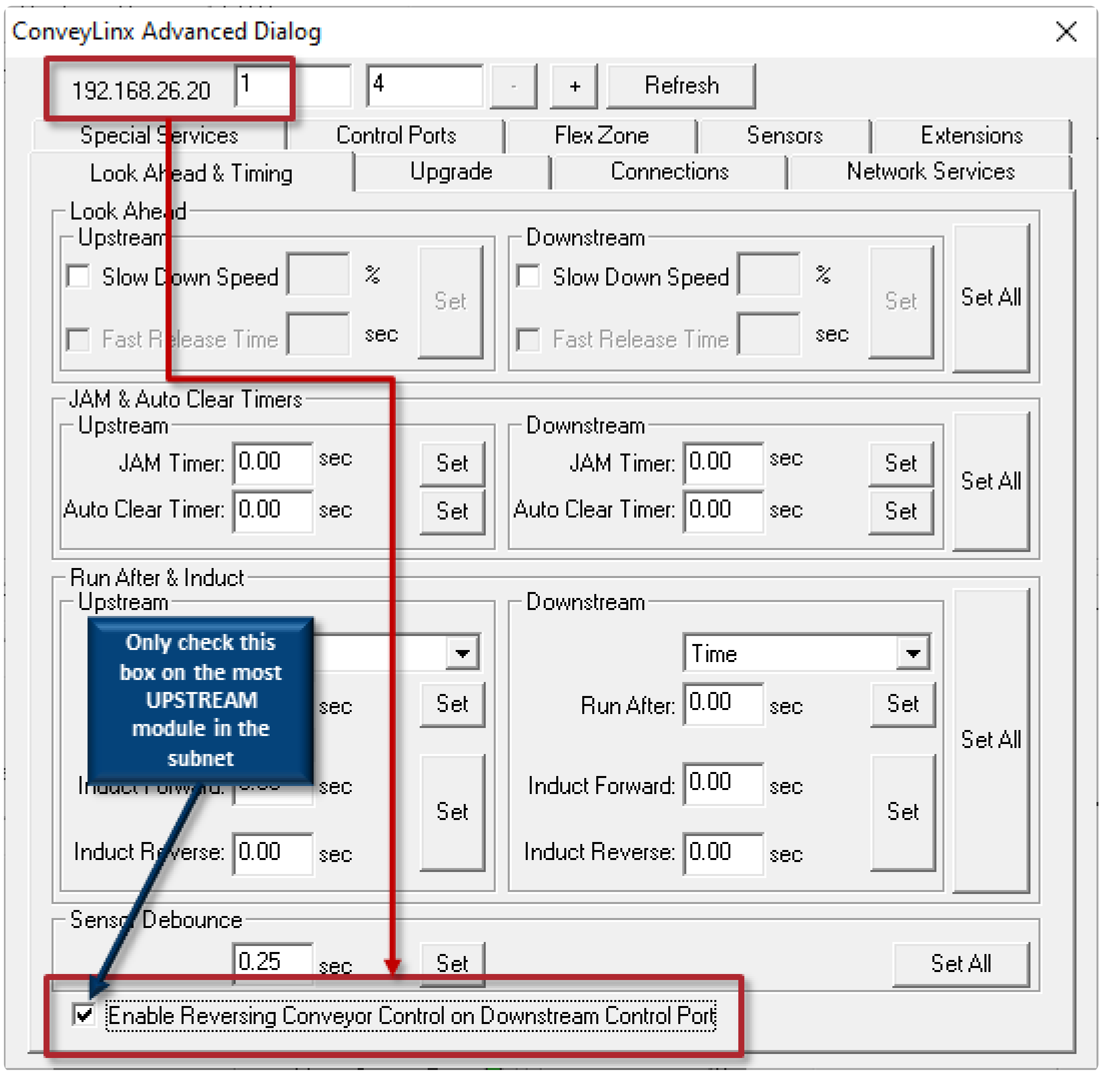
Enter the desired value in seconds and click the *Set* button. The default is 0.1 seconds and the valid range is 0 to 2 seconds. Please note this setting applies to **both** the **Left and Right Sensor** ports. In our example we entered 0.2 seconds. The *Set All* button will apply these same settings for the range of modules indicated at the top of the dialog



8.6.1.7. Reversing Control in ZPA Mode

There is a checkbox on the *Look Ahead & Timing* tab for *Enable Reversing Conveyor Control on Downstream Control Port* located at the bottom of the dialog. Reversing conveyor control applies to an entire subnet of modules and the Control Port used for this functionality is *always assigned to the Downstream zone of the* ***most Upstream module***.

This checkbox should only be checked on the module that is the **most upstream in a subnet**



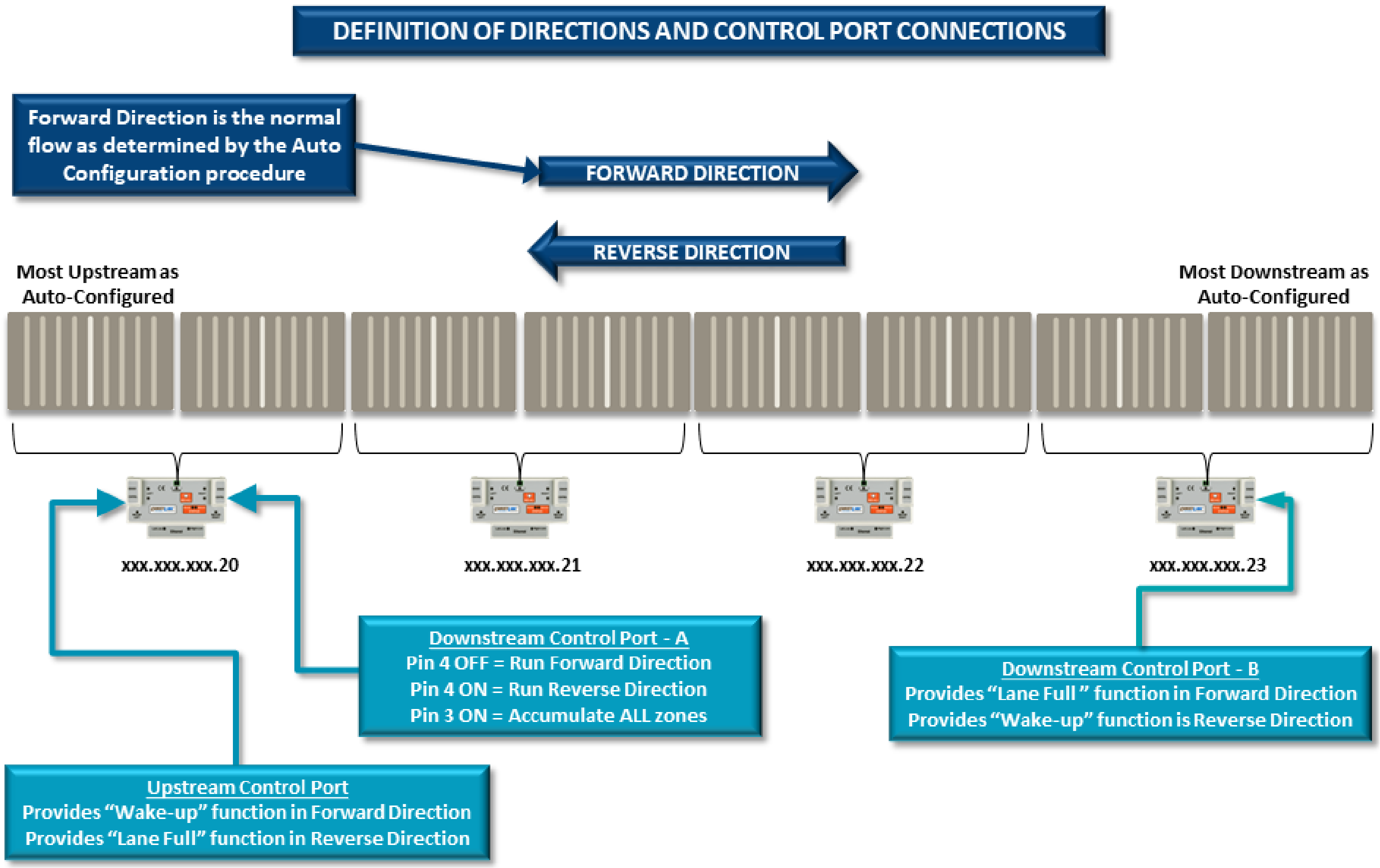
Practical Prerequisites for Reversing ZPA Control

- Sensors should be located very close to or in the center of each zone
- **Length** of all products should be **identical** or very **close to identical**
- *Induct Forward* and *Induct Reverse* need to be set

Reversing Control Example

Reversing conveyor utilizing hard-wired signal to Control Ports is enabled in EasyRoll as indicated in section Reversing Control on page 91. Figure 61 depicts a simple 8 zone example and indicates which signals on which Control Ports will provide the control. Reversing function applies to all modules in a subnet will work for any number of zones up to the maximum that can be included in a single subnet.

> Please note that you can utilize reversing control for a portion of a subnet by utilizing a remote PLC via the Ethernet network. See how to use a PLC for reversing ZPA control



Sequence of Operation

Forward Direction

- All signals are OFF on Downstream Control Port – A of the xxx.xxx.xxx.20 module
- You can use the Upstream Control Port to wake up the most upstream zone
- You can use the Downstream Control Port-B to hold and accumulate the most downstream zone

Switching to Reverse Direction

Energize Pin 3 on Downstream Control Port-A to cause all zones in the subnet to accumulate Wait for all zones to stop and accumulate (time value dependent upon zone length and speed – 2 or 3 seconds is typical) Energize Pin 4 on Downstream Control Port-A to logically switch the direction to Reverse De-energize Pin 3 on the Downstream Control Port-A to remove accumulation for all zones in subnet to start reversing operation Note that the Upstream Control Port and Downstream Control Port-B have also reversed their respective functions. Upstream Control Port now provides the Downstream Interlock (lane full) function and the Downstream Control Port-B provides the Upstream Interlock (wake-up) function

Switching Back to Forward Direction

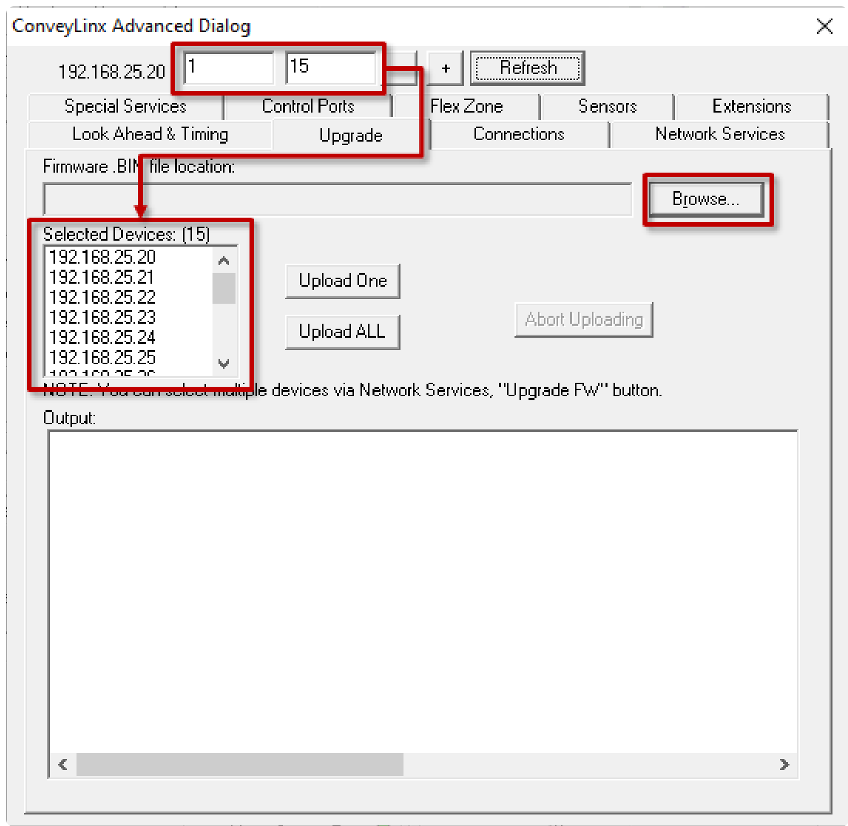
Energize Pin 3 on Downstream Control Port-A to cause all zones in the subnet to accumulate Wait for all zones to stop and accumulate (time value dependent upon zone length and speed – 2 or 3 seconds is typical) De-energize Pin 4 on Downstream Control Port-A to logically switch the direction to Forward De-energize Pin 3 on the Downstream Control Port-A to remove accumulation for all zones in subnet to start forward operation

8.6.2. Upgrade Tab

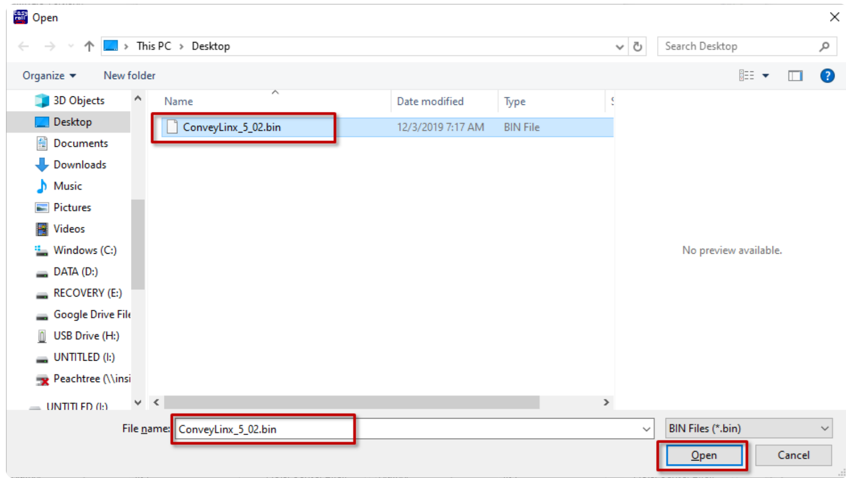
The *Upgrade* tab screen gives access to the module firmware utility. Over time, enhancements and features may be added to the ConveyLinx family of products. These features and enhancements are typically made available to customers in the form of firmware upgrade files that need to be uploaded to your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

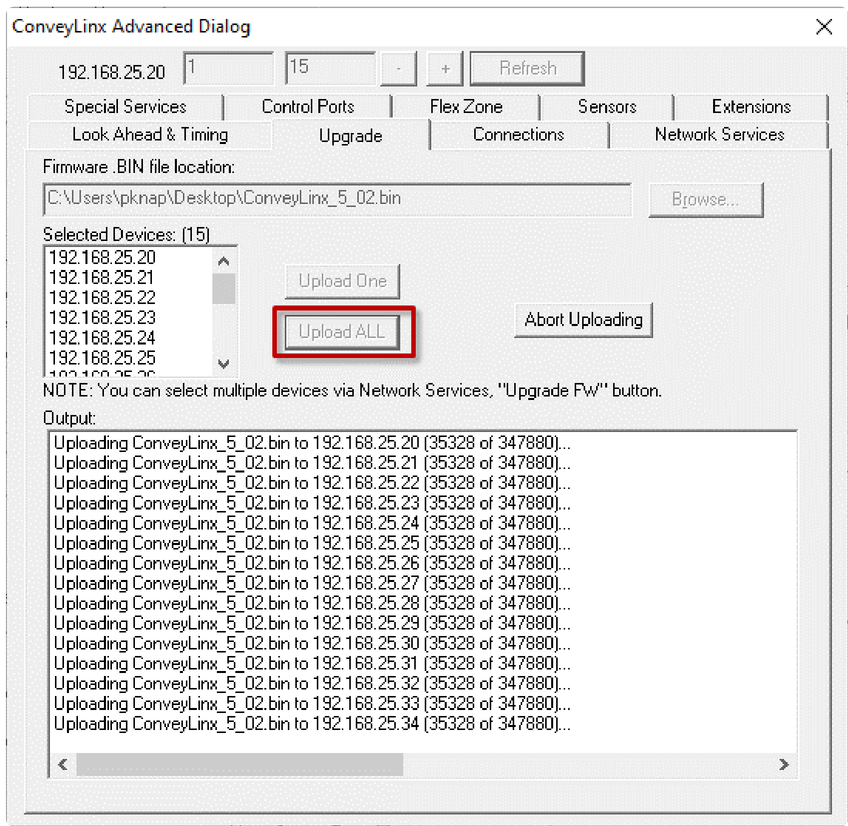
Upon selecting the *Upgrade* screen tab, *EasyRoll* fills in the I.P. address of the range of Nodes entered on the main screen. Click the *Browse* button to open a file selection dialog window



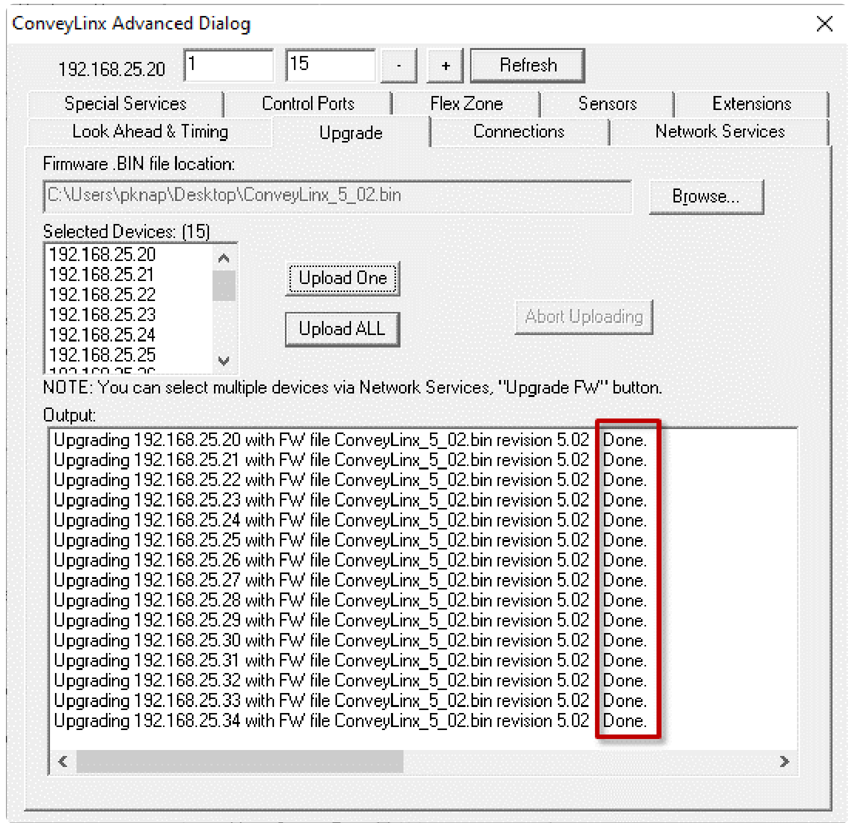
With the *Open* dialog displayed, navigate to the location on your PC where you placed the firmware upgrade file you received. Select the file and click *Open*



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



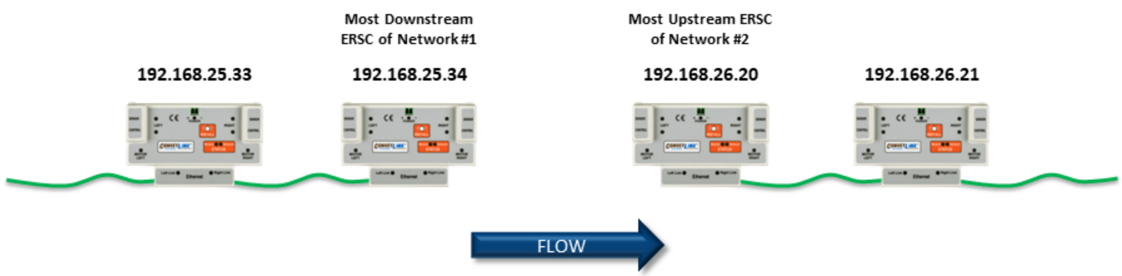
Please note that you typically arrive at the *Upgrade* tab from the ***Network Services*** tab where you select the modules first and click the *Upgrade* button there. That will automatically take you to the *Upgrade* tab and populate the *Selected Devices* list box on the *Upgrade* tab with your selection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

8.6.3. Connections Tab

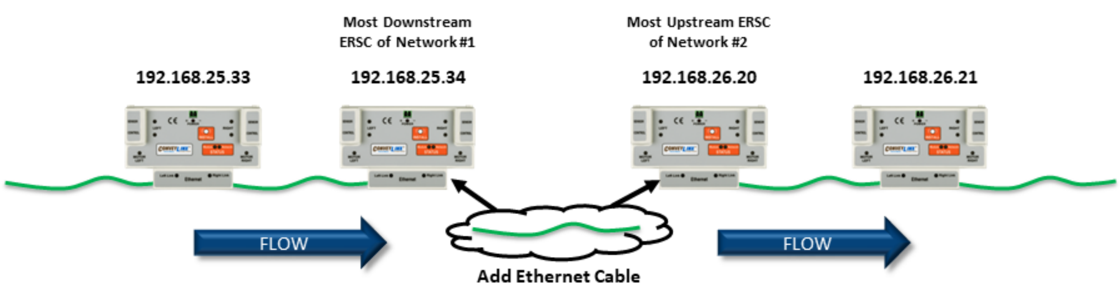
The *Connections* utility uses *EasyRoll* to instruct a given module to make a logical connection to another module that it otherwise would not have made during the *Auto-Configuration Procedure*. For applications where you have more than one *Subnet*, this would be the way to logically connect the most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

Connecting Two Subnets Together

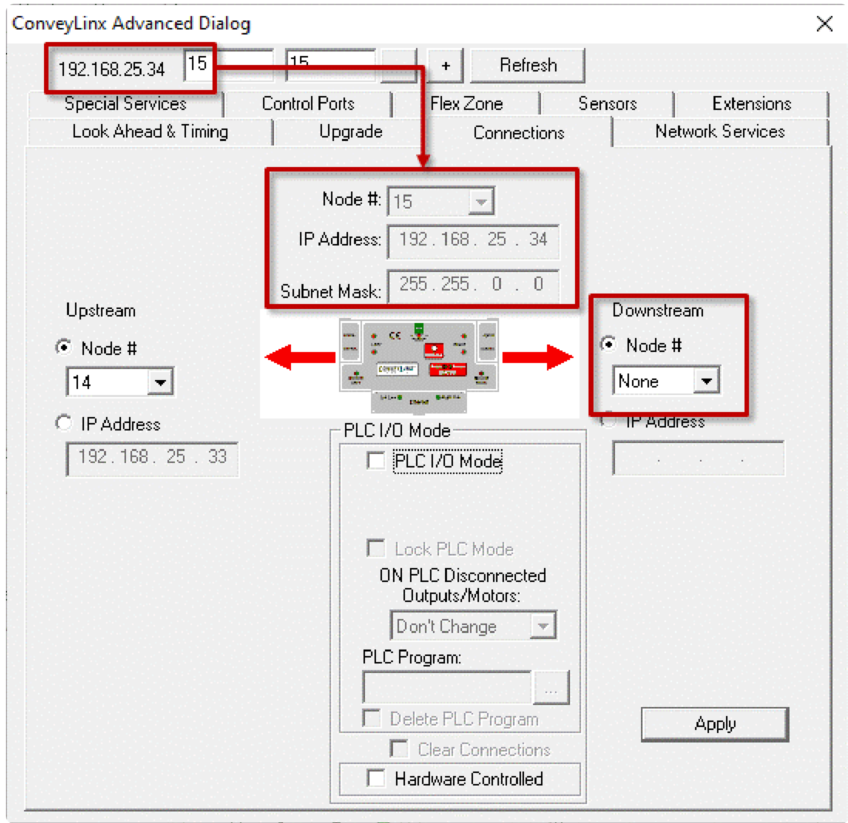
Here is a typical boundary between two *Subnets*. The most downstream Node of the first *Subnet* has an I.P. address of 192.168.25.34 and the most upstream Node of the second *Subnet* has an I.P. address of 192.168.26.20.



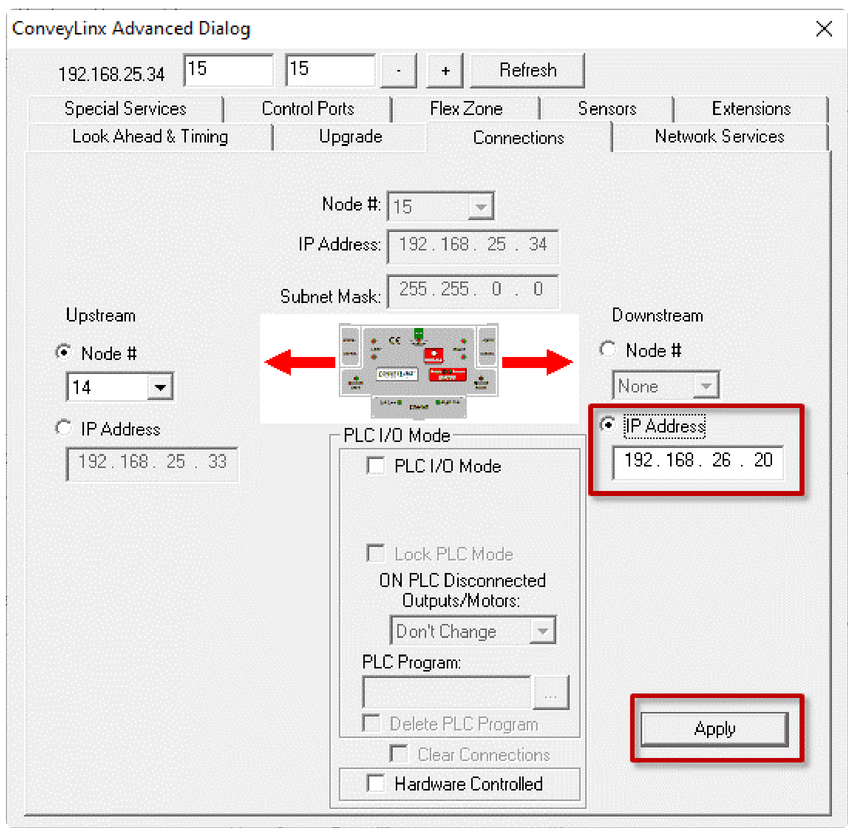
By simply connecting an Ethernet cable between these two boundary Nodes and then using *EasyRoll* to establish the “logical” connection between the two *Subnets*; you can achieve seamless flow between the two networks. The procedure requires that you have to instruct Node at 192.168.25.34 to convey cartons to Node at 192.168.26.20, and likewise you have to instruct Node at 192.168.26.20 to accept cartons from Node at 192.168.25.34



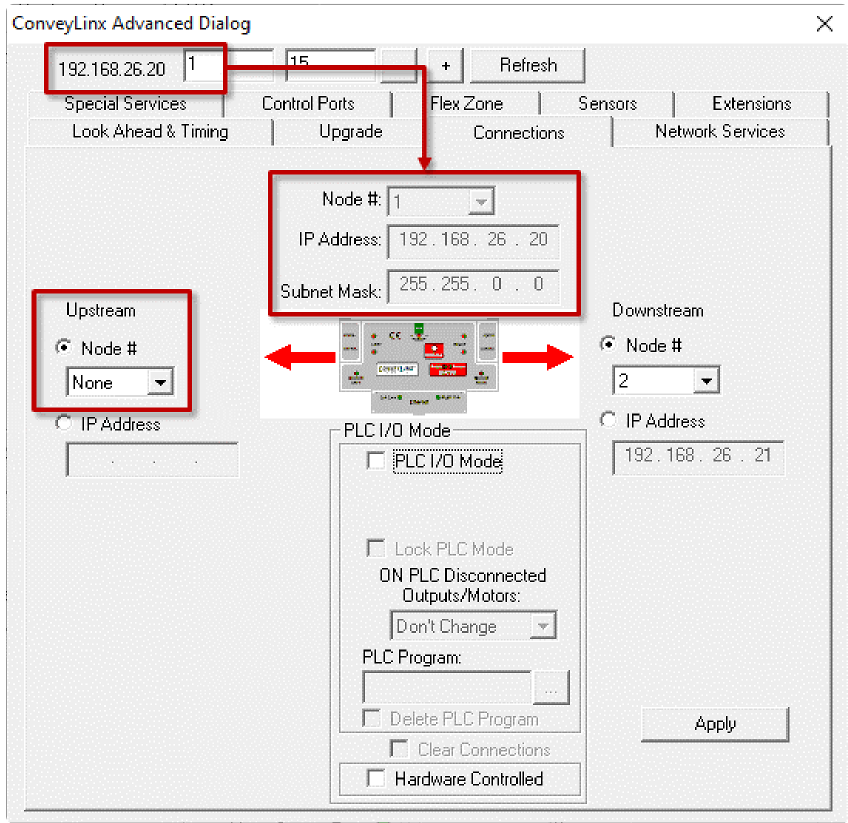
Navigate to Node 15 of the 192.168.25 subnet (i.e. 192.168.20.34). Notice that the module’s network data appears in the center. Also notice that the Downstream connection for this Node is *None*



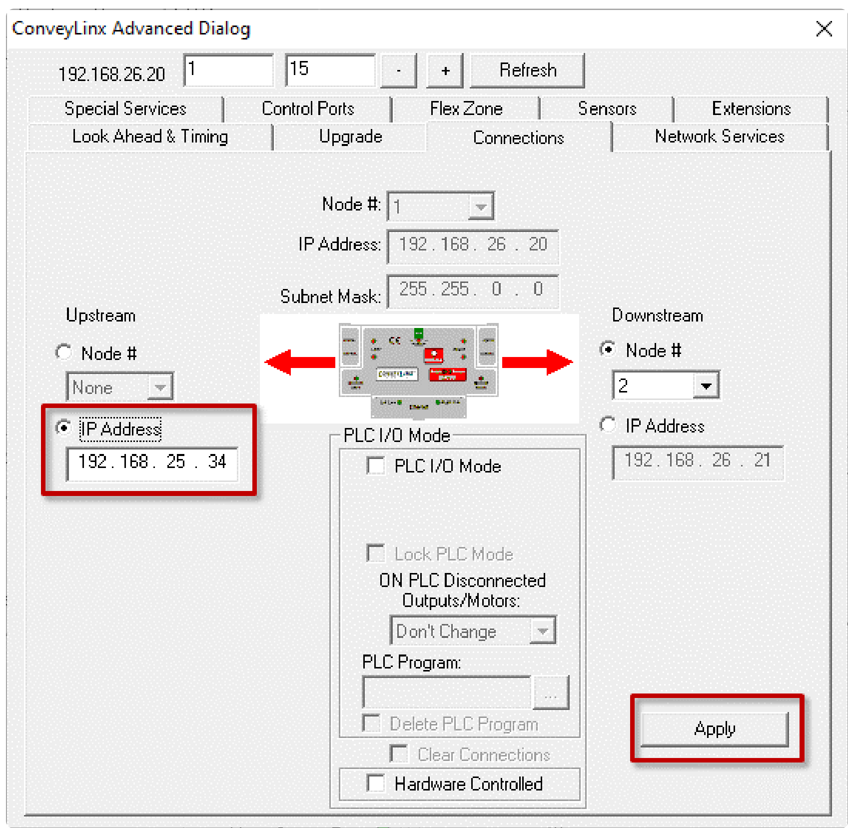
Enter *192.168.26.20* as the I.P. address for Node 15’s new Downstream connection. Click *Apply* to make the change. Please note that it will take a few seconds for this to complete



Navigate to Node 1 of the 192.168.26 subnet (i.e. 192.168.26.20). Notice that the module's network data appears in the center. Also notice that the Upstream connection for this Node is *None*



Enter *192.168.25.34* 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



8.6.4. Network Services Tab

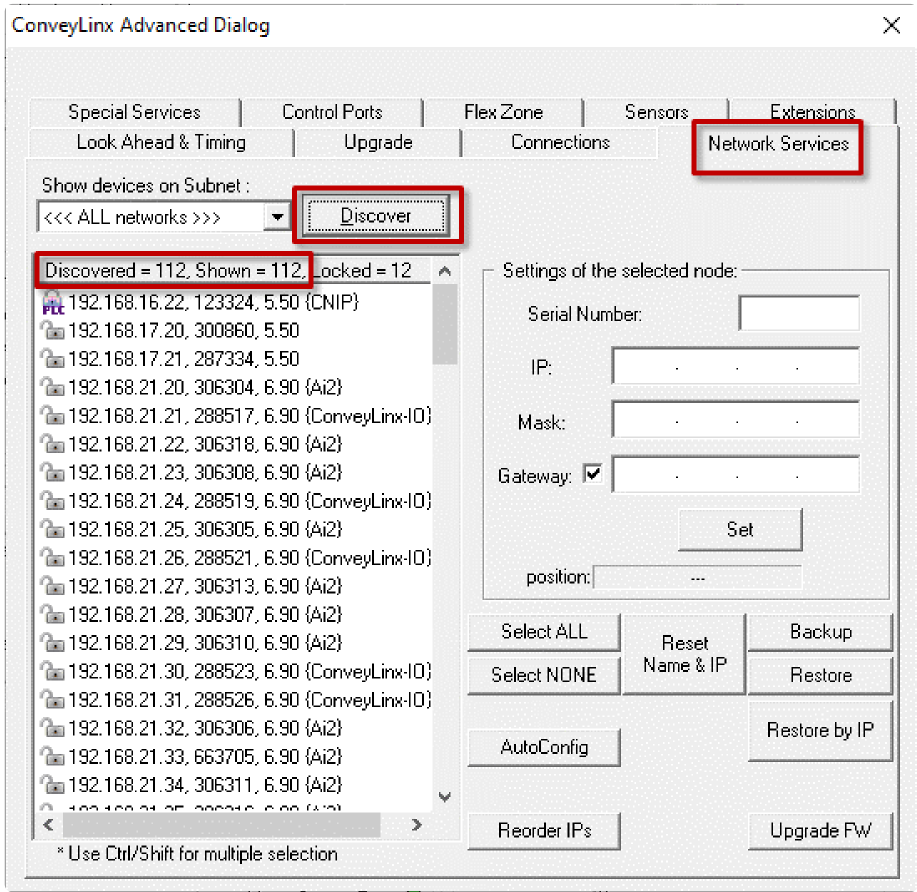
The Network Services screen provides multiple functions related to module and network management. These functions are:

- Discover and IP Address Set
- Network Lock/Unlock
- Backup and Restore
- Firmware Upgrade

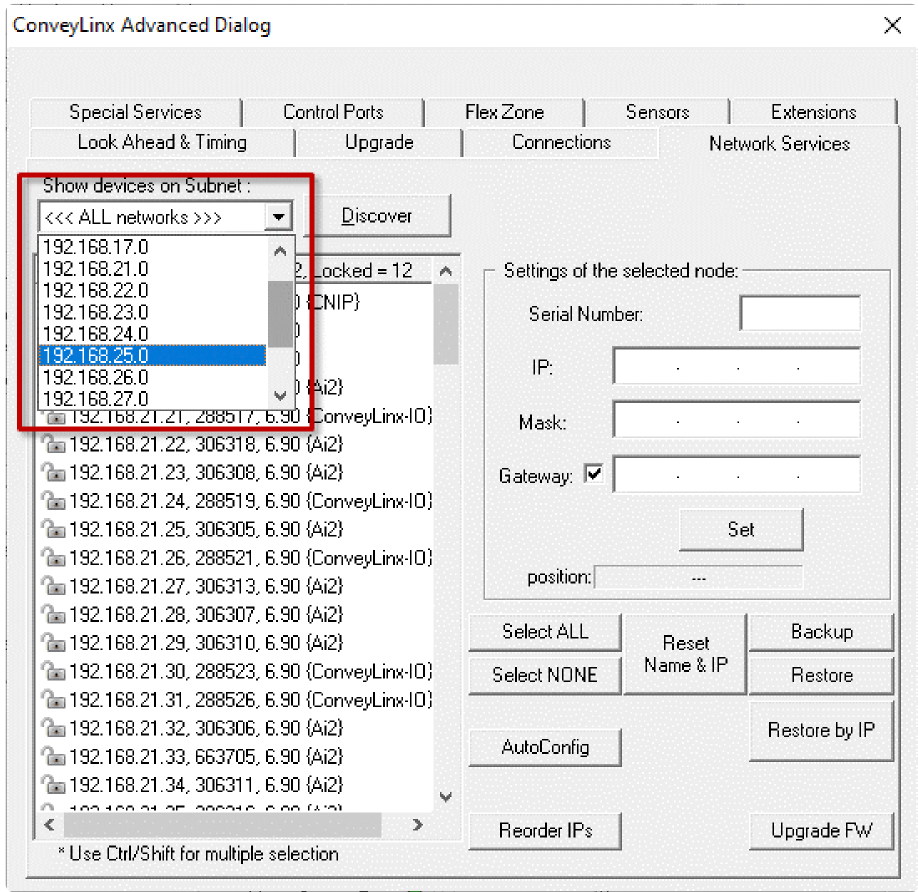
8.6.4.1. Discover and IP Address Set

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

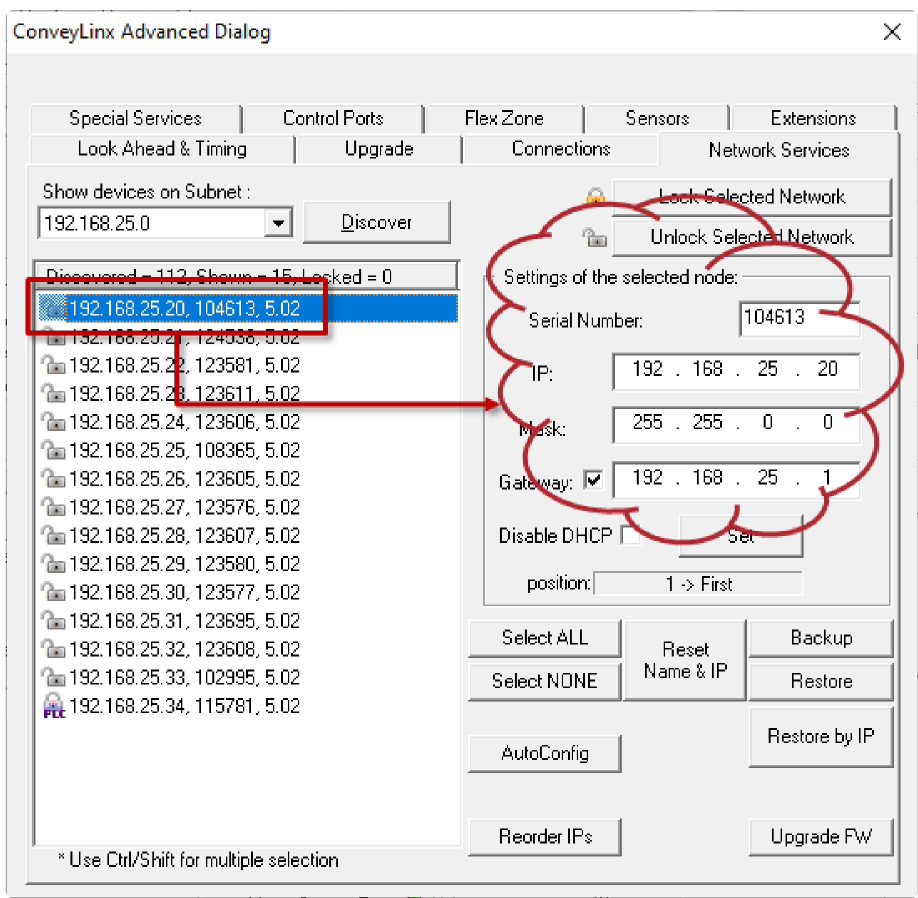
Clicking the *Discover* button will cause the list box to populate with all the modules in all *Subnets* that your PC can see out on the network. In this example you can see that 112 Nodes were discovered.



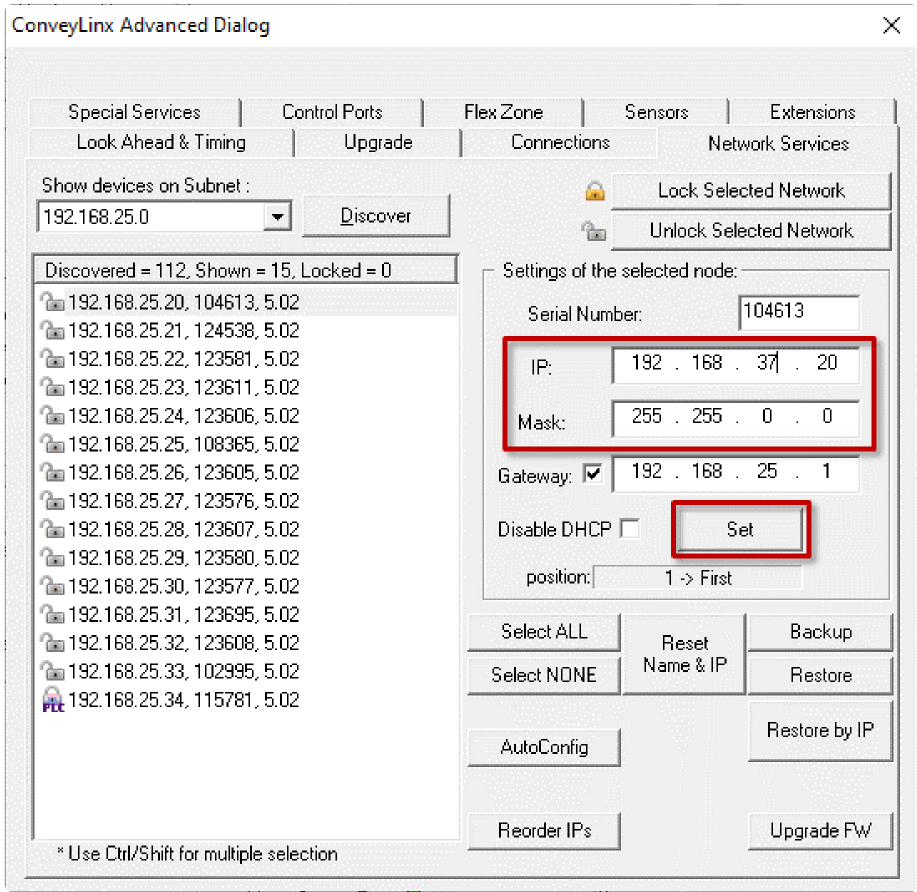
The drop-down box at the top of the list box lets you filter the list by *Subnet*. In this example we are selecting the 192.168.25 Subnet.



Double-click any one of the line items in the list to display its properties. In this example we double clicked the first item to show its properties data.



Once a modules data has been selected, you can change its properties. In this example we changed the 3rd octet of the I.P. address from 25 to 37. Once you have edited the data, click the *Set* button and the changes are sent to the module.



8.6.4.2. Position and DHCP

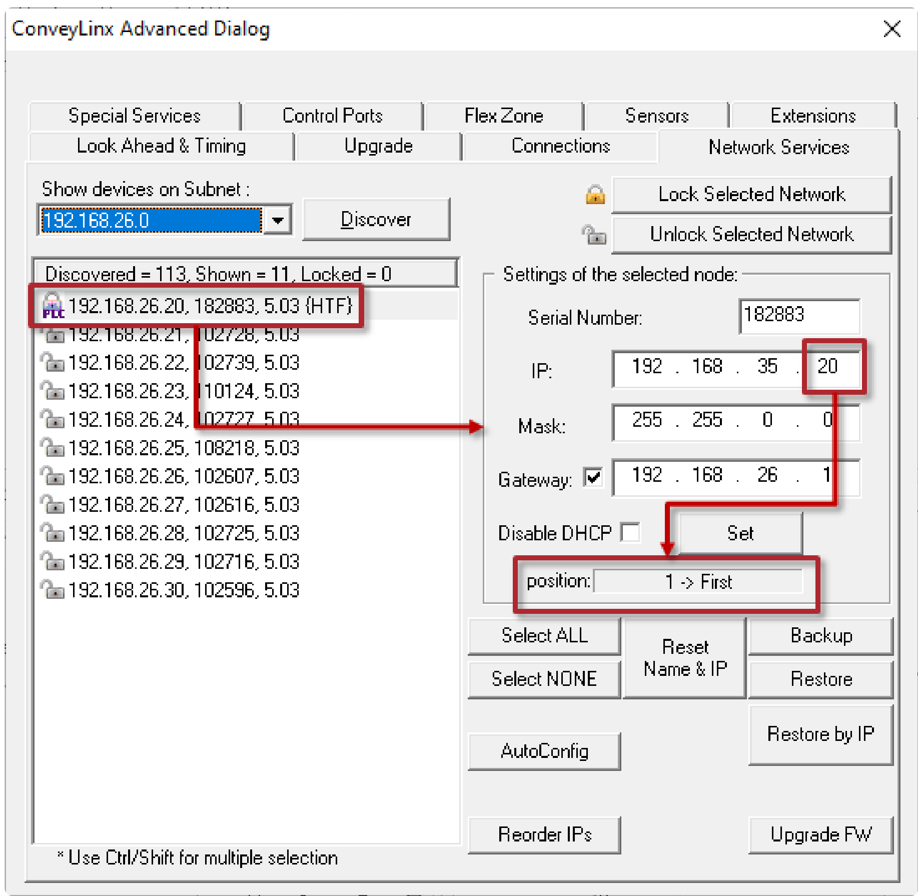
Position Identifier

Within the “Settings of the selected node” area, at the bottom, is a read-only box that shows the position of the selected module within the network. This information is provided so you can check the integrity of your Subnet. The data shown is a position indicator or *Flag*. The possible values are *1 -> First*, *2 -> Last*, and *0 -> unknown*.

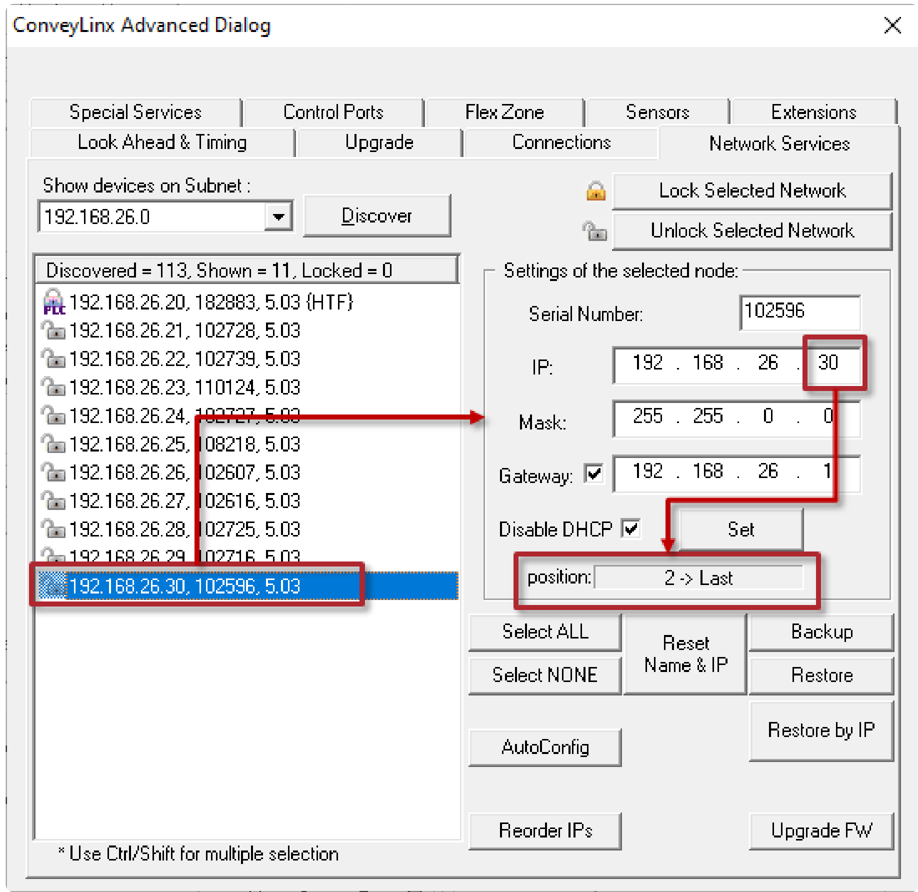
For a properly configured *Subnet* you must have:

- Only one Module with a *1 -> First* position *Flag*
- Only one Module with a *2 -> Last* position *Flag*
- All modules with the exception of the First and Last should indicate *0 -> unknown*

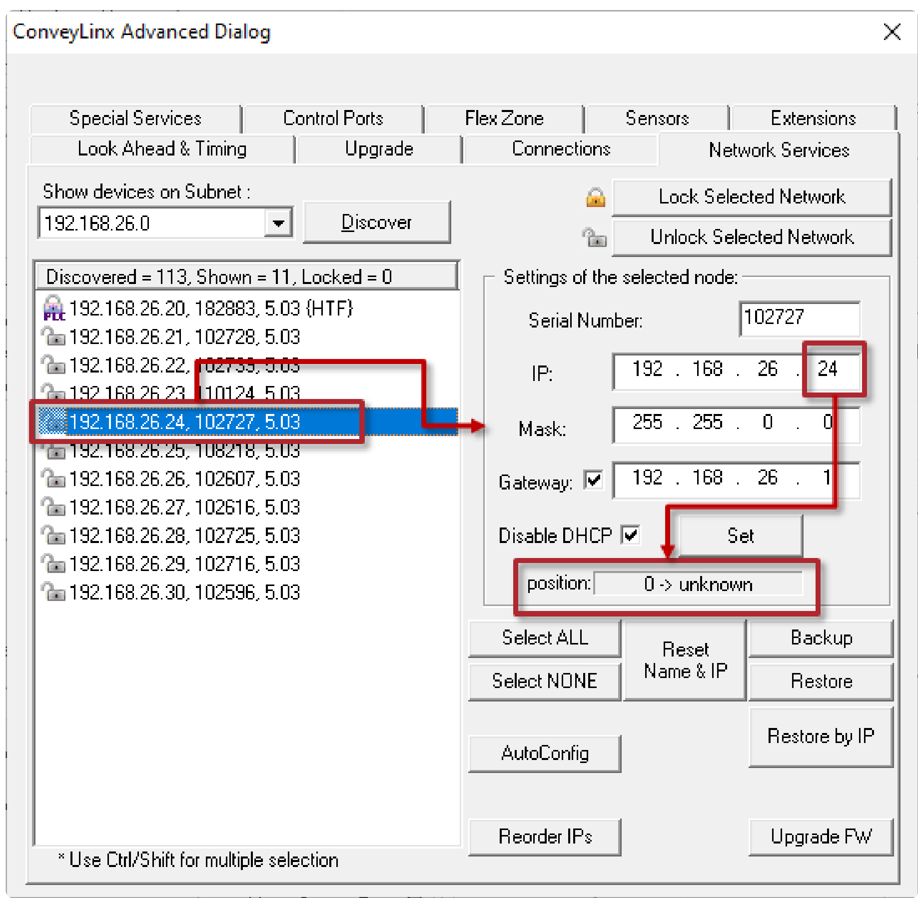
With your list filtered so that it shows only one *Subnet* , the first item in the list should be the .20 module of the *Subnet* and its position *Flag* should always indicate *1 -> First*. In our example we selected the 192.168.26.20 node and its *Flag* is shown as_1 -> First_.



The last module in the *Subnet* should have a position *Flag* value of *2 -> Last*. In our example we selected module 192.168.26.30 and we can see its position *Flag* is *2 -> Last* as shown.



Any module that is not the First or Last module in the *Subnet* should have a *Flag* value of *0 -> unknown*. In our example we selected module 192.168.26.24 and we can see its position *Flag* is *0 -> unknown* as shown.

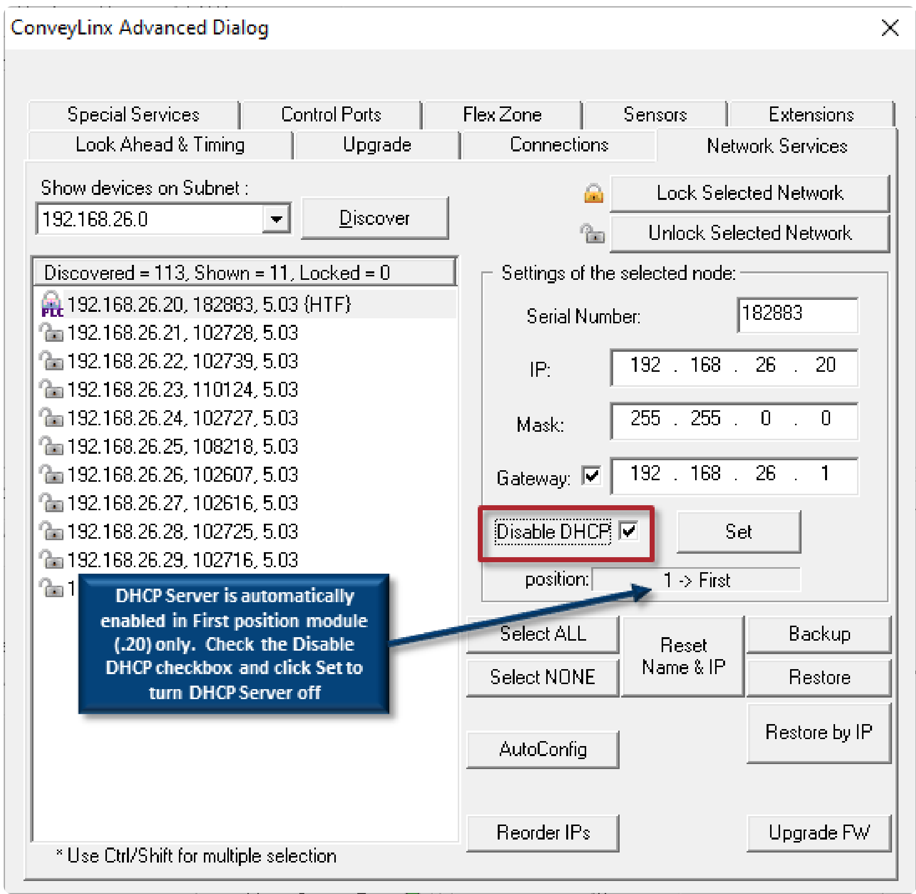


> **!** If you have more than one module in a Subnet with a First or Last Flag, or if there is no First or Last module in the Subnet, your network will not properly preserve its data for Automatic Module Replacement and the Subnet may have problems negotiating Ethernet switches and routers for PLC connectivity. You MUST properly Auto-Configure all Subnets so that these position Flags are correct!

DHCP Server

By default, the most upstream module of every Subnet (I.P. Address xxx.xxx.xxx.20) enables a simple DHCP server as a convenience so that if your PC is set up to receive an IP address from the network, your ConveyLinx Subnet will issue one. On larger systems with perhaps multiple Subnets and/or customer managed network infrastructures, this DHCP server may interfere with the function of the network. You can disable this DHCP server from the Network Services tab.

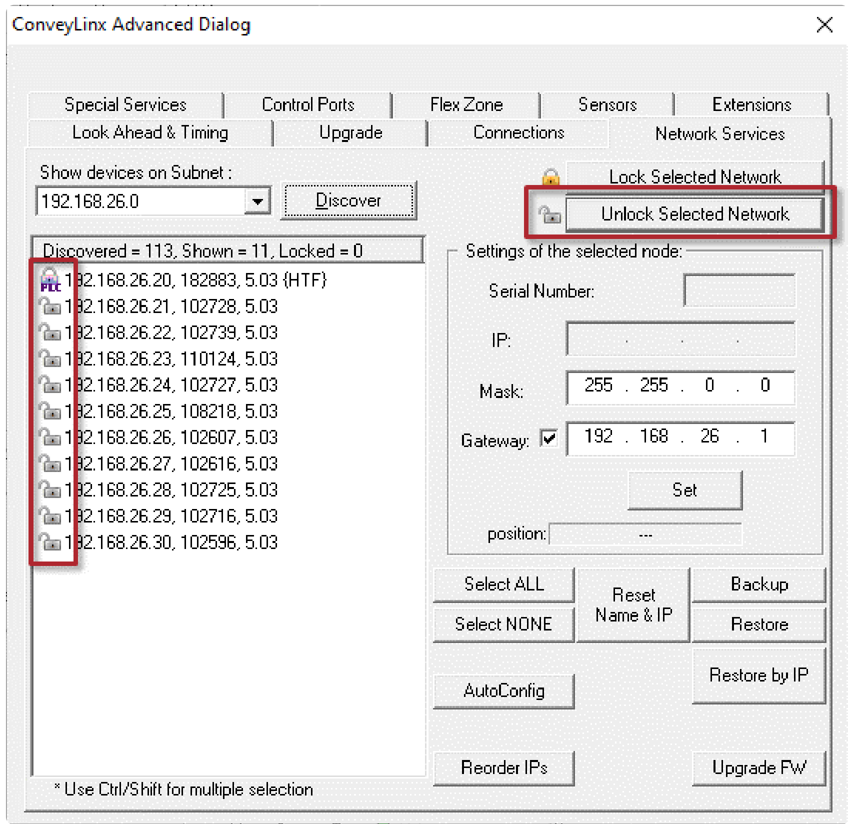
Once you have double clicked the module form the list, you can check the *Disable DHCP* checkbox to disable the DHCP server. You have to click the *Set* button to make the change to the module.



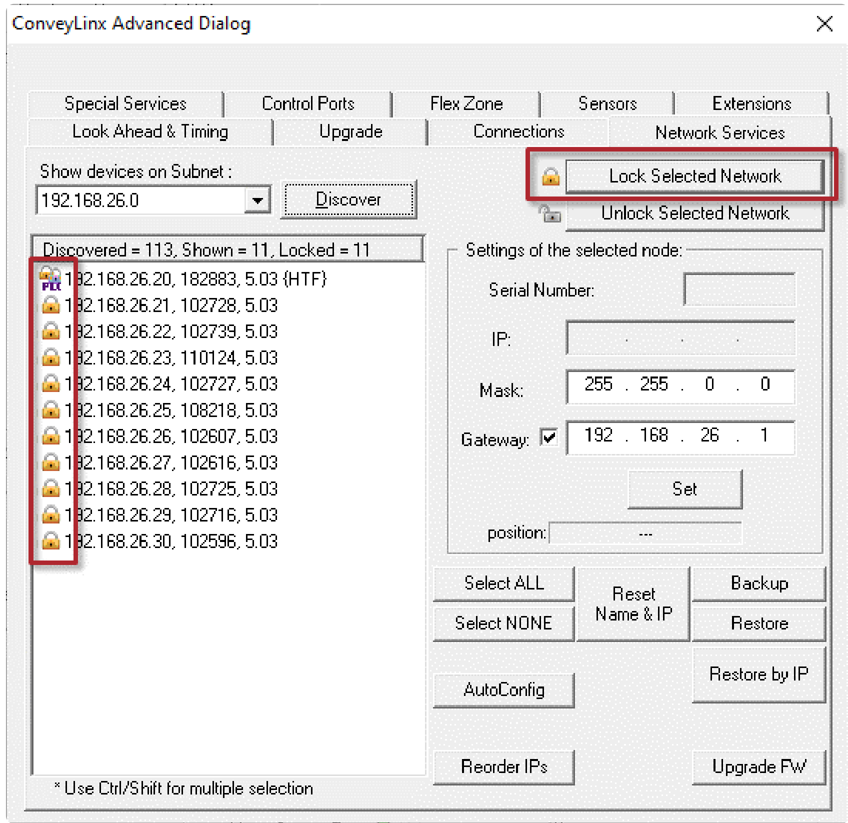
8.6.4.3. Network Lock Feature

The *Lock Feature* allows you to “lock” a specific *Subnet* so that no accidental pressing of the Install button on any of the modules will inadvertently cause an *Auto Configuration Procedure* to be activated.

In this example, the 192.168.29 **Subnet** is selected and you can see by the icons next to each module entry in the list that they are “Unlocked”



By clicking the *Lock Selected Network* button, all modules in the Subnet listed will be "Locked" as indicated by the icons in the list. Clicking the *Unlock Selected Network* button will "Unlock" all the modules in the list

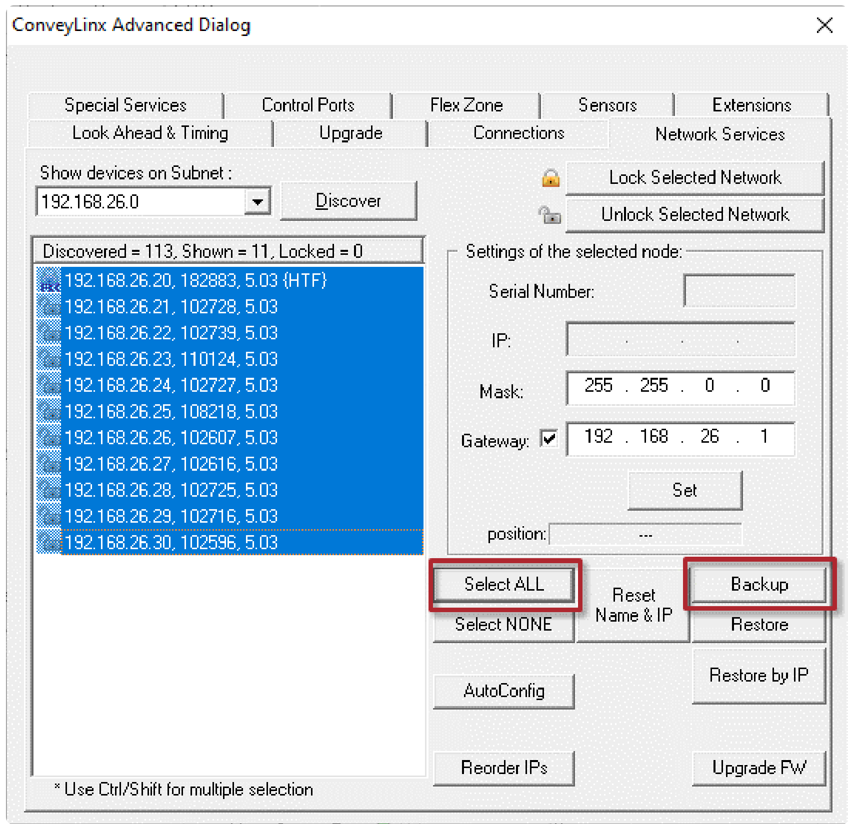


! Please note that this function applies to all the modules within a selected *Subnet* and cannot be activated for an individual module

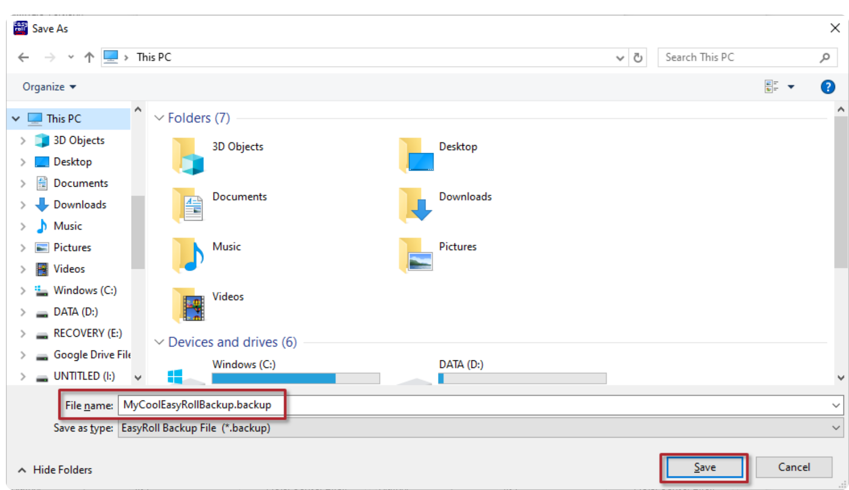
8.6.4.4. Backup and Restore

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

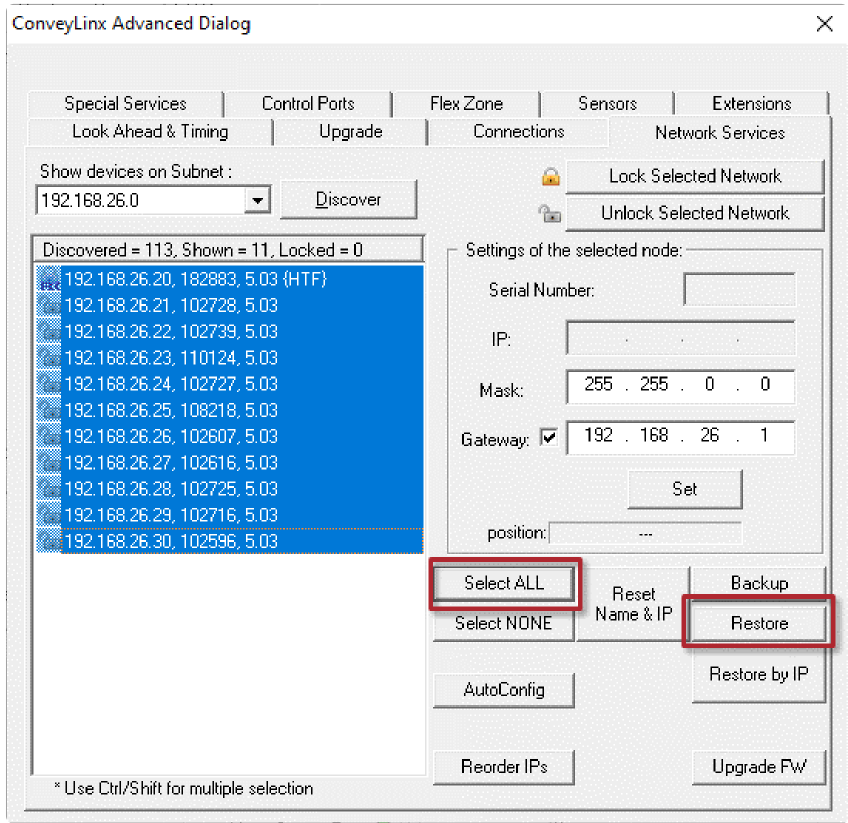
To perform a backup, click the ***Select ALL*** button to select all the items in the list. In this example we are going to backup **Subnet** 192.168.29. Once all modules are highlighted, click the ***Backup*** button. A “Save As” file dialog will appear and you provide a filename and location to save the file



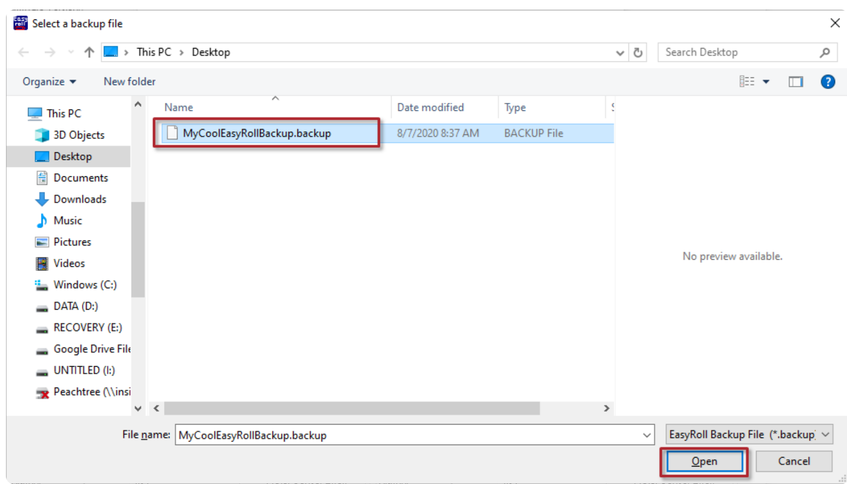
A ***Save As*** dialog will appear and you provide a filename and location to save the file



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



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

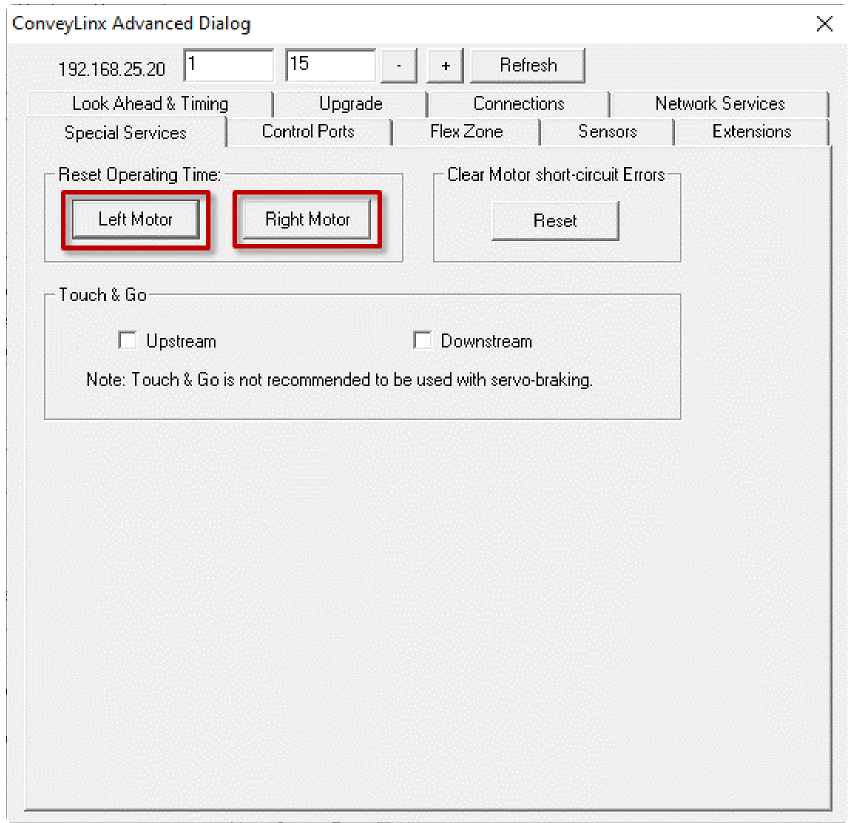
Sometimes there are situations where just because a software application will allow you to do something does not always mean that you should. The *Backup* and *Restore* functions in *EasyRoll* can fall into this scenario. Even though you can technically create a backup file for a few or even a single 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

8.6.5. Special Services Tab

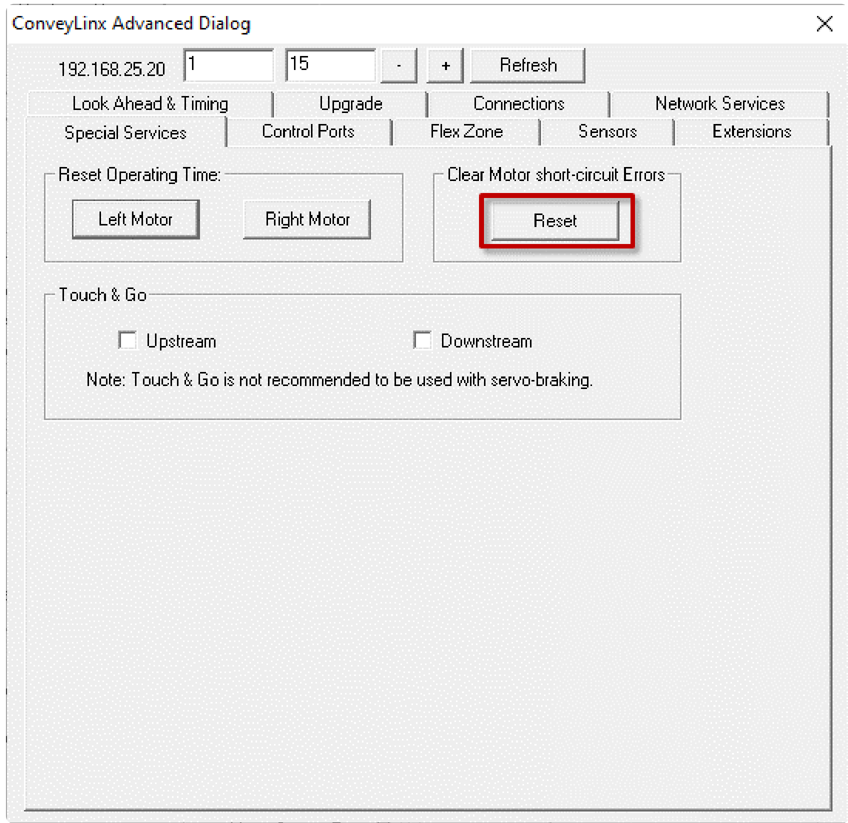
Reset Operating Time

Each module maintains a non-volatile data register for each MDR whose value is the number of minutes the motor has been run. This value is displayed as Operating Time on the Main Screen. This screen on the ConveyLinx Advanced Dialog allows you to reset this meter in the event you have to replace a given MDR. Click either the Left Motor or Right Motor buttons in the Reset Operating Time area to reset the corresponding value to 0.



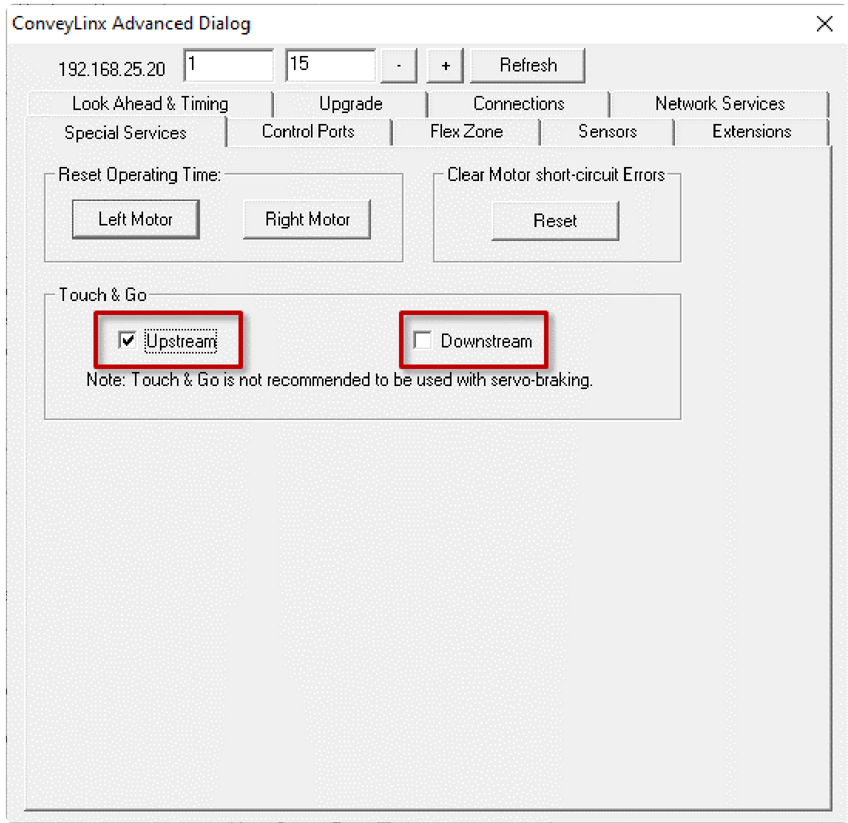
Clear Motor Short Circuit Error

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.



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.



! Touch & Go is only applicable for ZPA mode. Do NOT use Touch & Go with Servo Braking

If you are using Firmware 4.27 and later or ODVA Firmware 5.07 and later and EasyRoll version 4.21 or later; there is a ***Motor Slave*** function available on this screen tab

8.6.5.1. Motor Slave Function

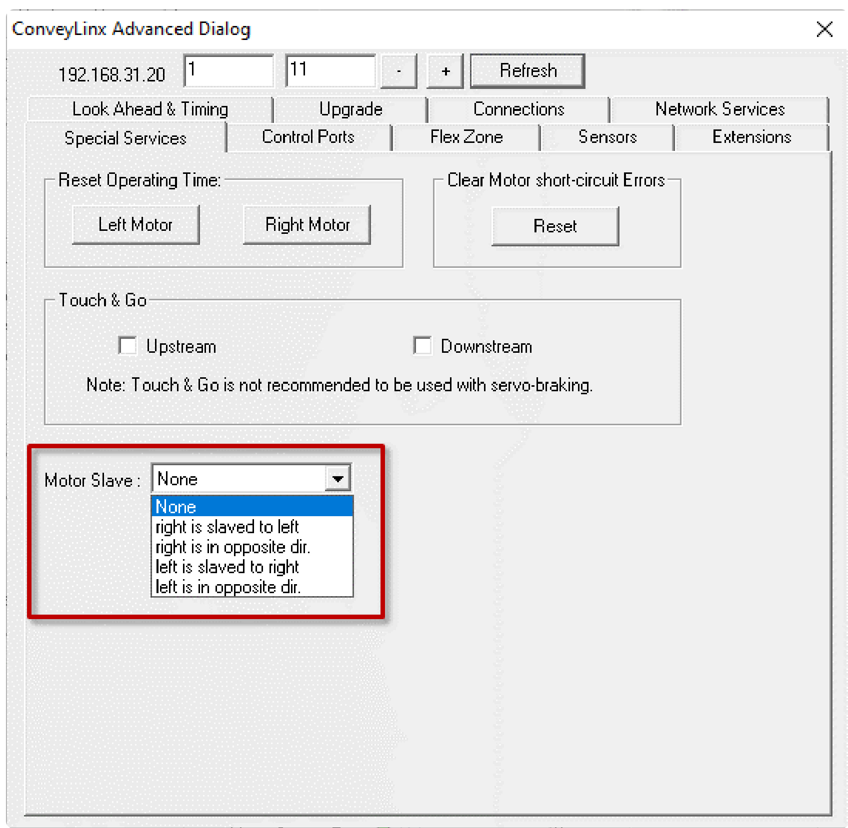
! The functionality in this topic requires Firmware 4.27 and later or ODVA Firmware 5.07 and later as well as EasyRoll version 4.21 or later

The ***Motor Slave*** function from EasyRoll is only applicable to modules in ***ZPA mode*** that have been Auto-Configured as a ***single zone*** (only one sensor connected) and two motors connected.

Motor Slave function **automatically** adjusts slave motor's speed and performance in ***Closed Loop*** mode to properly assist the master motor

The purpose of the *Slave Function* is to select one of the two motors (Left or Right) to be the master and the other to to follow the run/stop and speed of the opposite motor. When you select one of the motors to be the master, only its settings are accessible in EasyRoll and whatever setting you modify (Speed, Accel, Decel, etc.) will also apply to the slave motor.

From the *Special Services* Tab, if you have selected a Node that has been *Auto-Configured* as a *Single Zone ZPA* , then the *Slave Function* drop down box is enabled for selection.

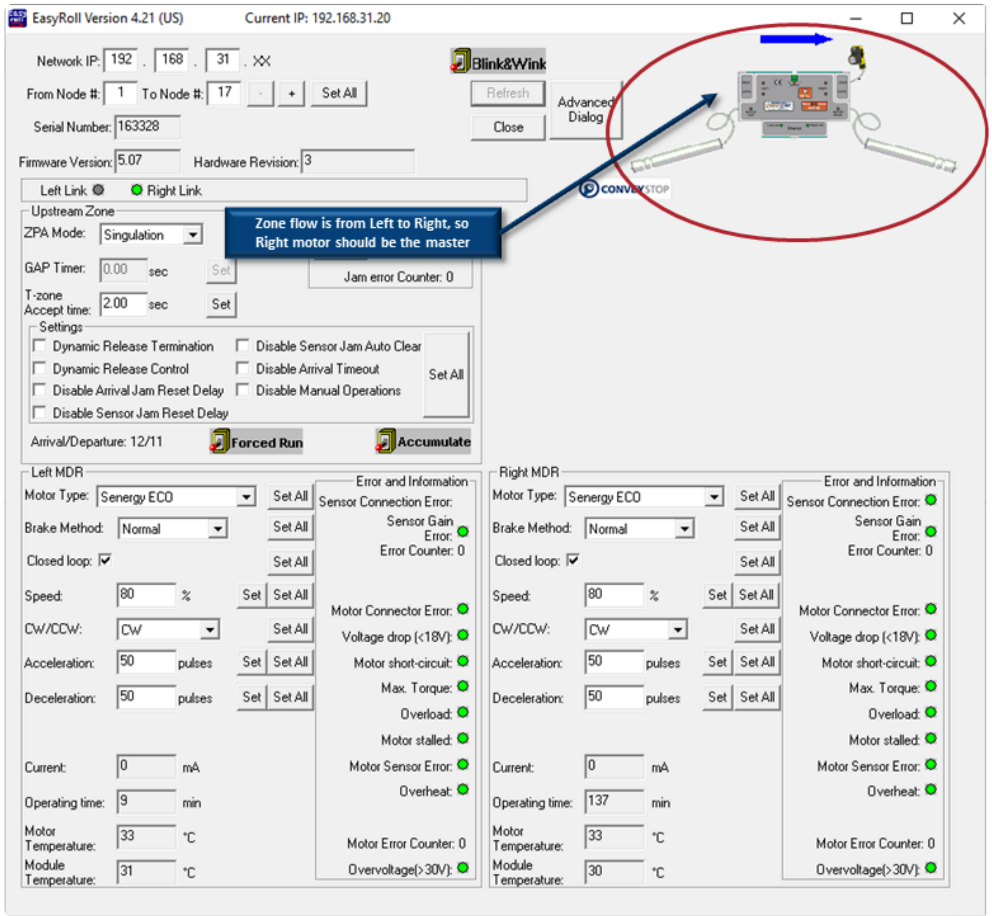


Slave Function Selections

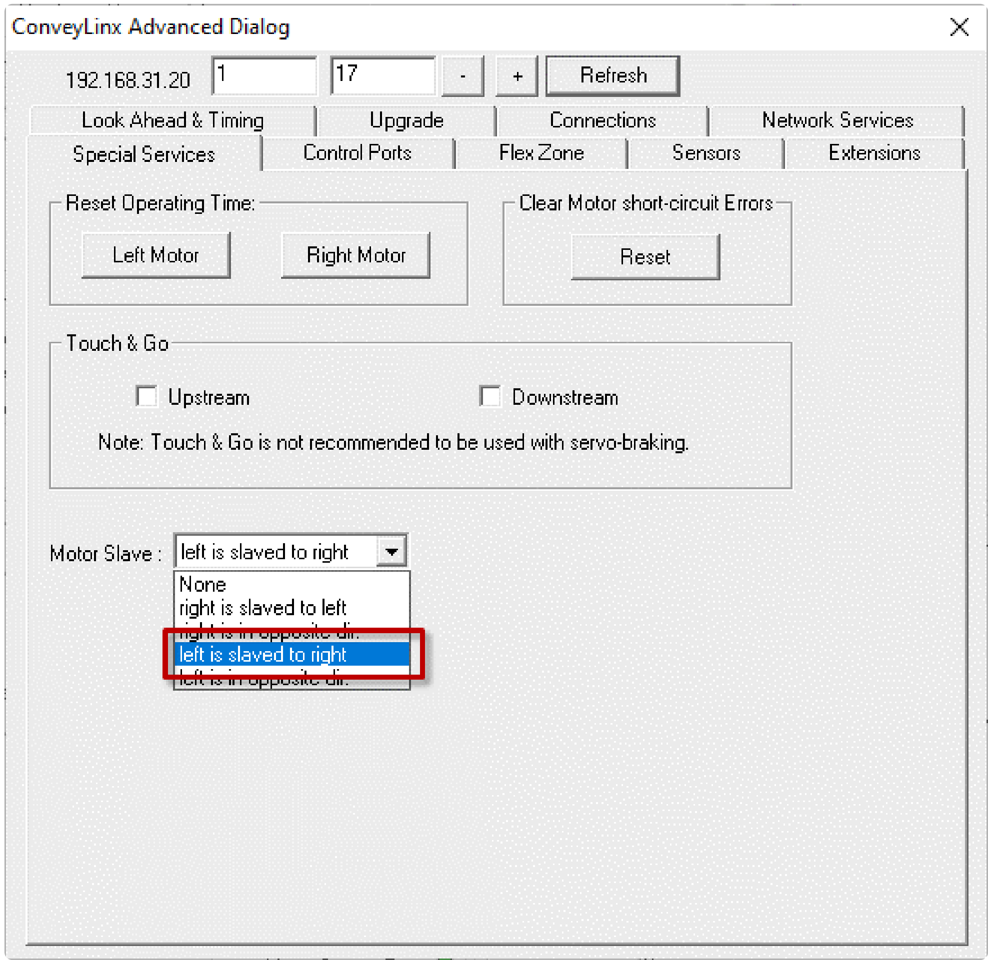
Selection	Description
None	No Slave Function is applied and Node functions as Auto-Configured as single zone ZPA. Both motors run together when the zone is activated and the two motors have their settings independently accessible from EasyRoll
right is slaved to left	Left motor becomes the master and the Right motor becomes the slave. Only the Left motor’s settings are accessible in EasyRoll and the Right motor will utilize all the settings made to the Left motor
right is in opposite direction	Identical to *right is slaved to left* except that when the Left motor runs, the Right motor runs in the opposite direction of the Left motor. This function is used for *Dual Drive* MDRs
left is slaved to right	Right motor becomes the master and the Left motor becomes the slave. Only the Right motor’s settings are accessible in EasyRoll and the Left motor will utilize all the settings made to the Right motor
left is in opposite direction	Identical to *left is slaved to right* except that when the Right motor runs, the Left motor runs in the opposite direction of the Right motor. This function is used for *Dual Drive* MDRs

Example

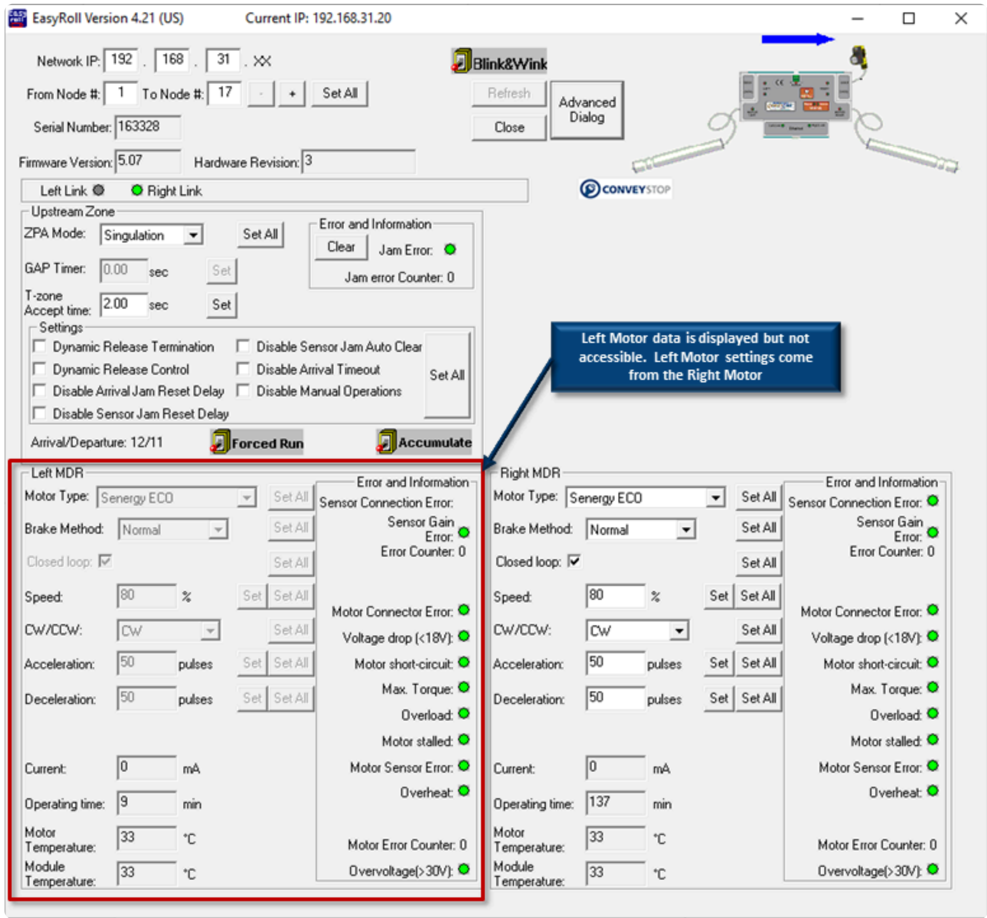
You have a *single zone ZPA* module where the flow is from *Left to Right*, so lets make the *Right* motor the master and the *Left* motor the slave



From the *Slave Function* drop down selector, select *left is slaved to right*



When you go back to the *Main Screen*, you can see that the Left motor parameters are greyed out and not accessible. Any changes you make to the Right motor will automatically be applied to the Left motor



8.6.6. Control Ports Tab

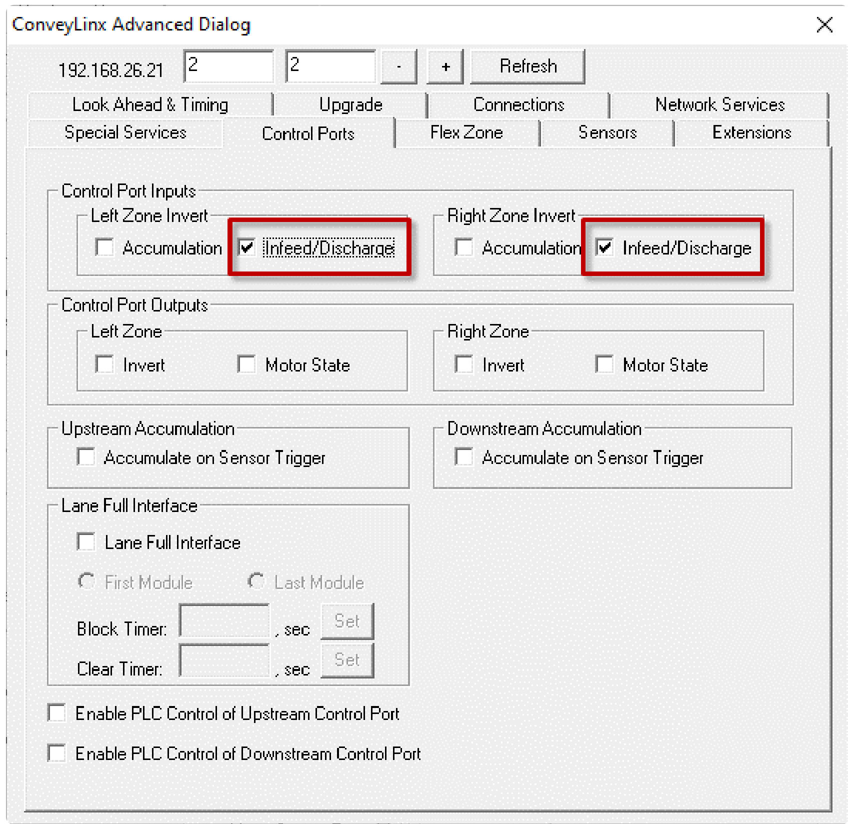
The Control Ports tab provides you with the ability modify the function of one or both of the Control Ports. This tab is organized in the following areas:

- Control Port Inputs
- Control Port Outputs
- Upstream/Downstream Accumulation
- Lane Full Interface
- PLC Control of Control Ports

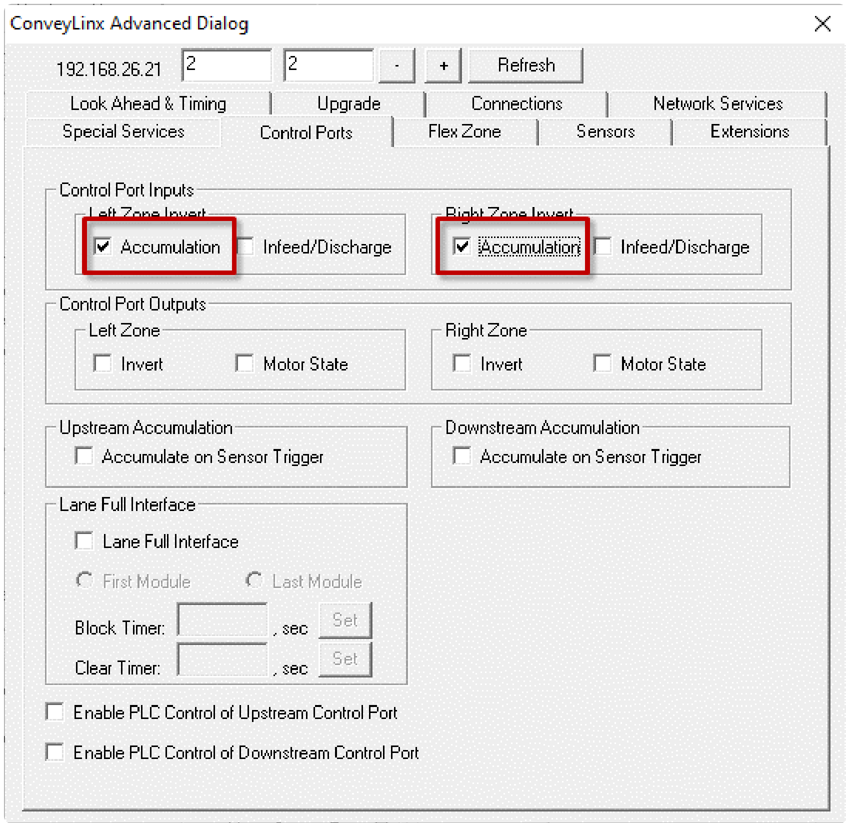
8.6.6.1. Control Port Inputs

Each of the two *Control Ports* on the module has a 2 input signals: *Local Accumulation* and *Infeed/Discharge* Interlock. Each of these inputs has a default logical state (on or off) that enables the function. The *Control Port Inputs* area contains check boxes to invert the logical state of these inputs to accommodate your particular needs. Please note that these checkboxes are for **Left** and **Right** ports on the module and **not** the logical **Upstream** or **Downstream** zones

By default, the module interprets the *Infeed/Discharge* input signal (Interlock Pin 4 on RJ-12 Control Port or P4 on a SE-4) when energized or ON to mean a logical "1". By clicking the check box to invert the *Infeed/Discharge* input signal, a de-energized or "OFF" condition will mean a logical "1" to the module for the Interlock Pin 4 *Control Port* signal

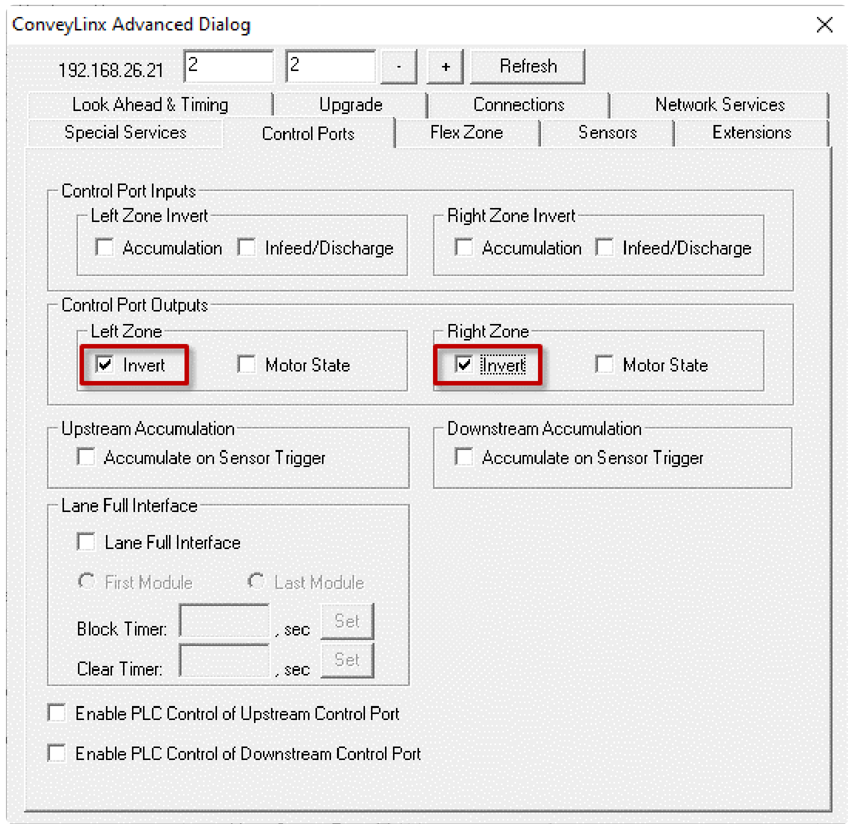


By default, the module interprets the *Accumulation* input signal (Pin 3 on RJ-12 Control Port or P3 on a SE-4) when energized or ON to mean a logical “1”. By clicking the check box to invert the *Accumulation* input signal, a de-energized or “OFF” condition will mean a logical “1” to the module for the Pin 3 *Control Port* signal



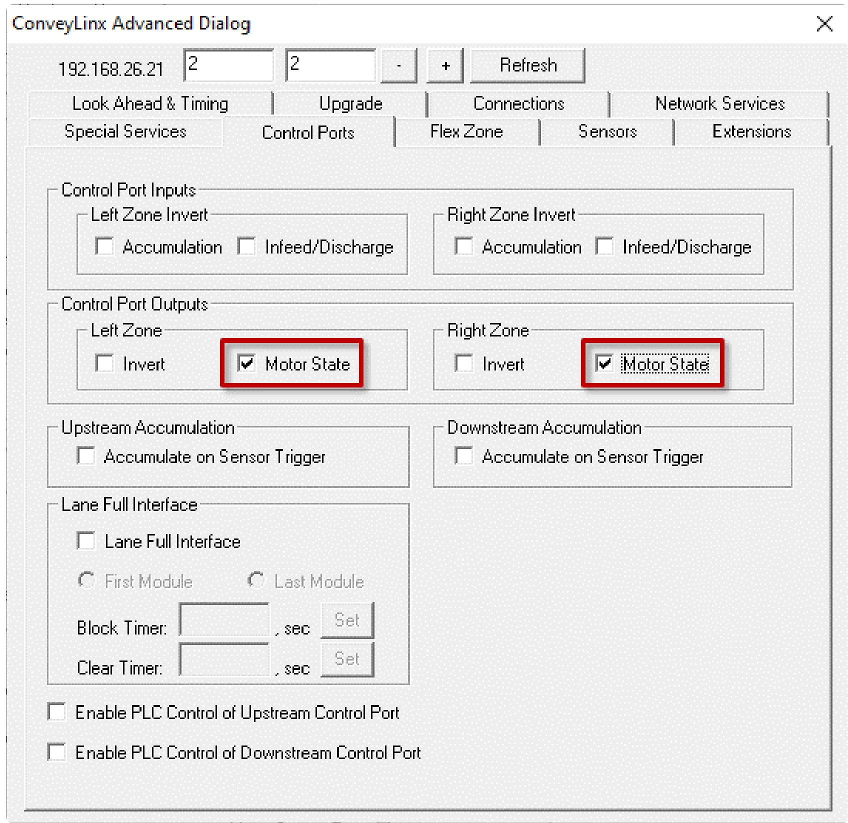
8.6.6.2. Control Port Outputs

> **!** Control Port outputs are very low power. You need to use an SE Module to connect your external device.



The default *Control Port Output* configuration signal is “ON” or logical “1” to indicate to external controls that it is associated zone is occupied. By clicking the associated *Invert* check box, the module will make the output “OFF” or logical “0” when its associated zone is occupied

The default operation of each *Control Port Output* signal is to indicate whether the zone is blocked. You can change this operation to indicate that the motor is running instead of zone blocked by clicking the *Motor State* checkbox. Please note that the *Invert* function applies to this option as well



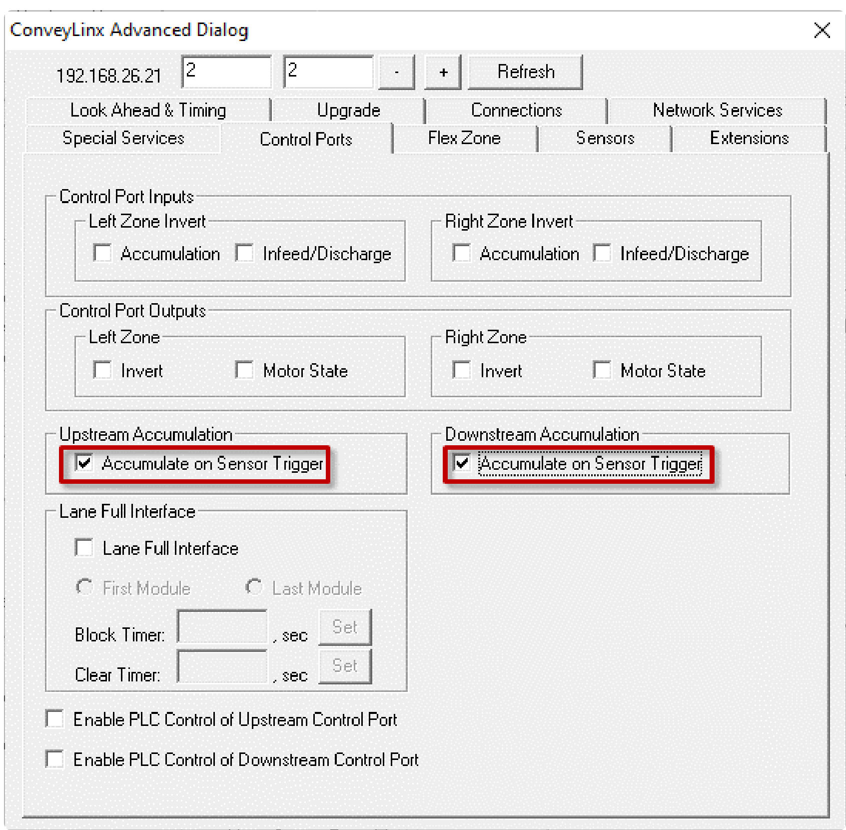
8.6.6.3. Upstream/Downstream Accumulation

One of the *Control Port* input signals previously mentioned above is the Local Accumulate signal (Pin 3 on RJ-12 Control Port or P3 on a SE-4). By default, this signal causes the local zone to accumulate when its zone sensor is blocked. If this signal is removed when a carton is blocking the sensor, the carton will discharge to the next downstream zone (if of course it is clear). While the carton is discharging and still blocking the zone sensor, if the Accumulate signal is re-energized, the zone will stop.

Accumulate on Sensor Trigger

In situations (like an operator workstation) where you for example want every carton to accumulate upon its arrival, you would leave the Accumulate signal energized all the time and then have an operator momentarily de-energize the signal to cause the carton to discharge. In this situation you probably do not want the operator to have to watch and wait for the carton to fully discharge from the zone in order for them to re-energize the Accumulate signal so that the next carton will stop at the zone. The *Accumulate on Sensor* Trigger checkboxes for the Upstream and Downstream zones allow you to tell either or both the Upstream or Downstream zones on the module to remember that the Accumulate signal was momentarily de-energized so the carton will discharge without any further operator intervention and automatically know to accumulate the zone upon the arrival of the next carton.

Click the *Accumulate on Sensor Trigger* checkbox so that a momentary removal of the *Pin 3* signal will release the carton from the zone. If the *Pin 3* signal comes back on and the sensor is still blocked, the control will still release the carton and will not accumulate again until the sensor becomes clear and then blocked again

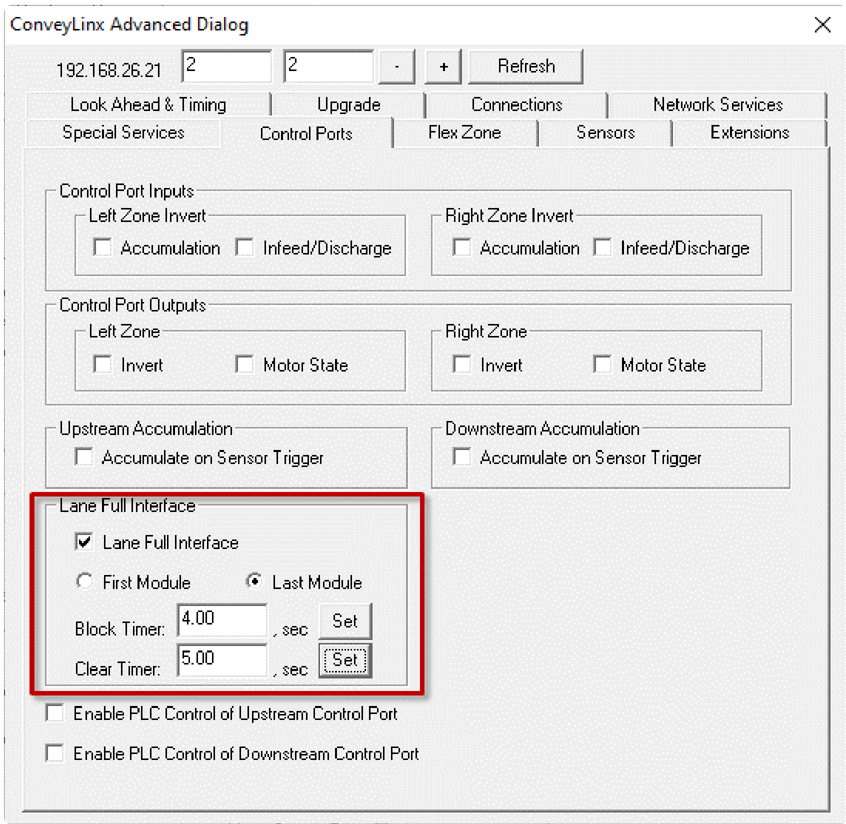


8.6.6.4. Lane Full Interface

The *Control Ports* tab provides selection and adjustment to the *Pin 4* Interlock Signal on the most *Downstream* zone to provide a *Lane Full Interface* that is typical in conveyor applications. With *Lane Full Interface* checkbox checked; the *Downstream Interlock* will change its default functionality to ignore the confirmation signal from the downstream controls and thus eliminate the logical Arrival Jam condition. This will allow the downstream zone to continually release as long as the sensor signal is OFF (default). This option also allows the user to set a block and clear time for the *Pin 4* signal for added flexibility

> The *Lane Full Interface Block and Clear Timer* capability is designed so you can plug a Sensor direct into the *Control Port*

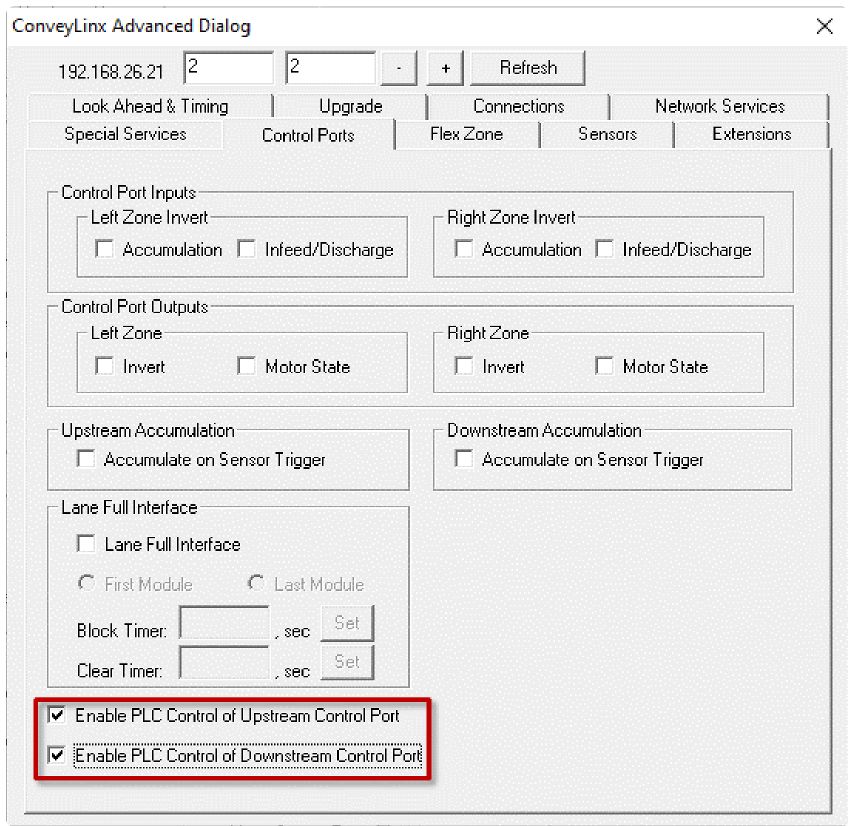
When a sensor is plugged into the **downstream** Control Port, you can check the *Lane Full Interface* checkbox. This will cause the zone to no longer require a downstream arrival interlock (i.e. *Ignore Arrival Jam*) and use the sensor signal to stop and start releasing from the zone. *Block and Clear Timers* are provided to adjust the behavior of product flow based upon the blocking and clearing of the sensor. In this example we entered 4 seconds for Block Time and 5 seconds for Clear Time.



8.6.6.5. PLC Control of Control Ports

If you have a PLC connected to a module while it is in ZPA mode, you have the option of disabling the default ZPA functionality of either *Control Port* (or both) and allow your external PLC to access the 2 input signals and the one output signal available on each port. This means that when selected for a given zone's *Control Port*; energizing either input signal will not indicate Local Accumulation or Infeed/Discharge Interlock functions and the output signal will not become energized when the zone is occupied. Your PLC program will have access to these inputs and output to use as remote I/O over the ConveyLinx Ethernet network

You can select either or both the Upstream and/or Downstream Control Ports to be controlled by your remote PLC over the ConveyLinx Ethernet network. Checking the appropriate checkbox will immediately make the change. If a checkbox is already checked, unchecking it will return the associated port back to its ZPA functionality

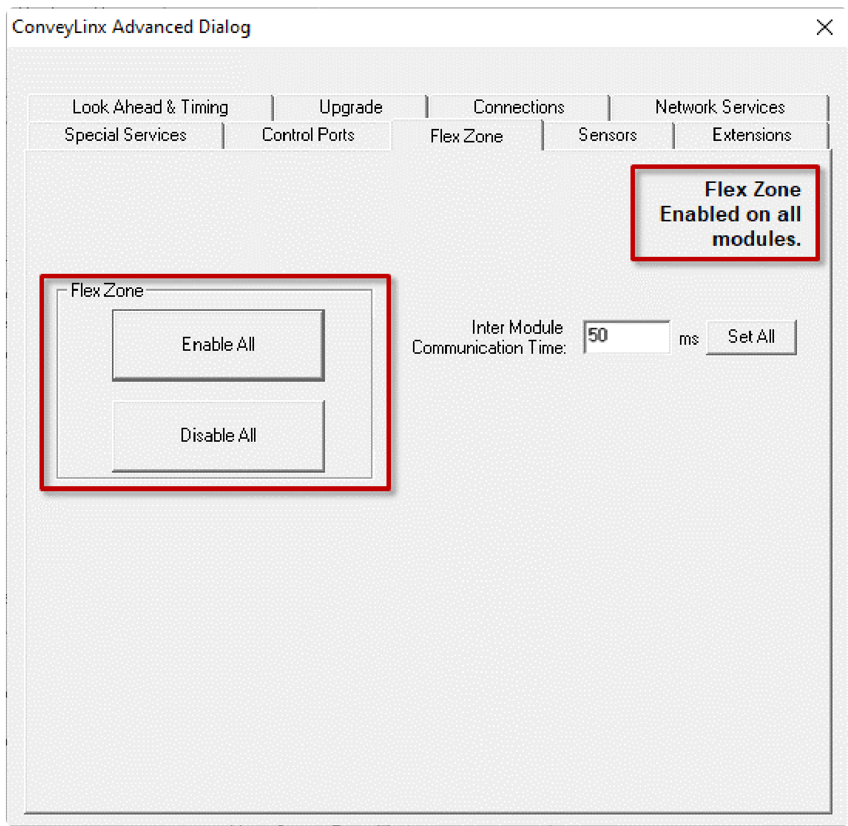


8.6.7. Flex Zone Tab

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.

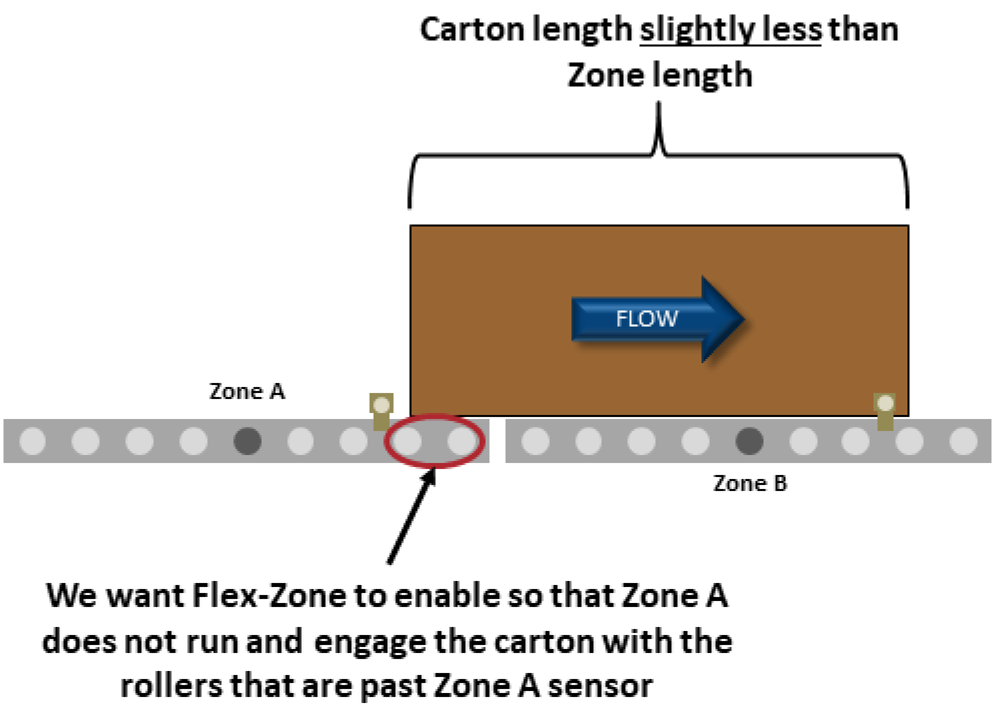
Click the appropriate button to either *Disable* or *Enable* Flex Zone based upon the subnet’s current status as indicated. Because this function applies to all modules in the *Subnet*, it does not matter what module you were connected to when you invoked the *Advance Dialog* in order to disable or enable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



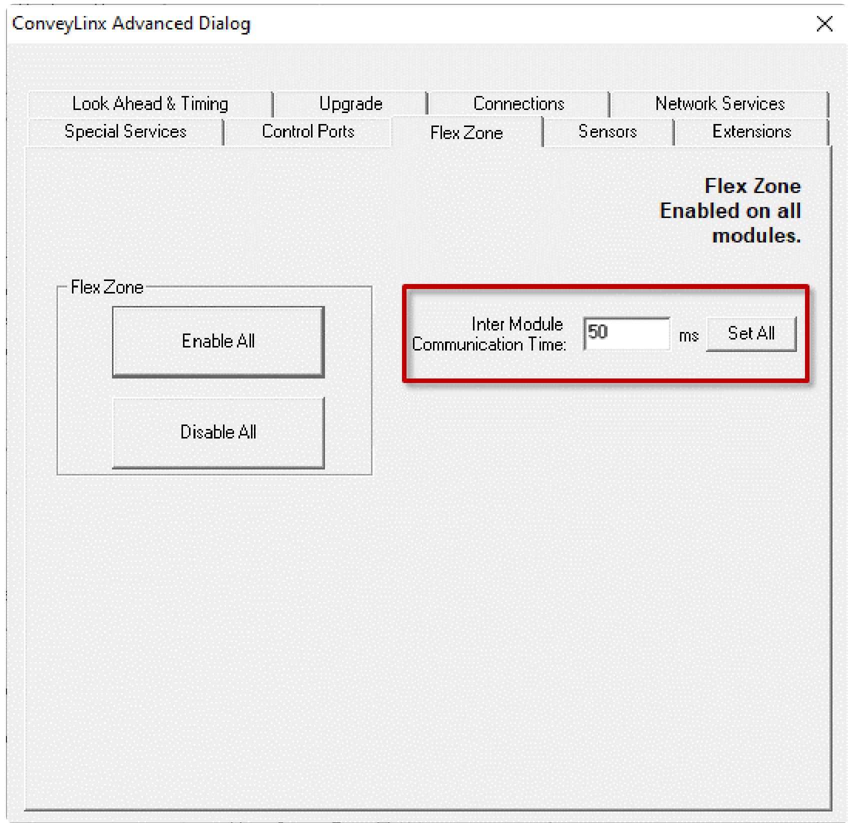
Inter Module Communication Time

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*.

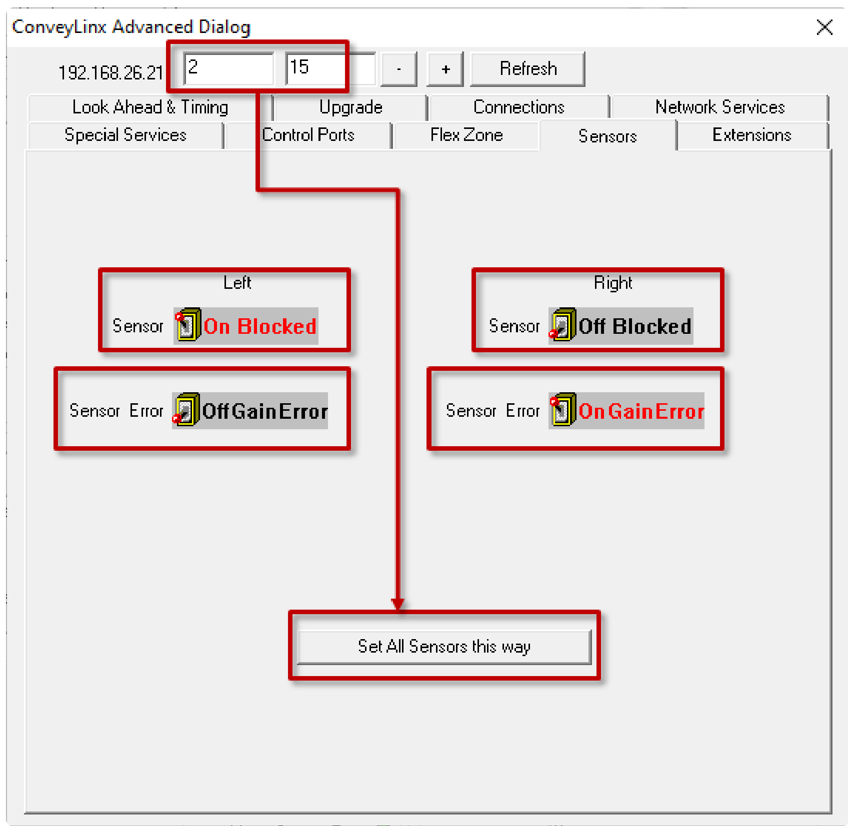


Please note this value has nothing to do with speeding up or slowing down the actual speed of communications between modules

8.6.8. 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*. Similarly, if the sensor is equipped with a separate health or low gain signal and this signal is on when there is no error, the graphic on this tab will show *Off Gain Error*.

Click the appropriate icon to change the sensor's blocked and/or gain error signal state. Note that you can make the same change for a group of modules by clicking the *Set All Sensors this Way* button



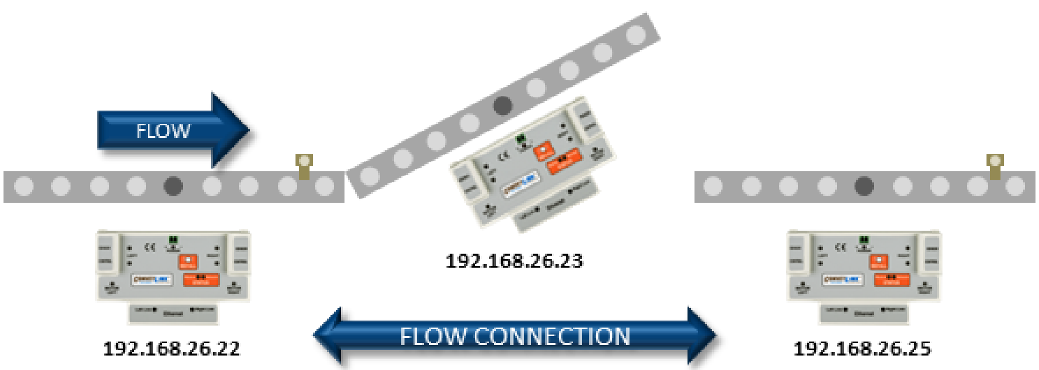
8.6.9. Extensions Tab

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.

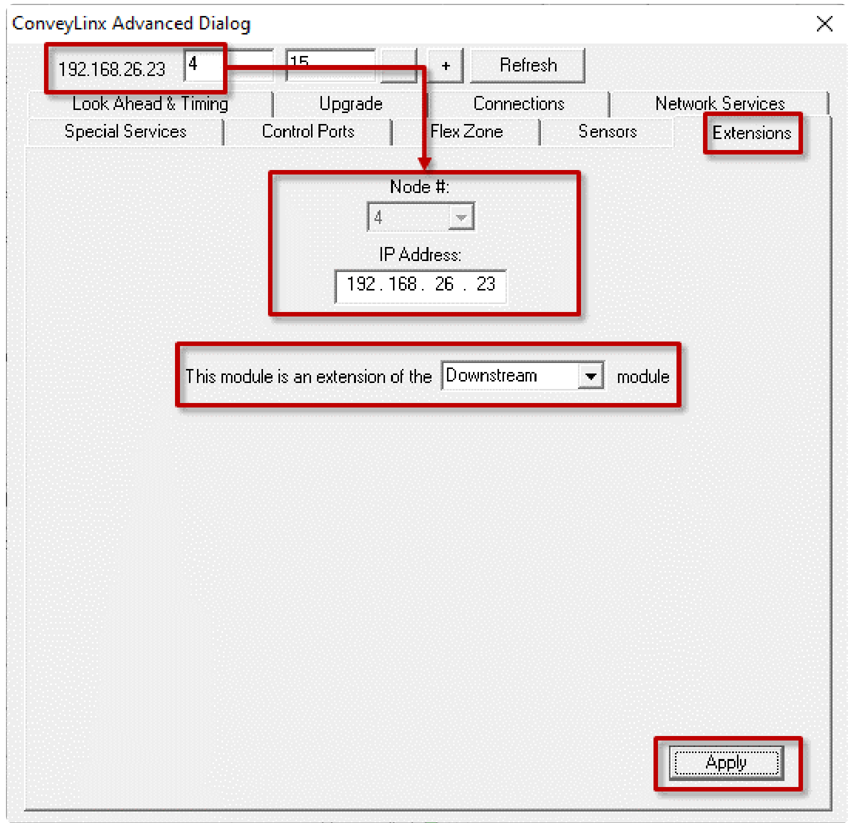
Example for Using Extensions for a Lift Gate

A common use of the Extension mode configuration available from the Connections tab selection is for a powered lift gate.

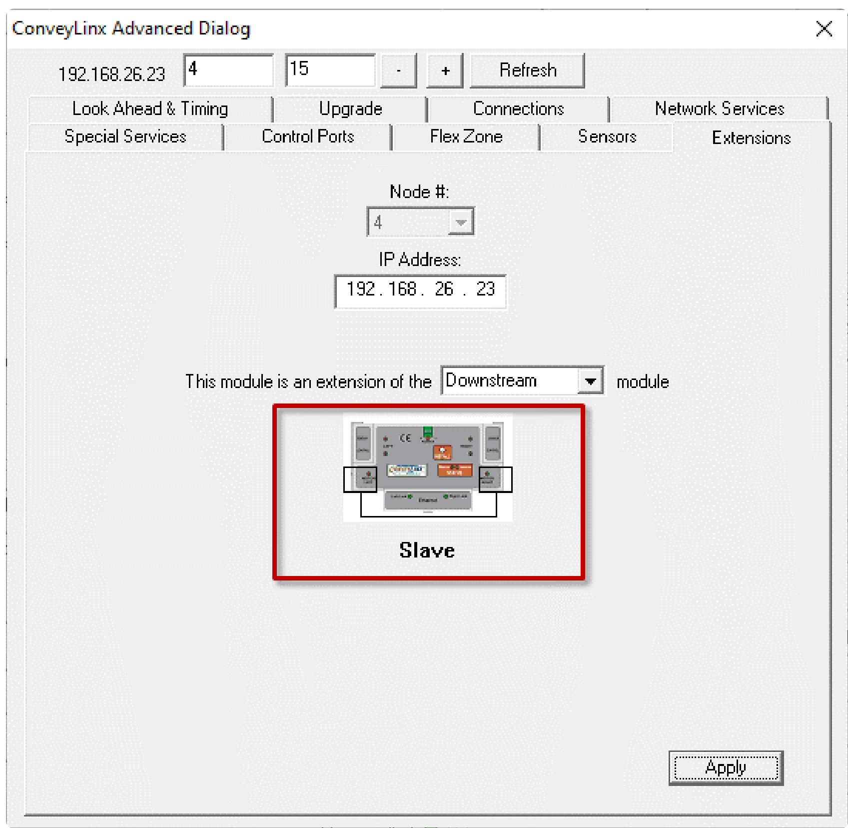
> In this example the module on the lifting or gate portion of conveyor has an MDR and no photo-sensors. Normal operation when the gate is down is for the MDR on the gate to run when its immediate downstream zone runs so as to create "one long logical zone". This means that if a carton is accumulated on the upstream zone of Node 192.168.26.24; a carton arriving at the downstream zone of Node 192.168.26.22 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 192.168.26.23 to be an Extension of its downstream neighbor Node 192.168.26.24



In the *Advanced Dialog*, navigate to Node 4 and click the *Extensions* Tab. Notice that Node 4's IP address 192.168.20.24 is filled in. In the drop-down box, select *Downstream* and click *Apply*. Note that this may take a few seconds to complete

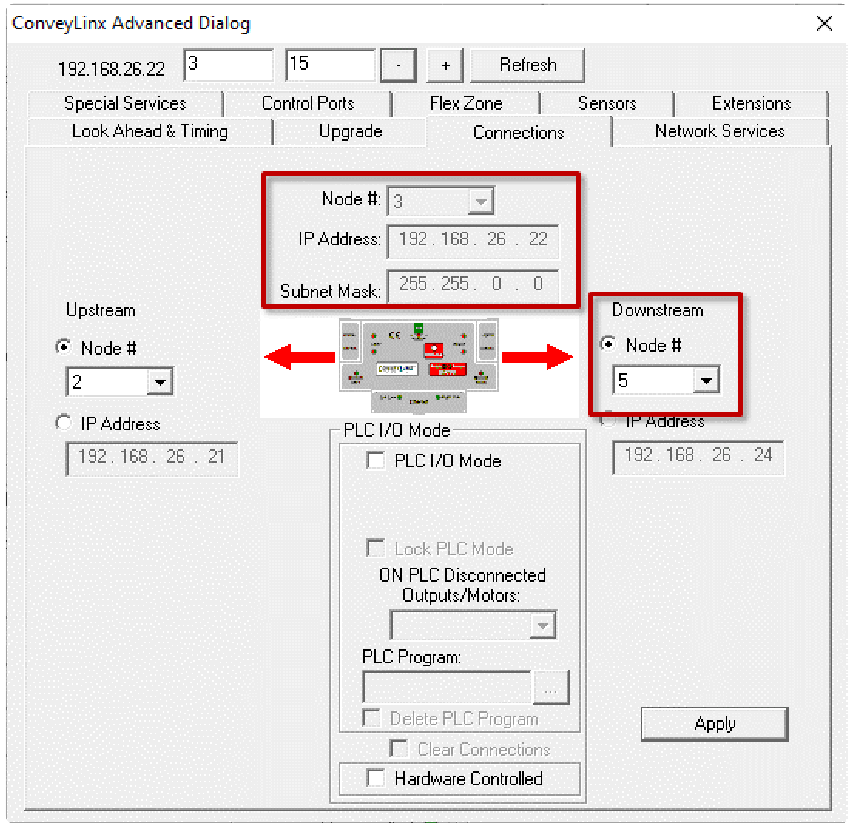


After clicking *Apply*, the screen will update and show you that the module has been configured as an extension or "slave" of its downstream neighbor

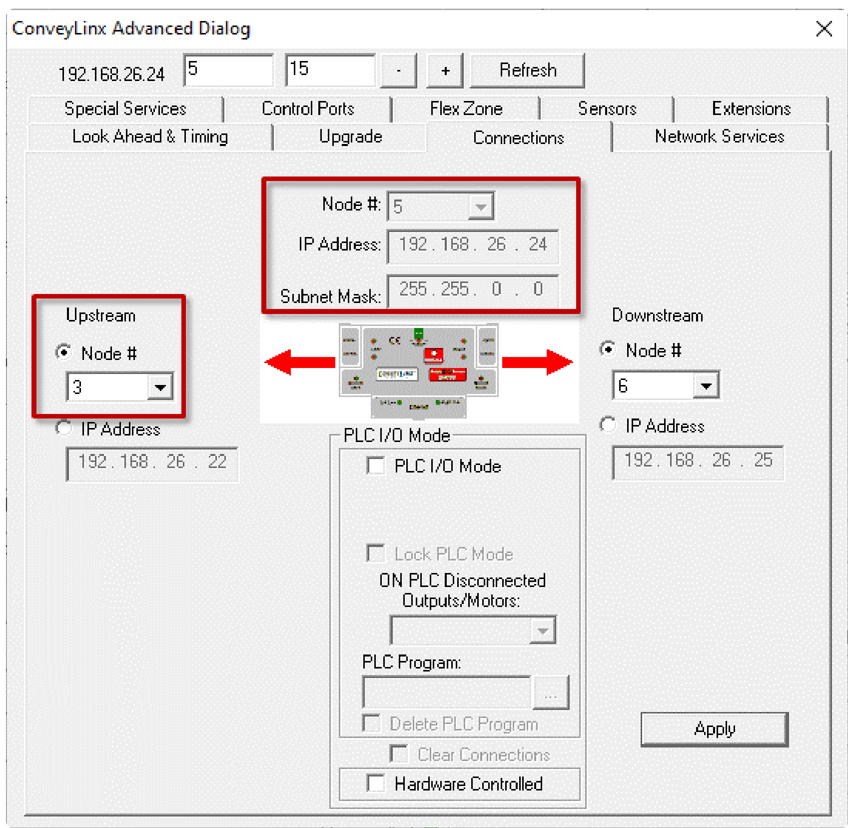


!

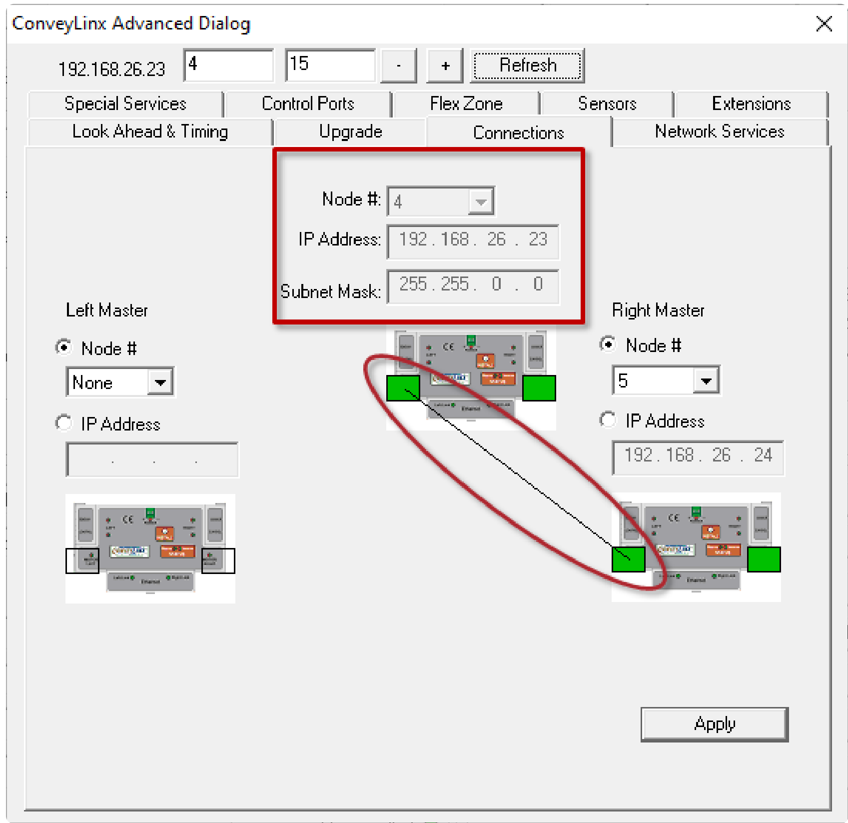
Now if you navigate back to Node 3 and look at the *Connections* Tab, you can see that the downstream connection of Node 3 is now Node 5 instead of Node 4



And then when you navigate to Node 5, you can see that its upstream connection is now Node 3 instead of Node 4



And when you navigate to Node 4, you can see that it is configured as an Extension of Node 5

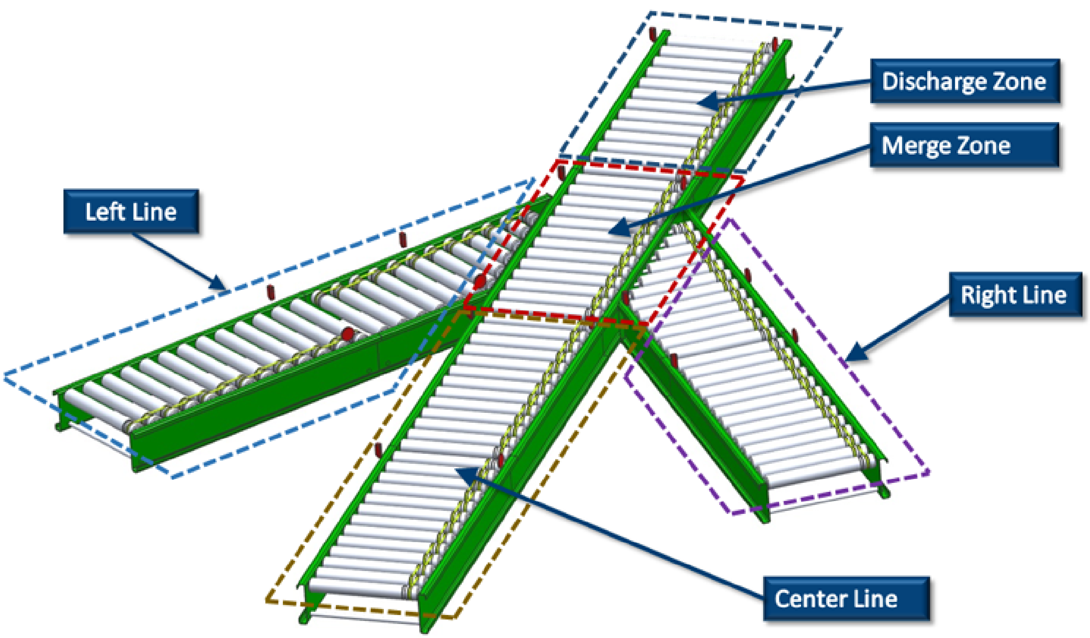


9. ConveyMerge

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

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

- *EasyRoll* version 4.19 or greater
- 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*

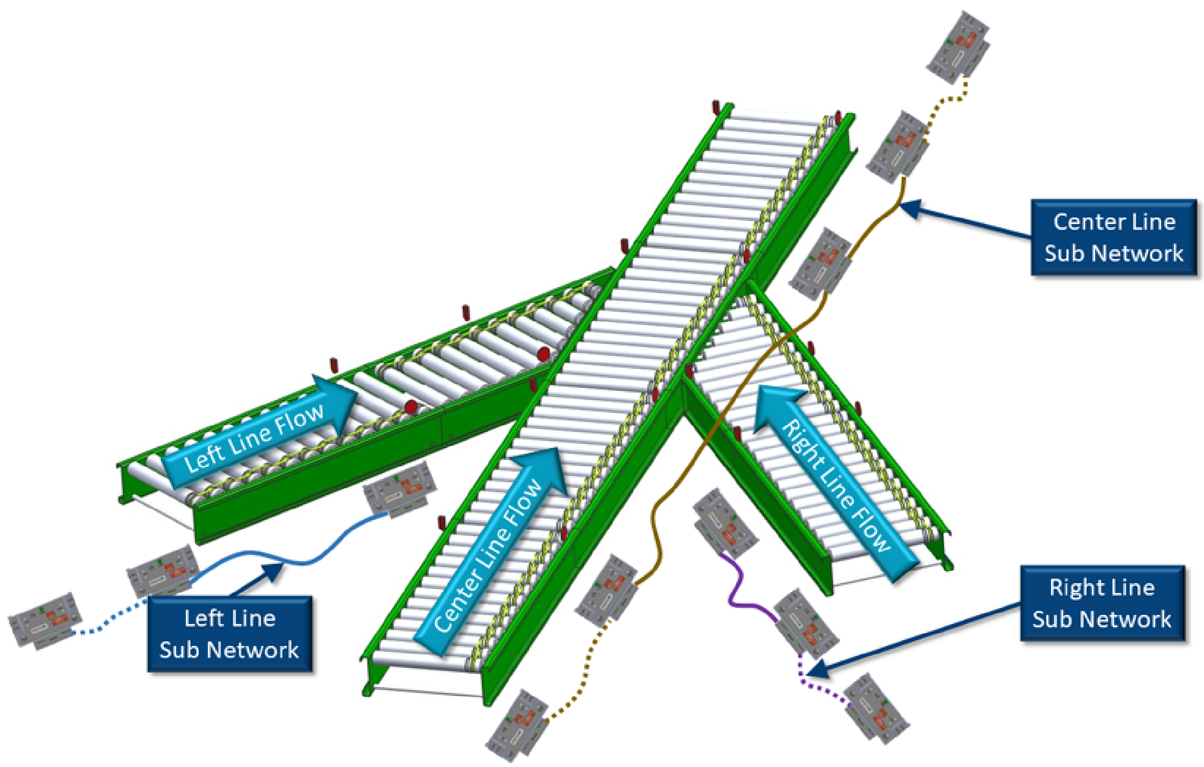
✻ If *Left Line* discharge, *Right Line* discharge, or *Center Line* discharge modules are in PLC I/O mode, you are responsible for programming logic to recognize and respond to ZPA zone status values between the *Merge Zone* module for proper jam free singulation control

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 Modules* and the actual position of the *Merging Zone* can be on any *ConveyLinx Module* within the *subnet*

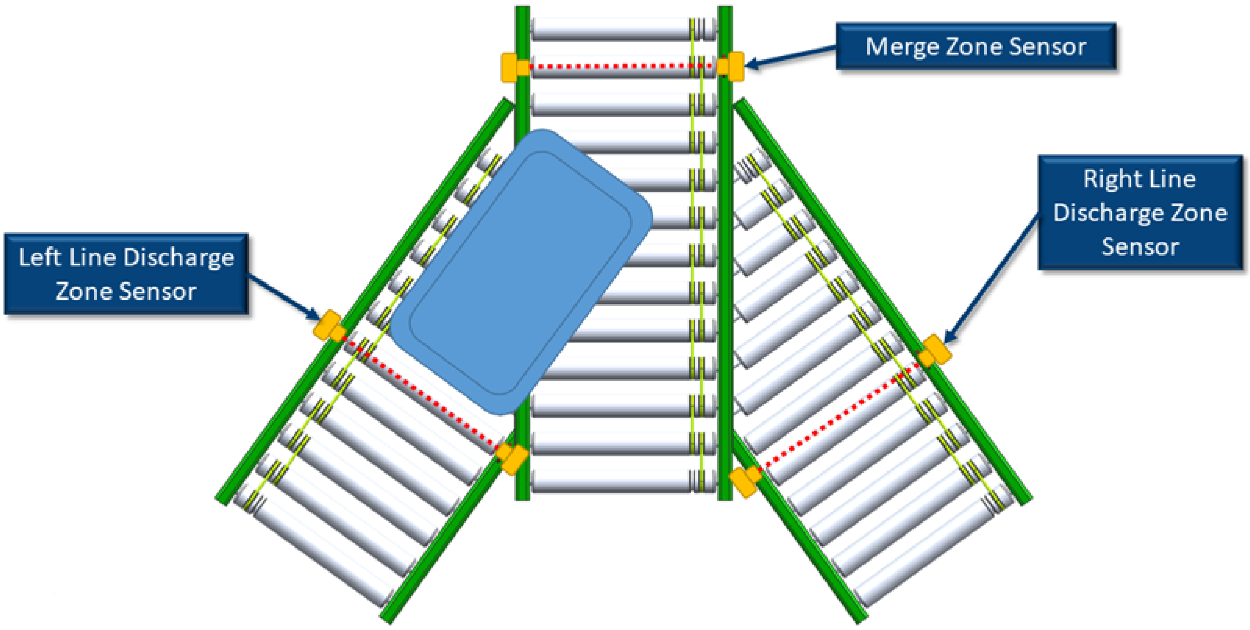
- The *Discharge Zone* has to be the zone immediately downstream of the *Merge Zone*. However, the *Discharge Zone* can be on either the same ConveyLinx Module as the *Merge Zone* or on the adjacent downstream ConveyLinx Module from the *Merge Zone's* ConveyLinx Module
- The *Left Line* is a separate *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 only after each *subnet* has been *Auto-Configured*.

*

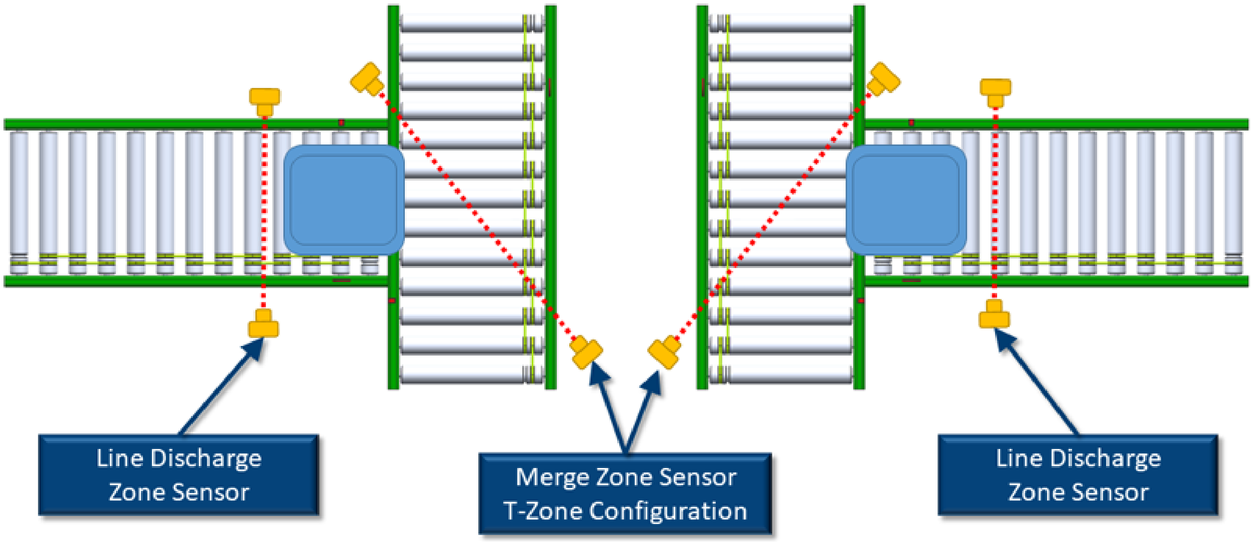
9.3. Sensor Placement

> **!** 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



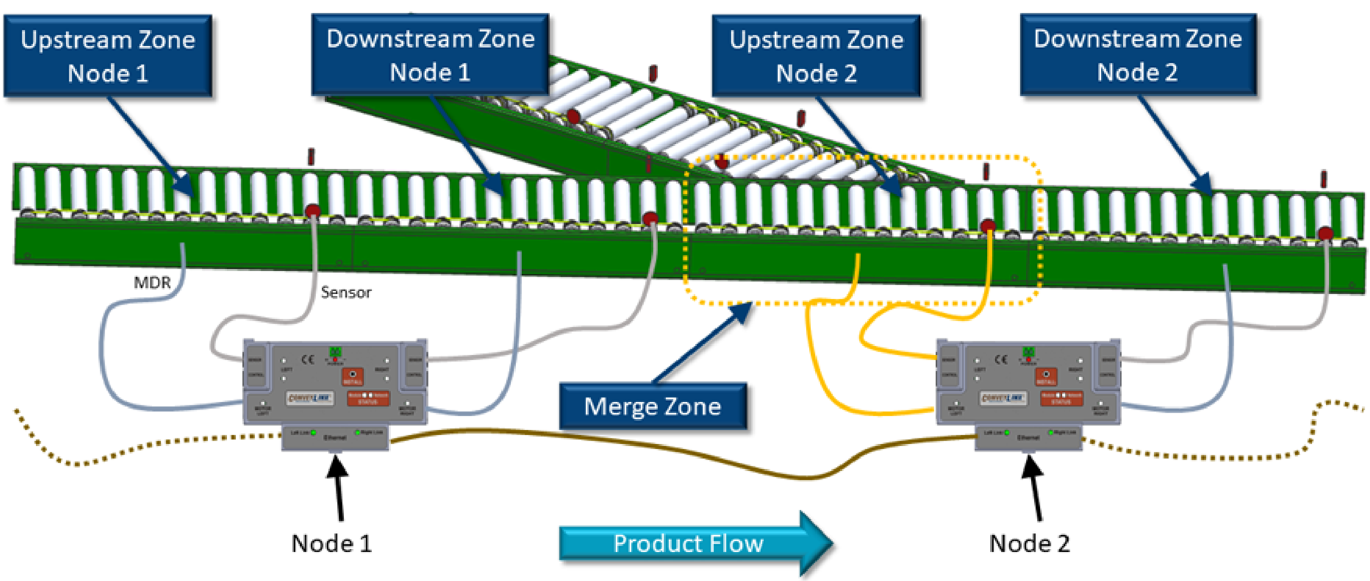
9.4. Merge Zone Module

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 Module. It can also be assigned to a ConveyLinx Module that is configured as a single zone ZPA controller.

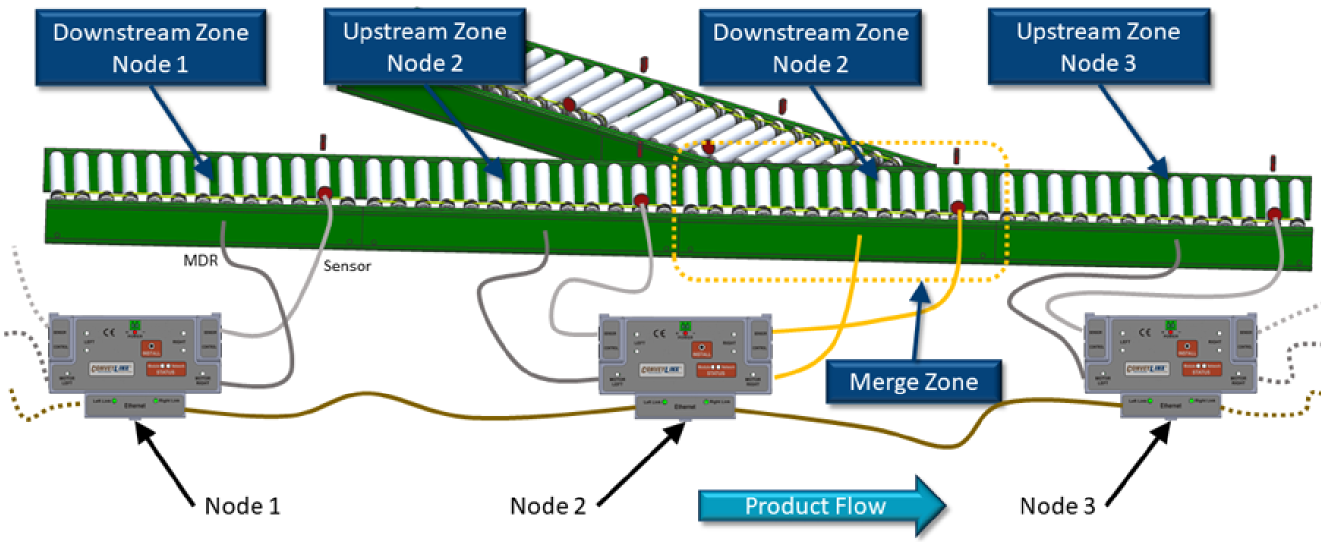
In all 4 of these examples, *Node 2* is the Merge Zone's controlling ConveyLinx Module

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

Merge Zone on Upstream Zone

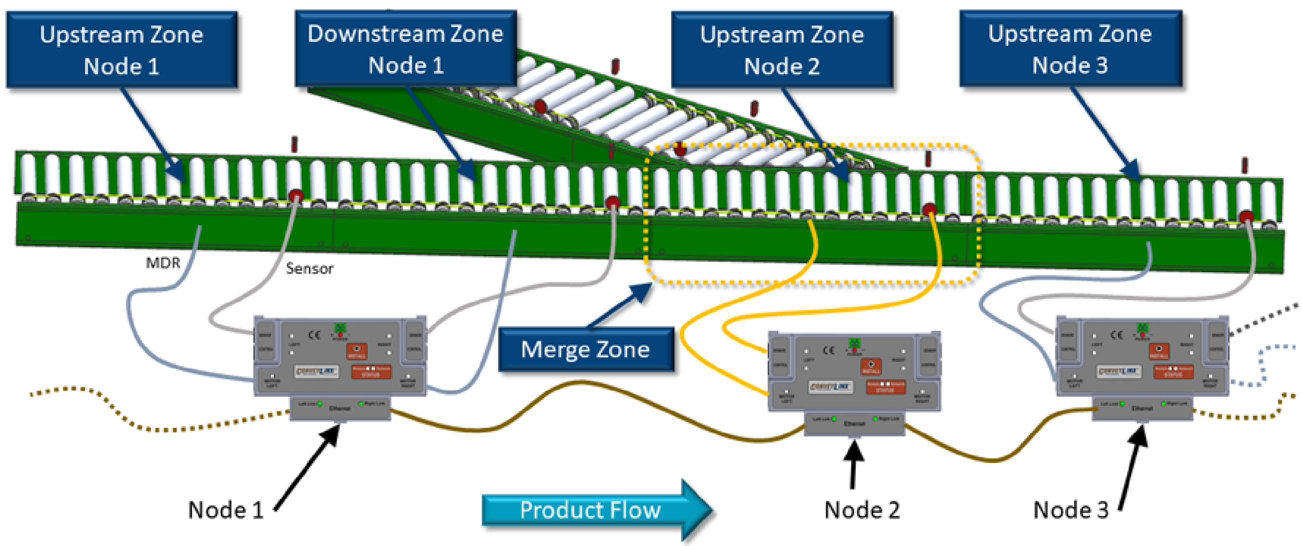


Merge Zone on Downstream Zone

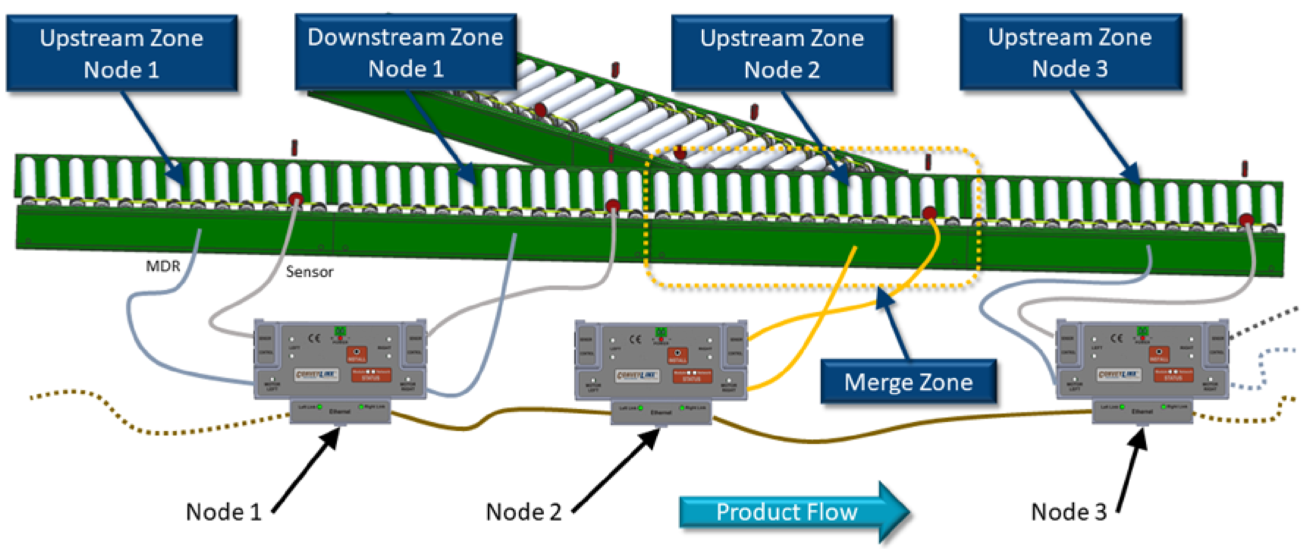


Single Zone Module

Whenever an ConveyLinx Module 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 Module, 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

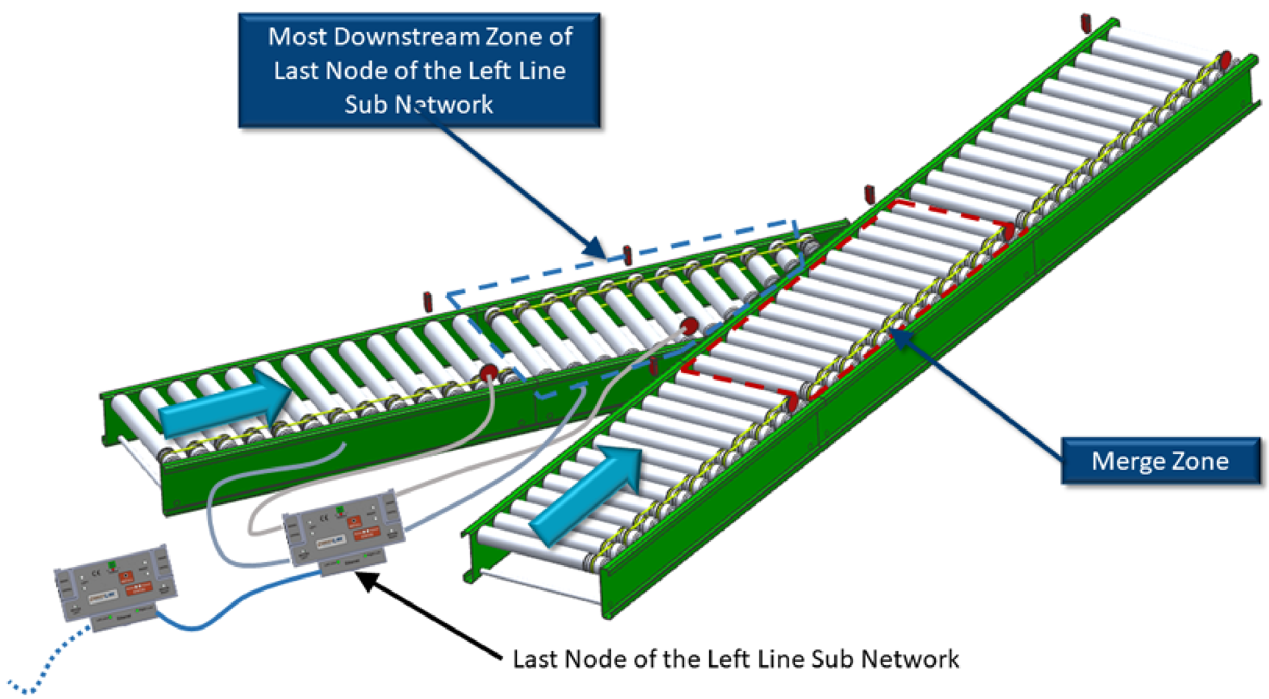


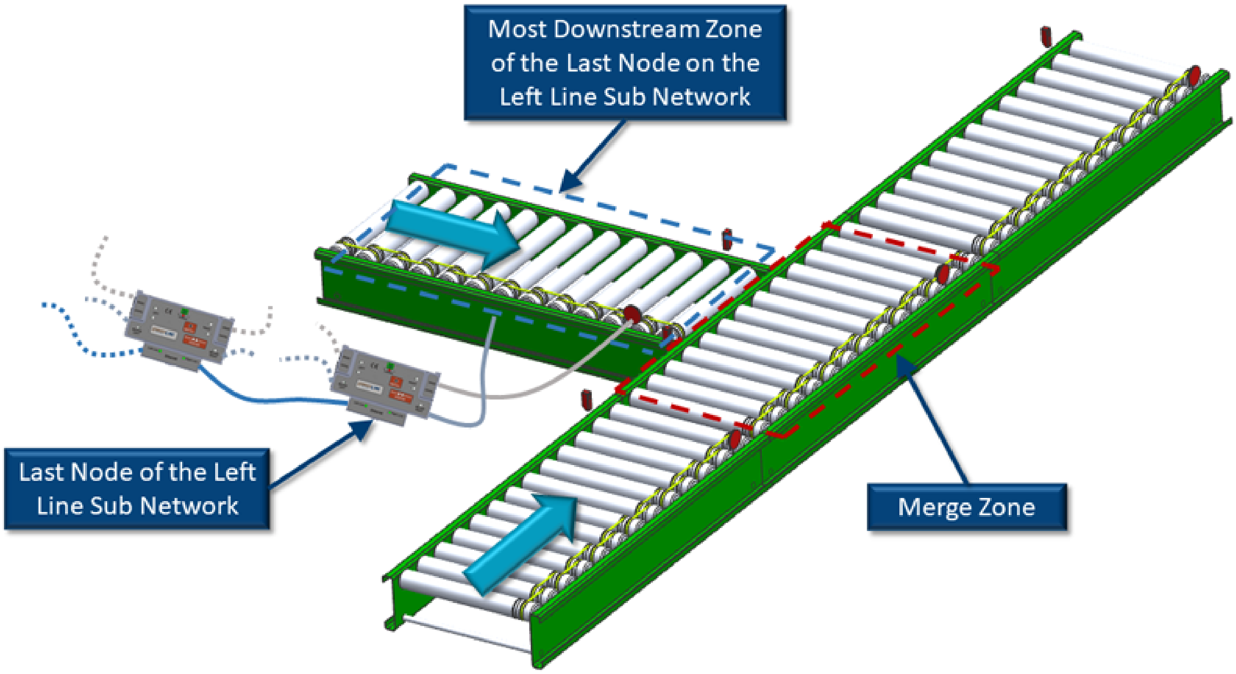
Merge Zone on a Single Zone Module's Right Side

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* must be the most downstream zone if its particular subnet. 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.

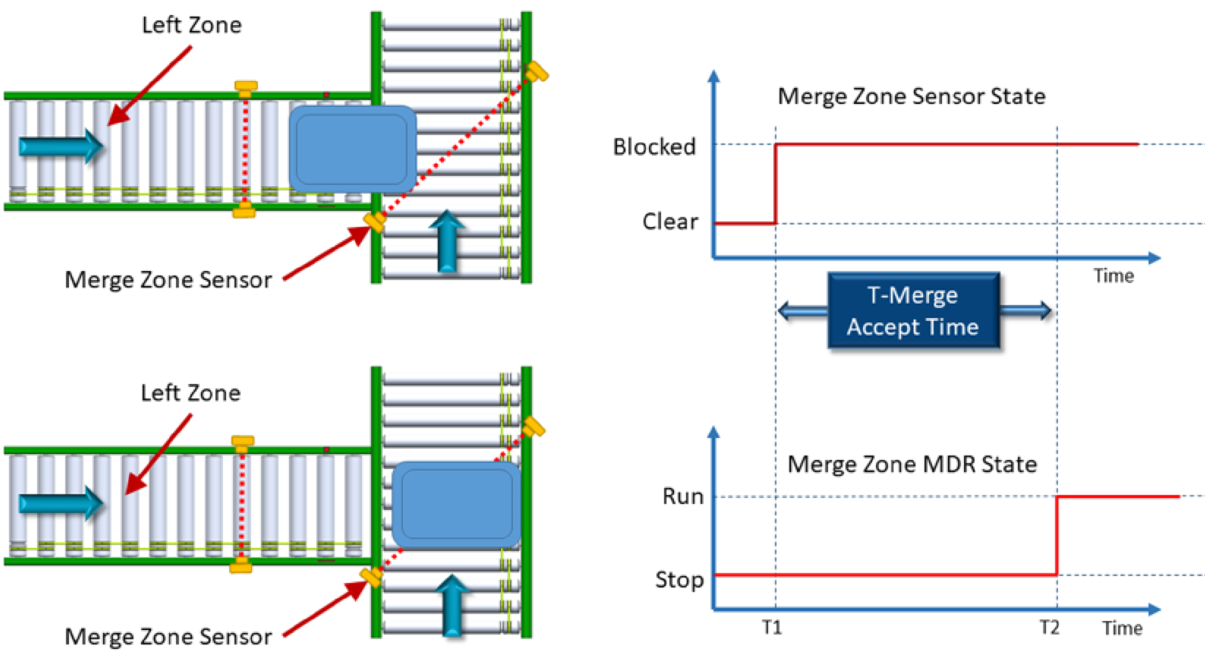
EasyRoll Priority Selection	Description
First Come First Served	The logic monitors the arrival status in each of the zones that discharge in to the *MergeZone*. The first product that arrives at any of these lines will be allowed to convey into the *MergeZone*. When the *MergeZone* is clear, the next 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.

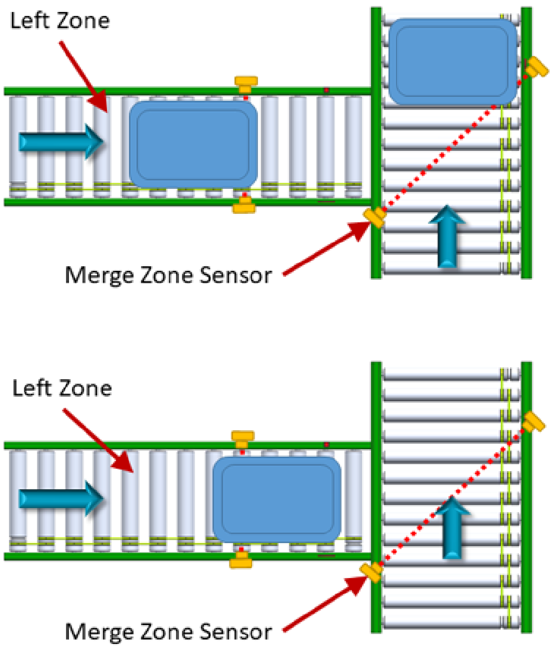
Accept Timeout

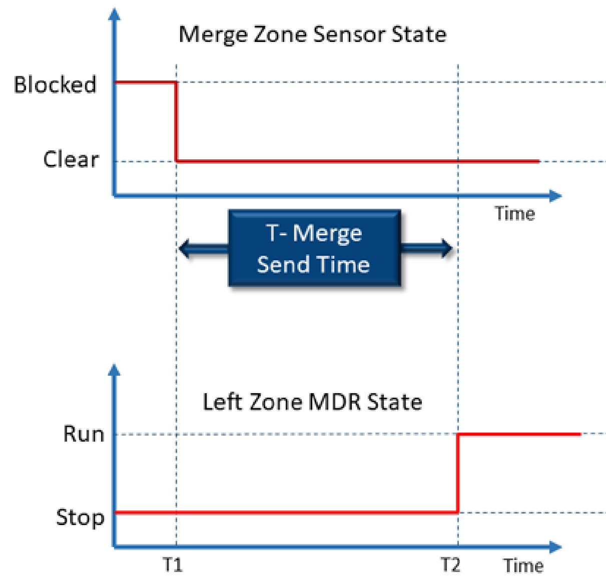
This timer specifies how much time the *Merge Zone* will wait after its sensor is blocked until the logic allows it’s motor to run. Once the timer has expired, the Merge Zone’s motor will be allowed to run. Keep in mind that conditions downstream of the *Merge Zone* also dictate if the motor is allowed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


Left Zone
Merge Zone Sensor
Left Zone
Merge Zone Sensor


Merge Zone Sensor State
Blocked
Clear
Time
T- Merge
Send Time
Left Zone MDR State
Run
Stop
T1
T2
Time

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

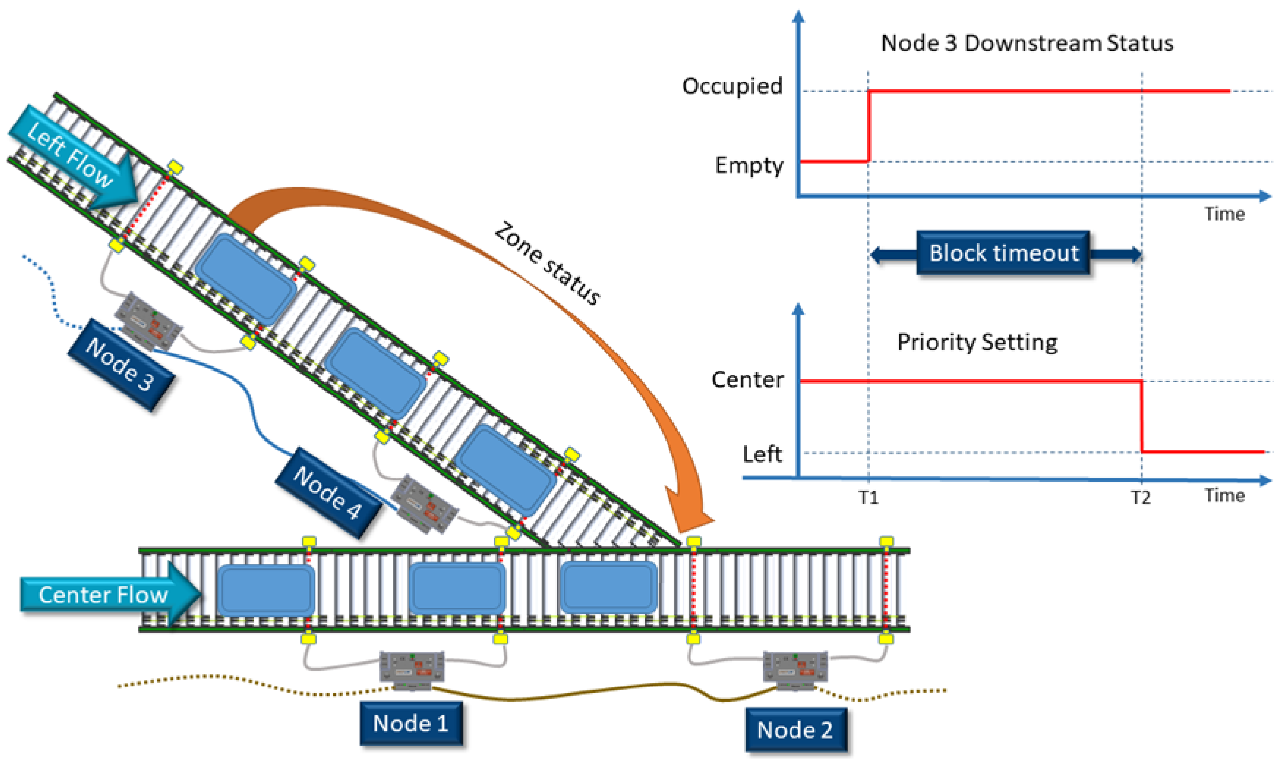


Figure A - Merge Line Full Block Timer

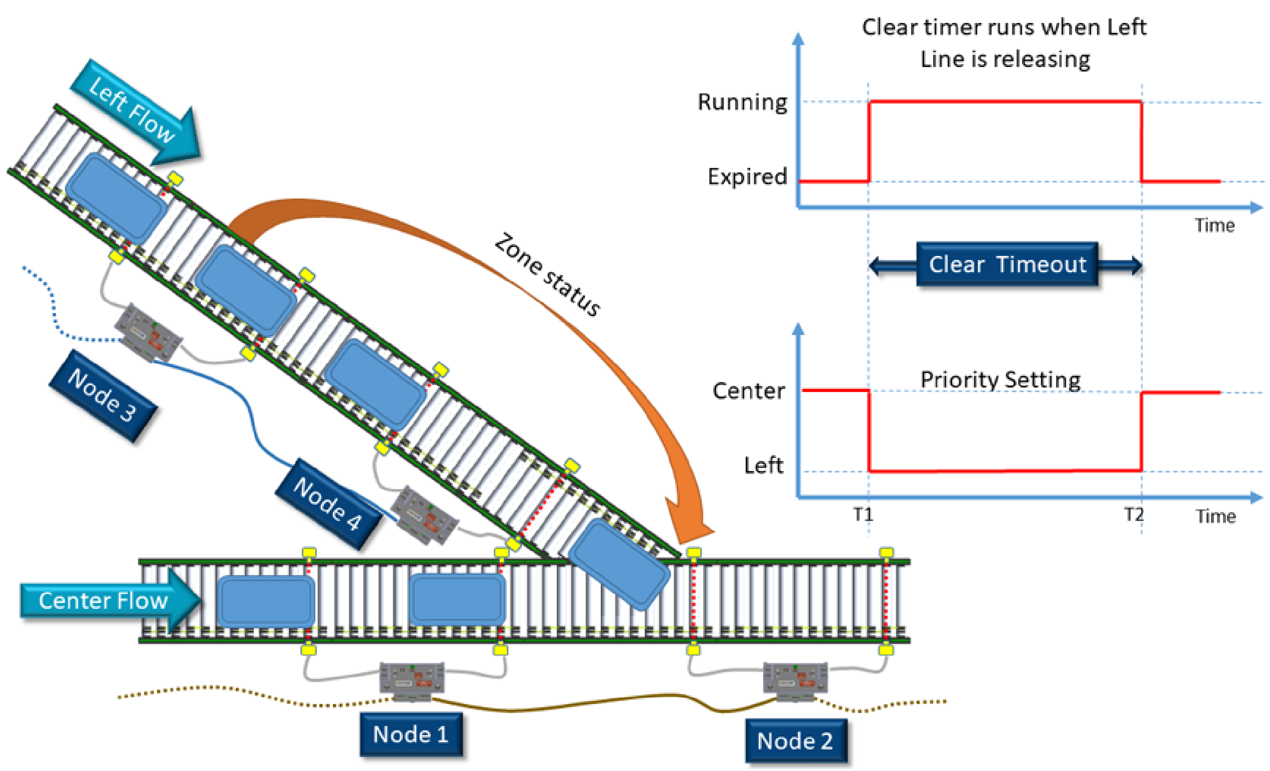
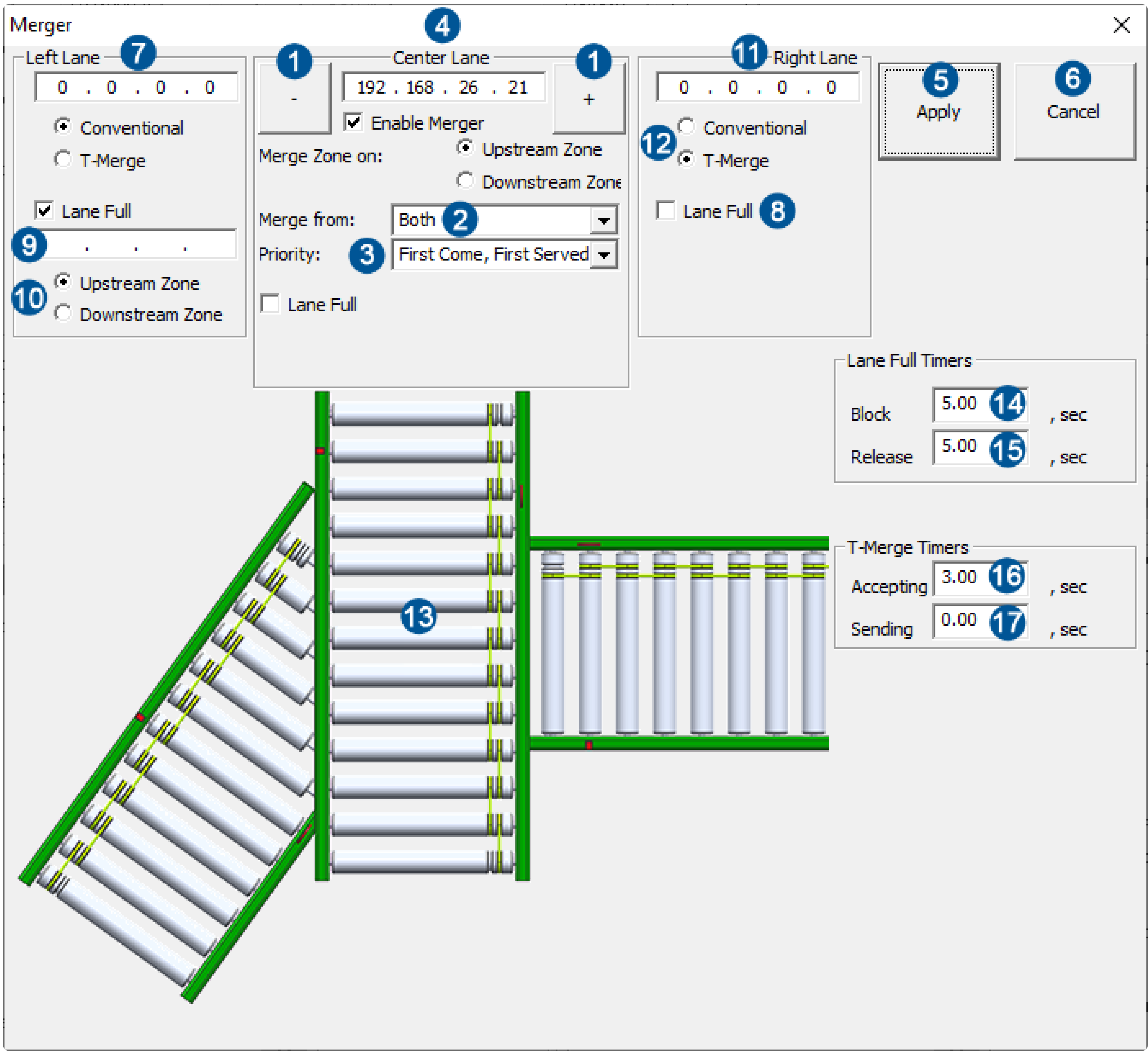


Figure B - Merge Line Full Clear Timer

9.10. Enabling ConveyMerge from EasyRoll

- Open *EasyRoll*
- Connect and navigate to the ConveyLinx Module that will control the *Merge Zone*.
- Press *CTRL+SHIFT+M* to open the *ConveyMerge Configuration Screen*



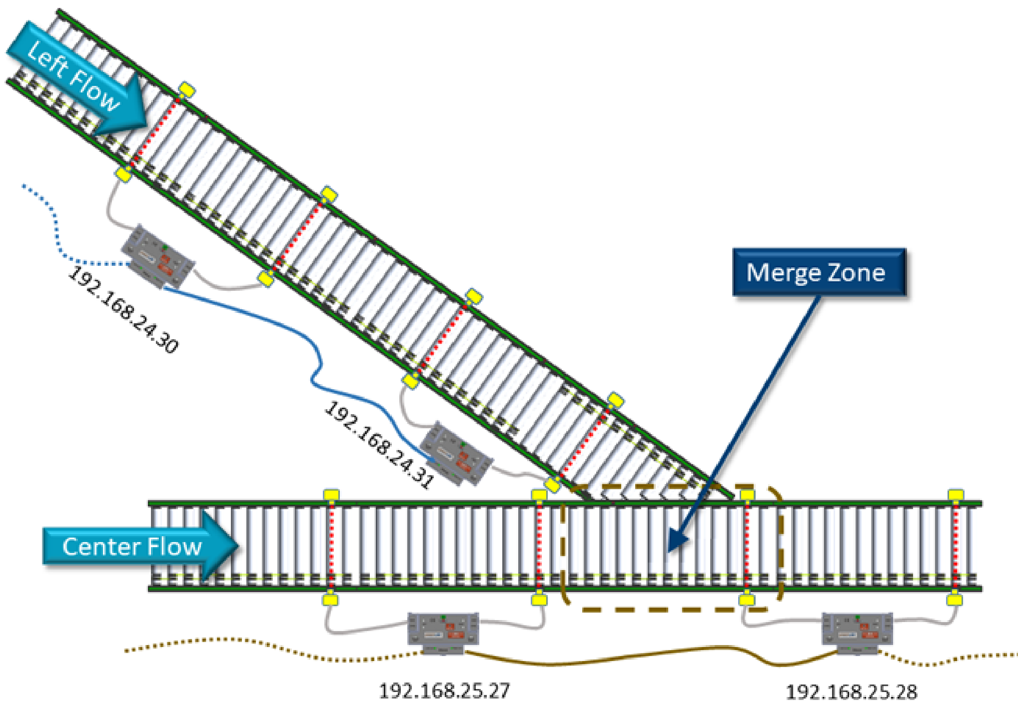
Item	Description
1	Navigation Buttons for Node selection

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 Module 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 Module
8	Checkbox to enable *Merge Line Full* functions
9	IP Address of the ConveyLinx Module 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 Module 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 Module
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

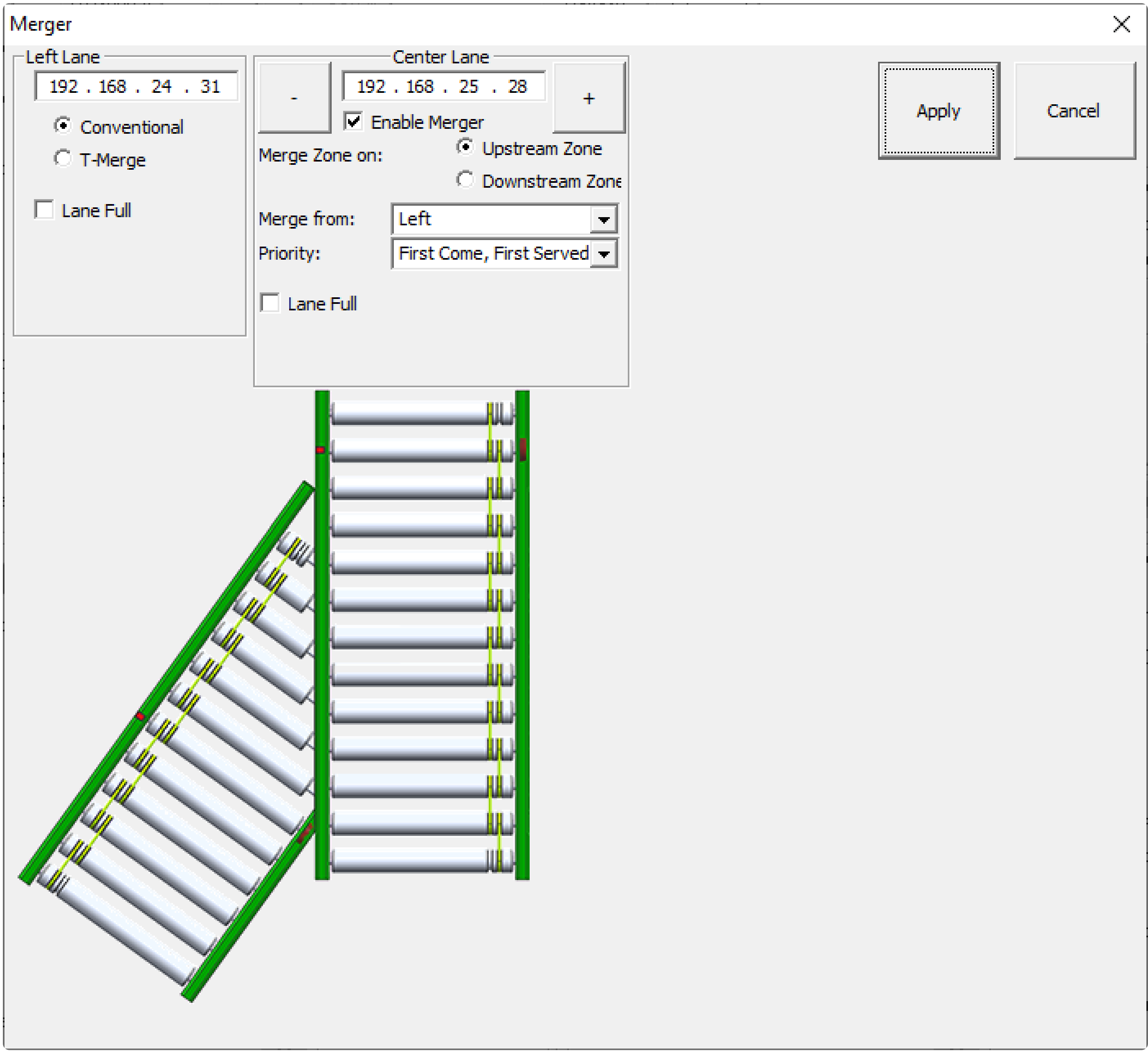
9.11. Conventional Spur Merge Example

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 Module for the *Left Line* is at 192.168.24.31. Here is the *ConveyMerge Configuration Screen* with these values entered



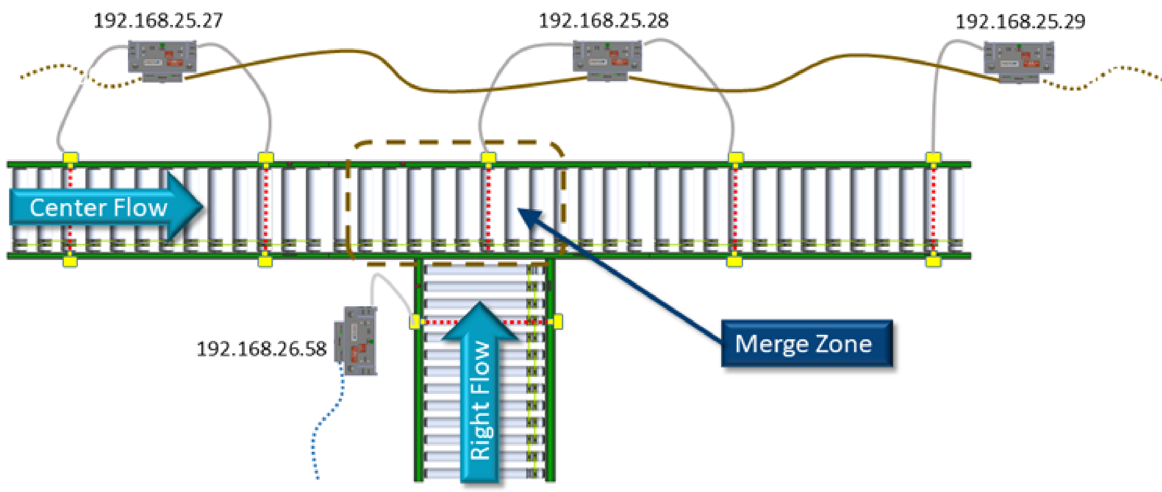
The last step is to click the *Apply* button to send the changes to the affected ConveyLinx Modules

> 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

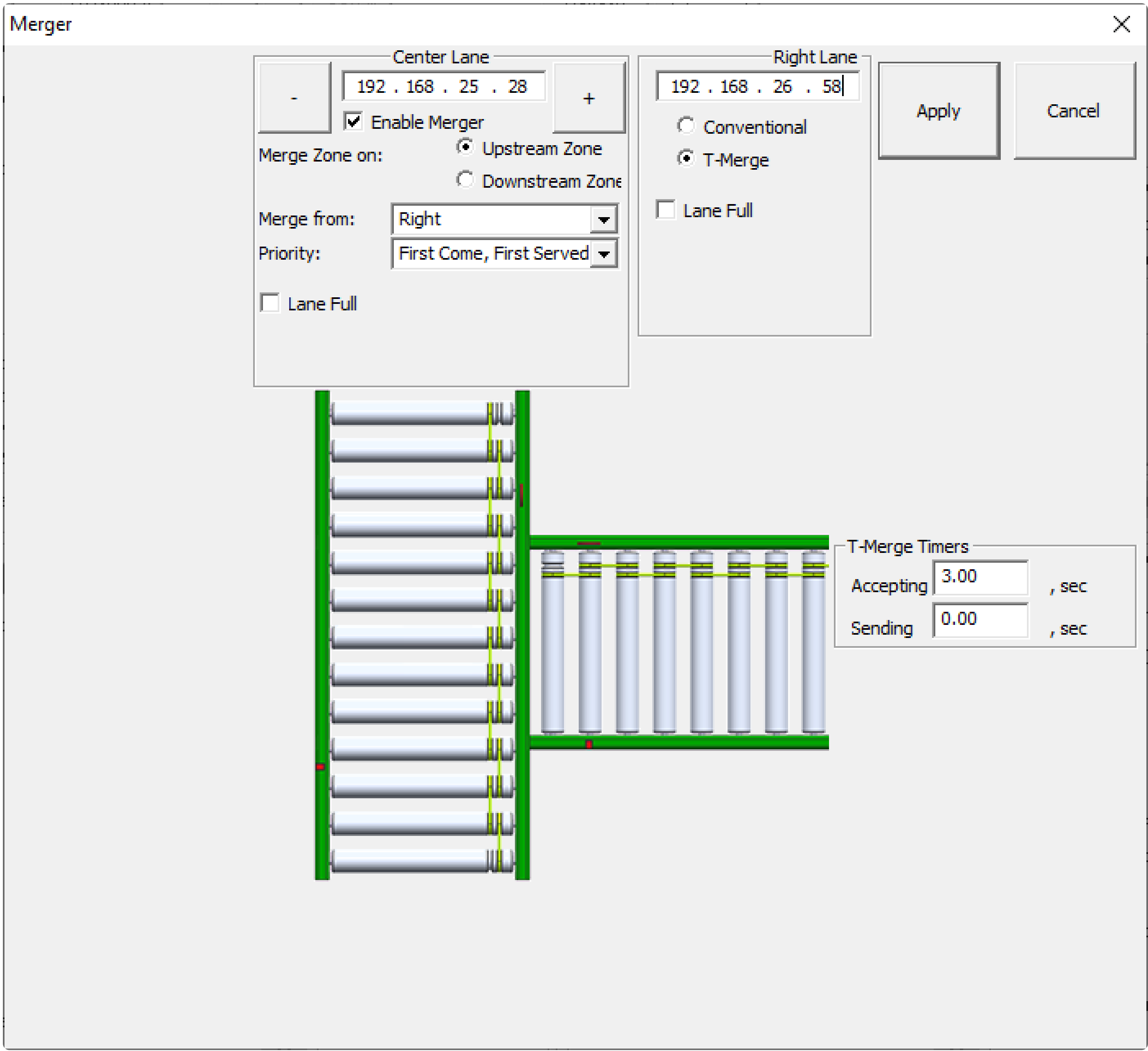
9.12. T-Merge Example

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



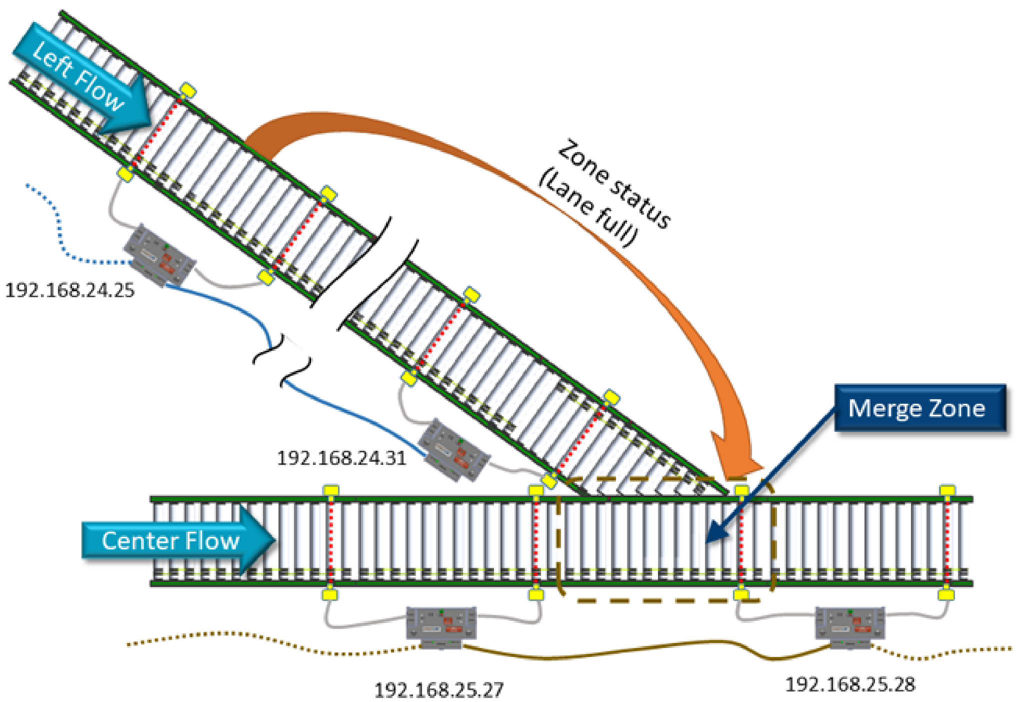
The *Merge Zone* is the upstream zone of the module at 192.168.25.28 and the most downstream ConveyLinx Module for the *Right Line* is at 192.168.26.58. Here is the *ConveyMerge Configuration Screen* with these values entered


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

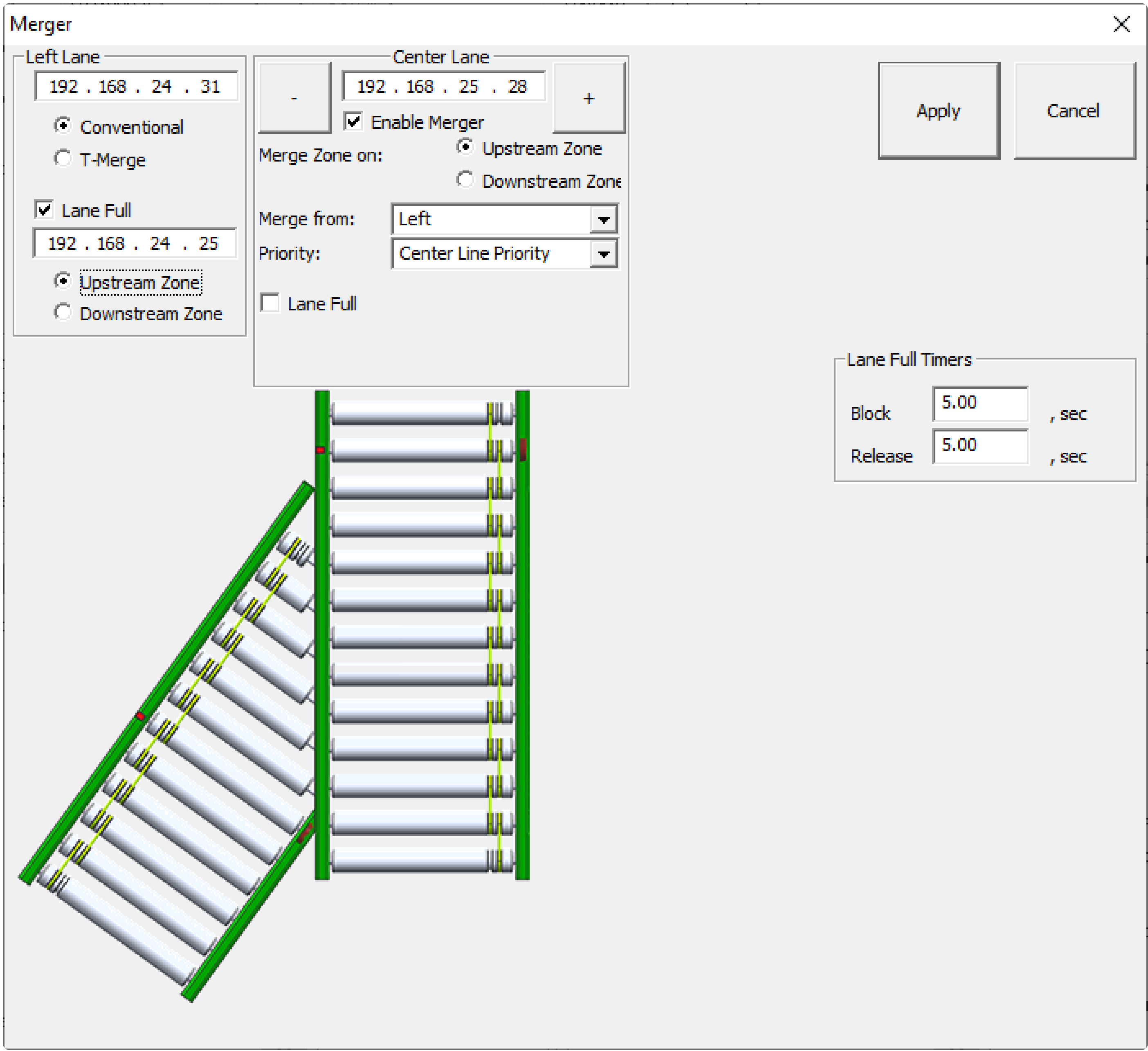
9.13. Merge Line Full Example

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

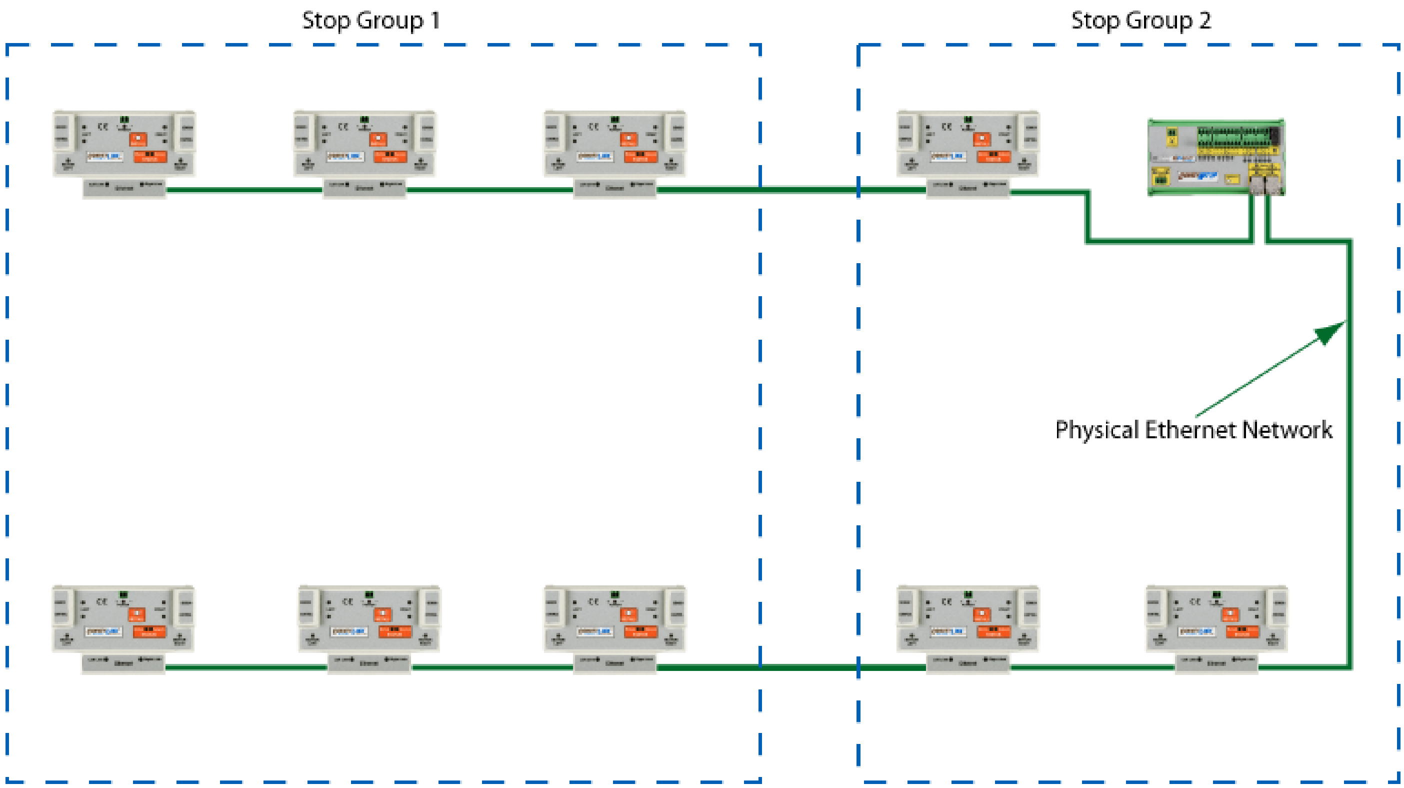


The *Merge Zone* is the upstream zone of ConveyLinx Module 192.168.25.28 and the *Left Line's* most downstream node is 192.168.24.31 and we want this line to have the *Merge Line Full* function enabled. On the *Left Line*, we would like for the upstream zone of ConveyLinx Module 192.168.24.25 to be the 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

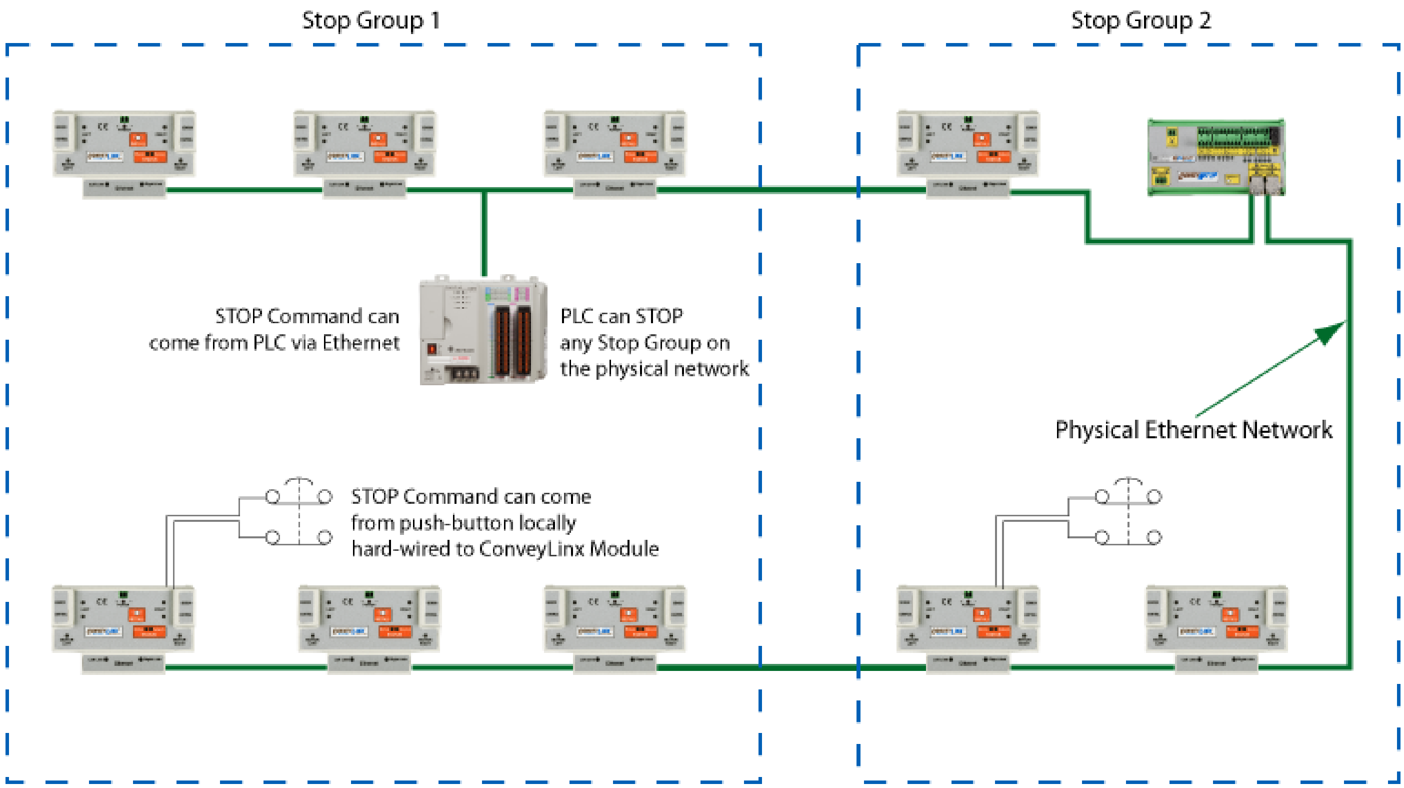
10. ConveyStop

ConveyStop is a *ConveyLinx* function that establishes a network based means to signal all *ConveyLinx Modules* with a *Stop Command* in order to physically stop their motion and/or output functions and remain in a stopped state until a separate *Start Command* network signal is received to return all connected modules back to their normal function. This is accomplished by utilizing the built-in Ethernet network that already inter-connects the ConveyLinx control system

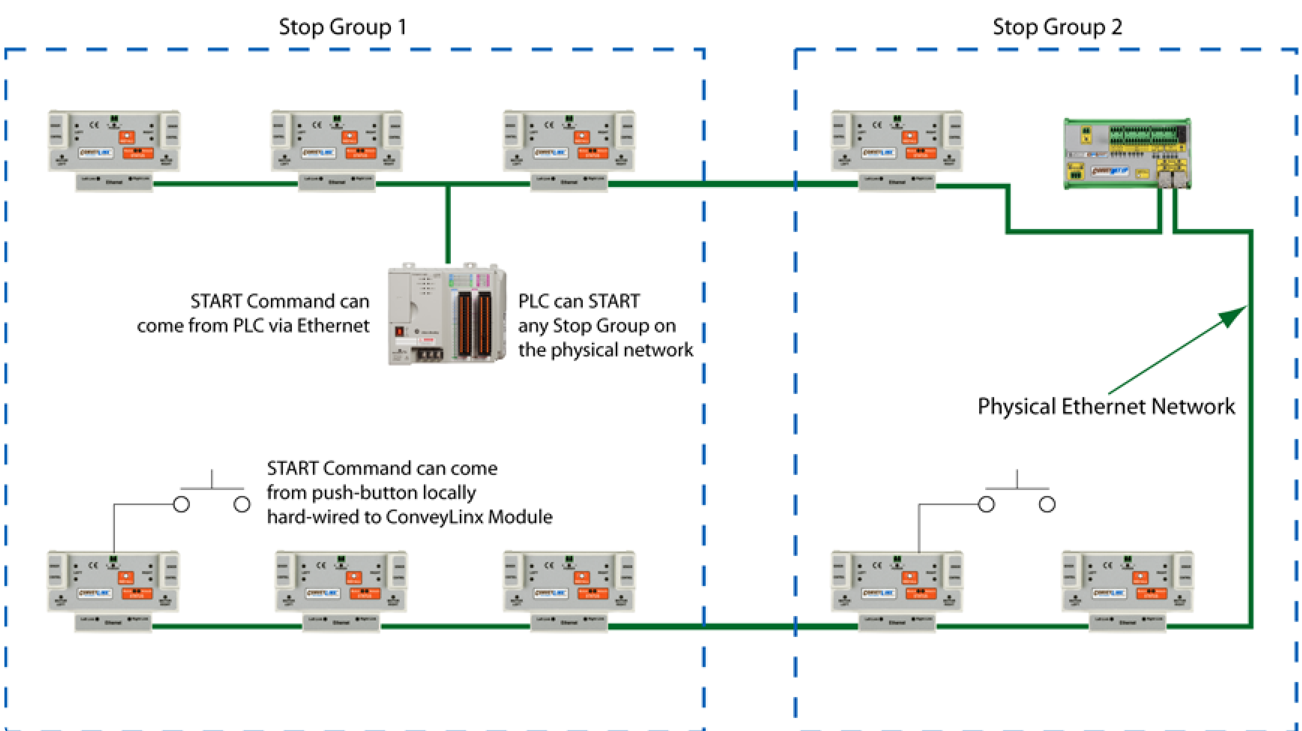


Single Physical Network with 2 Stop Groups

In a similar fashion to how ConveyLinx modules establish logical connections based upon conveyor flow; *ConveyStop*, when applied, establishes *Stop Group* connections between modules such that only modules within a given *Stop Group* are affected by STOP and START network commands. With ConveyStop, any given system can be segregated into as many *Stop Groups* as desired as long as all modules within a *Stop Group* are physically connected over Ethernet.



Typical STOP Command Sources



Typical START Command Sources

Each *Stop Group* is independent in that a *Stop Command* or *Start Command* occurring in one *Stop Group* does not affect the modules in another *Stop Group* even if they are on the same Ethernet network. In fact, modules within the same subnet that have logical conveyor flow connections can be in different *Stop Groups*. This means that a ConveyLinx-ERSC module in a non-stopped *Stop Group* will automatically detect if it is discharging into a stopped *Stop Group* and automatically inhibit product flow.

10.1. Benefits of ConveyStop

Replaces Separate Stop Circuit

One important feature of a *ConveryLinx* control system with *ConveyStop* enabled is that local operator buttons or switches can wire directly to a nearby *ERSC* or *CNIP* module. With these locally wired devices, anyone can initiate a controlled stop of a given *Stop Group* without requiring a separately wired stop circuit. Depending on system size and complexity, this can result in substantial savings in both installation cost and implementation time.

Detects Network Connectivity Loss

Another important feature of a *ConveyStop* enabled system is that loss of network communication and/or loss of connection to a PLC (if originally connected) will automatically cause a *Stop Command* to be initiated. Very often in large and/or complex network based control systems; the loss of communications is not easily detected nor does adjacent unaffected devices react in predictable ways. With *ConveyStop*, not only does every device stop upon communication loss, there can be information taken from the modules to help pin-point where the communication loss occurred.

Provides More Reliable Recovery

When any *Stop Command* is initiated in an *ERSC* module, not only is all motor commutation stopped, but the *ERSC* retains pertinent data on its status at the time the *Stop Command* was initiated. The *ERSC* will remember that it was discharging or accepting a package along with the packages tracking data. Upon getting a *Start Command*, the *ERSC* will pick up where it left off and attempt to finish what it was doing prior to the *Stop Command*.
This same scenario applies to situations when power is disconnected to all modules within a *Stop Group*. As part of the power loss procedure in all *ERSC* modules with *ConveyStop* enabled; certain pertinent data regarding module status and package tracking is saved to flash memory such that upon power-up and subsequent *Start Command*, each *ERSC* will attempt to complete what it was doing at the time of initial power loss.

10.2. Using ConveyStop in an Integrated Stop System

The means and methods of stopping automated equipment in an integrated material handling system are governed by many factors including, but not limited to:

- Location and usage of the equipment per the application
- Operator or non-maintenance personnel access to moving equipment
- Federal, State, and/or local ordinance or code
- Accepted electrical design practice
- Customer or end user preference

With all of these variables in play; *ConveyStop* cannot claim or be advertised as a de facto "approved" stopping method for all conveyor system stopping situations including an emergency stop situation. The purpose of this section is to define a recognized standard for control system stop classifications or categories and describe how *ConveyStop* can be applied for each.
It is the responsibility of the integrator of a ConveyLinx equipped system utilizing *ConveyStop* to assess all of the aforementioned factors before deeming a *ConveyStop* generated *Stop Command* as being applicable and suited for an emergency stop situation.

NFPA® 79

The National Fire Protection Association (NFPA®) 79 Electrical Standard for Industrial Machinery 2012 Edition contains accepted definitions for stopping functionality and emergency operations that are, in general, applicable to the conveyor and material handling industry.

Emergency Operations

NFPA® 79 section 9.2.5.4 Emergency Operations is as follows:

> (1) This section specifies the requirements for the emergency stop and the emergency switching-off functions of the emergency operations, both of which are initiated by a single human action
>
> (2) Once active operation of an emergency stop or emergency switching off actuator has ceased following a command, the effect of this command shall be sustained until it is reset. This reset shall be
> possible only at the location where the command has been initiated. The reset of the command shall not restart the machinery but only permit restarting
>
> (3) It shall not be possible to restart the machinery until all emergency stop

commands have been reset. It shall not be possible to re-energize the machinery until all emergency switching off commands have
been reset.

How ConveyStop Applies to Emergency Operations

The section 9.2.5.4 Item (1) criterion is met when physical buttons or switches are assigned and enabled with *ConveyStop*. Please note that a networked PLC or PC can generate a *Stop Command* and as such can occur programmatically and not necessarily by “single human action”. In this case, the integrator would be responsible for assuring that the PLC or PC based *Stop Command* is always initiated from a “single human action” integrated with said PLC or PC controls.
The section 9.2.5.4 Item (2) criterion is met when physical buttons or switches are assigned and enabled with *ConveyStop* and these devices are proper maintained contact type. As long as the button or switch is in its “emergency” position; *ConveyStop* will not issue or respond to any *Start Command* regardless of source (hard-wired button or networked PLC or PC). Also, resetting the physical device to its “non-emergency” state will not restart the modules to operation nor will this resetting of the device initiate any *Start Command*.
The section 9.2.5.4 Item (3) criterion is met in *ConveyStop* by design. If multiple physical buttons or switches are assigned and enabled with *ConveyStop*; all have to be placed into their “non-emergency” state before *ConveyStop* will issue or respond to any *Start Command* regardless of source. This is true regardless of which device first initiated the stop.

Stop Function

NFPA® 79 section 9.2.2 Stop Functions defines stop functionality as:

Stop functions shall operate by de-energizing that relevant circuit and shall override related start functions. The reset of the stop functions shall not initiate any hazardous conditions.

This section further defines three (3) Categories for stop functionality:

Category 0 – An uncontrolled stop by immediately removing power to the machines actuators
Category 1 – A controlled stop with power to the machine actuators available to achieve the stop then remove power when the stop is achieved
Category 2 – A controlled stop with power left available to the machine actuators

ConveyStop and Stop Function Definition

In general, *ConveyStop* follows the intent of the Stop Function definition for section 9.2.2. The manner in which “de-energizing that relevant circuit” it is accomplished differs between the *ERSC* and *CNIP* modules.

CNIP and Stop Function

Four of the six digital output circuits on the *CNIP* modules have their control power source internally wired to a contact relay. When a Stop Command is active, this relay is de-energized and control power is disconnected. In this state, the digital output circuit is de-energized regardless of the state of the logical output. This is a common stop circuit design for PLC based I/O systems.

ERSC and Stop Function

In situations where a Stop Command is active and control power is maintained to an *ERSC*, the de-energizing of the relevant circuit is accomplished by the on board processor. The *ERSC* utilizes a single processor and this processor directly controls (among other things) the power MOSFET transistor gates that commutate the motor. When a Stop Command is active, the processor places all MOSFET gates in their open or non-conductive state and then by-passes the task in the processor that produces motor commutation.

ConveyStop and Stop Function Categories

Category 2

Utilizing *ConveyStop* as designed and intended adheres to the criterion of Category 2. Maintaining control power to all ConveyLinx modules is desirable because they maintain their “pre stopped” state making for faster and more reliable recovery. Another added benefit of maintaining control power to all ConveyLinx modules is that the diagnostic features of both ConveyLinx and *ConveyStop* are available to PLC and/or PC (including the *ConveyStop* PC software monitoring capability) as an aid in troubleshooting and event logging.

Category 1 And Category 0

Both of these categories involve disconnecting power to ConveyLinx modules and either of these can be implemented on a ConveyLinx system with or without *ConveyStop* being enabled. If *ConveyStop* is implemented, the behavior would essentially be as described in section 0 Power Loss. *ERSC* modules will retain their state to flash memory as power is being dropped. When power is restored; each *Stop Group* affected by the power disconnect will have to receive a *Start Command* from either an assigned and enabled button or switch or from a network source (PLC or PC).
For *CNIP* modules, each is equipped with separate power terminals. One set of terminals is for module logic and input bus power and the other set powers only the output bus. In a Category 1 or Category 0 system, the motion producing control power can be disconnected from the *CNIP’s* output bus while leaving logic and input control power on. This is a typical strategy applied to PLC I/O systems and can be implemented with *CNIP* modules with or without *ConveyStop* installed and enabled. It must be noted that this strategy requires 2 separate power systems to be field wired.

What Happens When You Do Not Use ConveyStop?

Operationally, *ConveyStop* is not required for any ConveyLinx system to function. If *ConveyStop* is not implemented, disconnecting power does cause all motor motion to stop for *ERSC*'s and all outputs to be de-energized for CNIP's. However it is important to note that without *ConveyStop* enabled and when power is restored; *ERSC* modules will be enabled to run and if conveyor conditions dictate, motors will run and packages will begin to move with no other or separate start signal or command.

Another important note is that ConveyLinx devices are Ethernet based, and upon cycle of power, modules will individually reinitialize at different rates making exact power up behavior unpredictable.

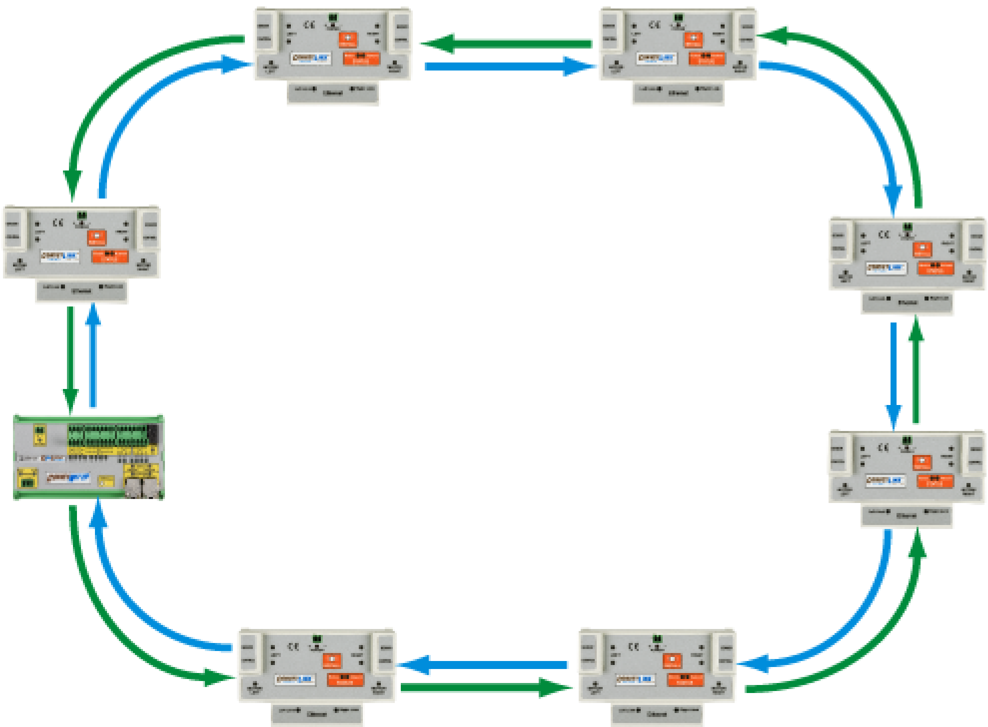
Integrator Responsibility

By no means is NFPA® 79 the only specification or criteria for defining the stop function of an automated system. The NFPA® 79 is a general standard for the USA and the descriptions above are based upon general experience for US installations.

The bottom line is that it is always up to the integrator to understand and adhere to the applicable specifications, codes, and standards on a per system basis. *ConveyStop* can be a valuable tool to achieve desired system stop functionality, enhanced diagnostics, and lower installed cost.

10.3. ConveyStop Architecture

The ability to respond to *Stop* Commands and *Start Commands* is built in to every *ERSC* and *CNIP* module. The activation of the *ConveyStop* function requires the *ConveyStop* PC software package to first define one or more *Stop Groups* of modules and then enable the *ConveyStop* functions in these modules. When enabled, the modules create 2 logical network connection rings within their configured *Stop Group* as shown:



! Please note this is only a logical group of connections between modules. ConveyLinx modules within a *Stop Group* are NOT physically connected in a closed loop

These connections create two logical rings. Each module expects to get data and heartbeat keep-alive messages every 200 ms over both connection rings. If any single module in the group does not receive data or heartbeat message either of its neighbors; the module will immediately disable the gates of its MOSFET transistors and sends a message to all other module over the 2 connection rings to do the same. The disabling of the MOSFET gates immediately halts any possible commutation and rotation of connected motors.

For ConveyLinx ERSC modules, the ConveyStop *Stop* Command causes all motor commutation to immediately stop. For ConveyLinx ConveyNet I/P (*CNIP*)) *modules, the _*Stop** Command causes the *CNIP* processor to de-energize the output bus relay, thus disconnecting control power to the affected discrete outputs.

Once a *Stop Group* has been stopped and any and all conditions that caused the *Stop Command* have been cleared; the *Stop Group* has to be given a *Start* command to restart the function of the logical ring connections.

10.4. ConveyStop Software

Before any *ConveyStop* functionality can be realized; you must first configure one or more *Stop Group*_s on your system. In order to do this, you must have the _*ConveyStop* Software. This section details how to:

- Install and enable the software
- Create a new Project
- Create *Stop Groups*
- Assign *STOP* Command inputs
- Assign *START* Command inputs
- Utilize monitoring functions

The *ConveyStop* PC software connects to a ConveyLinx network and shows all available *ERSC* and *CNIP* devices. From this list, the system integrator creates and populates one or more *Stop Groups*. Within each *Stop Group*, one or more *ERSC*’s can be configured to have either its Left or Right Control Port assigned to contain a hard-wired button or switch to issue a *STOP* Command. Similarly, any *CNIP* module can be similarly assigned a hard-wired button or switch to a specific dedicated input to issue a *STOP* Command. Also within each *Stop Group*, one or more *ERSC*’s can be configured to have either its Left or Right Control Port assigned to contain a hard-wired button or switch to issue a *START* Command. Similarly, any *CNIP* module can be similarly assigned a hard-wired button or switch to a specific dedicated input to issue a *START* Command.

10.4.1. User Accounts

Guest

Guest access allows access to *ConveyStop* software monitoring capabilities only and does not allow any access to add, delete, or modify any items

Administrator

Only *Administrator* accounts are allowed to modify any configuration or assignments. Administrator access is required to:

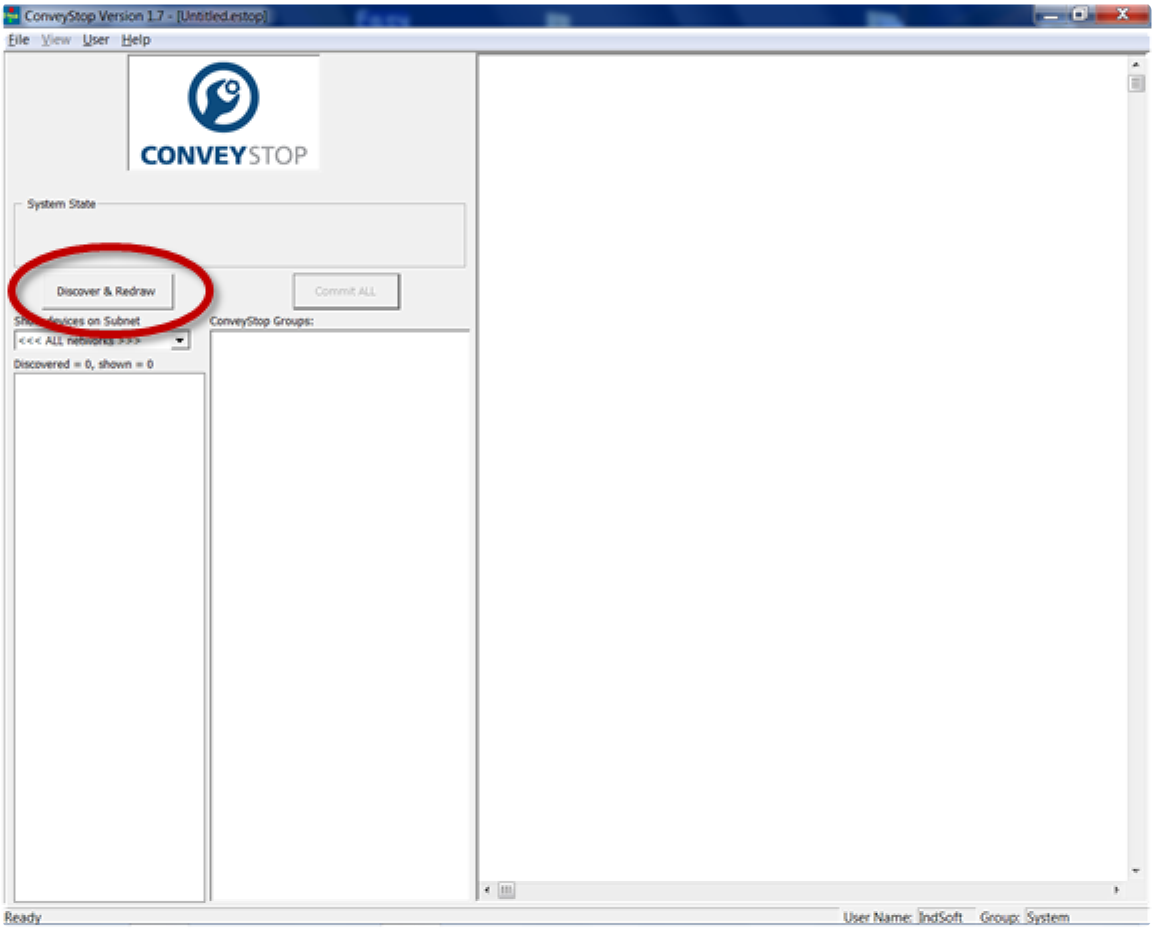
- Add or remove modules from a *Stop Group*
- Add or Remove *Stop Groups*
- Add or Remove *STOP* Command inputs
- Add or Remove *START* Command inputs

10.4.2. Creating a New Project

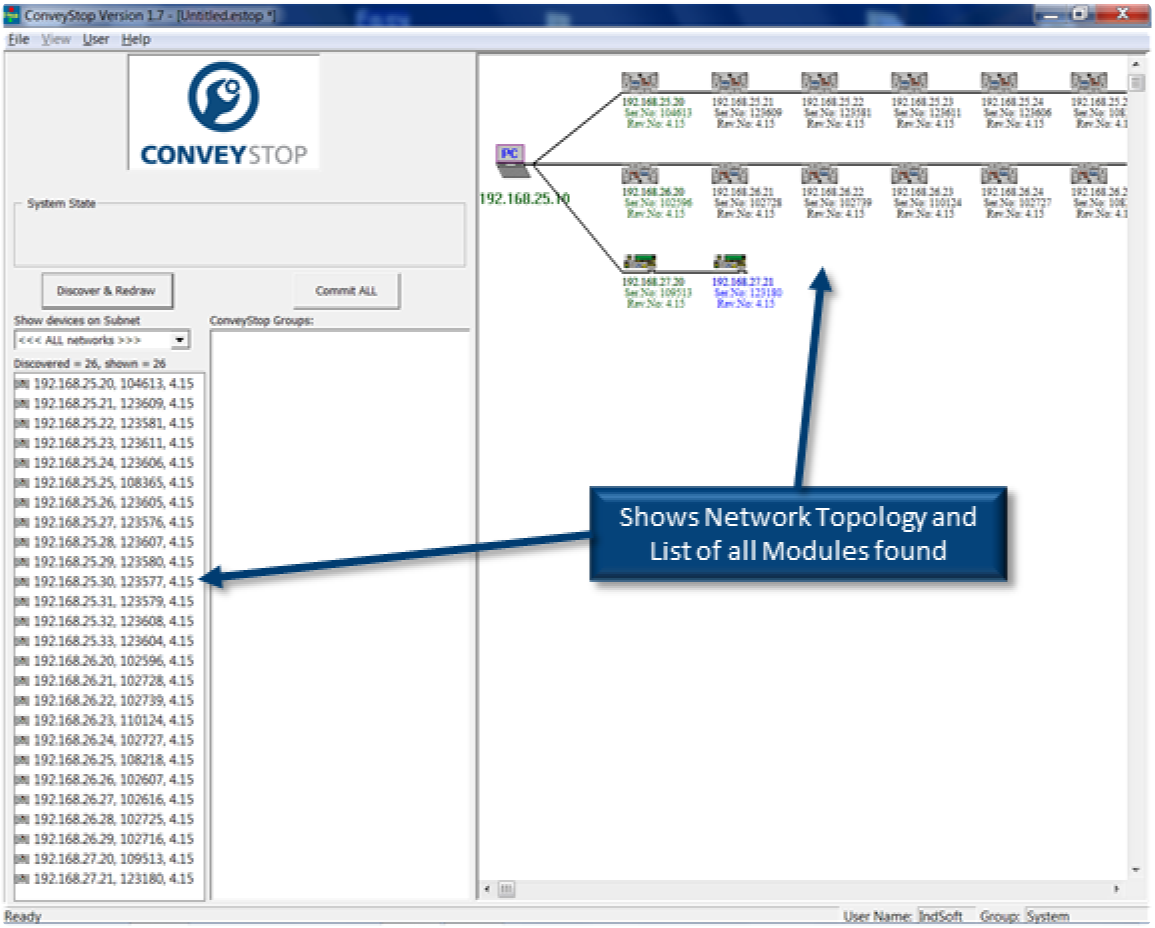
Create a new blank project by selecting “New” from the File menu.

10.4.3. Discovering Modules

Starting with a blank project as shown below, click the “Discover and Redraw” button will cause *ConveyStop* to search for any and all ConveyLinx devices it can find.



After a few seconds depending on the number of devices located, a list of discovered devices will be displayed in the window on the far left. *ConveyStop* will also create a graphical image of its discovery as shown:

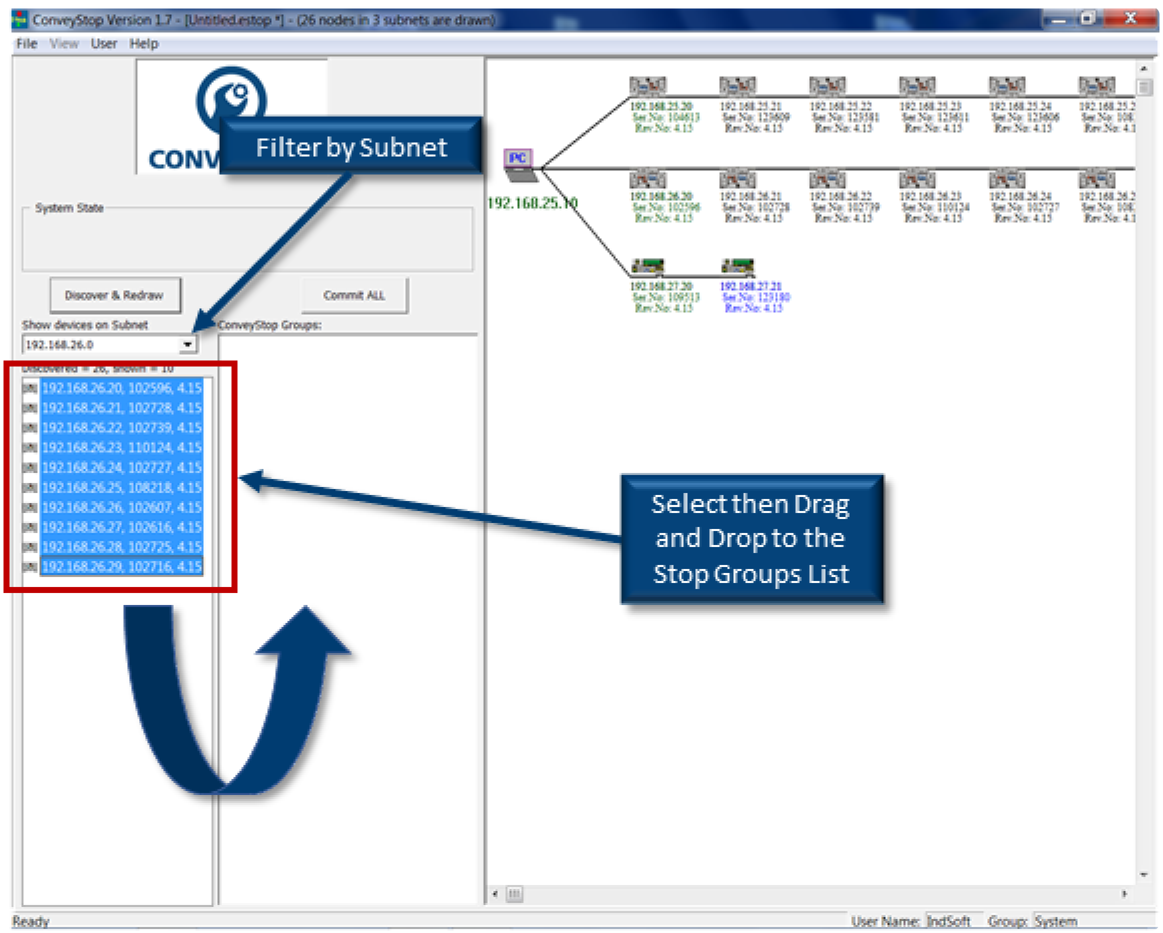

ConveyStop Version 1.7 - [Untitled.estop *]
File View User Help
CONVEYSTOP
System State
Discover & Redraw
Commit ALL
Show devices on Subnet
<<< ALL networks >>>
ConveyStop Groups:
Discovered = 26, shown = 26
192.168.25.20, 104613, 4.15
192.168.25.21, 123609, 4.15
192.168.25.22, 123581, 4.15
192.168.25.23, 123611, 4.15
192.168.25.24, 123606, 4.15
192.168.25.25, 108365, 4.15
192.168.25.26, 123605, 4.15
192.168.25.27, 123576, 4.15
192.168.25.28, 123607, 4.15
192.168.25.29, 123580, 4.15
192.168.25.30, 123577, 4.15
192.168.25.31, 123579, 4.15
192.168.25.32, 123608, 4.15
192.168.25.33, 123604, 4.15
192.168.26.20, 102596, 4.15
192.168.26.21, 102728, 4.15
192.168.26.22, 102739, 4.15
192.168.26.23, 110124, 4.15
192.168.26.24, 102727, 4.15
192.168.26.25, 108218, 4.15
192.168.26.26, 102607, 4.15
192.168.26.27, 102616, 4.15
192.168.26.28, 102725, 4.15
192.168.26.29, 102716, 4.15
192.168.27.20, 109513, 4.15
192.168.27.21, 123180, 4.15
PC
192.168.25.10
Shows Network Topology and List of all Modules found
Ready
User Name: IndSoft
Group: System

10.4.4. Creating Stop Groups

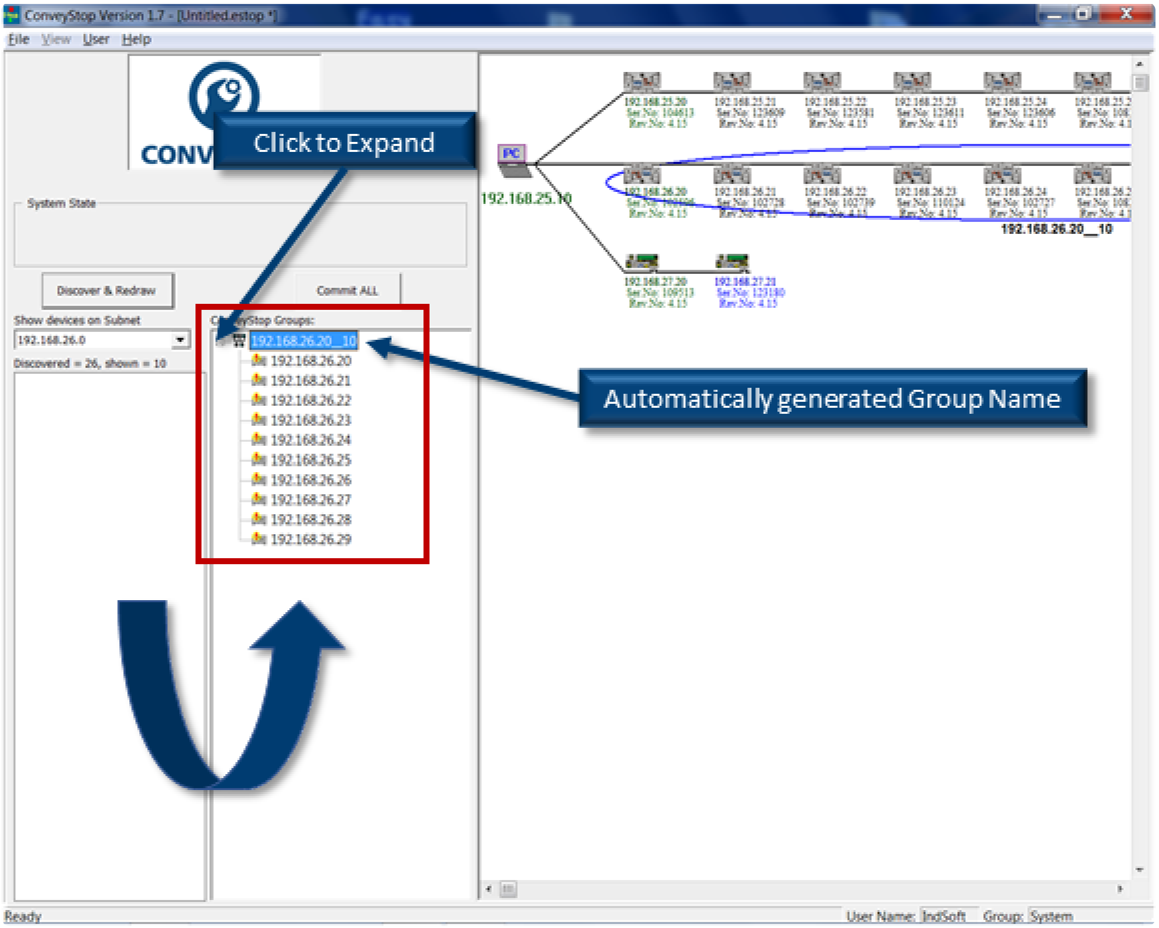
In this example, there are 3 *subnets* – one with 10 *ERSC* modules, one with 14 *ERSC* modules, and one with 2 *CNIP* modules. We want to create 2 *Stop Groups* – one with the 10 *ERSC* subnet and the other one with the 14 *ERSC* subnet combined with the 2 *CNIP* subnet.

First Stop Group

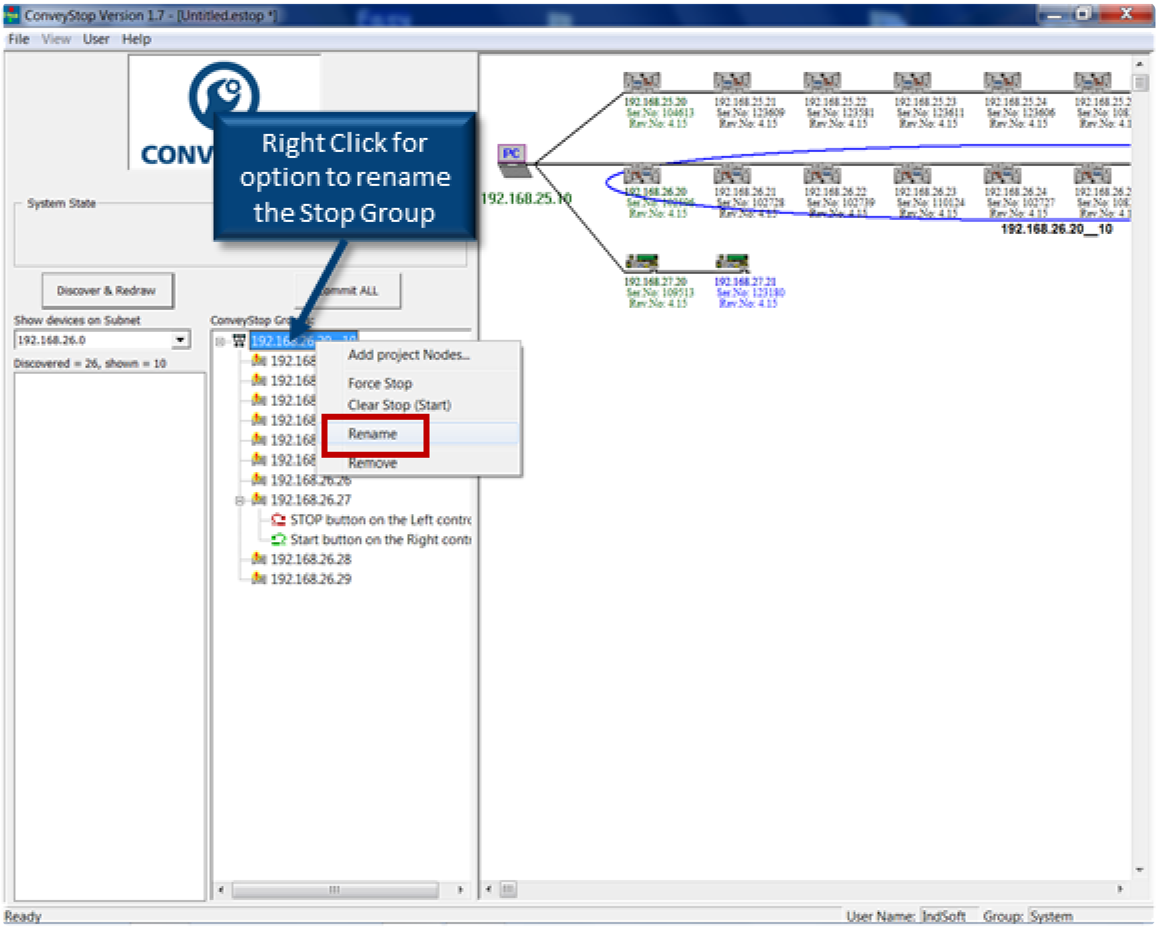
First we use the “Show devices on Subnet” to filter the list for the 10 ERSC subnet. Then we select all the items in the list as shown:



Then drag the selection into the ConveyStop Groups list window as shown. This creates a *Stop Group* from the selected modules and automatically generates a group name.



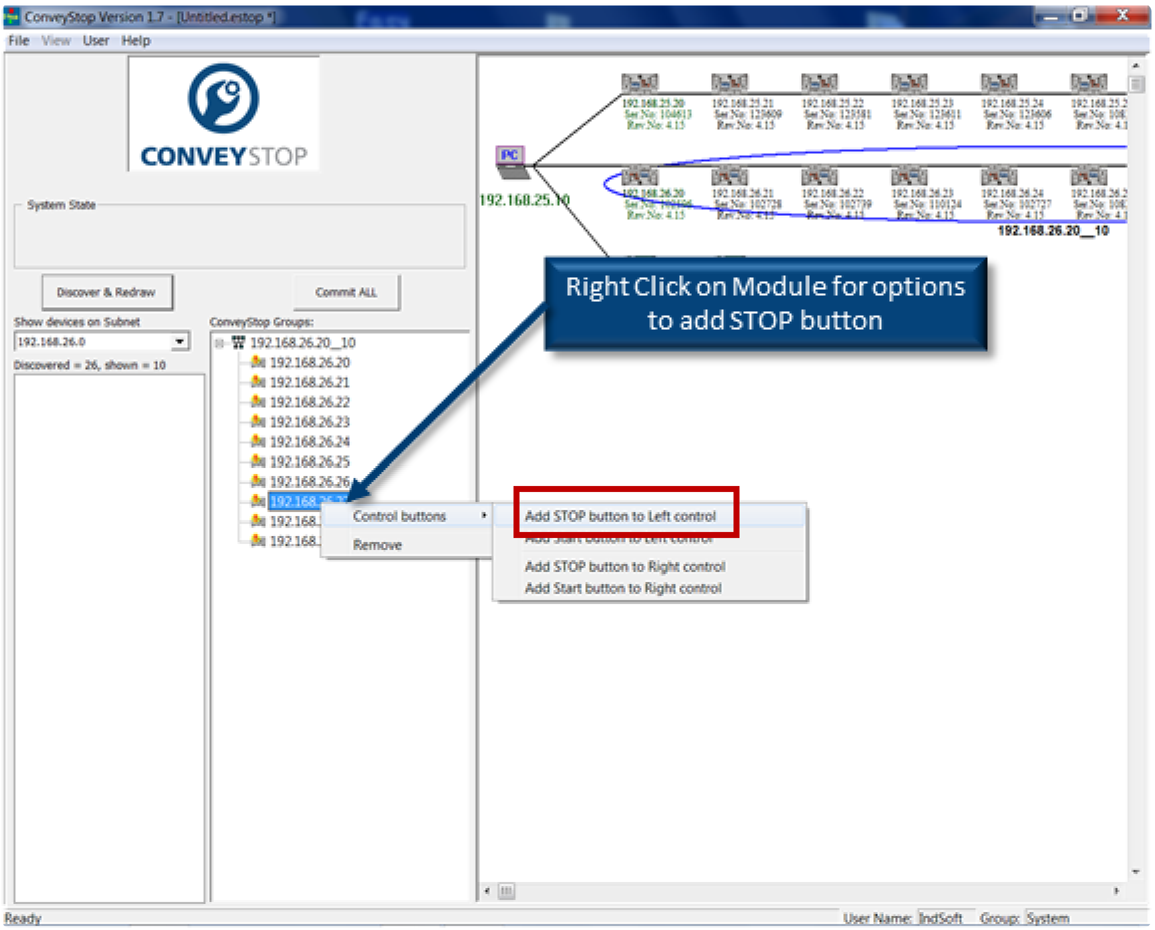
Right click on this name to give it as meaningful name for your project. In this example we will call it “Line A”. You can also expand and collapse the detail of a *Stop Group* by clicking the + or – icon to the left of the group name:


ConveyStop Version 1.7 - [Untitled.estop *]
File View User Help
Right Click for option to rename the Stop Group
System State
Discover & Redraw
Commit ALL
Show devices on Subnet
192.168.26.0
Discovered = 26, shown = 10
ConveyStop Groups
Add project Nodes...
Force Stop
Clear Stop (Start)
Rename
Remove
192.168.26.26
192.168.26.27
STOP button on the Left contro
Start button on the Right contr
192.168.26.28
192.168.26.29
PC
192.168.25.10
192.168.26.20__10
Ready
User Name: IndSoft
Group: System

10.4.5. Add Stop and Start Buttons

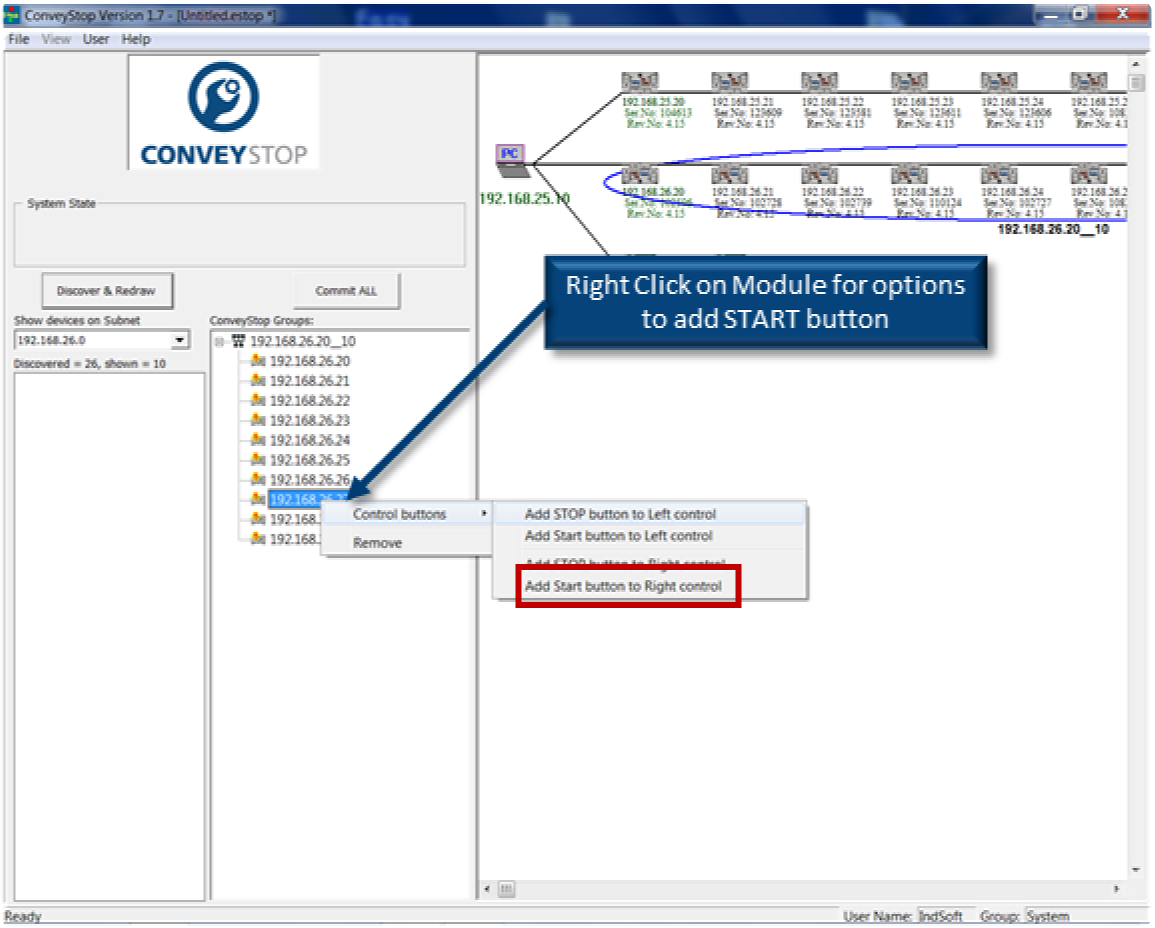
Stop Button

In our example, we want to add a *Stop* Button to node 8 at address 192.168.26.27 and we want the button to connect to the Left Control Port. You do this by right clicking on the desired node and selecting the option from the pop-up menu as shown:



Start Button

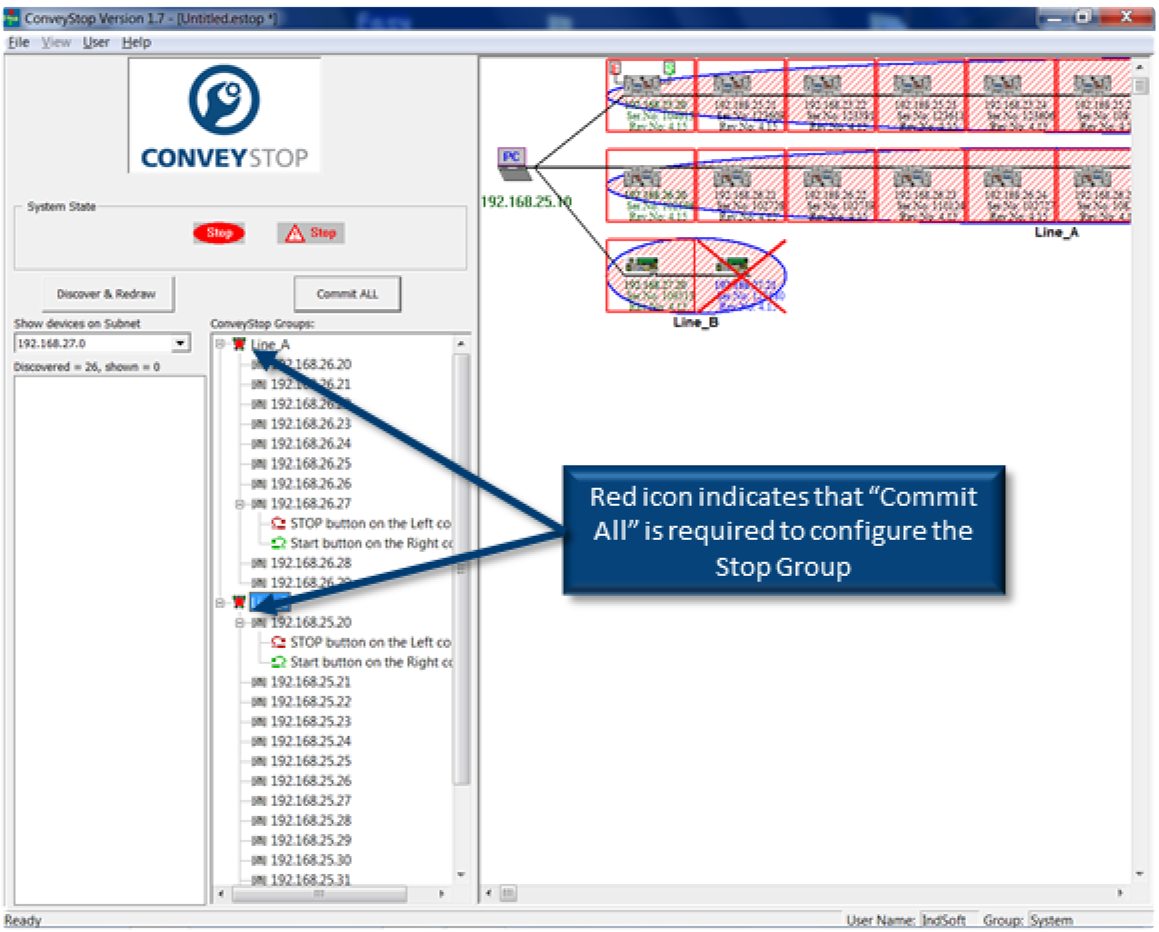
In our example we want to add a *Start* button to the Right Control Port of the same node. Right click on the node and select the desired option from the pop-up menu as shown:


ConveyStop Version 1.7 - [Untitled.estop *]
File View User Help
CONVEYSTOP
System State
Discover & Redraw
Commit ALL
Show devices on Subnet
192.168.26.0
Discovered = 26, shown = 10
ConveyStop Groups:
192.168.26.20__10
192.168.26.20
192.168.26.21
192.168.26.22
192.168.26.23
192.168.26.24
192.168.26.25
192.168.26.26
Control buttons
Remove
Add STOP button to Left control
Add Start button to Left control
Add Start button to Right control
192.168.25.10
192.168.26.20__10
Right Click on Module for options to add START button
Ready
User Name: IndSoft
Group: System

10.4.6. Commit All Button

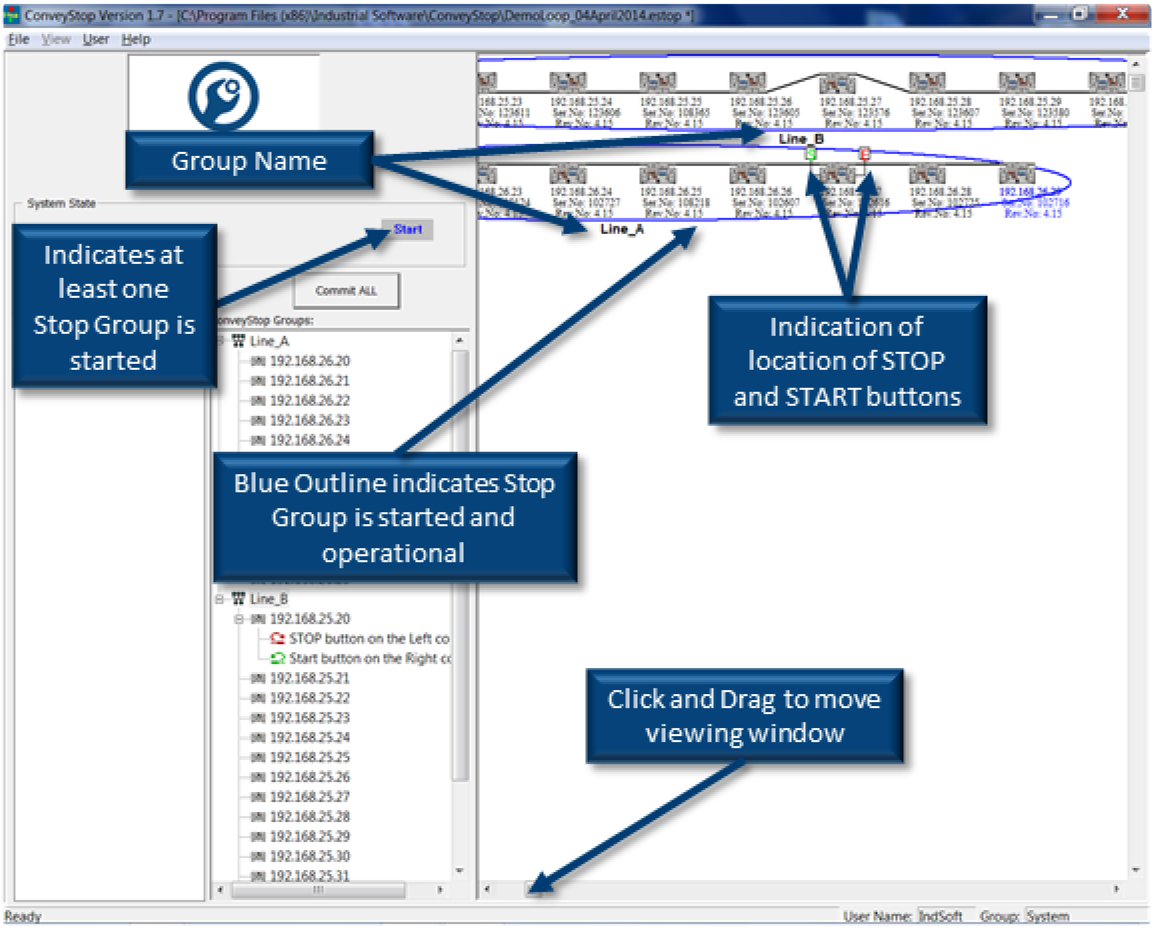
Completing Our Example

By repeating the same procedures, we want to add a 2nd *Stop Group* consisting of the remaining unused devices from the Discover window. We want to rename this *Stop Group* as “Line_B”. We then want a *STOP* Button on node 1 at I.P. Address 192.168.25.20 connected to the Left Control Port and a *START* Button on the same node connected to the Right Control Port. When this is complete it should look like as shown:



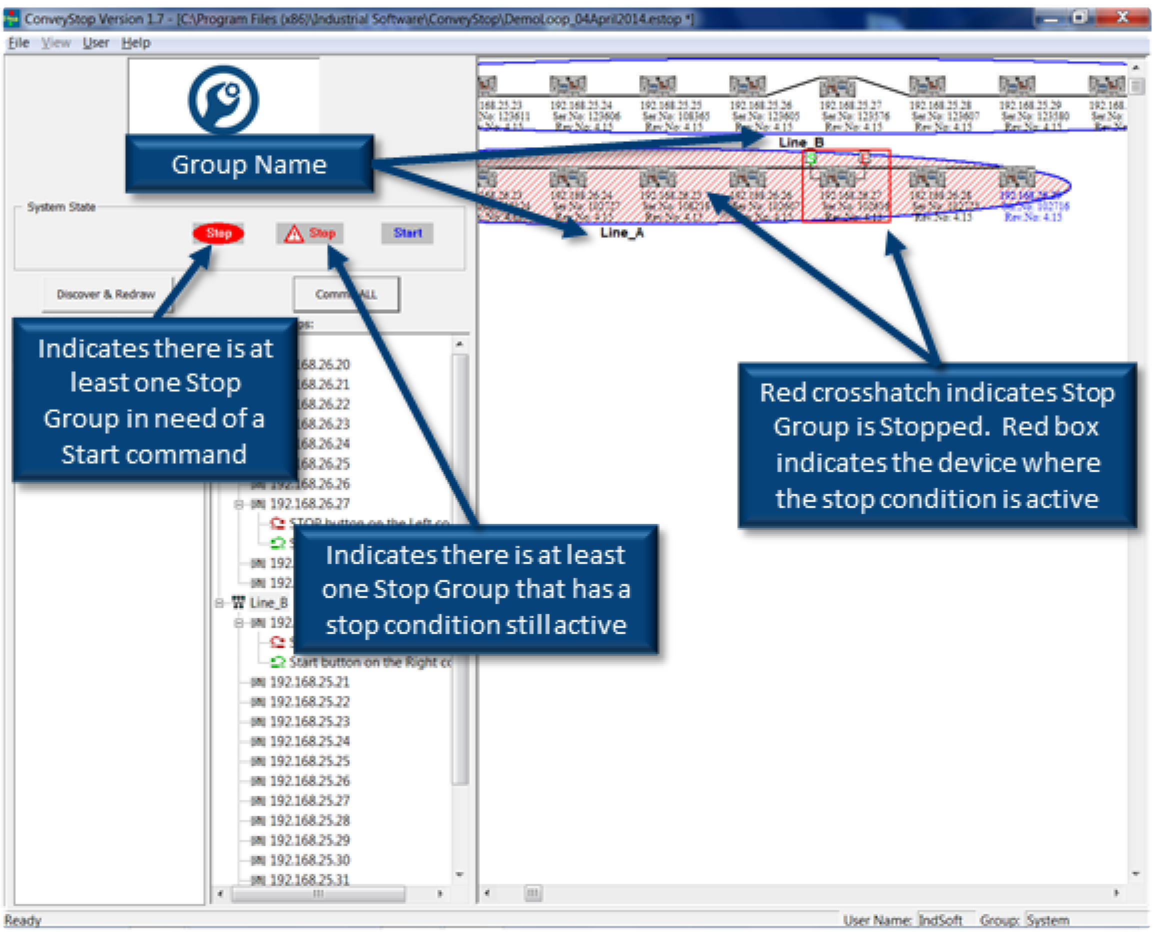
Commit All Button

The process is not complete because we have only configured *ConveyStop* within the software; we still need to *Commit* the functionality to the physical ConveyLinx modules. Clicking the “Commit All” button sends the current configuration of *Stop Groups*, buttons, etc. to the actual devices. When complete, *ConveyStop* screen displays the status of the connected devices. Shown below is what the screen should look like for our example when both *Stop Groups* are operational and have no stop conditions active and no stop groups awaiting Start:


Group Name
Indicates at least one Stop Group is started
Indication of location of STOP and START buttons
Blue Outline indicates Stop Group is started and operational
Click and Drag to move viewing window
Line_B
Line_A
System State
Start
Commit ALL
Ready

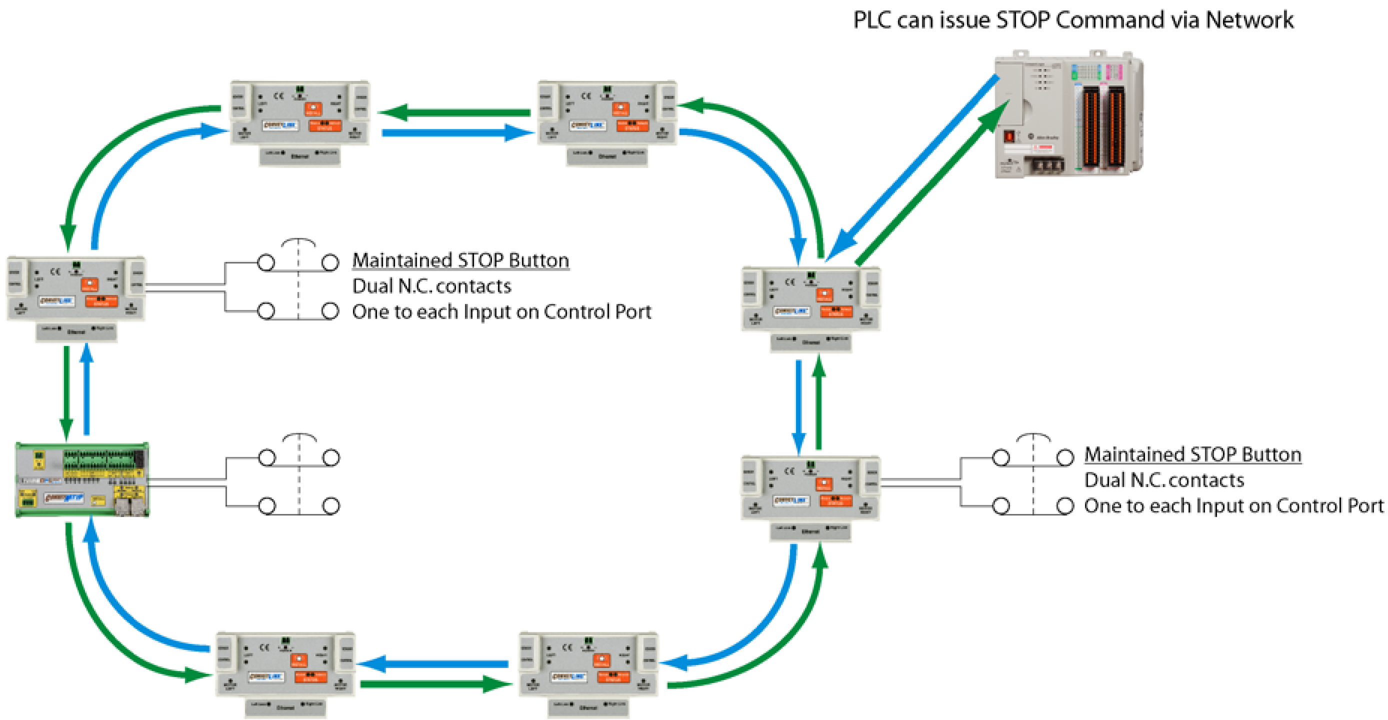
10.4.7. Status Monitoring

The *ConveyStop* screen gives indication when there is a Stop active and the individual node(s) where the Stop was initiated. The System State area on the screen provides quick status at a glance as to whether any group is in an active stopped state with the stop condition still true, in a stopped state with the condition reset and awaiting Start, and whether at least one Stop Group is started. Below shows our example system with Line_A in both a stopped state and the stop condition is still active. It also shows Line_B in a started state with no stop conditions active:



10.5. Issuing a Stop Command

Typical methods for an operator or the control system to issue a *Stop* Command is either via hardwired Stop push-button or via network command from PLC as shown:



Hard-Wired Push-Button

ERSC Module

You can connect *Stop Command* buttons to either the Left or Right Control Ports on an ERSC in the *Stop Group*. *Stop* buttons require 2 sets of contacts and connect to the 2 inputs on the control port. If ANY of these contacts become opened on a given module; it will generate a *STOP Command* to ALL modules within the group over the 2 communication rings.

CNIP Module

For the *CNIP* module, there are dedicated inputs to use to connect a *Stop Command* button. See Wiring Examples for details on the inputs reserved for this function

Remote PLC

If a PLC is connected to at least one module in a *Stop Group*; it can issue a *Stop Command* to that module and all modules within that group will recognize this as a *Stop Command*. Even if the PLC is connected to more than one module in a *Stop Group*, it only has to issue the *Stop*

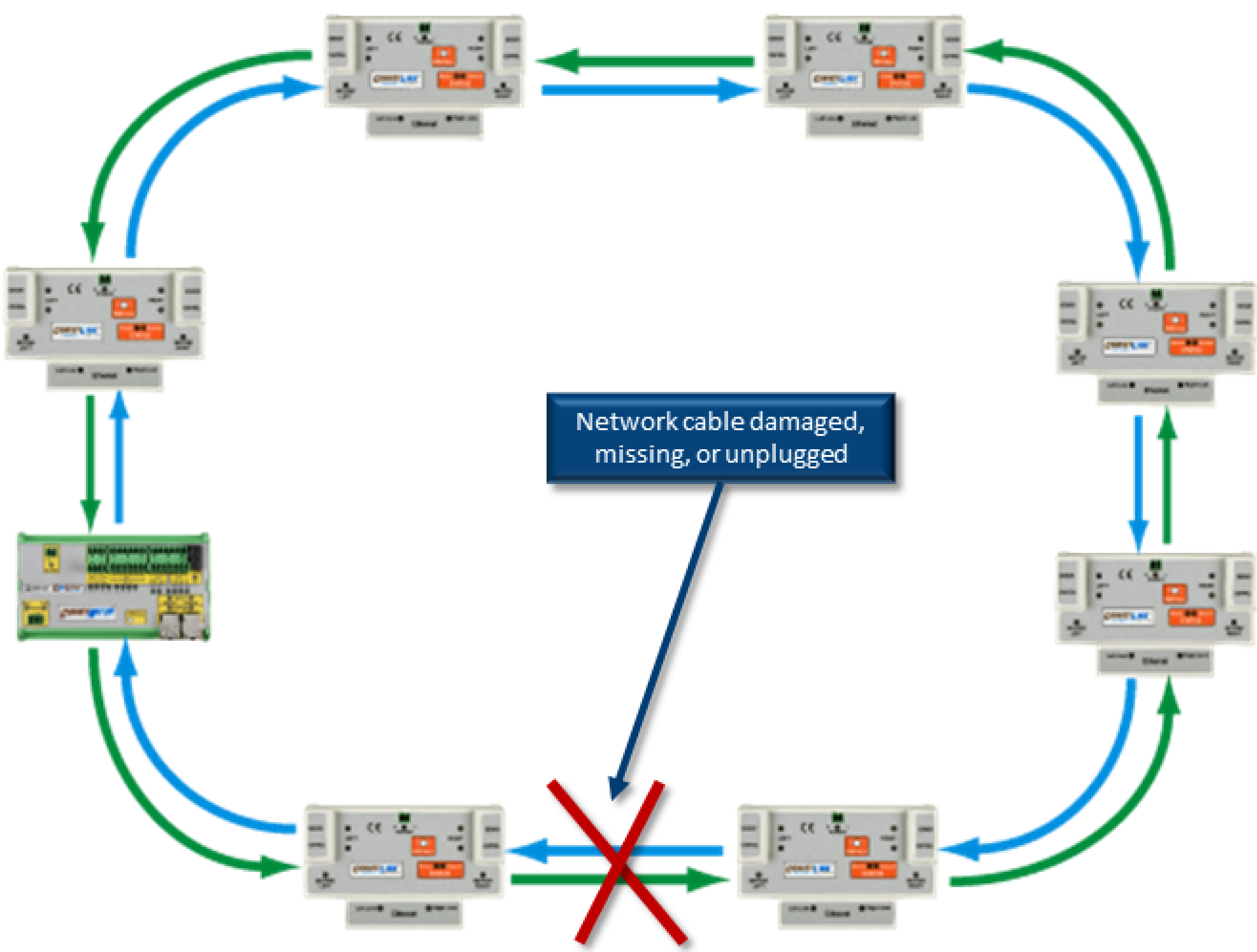
Command to only one module in order to *Stop* the whole *Stop Group*.

10.5.1. Other Conditions that cause Stop

There are other conditions, not necessarily initiated by an operator or the control system, that will cause a *Stop Command* to be interpreted by a given *Stop Group*

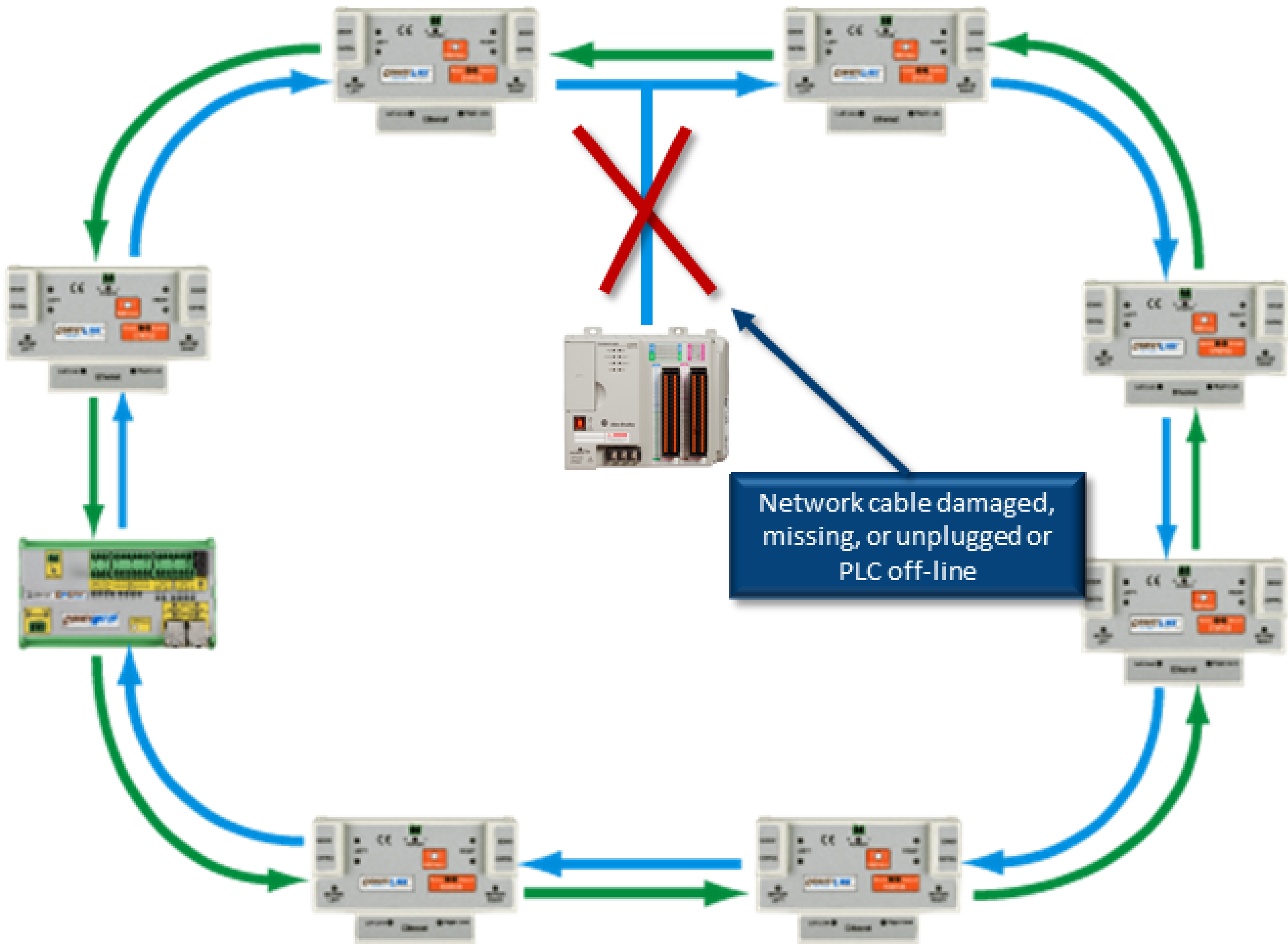
Missing Connection

If any single device (ERSC or ConveyNet I/P) is missing from a configured *Stop Group* due to for example unplugged or damaged cable; a *Stop Command* will be issued to all modules in the *Stop Group*



Missing PLC

If a PLC establishes a *structured instance* connection to any single device (ERSC or ConveyNet I/P) within a *Stop Group* and that connection is subsequently lost to the PLC, a *Stop Command* will be issued to all modules in the *Stop Group*



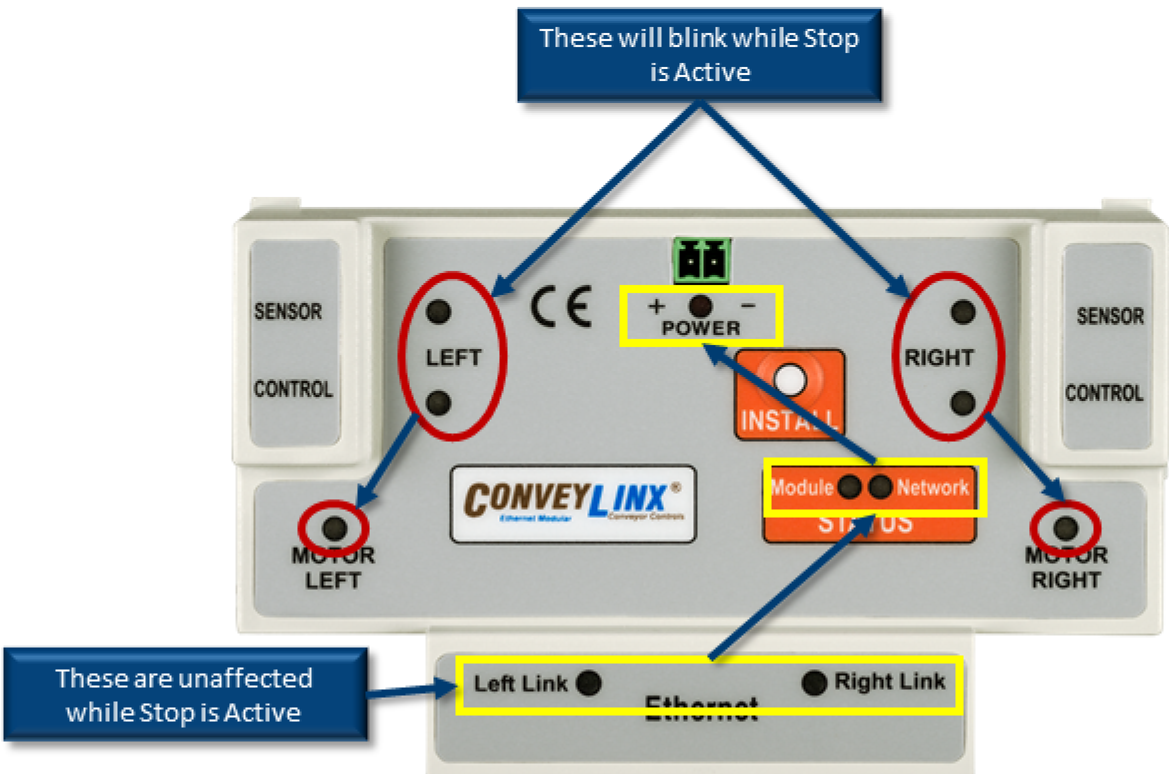
Power Loss

If any individual or multiple modules within a *Stop Group* has its power disconnected; all remaining powered modules within that *Stop Group* will recognize this as a *Stop Command*

10.5.2. Indications a Stop is Active

ERSC Indicators

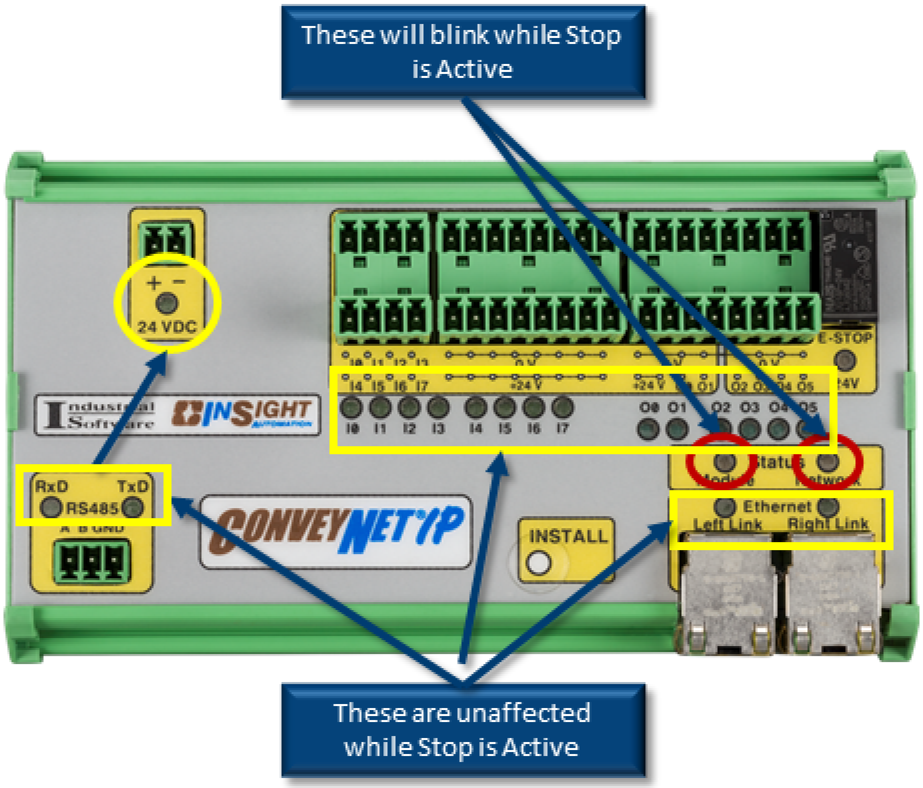
Any powered ERSC modules within a *Stop Group* that has an active *Stop Command* will flash Red LEDs on the Motor, Sensor, and Control Ports. Please note that the Link, Network, and Status LEDs will be unaffected by the *Stop Command*



If a given ERSC within a *Stop Group* has a *Stop Button* assigned; when a *Stop Command* is active, the digital output on the Control Port for which the stop button is assigned will toggle on and off. This signal can be used to illuminate an indicator light or be used as an input to a remote PLC. See Wiring Examples for more details.

CNIP Indicators

Any powered *CNIP* modules within a *Stop Group* that has an active *Stop Command* will flash their Module and Network Status LEDs in Red. All other LEDs will remain unaffected.



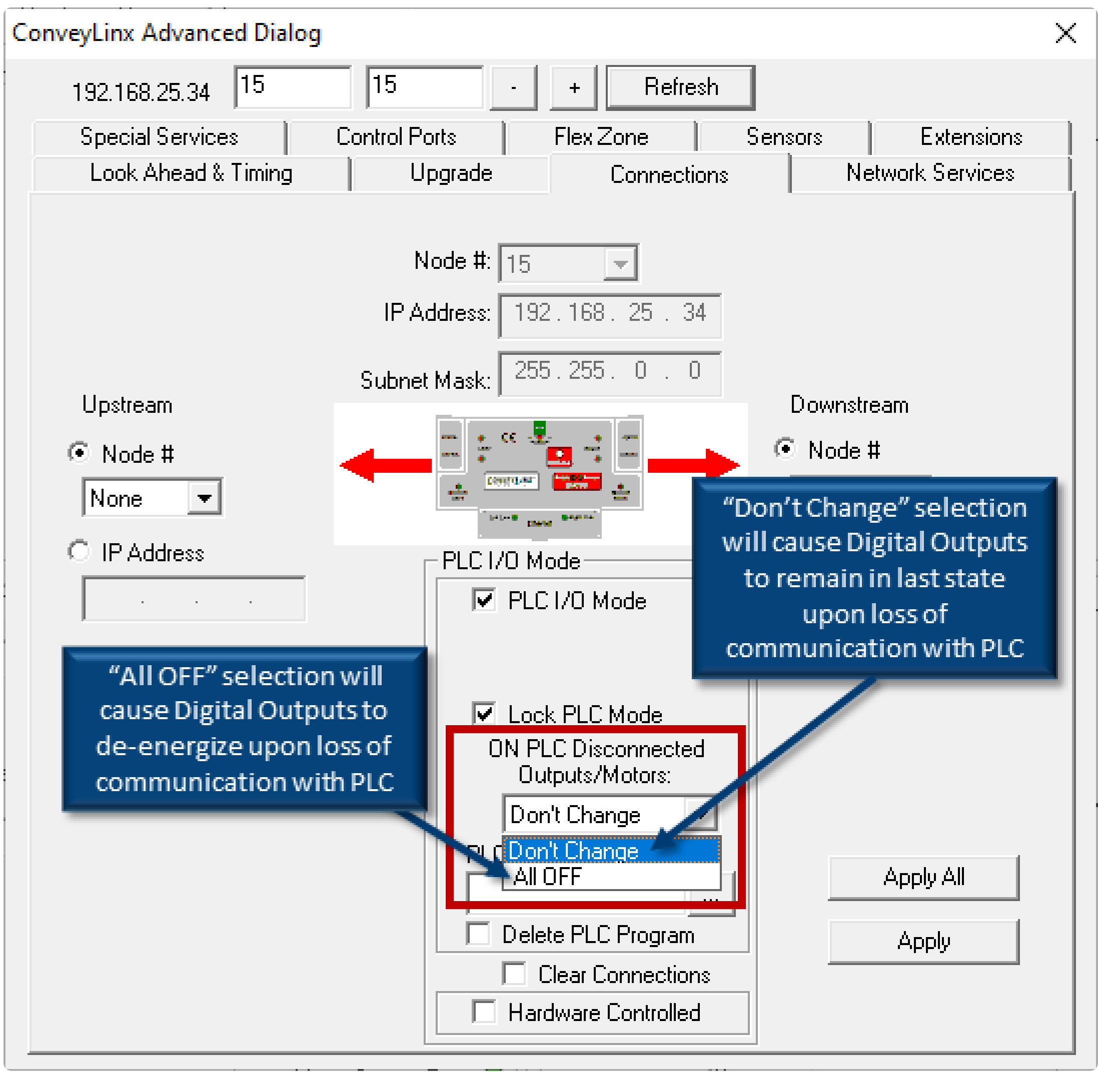
PLC Indication

In order for a PLC to be able to either initiate a *Stop or Start Command*; it has to establish an instance connection to at least one *ERSC* or *CNIP* within a given *Stop Group*. Each instance assembly available for both the ERSC and CNIP provides an input register for *ConveyStop Status* and an output register to issue ConveyStop *Stop* and *Start Commands*.
If the PLC is connected to multiple ERSC's or CNIP's in a given *Stop Group*; any connected ERSC or CNIP can provide ConveyStop status and can accept *Stop* and *Start Commands* from the PLC.

10.5.3. ERSC Functions Affected by Stop Command

Regardless of mode (ZPA, PLC I/O, ConveyLogix program); a *Stop Command* completely inhibits any rotation of a motor connected to either Motor Port.

How the module's digital outputs respond to a *Stop Command* is dependent upon what mode the module is in and its settings in *EasyRoll*. When placing a module into PLC I/O Mode from the Connections tab from the Advanced Dialog; there is a drop down box for the user to select how to respond when there is a communications loss with the PLC. The choices are to turn off the outputs of leave them in their last state. This setting also affects how the outputs respond during a *Stop Command*



EasyRoll Connections Tab with PLC Disconnect Option

Control Port Outputs

The state of the Control Port digital outputs during a Stop Command depends upon the following:

Whether the module is in *ZPA Mode* or *PLC I/O Mode*
Whether or not a *Stop Button* was assigned during ConveyStop *Stop Group* configuration
The selection of the ** On PLC Disconnected** option from the Connections Tab in *EasyRoll*

The following chart indicates the state of the corresponding Left or Right Control Port digital output while a Stop Command is Active based upon these factors:

ERSC Mode	Stop Button on Port	Port Output State
ZPA	No	Unaffected by *Stop Command*. Output will reflect the blocked/clear status of the zone
ZPA	Yes	Energized while *Stop Command* is Active, otherwise OFF
ZPA with PLC control of Control Port option enabled in *EasyRoll*	N/A	State determined by *On PLC Disconnected* option setting in *EasyRoll*
PLC I/O	No	State determined by *On PLC Disconnected* option setting in *EasyRoll*
PLC I/O Mode	Yes	Energized while *Stop Command* is Active, otherwise OFF

Motor Ports in Digital I/O Mode

For the Left and Right Motor Ports in Digital Mode, the state of these outputs is determined by the *On PLC Disconnected* setting on the Connections Tab in EasyRoll

! WARNING !!! Motor Port in Digital Mode output state upon STOP Command is determined by On PLC Disconnected setting in EasyRoll and that these outputs could remain energized during a STOP Command. In any case where output state is determined by EasyRoll setting; it is up to the Integrator to insure that control system programming understands that a STOP Command has no effect on these outputs. It is possible that based upon this setting, any given digital output could remain energized even though there is STOP Command active.

10.5.4. CNIP Functions Affected by Stop Command

Upon a *Stop Command*, the on-board relay is de-energized which in turn disconnects the control power bus that feeds Outputs O2 through O5. This ensures that Outputs O2 thru O5 de-energize during a *Stop Command* regardless of any logic instructing these outputs to remain energized.

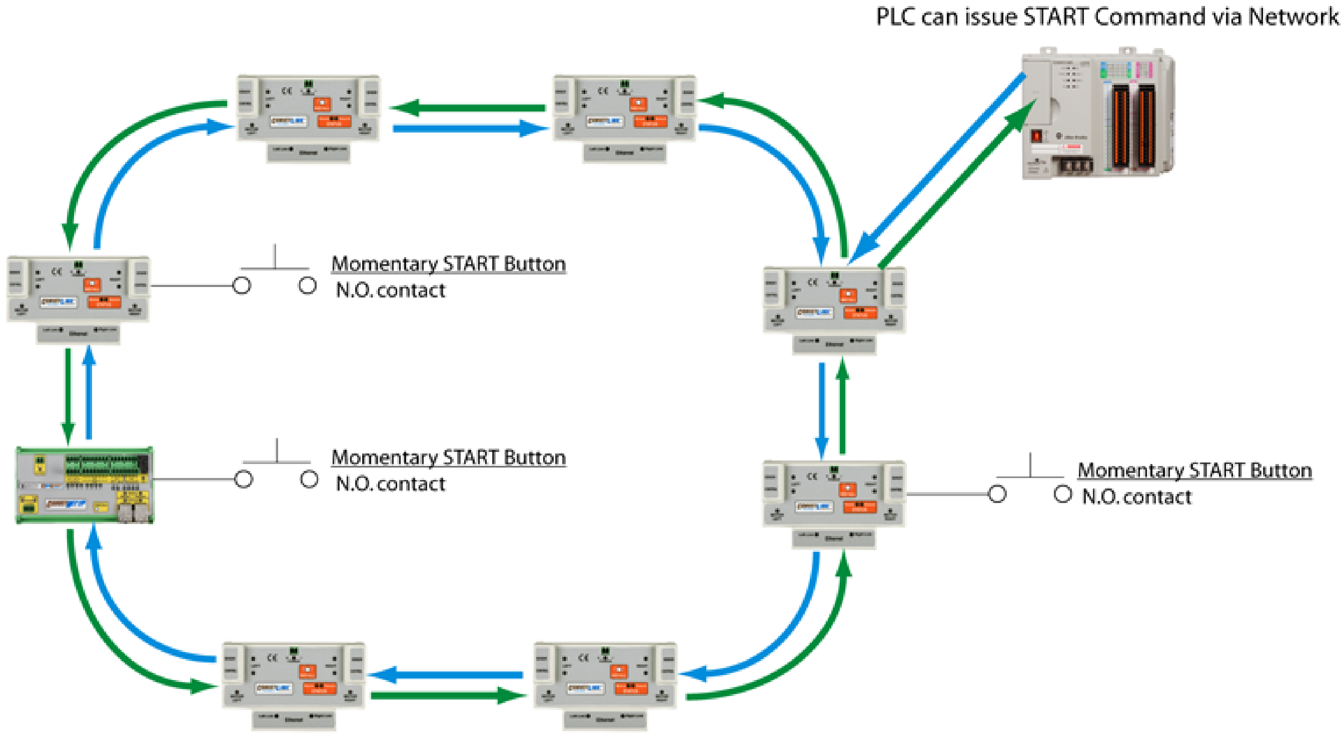
> ! Outputs O0 and O1 are unaffected by a *Stop Command* and will still respond to any logic from either a remote PLC or on-board ConveyLogix program

10.6. ConveyLinx Function at Stop Group Boundaries

If a given system application utilizes multiple *Stop Groups* (or non-ConveyStop configured modules) that are physically and logically connected such as a conveyor system; ConveyLinx has built in mechanisms for flow control. If the upstream or in feed end of a *Stop Group* is being fed by the discharge of ConveyLinx controlled conveyor; a *Stop Condition* will cause the most upstream module of the *Stop Group* to indicate to its upstream zone to not allow any item to be conveyed into the *Stop Group's* in feed zone.

10.7. Issuing a Start Command

Typical methods for an operator or the control system to issue a *Start* Command is either via hardwired Start push-button or via network command from PLC as shown:



Before a Start Command will work you must make sure:

- All modules in the *Stop Group* must be powered
- All *Stop Buttons* must be physically reset
- All expected communications between modules must be present
- All expected PLC connections to modules must be present

Hard-Wired Push-Button

ERSC Module

You can connect *Start Command* buttons to either the Left or Right Control Ports on an ERSC in the *Stop Group*. *Start* buttons require a single momentary normally open (N.O.) contact. Once the contact is closed, if all *Start Command* criteria are met, the *Stop Group* will resume its normal function.

CNIP Module

For the *CNIP* module, there are dedicated inputs to use to connect a *Start Command* button. See Wiring Examples for details on the inputs reserved for this function

Remote PLC

If a PLC is connected to at least one module in a *Stop Group*; it can issue a *Start Command* to that module and all modules within that group will recognize this as a *Start Command*. Even if the PLC is connected to more than one module in a *Stop Group*, it only has to issue the *Start Command* to only one module in order to *Start* the whole *Stop Group*.

10.7.1. Indications Start is Active

ERSC Indication

Only ERSC modules that have *Start Buttons* assigned will give an active indication that *ConveyStop* has been started. If a given ERSC has a *Start Button* assigned to a given Control Port, then the digital output signal of that Control Port will energize when *ConveyStop* is started and the *Stop Group* is functioning. If any *Stop Condition* exists, then this output is de-energized

CNIP Indication

There are no active indicators on a CNIP module that *ConveyStop* has been started. Even for modules that have a *Start Button* assigned; there is not default digital output on the module to indicate a started condition

10.8. Using a PLC with ConveyStop

All PLC communications assemblies for ConveyLinx modules include a reserved set of registers for *ConveyStop* command output and status input. Please refer to Appendix B – PLC Instance Usage for details on the particular PLC and instance connection you want to use.
Regardless of which network the PLC uses (Ethernet I/P, Modbus TCP, or Profinet), the data transfer between ConveyLinx modules (ERSC or CNIP) is organized in blocks of registers defined as *Assembiles*. Each *Assembly* is part of an input / output pair. The input assembly provides status input data to the PLC to read and the output assembly is used by the PLC to write data to the ConveyLinx module.
Within each *Input Assembly* is a *ConveyStop Status Register* and within each *Output Assembly* is a *ConveyStop Command Register*. Regardless of PLC, each register is a 16-bit value.

Ethernet I/P

If a PLC establishes an Ethernet I/P assembly instance connection to at least one ERSC or CNIP module within a *Stop Group*, then the *ConveyStop* control within the *Stop Group* will expect this connection to always be present. If this connection is ever lost, then a *Stop Condition* is triggered for the *Stop Group*. If the PLC has established connections to multiple modules within a *Stop Group*; the loss of connection to any one of these connections will trigger a *Stop Condition* for the *Stop Group*.
Please Note: ERSC and CNIP modules will also respond to unsolicited MSG instruction requests from the PLC. This form of communication does not require the ERSC or CNIP to be configured as a “permanent” Ethernet module on the PLC’s logic Ethernet backplane. If the PLC establishes communication with an ERSC or CNIP within a *Stop Group* with a MSG instruction request; the *ConveyStop* control does not treat this as an expected connection and does not generate a *Stop Condition* for any MSG instruction communication presence or loss.

Modbus TCP

Modbus TCP is a request/response type protocol and each ERSC or CNIP module acts as a Modbus TCP server and will respond to holding register read/write requests. Holding Register addresses 4:1000 and above are reserved for assembly instances. If a PLC establishes communications to at least one module in a *Stop Group* to read/write to any block of holding registers at address 4:1000 or above and these read/write communication requests from the PLC is missing for more than 100 msec; a *Stop Condition* will be generated.
If a PLC establishes a connection to read/write to Holding Registers whose addresses are in the range of 4:0001 thru 4:999; a loss of connection for this communication will not result in a *Stop Condition*

Profinet I/O

Profinet I/O connection works similarly to Ethernet I/P with *ConveyStop*. If a PLC establishes a connection to one or more modules within a *Stop Group*; then the ConveyStop control within the *Stop Group* will expect all of these connections to always be present. If any one of these connections is lost, then a *Stop Condition* is triggered for the *Stop Group*

10.8.1. ConveyStop Status Register

The data in this register is to be interpreted bit wise regardless of the PLC protocol used. The following chart lists the bit definitions for the *ConveyStop Status Register*:

Bit	Description
0 thru 4	Reserved
5	1 = *Stop Command* issued from a hardwired button on a module in the *Stop Group* 0 = No *Stop Command* issued from a hardwired button on a module in the *Stop Group*
6	1 = *Stop Command* issues because of loss of Ethernet connectivity 0 = Ethernet connectivity is OK
7	1 = *Stop Command* issued because of loss of connection to PLC 0 = PLC connectivity is OK
8	1 = *Stop Command* issued because Stop Button on local module's *Left* Control Port (or CNIP I0/I1) 0 = Stop button OK on local module's Left Control Port(or CNIP I0/I1)
9	1 = *Stop Command* issued because Stop Button on local module's *Right* Control Port (or CNIP I0/I1) 0 = Stop button OK on local module's Left Control Port(or CNIP I0/I1)
10	1 = *Stop Command* issued from PLC 0 = PLC is not issuing a stop command
11 thru 15	Reserved

✻ If all bits are reset (0) in the *ConveyStop Status Register* then there is no *Stop Condition* active within the *Stop Group*

10.8.2. ConveyStop Command Register

The PLC writes integer values to the *ConveyStop Command Register* to issue either a *Stop Command* or a *Start Command* .

Stop Command

Writing a value of “1” to the *ConveyStop Command Register* will cause a *Stop Command* to be issued to the STOP Group. In any module (ERSC or CNIP) that is connected to the PLC, bit10 will be set in their corresponding *ConveyStop Status Register*.

Start Command

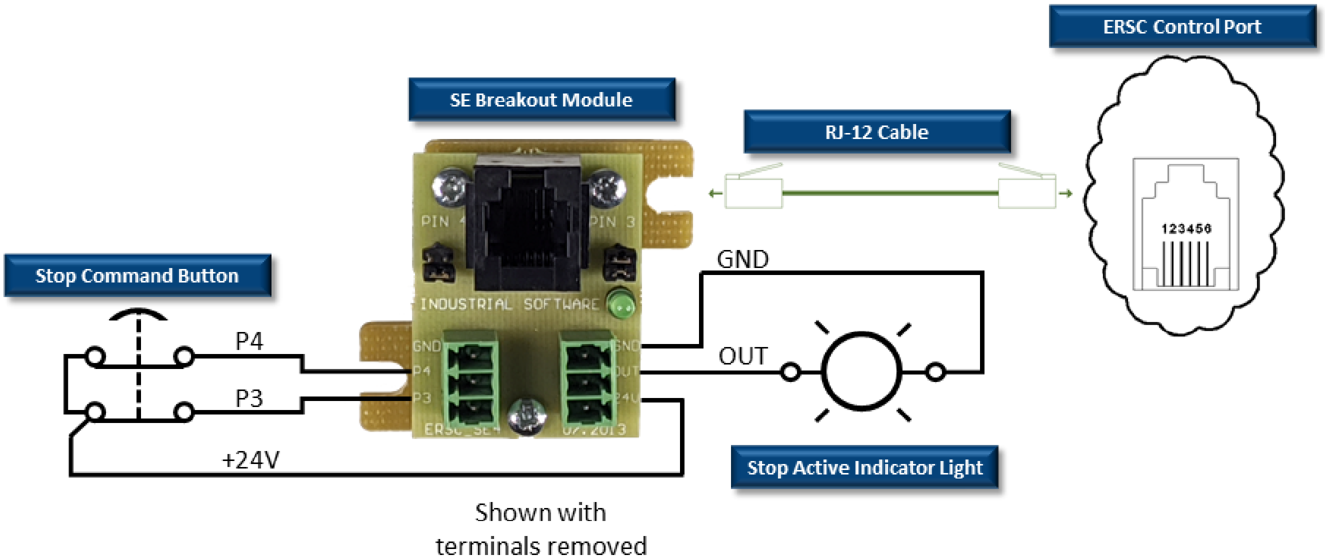
Before issuing a *Start Command*, the PLC should first verify that the *ConveyStop Status Register* is “0” indicating there is not an active STOP condition. One of these conditions is indicated in Bit 10 of the *ConveyStop Status Register*. So the procedure for issuing a *Start Command* is:

1. Write a 0 to the *ConveyStop Command Register*
2. Verify that the *ConveyStop Status Register* is equal to 0
3. Write a 2 to the *ConveyStop Command Register* to issue the *Start Command*
4. Wait at least 500 msec and then write a 0 to the *ConveyStop Command Register*

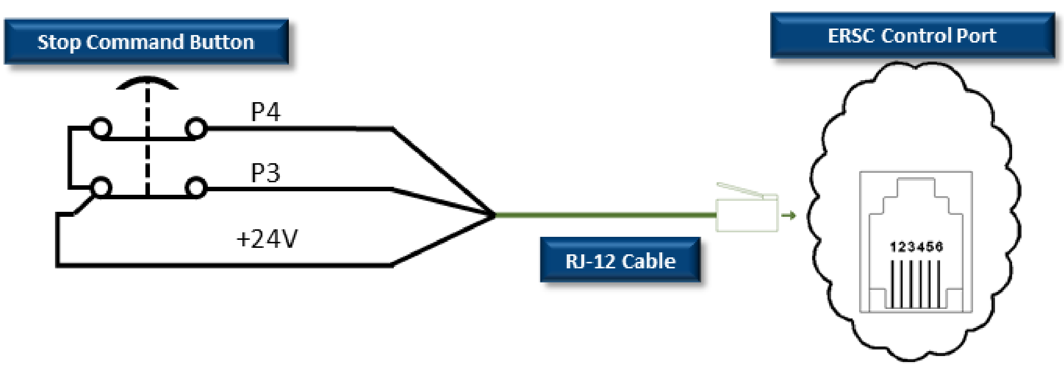
10.9. Wiring Examples

10.9.1. ERSC with Stop Button

Stop Button with SE Breakout Module



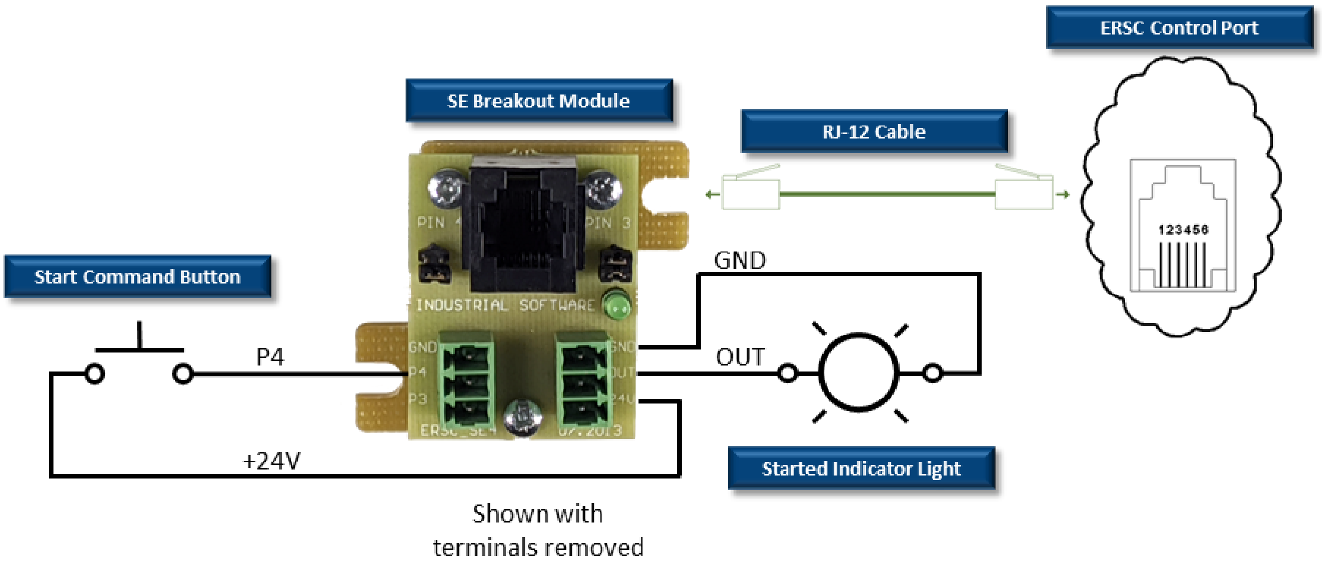
Stop Button with RJ-12 Pigtail Cable



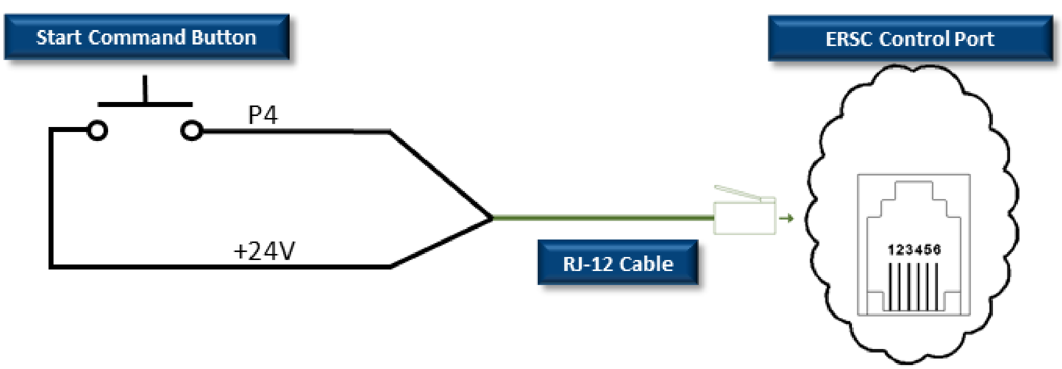
! Please note that you cannot use an indicator light with RJ-12 pigtail cable. You must use *SE Breakout Module* for an indicator light

10.9.2. ERSC with Start Button

Start Button with SE Breakout Module



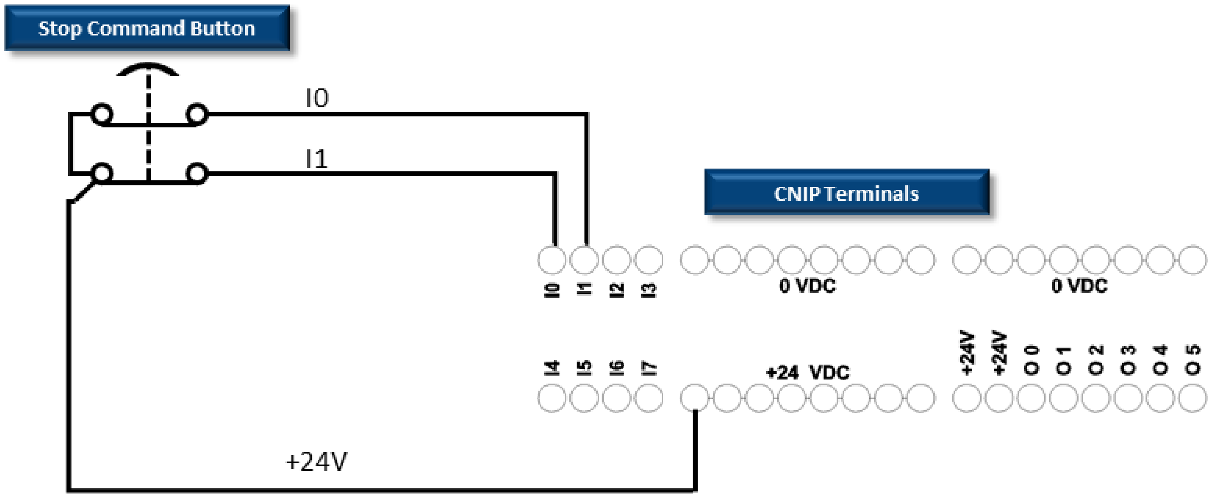
Start Button with RJ-12 Pigtail Cable



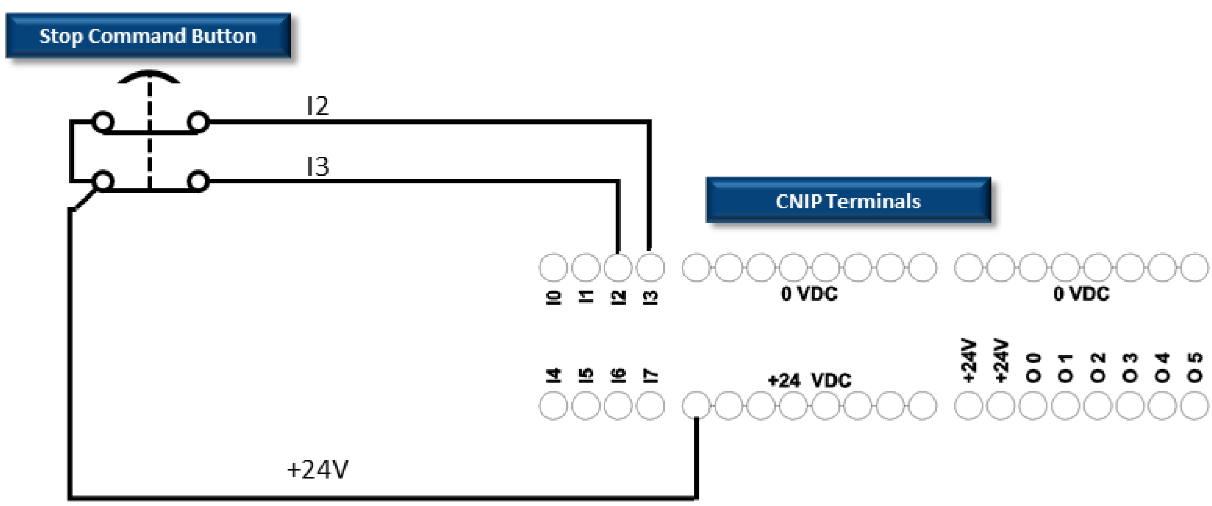
! Please note that you cannot use an indicator light with RJ-12 pigtail cable. You must use *SE Breakout Module* for an indicator light

10.9.3. CNIP with Stop Button

Stop Button on Inputs I0 and I1



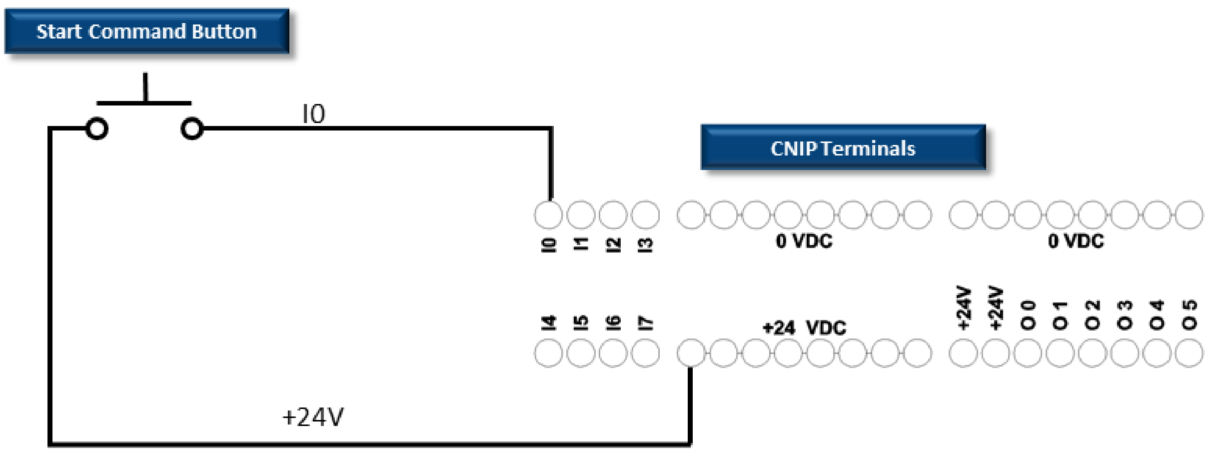
Stop Button on Inputs I2 and I3



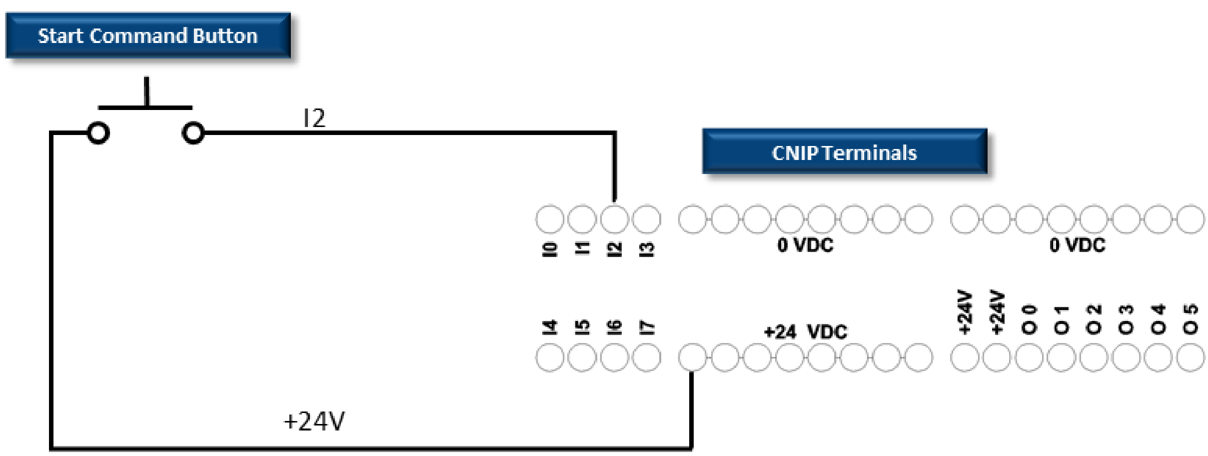
! Indicator Light option is not available on CNIP Module

10.9.4. CNIP with Start Button

Start Button on Input I0



Start Button on Inputs I2



! Indicator Light option is not available on CNIP Module

11. ERSC PLC Developers Guide

This section defines the data connectivity available between ConveyLinx Modules and remote control devices such as *PLCs* and *PCs*

Click Links to Learn More

11.1. Network Architecture

Each *ConveyLinx Module* communicates to its adjacent modules and to any connected PC or PLC via Ethernet physical media. ConveyLinx Modules recognize (3) TCP/IP based protocols: *Modbus TCP, Profinet I/O, and Ethernet I/P*. Modbus TCP is the “native” protocol for communications between *ConveyLinx Modules* and the *EasyRoll* PC software. When ConveyLinx Modules 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 Modules and any given ConveyLinx Module 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 Module by using a Pulseroller provided *EDS file*. *Profinet I/O* is recognized by ConveyLinx Modules and can be attached to any Profinet I/O capable PLC (applicable Siemens platforms) using a Pulseroller provided *GSDXML file*.

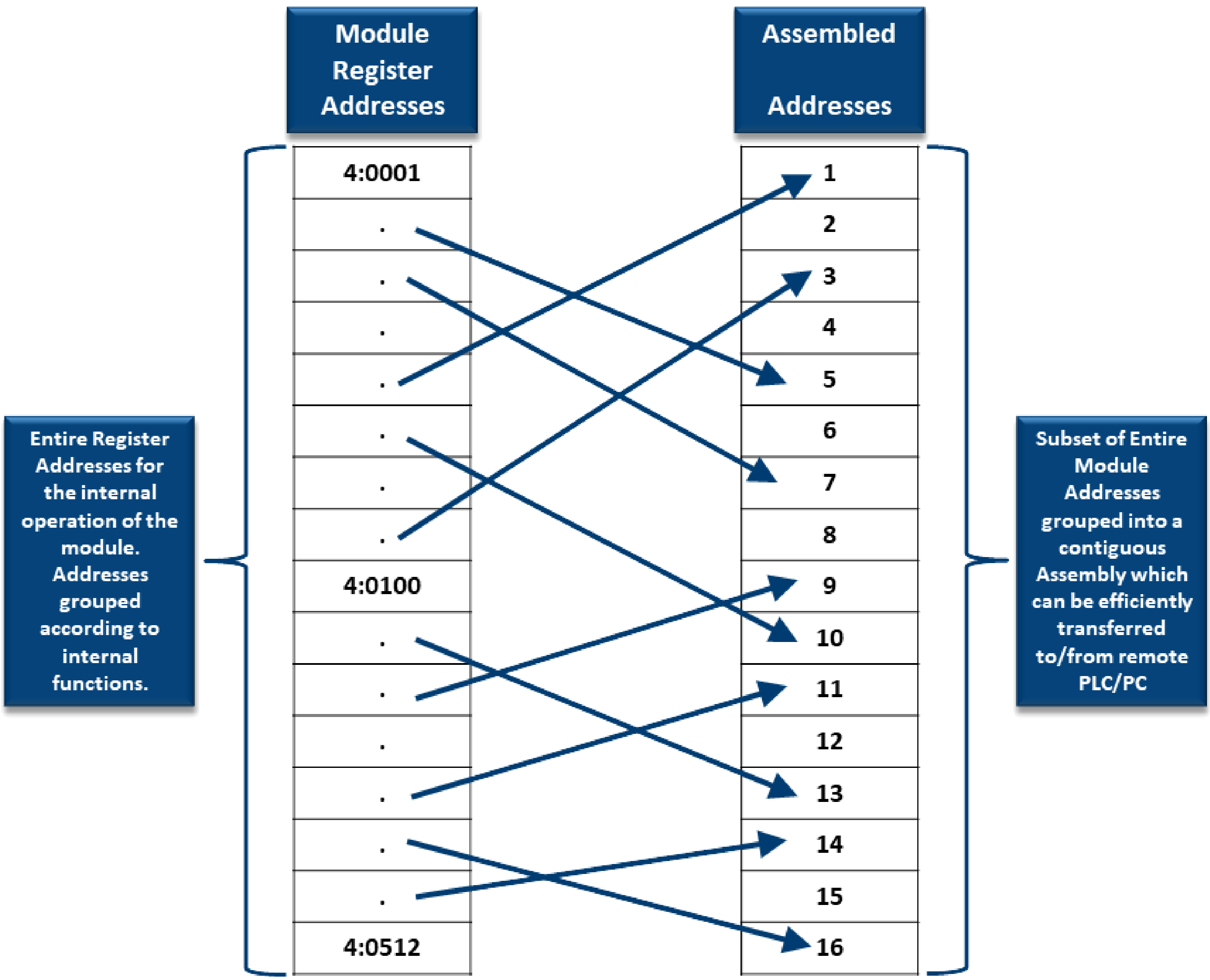
> All protocols access the same internal data locations on a given ConveyLinx Module.

Each ConveyLinx Module’s internal data structure is arranged and addressed as *Modbus Holding Registers*. The on-board communication and control processes attach logical meanings to each holding register and read and write data to specific registers to initiate and/or react to events. Certain registers contain information as to how the ConveyLinx Module 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 Module has a Carton ready to discharge to its neighboring downstream ConveyLinx Module, the upstream ConveyLinx Module will write a specific value to a specific address in the downstream ConveyLinx Module’s internal registers. The downstream ConveyLinx Module’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 Modules 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 Module 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 Module 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 Module 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 Module's memory structure contains a fixed array of internal data locations that are constructed as ***Modbus Holding Registers***. Each ConveyLinx Module has a fixed reserve of ***Holding Registers*** with each capable of holding a 16-bit numerical integer value. Modbus TCP protocol provides for read/write access to any available *Holding Register*. The structure of these registers allows for individual ConveyLinx Module's to read from and write to specific register address locations to achieve inter-module communications. Certain registers are read from and written to by the *EasyRoll* software in order to monitor and/or change default configuration values 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 Module, the Assembly Registers are "virtual" in that they are not "physical" Module Register Addresses and as such are not directly addressable. The ConveyLinx Module 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 Module 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 Module 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 Module will return an error

11.2.2. Ethernet I/P Assembly Instance Structure

When an ConveyLinx Module 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 ERSC is in its default ZPA mode, an external networked PLC or PC controller can connect to the ERSC 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 ERSC 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 ERSC 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 ERSC’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 ERSC 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 in ZPA Mode that is provided as input to the PLC.

Register Name / Module Address	Assembled Address for PLC	Links
Local Status Upstream Zone 4:0116	M: 4:1500 E: I.Data[0] P: Byte 0 (Hi) Byte 1 (Lo)	Local Zone Status
Local Status Downstream Zone 4:0196	M: 4:1501 E: I.Data[1] P: Byte 2 (Hi) Byte 3 (Lo)	Local Zone Status
Arrival Count Local Upstream Zone 4:0106	M: 4:1502 E: I.Data[2] P: Byte 4 (Hi) Byte 5 (Lo)	Arrival/Departure Counts
Departure Count Local Upstream Zone 4:0107	M: 4:1503 E: I.Data[3] P: Byte 6 (Hi) Byte 7 (Lo)	Arrival/Departure Counts
Arrival Count Local Downstream Zone 4:0186	M: 4:1504 E: I.Data[4] P: Byte 8 (Hi) Byte 9 (Lo)	Arrival/Departure Counts

Departure Count Local Downstream Zone 4:0187	M: 4:1505 E: I.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	Arrival/Departure Counts
Module Status Word 1 4:0088	M: 4:1506 E: I.Data[6] P: Byte 12 (Hi) Byte 13 (Lo)	Module Status
Module Status Word 2 4:0089	M: 4:1507 E: I.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	Module Status
Current Upstream Zone Tracking Word 1 4:0119	M: 4:1508 E: I.Data[8] P: Byte 16 (Hi) Byte 17 (Lo)	Tracking and Release Counts
Current Upstream Zone Tracking Word 2 4:0120	M: 4:1509 E: I.Data[9] P: Byte 18 (Hi) Byte 19 (Lo)	Tracking and Release Counts
Current Downstream Zone Tracking Word 1 4:0199	M: 4:1510 E: I.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	Tracking and Release Counts
Current Downstream Zone Tracking	M: 4:1511 E: I.Data[11] P: Byte 22 (Hi)	Tracking and Release Counts

Word 2 4:0200	Byte 23 (Lo)	
Current Release Count for Upstream Zone 4:0105	M: 4:1512 E: I.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)	Tracking and Release Counts
Current Release Count for Downstream Zone 4:0185	M: 4:1513 E: I.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)	Tracking and Release Counts
Get Tracking Forward Direction Word 1 4:0201	M: 4:1514 E: I.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	Forward and Reverse Tracking
Get Tracking Forward Direction Word 2 4:0202	M: 4:1515 E: I.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	Forward and Reverse Tracking
Get Tracking Reverse Direction Word 1 4:0121	M: 4:1516 E: I.Data[16] P: Byte 32 (Hi) Byte 33 (Lo)	Forward and Reverse Tracking
Get Tracking Reverse Direction Word 2 4:0122	M: 4:1517 E: I.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)	Forward and Reverse Tracking
Sensor & Control Port	M: 4:1518 E: I.Data[18]	Port Inputs and ConveyStop

Inputs 4:0035	P: Byte 36 (Hi) Byte 37 (Lo)	Status
Reserved N/A	M: 4:1519 E: I.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)	
ConveyStop Status 4:0019	M: 4:1520 E: I.Data[20] P: Byte 40 (Hi) Byte 41 (Lo)	Port Inputs and ConveyStop Status

11.3.1.1. Local Zone Status

<table>
<tr><th>Register Name / Module Address</th><th>Assembled Address for PLC</th><th>Description</th></tr>
<tr><td>Local Status Upstream Zone Forward Direction
4:0116</td><td>M: 4:1500 (Lo Byte)
E: I.Data [0] (Lo Byte)
P: Byte 1</td><td rowspan="4">Unsigned Value of Byte:

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)</td></tr>
<tr><td>Local Status Upstream Zone Reverse Direction
4:0116</td><td>M: 4:1500 (Hi Byte)
E: I.Data [0] (Hi Byte)
P: Byte 0</td></tr>
<tr><td>Local Status Downstream Zone Forward Direction
4:0196</td><td>M: 4:1501 (Lo Byte)
E: I.Data [1] (Lo Byte)
P: Byte 3</td></tr>
<tr><td>Local Status Downstream Zone Reverse Direction
4:0196</td><td>M: 4:1501 (Hi Byte)
E: I.Data [1] (Hi Byte)
P: Byte 2</td></tr>
</table>

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 = Upstream Zone Flow Direction – 0 = Configured Dir, 1 = Opposite of Configured Dir bit 2 = Over-Voltage – Input Power > 30V bit 3 = Left Motor Error – bit is set if any of the following bits are set: 7,8,10 thru 14 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 120°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 = Downstream Zone Flow Direction – 0 = Configured Dir, 1 = Opposite of Configured Dir bit 2 = Over-Voltage – Input Power > 30V bit 3 = Right Motor Error – bit is set if any of the following bits are set: 7,8,10 thru 14 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 120°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 & Control Port Inputs 4:0035	M: 4:1518 E: I.Data [18] P: Byte 36 (Hi) Byte 37 (Lo)	<u>Bitwise Value – Read Only:</u> bit 0 = Left Sensor Port – Pin 3 bit 1 = Left Control Port – Pin 3 bit 2 = Right Sensor Port – Pin 3 bit 3 = Right Control Port – Pin 3 bit 4 = Left Sensor Port – Pin 4 bit 5 = Left Control Port – Pin 4 bit 6 = Right Sensor Port – Pin 4 bit 7 = Right Control Port – Pin 4 bit 8 – 14 Reserved bit 15 = 2 sec on / 2 sec off heatbeat
Reserved	M: 4:1519 E: I.Data [19] P: Byte 38 (Hi) Byte 39 (Lo)	
ConveyStop Status 4:0019	M: 4:1520 E: I.Data [20] P: Byte 40 (Hi) Byte 41 (Lo)	<u>Bitwise Value – Read Only:</u> 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 = Stop active on local module’s Left Control Port bit 9 = Stop active on local module’s Right Control Port 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 when in ZPA Mode.

Register Name / Module Address	Assembled Address for PLC	Links
Set Local Upstream Zone Tracking Word 1 4:0132	M: 4:1600 E: O.Data[0] P: Byte 0 (Hi) Byte 1 (Lo)	Set Local Tracking
Set Local Upstream Zone Tracking Word 2 4:0133	M: 4:1601 E: O.Data[1] P: Byte 2 (Hi) Byte 3 (Lo)	Set Local Tracking
Set Local Downstream Zone Tracking Word 1 4:0212	M: 4:1602 E: O.Data[2] P: Byte 4 (Hi) Byte 5 (Lo)	Set Local Tracking
Set Local Downstream Zone Tracking Word 2 4:0213	M: 4:1603 E: O.Data[3] P: Byte 6 (Hi) Byte 7 (Lo)	Set Local Tracking
Accumulation Control for Local Upstream Zone 4:0104	M: 4:1604 E: O.Data[4] P: Byte 8 (Hi) Byte 9 (Lo)	Accumulation Control

Accumulation Control for Local Downstream Zone 4:0184	M: 4:1605 E: O.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	Accumulation Control
Set Left MDR Speed 4:0040	M: 4:1606 E: O.Data[6] P: Byte 12 (Hi) Byte 13 (Lo)	Speed Control
Set Right MDR Speed 4:0064	M: 4:1607 E: O.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	Speed Control
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)	Release and Status
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)	Release and Status
Set Status for Upstream Induct 4:0134	M: 4:1610 E: O.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	Release and Status

Set Status for Downstream Discharge 4:0232	M: 4:1611 E: O.Data[11] P: Byte 22 (Hi) Byte 23 (Lo)	Release and Status
Set Induct Tracking Forward Direction Word 1 4:0139	M: 4:1612 E: O.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)	Induct Tracking Forward and Reverse
Set Induct Tracking Forward Direction Word 2 4:0140	M: 4:1613 E: O.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)	Induct Tracking Forward and Reverse
Set Induct Tracking Reverse Direction Word 1 4:0237	M: 4:1614 E: O.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	Induct Tracking Forward and Reverse
Set Induct Tracking Forward Direction Word 2 4:0238	M: 4:1615 E: O.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	Induct Tracking Forward and Reverse
Clear Motor Error 4:0022	M: 4:1616 E: O.Data[16] P: Byte 32 (Hi) Byte 33	Set Outputs and Motor Clear

	(Lo)	
Set Control Port Outputs 4:0063	M: 4:1617 E: O.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)	Set Outputs and Motor Clear
Reserved N/A	M: 4:1618 E: O.Data[18] P: Byte 36 (Hi) Byte 37 (Lo)	
ConveyStop Command 4:0020	M: 4:1619 E: O.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)	ConveyStop and Clear Jams
Clear Sensor Jam Command for Local Upstream Zone 4:0109	M: 4:1620 E: O.Data[20] P: Byte 40 (Hi) Byte 41 (Lo)	ConveyStop and Clear Jams
Clear Sensor Jam Command for Local Downstream Zone 4:0189	M: 4:1621 E: O.Data[21] P: Byte 42 (Hi) Byte 43 (Lo)	ConveyStop and Clear Jams
Direction & Accumulation Mode Control for Local	M: 4:1622 E: O.Data[22] P: Byte 44	Direction and Accumulation Mode

Upstream Zone 4:0365	(Hi) Byte 45 (Lo)	
Direction & Accumulation Mode Control for Local Downstream Zone 4:0375	M: 4:1623 E: O.Data[23] P: Byte 46 (Hi) Byte 47 (Lo)	Direction and Accumulation Mode
ConveyMerge Interface 4:0387	M: 4:1624 E: O.Data[24] P: Byte 48 (Hi) Byte 49 (Lo)	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 ERSC 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 ERSC 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 ERSC 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 ERSC 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

<table>
<tr><th>Register Name / Module Address</th><th>Assembled Address for PLC</th><th>Description</th></tr>
<tr><td>Accumulation Control for Local Upstream Zone
4:0104</td><td>M: 4:1604
E: O.Data [4]
P: Byte 8 (Hi)
Byte 9 (Lo)</td><td rowspan="2">Bitwise Values:
bit 0 = Set/Clear Accumulation Mode for Local Zone
bit 8 = Accumulate adjacent upstream zone
bit 9 = 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</td></tr>
<tr><td>Accumulation Control for Local Downstream Zone
4:0184</td><td>M: 4:1605
E: O.Data [5]
P: Byte 10 (Hi)
Byte 11 (Lo)</td></tr>
</table>

Notes for Accumulation Control Bits

> Beginning with ConveyLinx firmware 3.45 and later; the function of this register has been expanded to include separate bit-wise functionality. Previous to firmware 3.45, only bit 0 function was available

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 product and equipment

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 ERSC had written a status value of “4”. This function would be useful for a merge onto a main line of ZPA conveyor.

> Wake up Local Zone functionality available in ERSC firmware versions 4.1 and higher

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 % PWM: Range: 0 to 1000 *Example: 400 = 40%* 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)	Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor: 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)	Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor: 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)	Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor: 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)	Only used when local module is accepting loads from Non-ConveyLinx controlled conveyor: 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 Control Port Outputs 4:0063	M: 4:1617 E: O.Data [17] P: Byte 34 (Hi) Byte 35 (Lo)	Bitwise Value: 1 energizes Output bit 1 = Left Control Port bit 3 = Right Control Port
Reserved	M: 4:1618 E: O.Data [18] P: Byte 36 (Hi) Byte 37 (Lo)	

Clear Motor Error

Errors deemed “fatal” for the ERSC (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 ERSC. 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 ERSC to recognize the reset command

Control Port Outputs

By default, in ZPA mode, the PLC does not have control of the local connected module’s Control Port output signals. The default function of the Control Port output signals is to be energized when the Control Port’s corresponding zone is occupied. 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. Using EasyRoll software, one or both Control Ports can have their function switched from internal ZPA logic control to be controlled by the PLC. See EasyRoll software Advanced Screen|Control Ports| Checkboxes for Enable PLC Controls

> Please note that the checkboxes in EasyRoll for selecting PLC control of Control Ports is based upon upstream or downstream configuration of the ERSC determined during Auto-Configuration and NOT by “left” or “right” Control Port designation

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

✻ Refer to Pulseroller publication *ERSC-1800 ConveyStop User's Guide* for details on using ConveyStop

Sensor Jam Clear Commands

These registers are applicable only to a Sensor Jam condition. By default, in ZPA mode, the ERSC will attempt 3 times to automatically clear a sensor jam and if the sensor is still blocked, the ERSC will stop the zone. To reset this condition the Carton must be manually removed after the 3rd attempt and the sensor cleared before the zone will return to normal operation. These registers allow the PLC to remotely attempt another auto-clear cycle in an attempt to clear the jam condition. Please note that this 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 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”
Direction & Accumulation Mode Control for Local Downstream Zone 4:0375	M: 4:1623 E: O.Data [23] P: Byte 46 (Hi) Byte 47 (Lo)	

11.3.2.9. ConveyMerge Interface

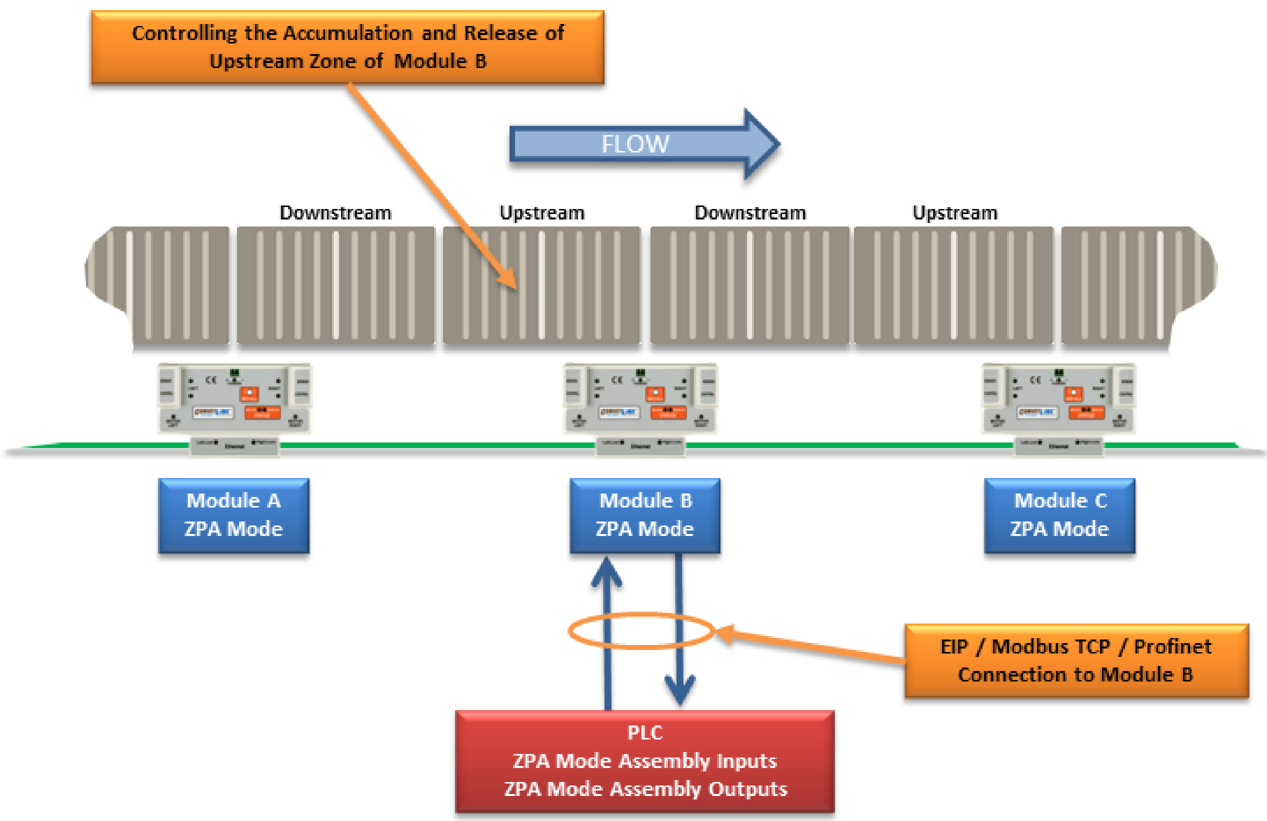
Register Name / Module Address	Assembled Address for PLC	Description
ConveyMerge Interface 4:0387	M: 4:1624 E: O.Data [24] P: Byte 48 (Hi) Byte 49 (Lo)	ONLY AVAILABLE for FIRMWARE 4.25 and HIGHER Used to dynamically modify a ConveyMerge configuration already established with EasyRoll Bitwise Values: 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

Please refer to publication *ERSC-1005 User's Guide Supplement for ConveyMerge* for details on configuring merging sections with EasyRoll software

11.3.3. ZPA Examples

11.3.3.1. Basic Accumulate and Release with Tracking Data

Here is a typical arrangement of an upstream or downstream zone on an ERSC 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.

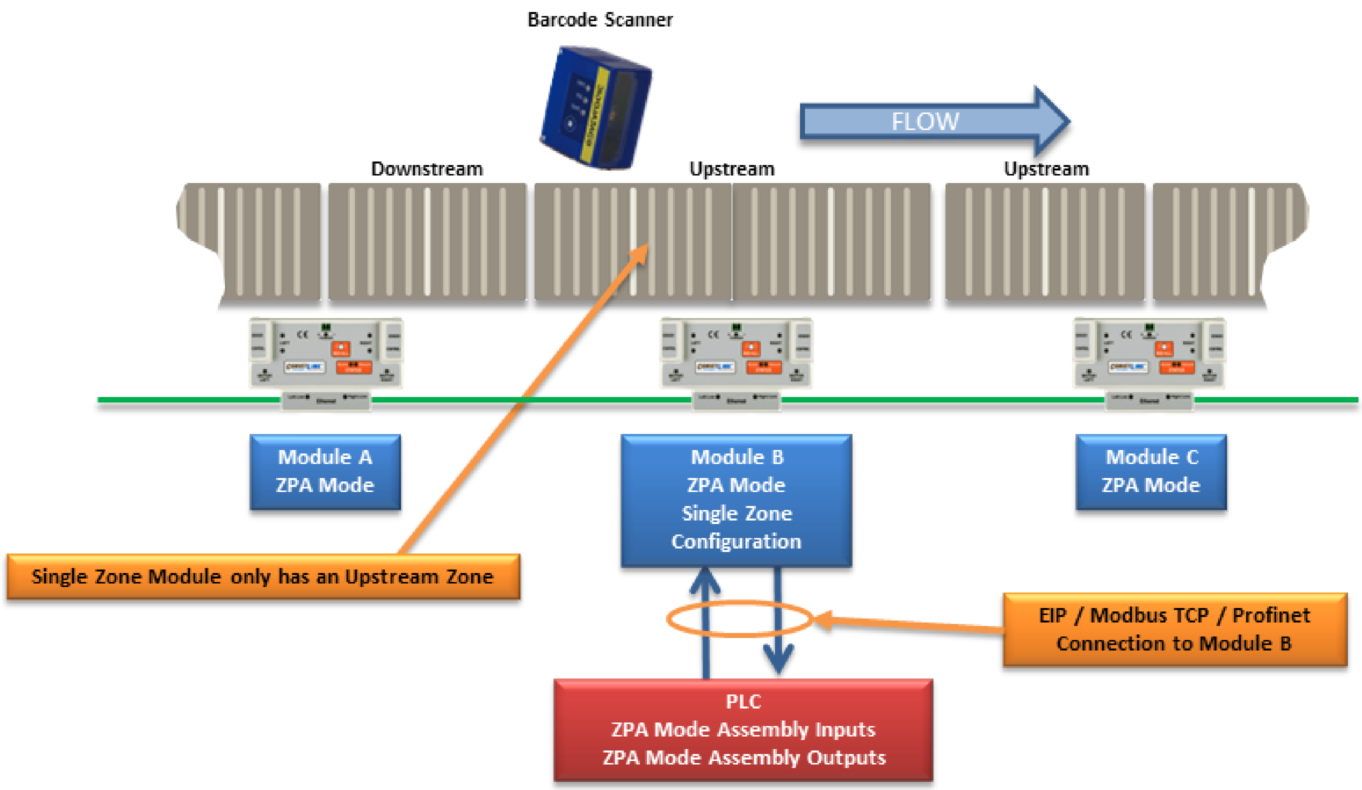
4. The PLC may then decide that this tracking data is to be updated. The PLC can then write new tracking data to registers *Set Local Downstream Zone Tracking Word 1* and *Set Local Downstream Zone Tracking Word 2*.
5. When the PLC is ready to release the Carton in the downstream zone, it should read 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



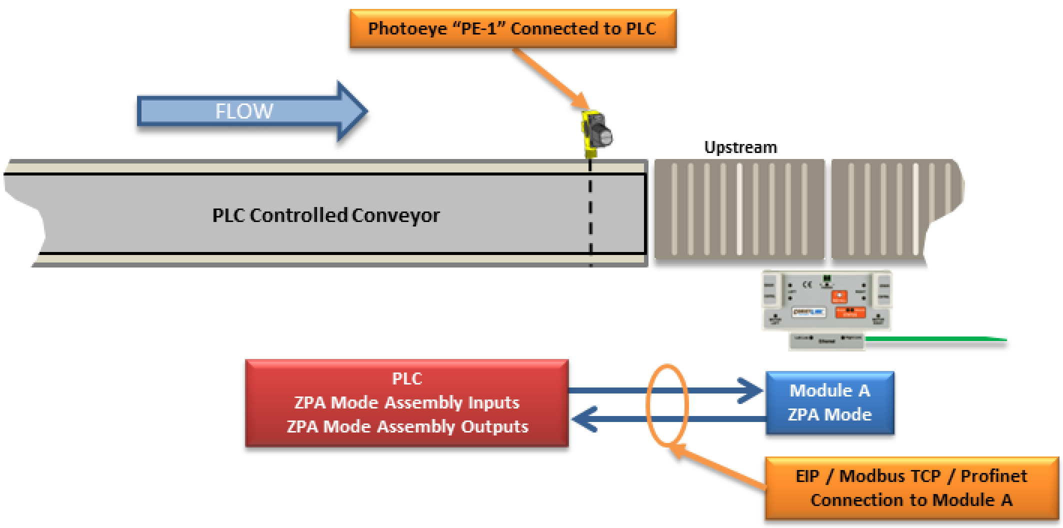
In this example, Module B must be logically configured as a single zone. This can be either a single long zone with one MDR and one sensor, or a single long zone with 2 MDRs and one sensor.

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:

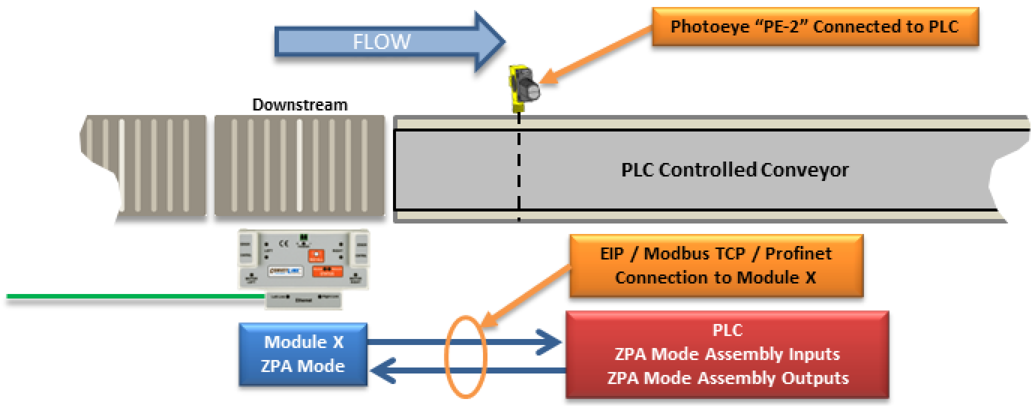
1. Write a “4” into *Set Status for Upstream Induct* register. This will cause the upstream zone of **Module A** to run to accept Carton.
2. When Carton clears **PE-1**, write tracking data to *Set Induct Tracking Forward Direction Word 1* and *Set Induct Tracking Forward Direction Word 2*
3. When Carton 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:

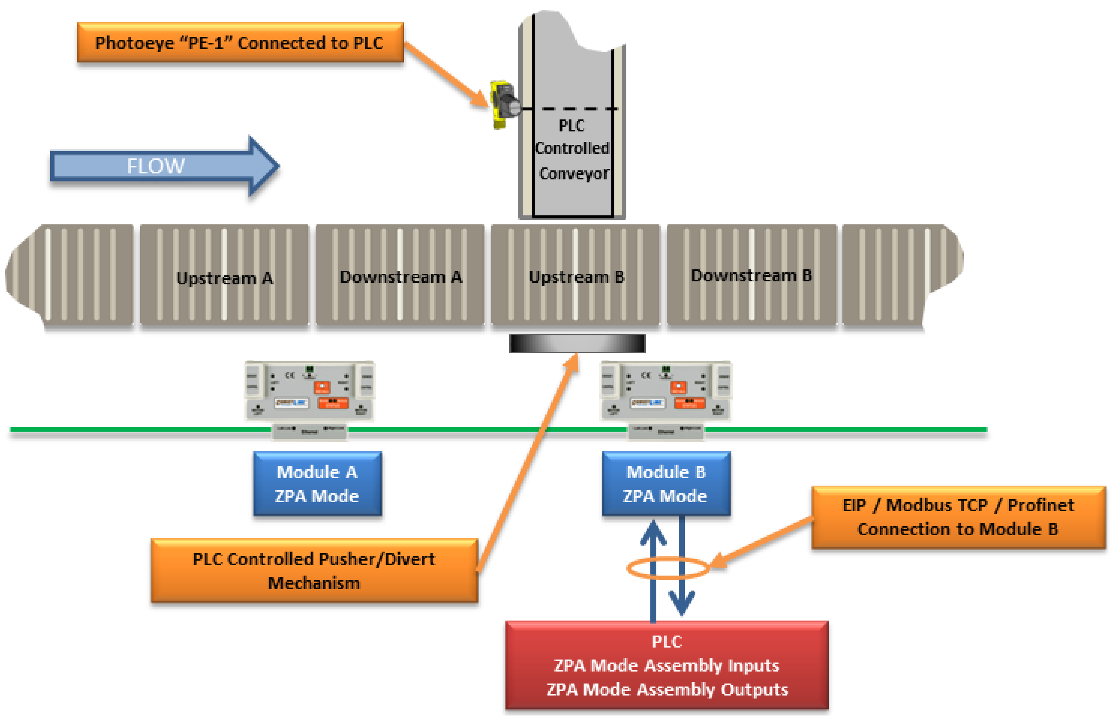
If the application is such that the PLC would like to inhibit a new Carton from entering **Module X** downstream zone for whatever reason, there is a way to accumulate the next upstream zone to the **Module X** downstream zone. The PLC can set bit 8 in the *Accumulation Control for Local Downstream Zone* 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 ERSC 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.

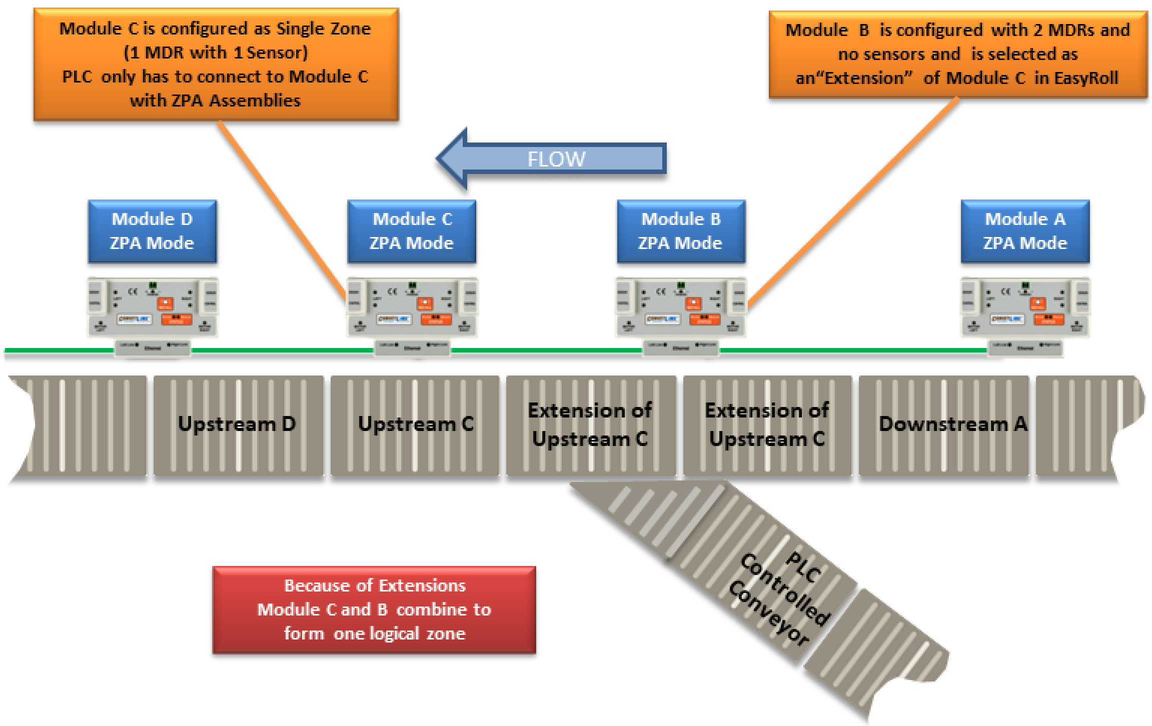
If the PLC determines that the newly arrived carton does not need to divert:

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 ERSC 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 & Control 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 Control Port 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 Module is in PLC I/O mode, all automatic functions of detecting loads and running motors are suspended by the local ConveyLinx Module’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 Module is in PLC I/O Mode:

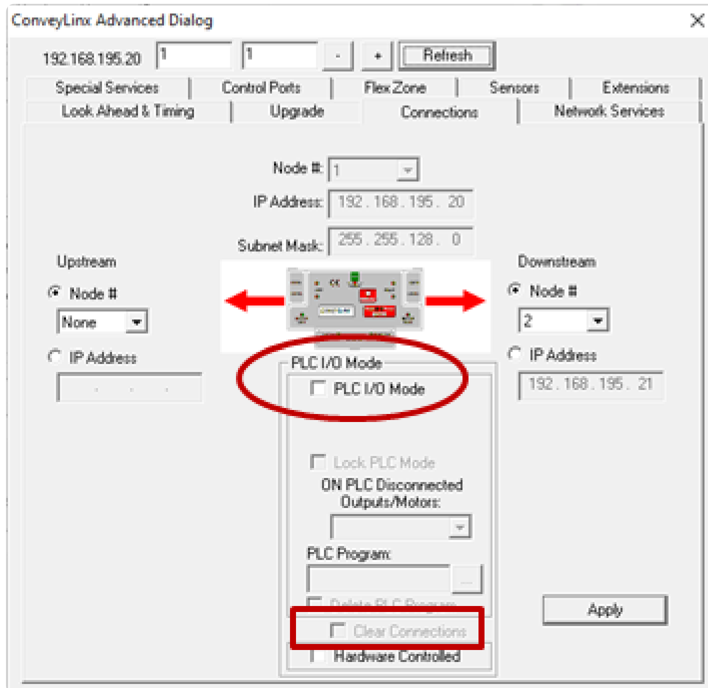
- Status of all available digital inputs on Sensor and Control ports (8 total inputs)
- Module voltage reading
- Left and Right motor status of frequency, current, and calculated temperature
- Left and Right motor diagnostic error status word
- Control of Control Port digital outputs
- Ability to independently run both Left and Right motors
- Ability to set speed, acceleration, deceleration, PI Mode, 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 ERSC is placed in PLC I/O mode; it suspends all of its internal ZPA logic control. Any sensors or motors connected to the ERSC require explicit interaction with an external controller. The external controller will have typical Ethernet-based remote I/O performance from an ERSC when in PLC I/O mode

11.4.1. Setting PLC I/O Mode in EasyRoll

Individual ERSC's 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. <u>Check or Uncheck the "Clear Connections" checkbox depending upon your application</u>. Click "Apply" to initiate the change. The ERSC will restart and this may take several seconds to complete

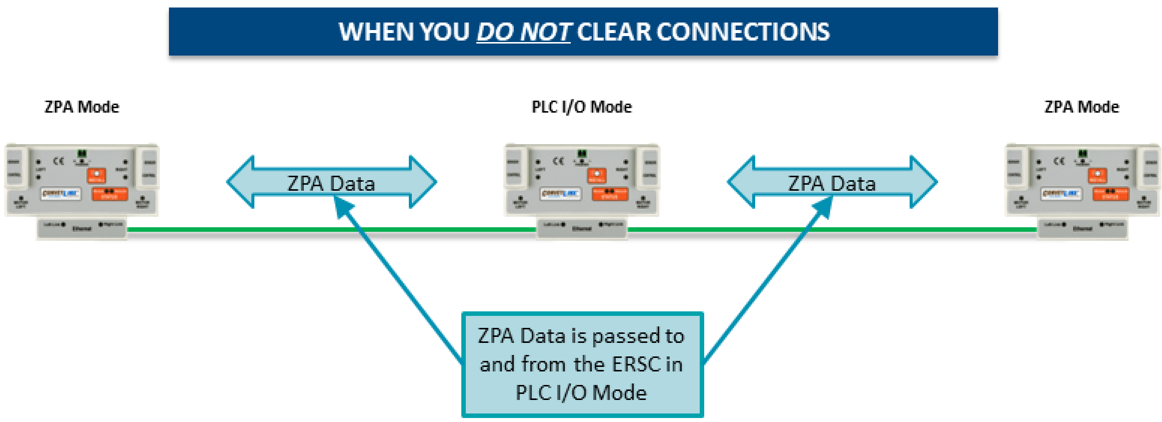


! IMPORTANT NOTE: Once a given ERSC has been placed in PLC I/O mode, the ONLY way to return it to ZPA mode is to perform an Auto-Configuration procedure or Restore from a backup file with EasyRoll. There is no "undo" or "reset" function for this action

11.4.2. Optional Clear Connections Choice

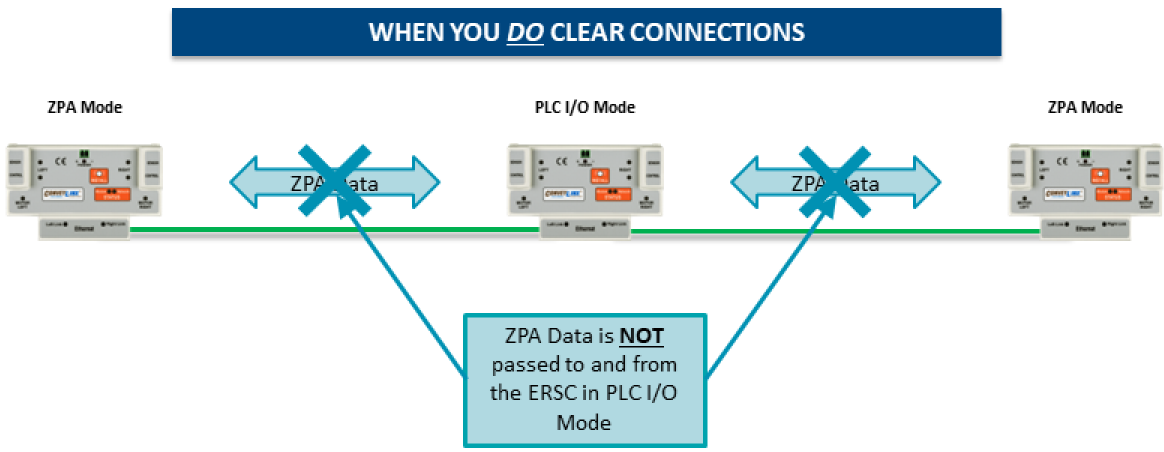
The decision to “Clear Connections” is based upon the application. When a string of ERSC modules are Auto-Configured, each successive ERSC in the string establishes a logical upstream / downstream connection with its neighbor ERSC’s. These connections provide the basis for the logical flow of inter-module status data for ZPA functionality. However, if a single ERSC 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 ERSC 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 an ERSC in PLC I/O mode. For example, if the PLC I/O mode ERSC is being either fed or feeds conveyors controlled by ERSC’s in standard ZPA mode, these ERSC’s 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 ERSC’s 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 an ERSC 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 ERSC’s 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.

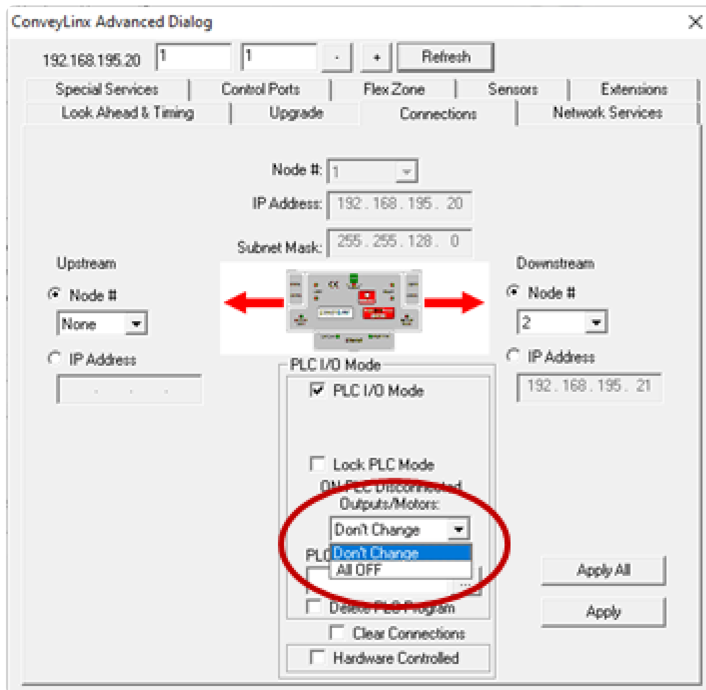
11.4.3. Configuring Action for Loss of Communication

When changing the mode of a given ConveyLinx Module to PLC I/O mode in *EasyRoll*, you are given the option to select the behavior of the ConveyLinx Module’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 Module 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 Module will automatically resume having its outputs and MDR’s controlled by PLC command



11.4.4. PLC Inputs for PLC I/O Mode

This chart shows the assembled registers for the data coming from the ConveyLinx Module in PLC I/O Mode and provided as input to the PLC:

Register Name / Module Address	Assembled Address for PLC	Links
ConveyStop Status Word 4:0019	M: 4:1700 E: I.Data[0] P: Byte 0 (Hi) Byte 1 (Lo)	ConveyStop Status
Sensor & Control Port Inputs 4:0035	M: 4:1701 E: I.Data[1] P: Byte 2 (Hi) Byte 3 (Lo)	Sensor & Control Ports
Sensor Detect 4:0036	M: 4:1702 E: I.Data[2] P: Byte 4 (Hi) Byte 5 (Lo)	Sensor & Control Ports
Module Voltage 4:0024	M: 4:1703 E: I.Data[3] P: Byte 6 (Hi) Byte 7 (Lo)	Sensor & Control Ports
Left Motor Current 4:0055	M: 4:1704 E: I.Data[4] P: Byte 8 (Hi) Byte 9 (Lo)	Left Motor Status

Left Motor Frequency 4:0056	M: 4:1705 E: I.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	Left Motor Status
Left Motor Temperature 4:0057	M: 4:1706 E: I.Data[6] P: Byte 12 (Hi) Byte 13 (Lo)	Left Motor Status
Left Motor Status 4:0058	M: 4:1707 E: I.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	Left Motor Status
Right Motor Current 4:0079	M: 4:1708 E: I.Data[8] P: Byte 16 (Hi) Byte 17 (Lo)	Right Motor Status
Right Motor Frequency 4:0080	M: 4:1709 E: I.Data[9] P: Byte 18 (Hi) Byte 19 (Lo)	Right Motor Status
Right Motor Temperature 4:0081	M: 4:1710 E: I.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	Right Motor Status
Right Motor Status 4:0082	M: 4:1711 E: I.Data[11] P: Byte 22 (Hi)	Right Motor Status

	Byte 23 (Lo)	
Left Motor Port Digital I/O Status 4:0060	M: 4:1712 E: I.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)	Motor Ports Digital Status
Right Motor Port Digital I/O Status 4:0084	M: 4:1713 E: I.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)	Motor Ports Digital Status
Upstream Module Status 4:0134	M: 4:1714 E: I.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	Upstream / Downstream Status & Tracking
Downstream Module Status 4:0232	M: 4:1715 E: I.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	Upstream / Downstream Status & Tracking
Current Tracking Word 1 for Adjacent Upstream Module 4:0139	M: 4:1716 E: I.Data[16] P: Byte 32 (Hi) Byte 33 (Lo)	Upstream / Downstream Status & Tracking
Current Tracking Word 2 for Adjacent Upstream Module 4:0140	M: 4:1717 E: I.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)	Upstream / Downstream Status & Tracking

Current Module Reset Counter N/A	M: 4:1718 E: I.Data[18] P: Byte 36 (Hi) Byte 37 (Lo)	Upstream / Downstream Status & Tracking
Left Motor Servo Position 4:0062	M: 4:1719 E: I.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)	Servo Control Status
Right Motor Servo Position 4:0086	M: 4:1720 E: I.Data[20] P: Byte 40 (Hi) Byte 41 (Lo)	Servo Control Status
Left Motor Servo Status 4:0011	M: 4:1721 E: I.Data[21] P: Byte 42 (Hi) Byte 43 (Lo)	Servo Control Status
Right Motor Servo Status 4:0016	M: 4:1722 E: I.Data[22] P: Byte 44 (Hi) Byte 45 (Lo)	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 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 = Stop active on local module’s Left Control Port bit 9 = Stop active on local module’s Right Control Port bit 10 = Stop active due to Stop Command from PLC bit 11 – 15 Reserved

Refer to Pulseroller publication *ERSC-1800 ConveyStop User’s Guide* for details on using ConveyStop

11.4.4.2. Sensor & Control Ports

Register Name / Module Address	Assembled Address for PLC	Description
Sensor & Control Port Inputs 4:0035	M: 4:1701 E: I.Data [1] P: Byte 2 (Hi) Byte 3 (Lo)	Bitwise Value – Read Only bit 0 = Left Sensor Port – Pin 3 bit 1 = Left Control Port – Pin 3 bit 2 = Right Sensor Port – Pin 3 bit 3 = Right Control Port – Pin 3 bit 4 = Left Sensor Port – Pin 4 bit 5 = Left Control Port – Pin 4 bit 6 = Right Sensor Port – Pin 4 bit 7 = Right Control Port – Pin 4 bit bit 8 – 14 Reserved bit 15 = 2 sec on / 2 sec off heartbeat
Sensor Detect 4:0036	M: 4:1702 E: I.Data [2] P: Byte 4 (Hi) Byte 5 (Lo)	Bitwise Value – Read Only 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 & Control Port Input Signal Condition Mask* register to arrive at the bit values seen in the *Sensor & Control 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 & Control 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 = Calculated motor temperature Low Byte = Temperature reading from the on board sensor
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

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.

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 = Calculated motor temperature Low Byte = Temperature reading from the on board sensor
Right Motor Status 4:0082	M: 4:1711 E: I.Data [11] P: Byte 22 (Hi) Byte 23 (Lo)	<u>Bitwise Value – Read Only:</u> 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

Motor Rotation Status

Motor Frequency

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 wire Motor Port for Digital Output Control

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)	Interger Value of Low or High Byte: 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 ERSC Adjacent to this local ERSC 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 ERSC Adjacent to this local ERSC module
Reserved	M: 4:1718 E: I.Data [18] P: Byte 36 (Hi) Byte 37 (Lo)	

Understanding Adjacent Modules

These registers only contain meaningful data if the ConveyLinx connections between upstream and/or downstream ERSC's are preserved when placing the local ERSC 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 ERSC's. Learn about clearing connections.

11.4.4.7. Servo Control Status

<table>
<tr><th>Register Name /
Module Address</th><th>Assembled
Address for PLC</th><th>Description</th></tr>
<tr><td>Left Motor Servo
Position
4:0062</td><td>M: 4:1719
E: I.Data [19]
P: Byte 38 (Hi)
Byte 39 (Lo)</td><td>Signed integer value that indicates the current position of the Left Motor in relation to its “0” position</td></tr>
<tr><td>Right Motor Servo
Position
4:0086</td><td>M: 4:1720
E: I.Data [20]
P: Byte 40 (Hi)
Byte 41 (Lo)</td><td>Signed integer value that indicates the current position of the Right Motor in relation to its “0” position</td></tr>
<tr><td>Left Motor Servo
Status
4:0011</td><td>M: 4:1721
E: I.Data [21]
P: Byte 42 (Hi)
Byte 43 (Lo)</td><td rowspan="2">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

Bit 2: Servo Command Status
Echoes state of Motor Servo Command bit 1</td></tr>
<tr><td>Right Motor Servo
Status
4:0016</td><td>M: 4:1722
E: I.Data [22]
P: Byte 44 (Hi)
Byte 45 (Lo)</td></tr>
</table>

See Servo Move example

11.4.5. PLC Outputs for PLC I/O Mode

This chart shows the assembled registers for the data from the PLC to the ConveyLinx Module in PLC I/O Mode:

Register Name / Module Address	Assembled Address for PLC	Links
ConveyStop Command 4:0020	M: 4:1800 E: O.Data[0] P: Byte 0 (Hi) Byte 1 (Lo)	ConveyStop Command & Clear Motor Error
Set Left Motor Port Digital Control 4:0060	M: 4:1801 E: O.Data[1] P: Byte 2 (Hi) Byte 3 (Lo)	Motor & Control Port Digital Output
Set Right Motor Port Digital Control 4:0084	M: 4:1802 E: O.Data[2] P: Byte 4 (Hi) Byte 5 (Lo)	Motor & Control Port Digital Output
Control Port Digital Output Control 4:0037	M: 4:1803 E: O.Data[3] P: Byte 6 (Hi) Byte 7 (Lo)	Motor & Control Port Digital Output
Left Motor Run / Reverse 4:0260	M: 4:1804 E: O.Data[4]	Left Motor Control

	P: Byte 8 (Hi) Byte 9 (Lo)	
Left Motor Brake Method 4:0261	M: 4:1805 E: O.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	Left Motor Control
Left Motor Speed Control Method 4:0262	M: 4:1806 E: O.Data[6] P: Byte 12 (Hi) Byte 13 (Lo)	Left Motor Control
Right Motor Run / Reverse 4:0270	M: 4:1807 E: O.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	Right Motor Control
Right Motor Brake Method 4:0271	M: 4:1808 E: O.Data[8] P: Byte 16 (Hi) Byte 17 (Lo)	Right Motor Control
Right Motor Speed Control Method 4:0272	M: 4:1809 E: O.Data[9] P: Byte 18 (Hi) Byte 19 (Lo)	Right Motor Control
Left Motor	M: 4:1810	Left Motor Control

Speed Reference 4:0040	E: O.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	
Right Motor Speed Reference 4:0064	M: 4:1811 E: O.Data[11] P: Byte 22 (Hi) Byte 23 (Lo)	Right Motor Control
Left Motor Acceleration Ramp 4:0043	M: 4:1812 E: O.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)	Left Motor Control
Left Motor Deceleration Ramp 4:0044	M: 4:1813 E: O.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)	Left Motor Control
Right Motor Acceleration Ramp 4:0067	M: 4:1814 E: O.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	Right Motor Control
Right Motor Deceleration Ramp 4:0068	M: 4:1815 E: O.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	Right Motor Control

Clear Motor Error 4:0022	M: 4:1816 E: O.Data[16] P: Byte 32 (Hi) Byte 33 (Lo)	ConveyStop Command & Clear Motor Error
Set Status to Downstream Module 4:0196	M: 4:1817 E: O.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)	Set Status & Tracking
Set Status to Upstream Module 4:0116	M: 4:1818 E: O.Data[18] P: Byte 36 (Hi) Byte 37 (Lo)	Set Status & Tracking
Sensor Port Input Signal Condition Mask 4:0034	M: 4:1819 E: O.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)	Set Sensor & Control Port Input Mask
Set Discharge Tracking Word 1 4:0201	M: 4:1820 E: O.Data[20] P: Byte 40 (Hi) Byte 41 (Lo)	Set Status & Tracking
Set Discharge Tracking Word 2 4:0202	M: 4:1821 E: O.Data[21] P: Byte 42 (Hi) Byte 43	Set Status & Tracking

	(Lo)	
Set Module Reset Count N/A	M: 4:1822 E: O.Data[22] P: Byte 44 (Hi) Byte 45 (Lo)	
Left Motor Servo Command Pulses 4:0008	M: 4:1823 E: O.Data[23] P: Byte 46 (Hi) Byte 47 (Lo)	Servo Control
Left Motor Servo Command Word 4:0009	M: 4:1824 E: O.Data[24] P: Byte 48 (Hi) Byte 49 (Lo)	Servo Control
Right Motor Servo Command Pulses 4:0013	M: 4:1825 E: O.Data[25] P: Byte 50 (Hi) Byte 51 (Lo)	Servo Control
Right Motor Servo Command Word 4:0014	M: 4:1826 E: O.Data[26] P: Byte 52 (Hi) Byte 53 (Lo)	Servo Control

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

Refer to Pulseroller publication *ERSC-1800 ConveyStop User's Guide* for details on using ConveyStop

Learn more about Clear Motor Error

11.4.5.2. Motor & Control Port Digital Output

<table>
<tr><th>Register Name / Module Address</th><th>Assembled Address for PLC</th><th>Description</th></tr>
<tr><td>Set Left Motor Port Digital Control
4:0060</td><td>M: 4:1801
E: O.Data [1]
P: Byte 2 (Hi)
Byte 3 (Lo)</td><td rowspan="2">Bitwise value – “1” Energizes Output:
bit 0 = Motor Port Pin 3
bit 1 = Motor Port Pin 4
bit 2 = Motor Port Pin 5
bit 3 thru bit 5 = Reserved
bit 6 = Brake Output Pin 9
bit 7 = Enable Brake Pin 9 when Motor Port is NOT in Digital mode
bit 8 = Clear over-current error
bit 9 thru 14 = Reserved
bit 15 = Digital Output Enable
0 = Use Port as Motor Control
1 = Use Port as Digital Outputs</td></tr>
<tr><td>Set Right Motor Port Digital Control
4:0048</td><td>M: 4:1802
E: O.Data [2]
P: Byte 4 (Hi)
Byte 5 (Lo)</td></tr>
<tr><td>Control Port Digital Output Control
4:0037</td><td>M: 4:1803
E: O.Data [3]
P: Byte 6 (Hi)
Byte 7 (Lo)</td><td>Bitwise Value – “1” Energizes Output:
bit 0 = bit 0 = Reserved
bit 1 = Left Control Port
bit 2 = Reserved
bit 3 = Right Control Port
All other bits reserved</td></tr>
</table>

Learn more about using Motor Ports for Digital Outputs

11.4.5.3. Left Motor Control

<table>
<tr><th>Register Name / Module Address</th><th>Assembled Address for PLC</th><th>Description</th></tr>
<tr><td>Left Motor Run / Reverse
4:0260</td><td>M: 4:1804
E: O.Data [4]
P: Byte 8 (Hi)
Byte 9 (Lo)</td><td>Bit 0:
1 = Run Command
0 = Stop Command
Bit 8:
0 = Run in Configured Direction
1 = Run opposite of Configured Direction</td></tr>
<tr><td>Left Motor Brake Method
4:0261</td><td>M: 4:1805
E: O.Data [5]
P: Byte 10 (Hi)
Byte 11 (Lo)</td><td><u>Integer Value:</u>
1 = Use Standard Brake Method
2 = Use Free Coast Brake Method
3 = Use Servo 1 Brake Method
4 = Use Servo 2 Brake Method
0 = Remain at last configured or last value entered</td></tr>
<tr><td>Left Motor Speed Control Method
4:0262</td><td>M: 4:1806
E: O.Data [6]
P: Byte 12 (Hi)
Byte 13 (Lo)</td><td><u>Low Byte Value:</u>
0 = Remain at last configured or last value entered
1 = Use Open Loop
2 = Use Closed Loop
<u>Hi Byte Value (requires FW 4.27/5.07 or later):</u>
0 = Ignore
1 = Independent Control
2 = Slave Mode ON, Left motor mirrors Right motor
3 = Slave Mode ON, Left motor runs opposite direction of Right motor</td></tr>
<tr><td>Left Motor Speed Reference
4:0040</td><td>M: 4:1810
E: O.Data [10]
P: Byte 20 (Hi)
Byte 21 (Lo)</td><td><u>Value in % PWM:</u>
Range: 0 to 1000
Example: 400 = 40%
0 = Remain at last non zero value entered</td></tr>
<tr><td>Left Motor Acceleration
4:0043</td><td>M: 4:1812
E: O.Data [12]
P: Byte 24 (Hi)
Byte 25 (Lo)</td><td rowspan="2"><u>When Speed Control Method is in Open Loop:</u>
Value in milliseconds
Range: 0 to 10000 (Deceleration)
Range: 30 to 10000 (Acceleration)
Example: 900 = 0.9 sec

<u>When Speed Control Method is in Closed Loop:</u>
Value in motor pulses
Range: 0 to 10000 (Deceleration)
Range: 30 to 10000 (Acceleration)</td></tr>
<tr><td>Left Motor Deceleration
4:0044</td><td>M: 4:1813
E: O.Data [13]
P: Byte 26 (Hi)
Byte 27 (Lo)</td></tr>
</table>

		For Either Open or Closed Loop 0 = Remain at last non-zero value entered

11.4.5.4. Right Motor Control

<table>
<tr><th>Register Name / Module Address</th><th>Assembled Address for PLC</th><th>Description</th></tr>
<tr><td>Right Motor Run / Reverse
4:0270</td><td>M: 4:1807
E: O.Data [7]
P: Byte 14 (Hi)
Byte 15 (Lo)</td><td>Bit 0:
1 = Run Command
0 = Stop Command
Bit 8:
0 = Run in Configured Direction
1 = Run opposite of Configured Direction</td></tr>
<tr><td>Right Motor Brake Method
4:0271</td><td>M: 4:1808
E: O.Data [8]
P: Byte 16 (Hi)
Byte 17 (Lo)</td><td>Integer Value:
1 = Use Standard Brake Method
2 = Use Free Coast Brake Method
3 = Use Servo 1 Brake Method
4 = Use Servo 2 Brake Method
0 = Remain at last configured or last value entered</td></tr>
<tr><td>Right Motor Speed Control Method
4:0272</td><td>M: 4:1809
E: O.Data [9]
P: Byte 18 (Hi)
Byte 19 (Lo)</td><td>Low Byte Value:
0 = Remain at last configured or last value entered
1 = Use Open Loop
2 = Use Closed Loop
Hi Byte Value (requires FW 4.27/5.07 or later):
0 = Ignore
1 = Independent Control
2 = Slave Mode ON, Right motor mirrors Left motor
3 = Slave Mode ON, Right motor runs opposite direction of Left motor</td></tr>
<tr><td>Right Motor Speed Reference
4:0064</td><td>M: 4:1811
E: O.Data [11]
P: Byte 22 (Hi)
Byte 23 (Lo)</td><td>Value in % PWM:
Range: 0 to 1000
Example: 400 = 40%
0 = Remain at last non zero value entered</td></tr>
<tr><td>Right Motor Acceleration
4:0067</td><td>M: 4:1814
E: O.Data [14]
P: Byte 28 (Hi)
Byte 29 (Lo)</td><td rowspan="2">When Speed Control Method is in Open Loop:
Value in milliseconds
Range: 0 to 10000 (Deceleration)
Range: 30 to 10000 (Acceleration)
Example: 900 = 0.9 sec

When Speed Control Method is in Closed Loop:
Value in motor pulses
Range: 0 to 10000 (Deceleration)
Range: 30 to 10000 (Acceleration)</td></tr>
<tr><td>Right Motor Deceleration
4:0068</td><td>M: 4:1815
E: O.Data [15]
P: Byte 30 (Hi)
Byte 31 (Lo)</td></tr>
</table>

		For Either Open or Closed Loop 0 = Remain at last non-zero value entered

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 ConveyLinx Module ZPA Status to <u>Downstream ERSC</u>: 4 = Instruct Downstream ConveyLinx Module to "wake-up" and run its most upstream zone 1 = Instructs Downstream ConveyLinx 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 ConveyLinx Module's ZPA Status to <u>Upstream ConveyLinx Module</u>: 5 = Instructs Upstream ConveyLinx Module's discharge zone to accumulate and hold any carton that arrives at its discharge zone. 1 = Instructs Upstream ConveyLinx 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 ConveyLinx Module needs to pass tracking data to a downstream connected ConveyLinx Module. 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 ConveyLinx Module's that are in ZPA mode, the connections <u>must NOT be cleared</u> when placing this ConveyLinx Module into PLC I/O mode from within EasyRoll.

Learn more about Clearing Connection Options

11.4.5.6. Set Sensor & Control 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)	Bitwise Value: bit 00 = Left Sensor Port – Pin 3 bit 01 = Left Control Port – Pin 3 bit 02 = Right Sensor Port – Pin 3 bit 03 = Right Control Port – Pin 3 bit 04 = Left Sensor Port – Pin 4 bit 05 = Left Control Port – Pin 4 bit 06 = Right Sensor Port – Pin 4 bit 07 = Right Control Port – Pin 4 bit 08 thru bit 15 = Reserved

How Input signals are conditioned in ZPA Mode

For an ConveyLinx Module in standard ZPA mode, the Auto-Configuration procedure sets values in this register to allow the ConveyLinx Module to correctly display the Sensor and Control port Input circuit 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 ConveyLinx Module's sensor input pin 4 when the zone is clear. The ConveyLinx 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 & Control 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 an ConveyLinx Module is placed in PLC I/O mode; the Sensor & Control Port Input Signal Condition Mask register is cleared of the values set during the Auto-Configure procedure. The Sensor & Control Port Input Signal Condition Mask register is made available for PLC I/O mode Sensor and Control 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 & Control 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 & Control 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 ConveyLinx Module's inputs. The same pattern applies to all 8 available ERSC inputs:

Example: Right Sensor Port – Pin 4 Signal			
Electrical Signal	*Sensor & Control Port Input Signal Condition Mask* Register bit 6	*Sensor & Control 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 Control Port – Pin 3 Signal			
Electrical Signal	*Sensor & Control Port Input Signal Condition Mask* Register bit 1	*Sensor & Control Port Inputs* Register bit 1	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 & Control Port Input Signal Condition Mask* in your Ethernet I/P PLC program. The input bit values in the *Sensor & Control Port Inputs* register can show unexpected or opposite values from expected until the PLC has updated the *Sensor & Control Port Input Signal Condition Mask* data. This update could take several program scans depending upon the Ethernet IP RPI settings. Take care to be sure that the *Sensor & Control Port Input Signal Condition Mask* data is written to the ERSC Before acting on any input values in the *Sensor & Control Port Inputs* register

* Please note that the Sensor/Control Port LEDs are tri-colored. In applications where both Pin 4 and Pin 3 are used on the same Sensor/Control Port; when the green and red LEDs are illuminated at the same time, the color will be amber

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 Pulse Count to "0" Bit 1: Servo Run Command 1 = Run in Motor from current pulse count to set pulse count in Left Motor Servo Command Pulses 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 Pulses 4:0008	M: 4:1823 E: O.Data [23] P: Byte 46 (Hi) Byte 47 (Lo)	Signed integer value in motor pulses of the position to move to on the next Servo Run Command Valid values are from -32767 to +32767
Right Motor Servo Command Pulses 4:0013	M: 4:1825 E: O.Data [25] P: Byte 50 (Hi) Byte 51 (Lo)	

! Servo Motor Control requires ConveyLinx firmware 4.14 or later

! Servo Control functionality requires that the Closed Loop option be enabled for the motor port in question. This can be done either by selecting the check box in EasyRoll or using the PLC to write a value of "2" in the appropriate Motor Speed Control register for the motor in question

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

Servo Control functionality utilizes motor pulses as generated by the BLDC motor's Hall Effect Sensors as the metric for movement and positioning. The quantity of pulses per revolution of the BLDC motor is based upon the number of motor poles X 3. The actual number of pulses per revolution of the MDR tube is based upon the gear reduction ratio of the gearbox in the MDR and the diameter of the tube. Learn about Motor Pulse to Distance Calculation

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 Module:

- Establish a zero or home position by external input to PLC (sensor or operator button)
- Rotate in the CCW direction for 7000 pulses at a speed of 80% PWM with acceleration ramp of 100 pulses and a deceleration ramp of 50 pulses
- Rotate the CW direction for 9000 pulses at a speed of 100% PWM with acceleration ramp of 50 pulses and a deceleration ramp of 10 pulses.
- Rotate CCW back to the zero or home position at a speed of 80% PWM with acceleration ramp of 50 pulses and a deceleration ramp of 10 pulses.
- 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 Module'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
MODE	Integer	Left Motor Speed Control Method

Sequence of Operation

<u>Step #1:</u> Place Left Motor in Closed Loop Mode by writing a “2” to *MODE*

Step #2: 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 #3: 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 #4: If *READY* is reset, then the PLC can set the *RUN* bit to begin the rotation. When the rotation is complete, the ERSC sets the *READY* bit. This will be the signal to the PLC to reset the *RUN* bit. Once the ERSC sees that the *RUN* bit has been reset, it will reset the *READY* bit.

Step #5: 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 700 pulses “forward” and we want to move 900 pulses “backward”. The position we want to end up is 7000-9000 = -2000.

- Write -2000 to POSITION

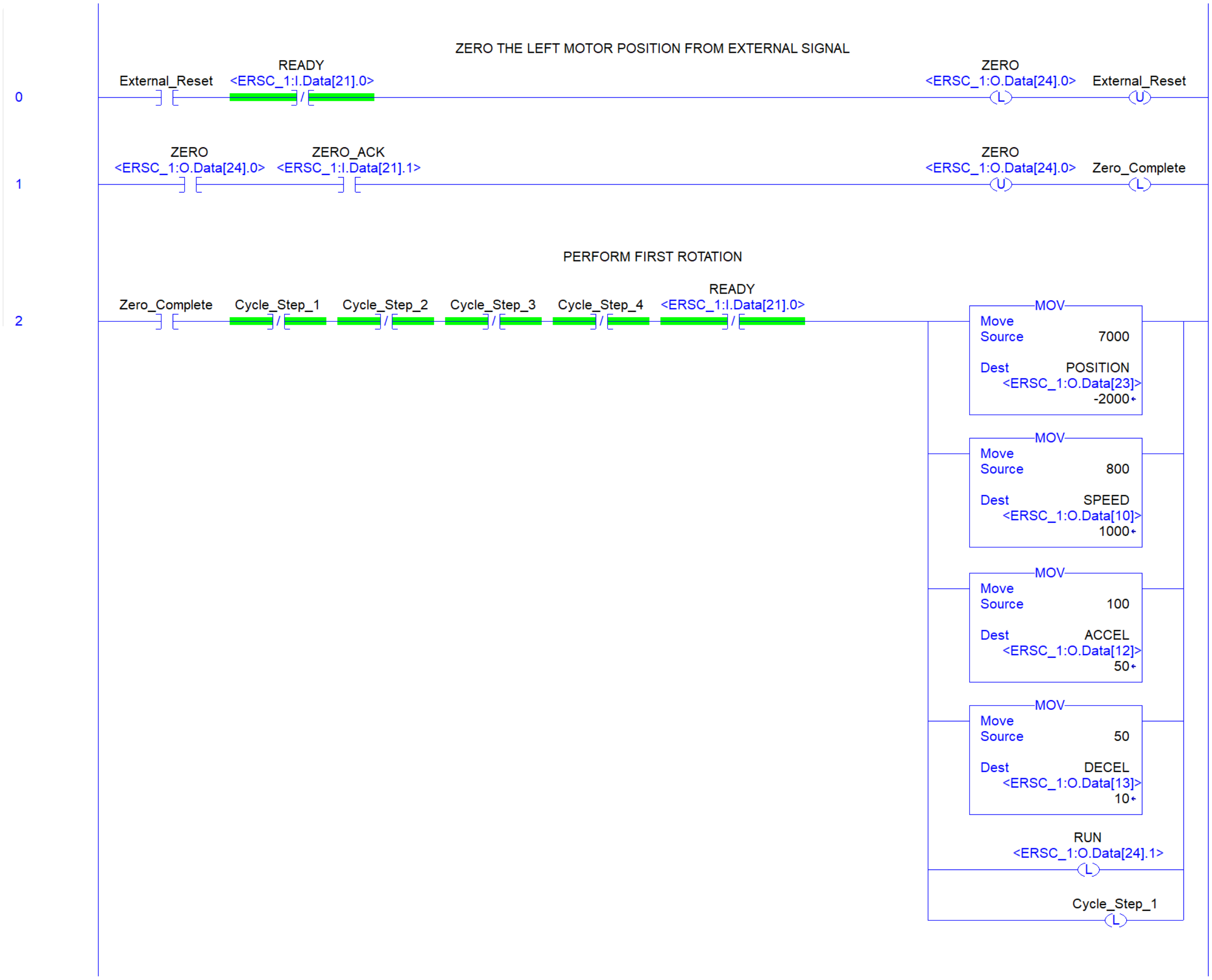
Step #6: Repeat Step #4

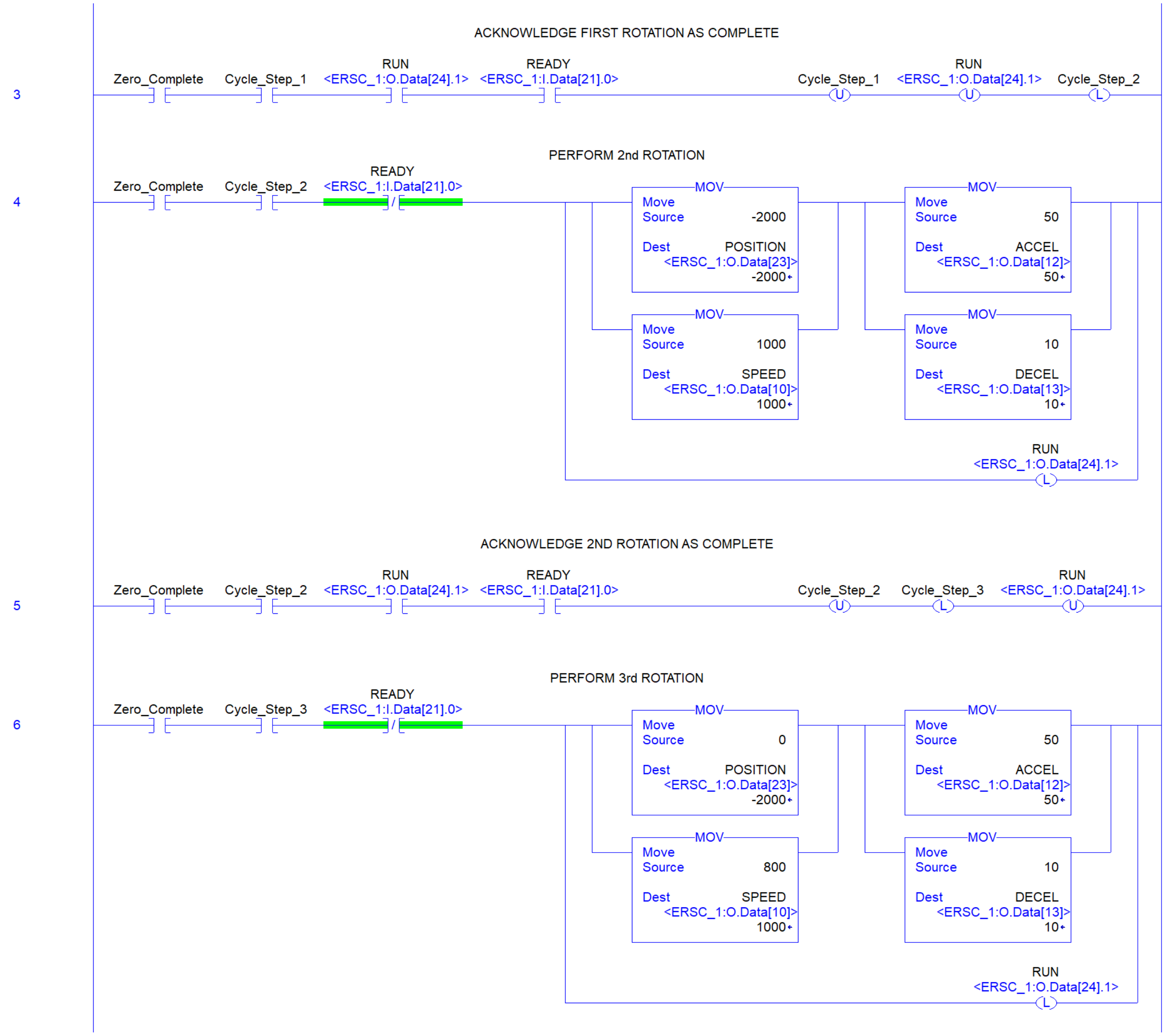
Step #7: For the 3rd rotation, we keep the ramp values from the 2nd rotation, but we need to set the speed and the position to rotate. In this case we want to go to the zero or “home position.

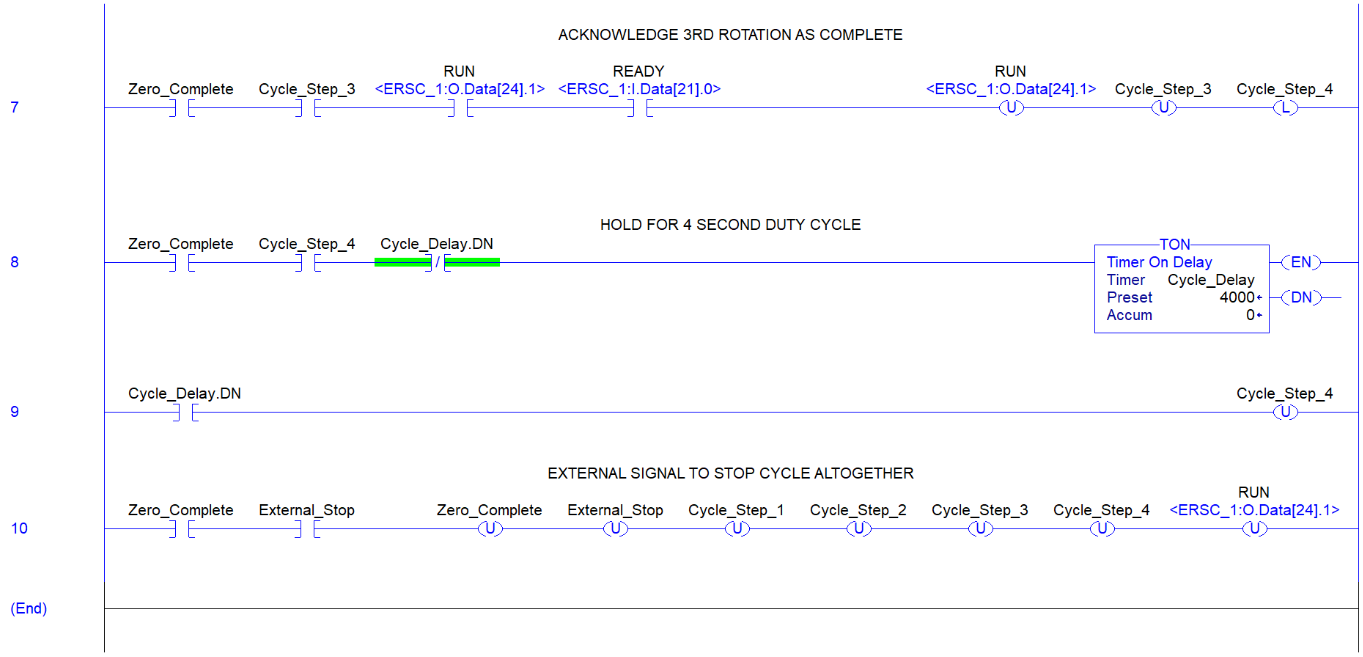
- Write 800 to SPEED
- Write 0 to POSITION

Step #8: Repeat Step #4

PLC Ladder Diagram


0
ZERO THE LEFT MOTOR POSITION FROM EXTERNAL SIGNAL
External_Reset
READY
<ERSC_1:I.Data[21].0>
ZERO
<ERSC_1:O.Data[24].0>
L
External_Reset
U
1
ZERO
<ERSC_1:O.Data[24].0>
ZERO_ACK
<ERSC_1:I.Data[21].1>
ZERO
<ERSC_1:O.Data[24].0>
U
Zero_Complete
L
2
PERFORM FIRST ROTATION
Zero_Complete
Cycle_Step_1
Cycle_Step_2
Cycle_Step_3
Cycle_Step_4
READY
<ERSC_1:I.Data[21].0>
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>
L
Cycle_Step_1
L


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 50
Dest ACCEL
<ERSC_1:O.Data[12]>
50
MOV
Move
Source 1000
Dest SPEED
<ERSC_1:O.Data[10]>
1000
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 50
Dest ACCEL
<ERSC_1:O.Data[12]>
50
MOV
Move
Source 800
Dest SPEED
<ERSC_1:O.Data[10]>
1000
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
Timer Cycle_Delay
Preset 4000
Accum 0
EN
DN
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 Modules 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 Module that is running a ConveyLogix program.

> With ConveyLogix, because a custom user generated program is running in the ConveyLinx Module, 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.

> Refer to Pulseroller publication ERSC-1200 ConveyLogix Programmer's Guide for details on how to create and download a program to your ConveyLinx Module

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.

> ! These assemblies are only available for ConveyLinx Module firmware 4.25 and later and ConveyLogix version 2.02 or later is required to create programs that can access these assemblies

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 an ERSC 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 ERSC module for recovery.

When an ERSC 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 ERSC'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 ERSC. Upon power cycle to the ERSC, 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 ERSC'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 ERSC due to power cycle and because the PLC has not established communications to set the accumulate command bit, the ERSC releases the zone because there is no command present to accumulate.

For ERSC's in PLC I/O mode, Reset Protection may not be as much a concern as for a module in ZPA mode, 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 ERSC 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: I.Data [0]	M: 4:3600
4:0133	Set Local Upstream Zone Tracking Word 2	E: I.Data [1]	M: 4:3601
4:0212	Set Local Downstream Zone Tracking Word 1	E: I.Data [2]	M: 4:3602
4:0213	Set Local Downstream Zone Tracking Word 2	E: I.Data [3]	M: 4:3603
4:0104	Accumulation Control for Local Upstream Zone	E: I.Data [4]	M: 4:3604
4:0184	Accumulation Control for Local Downstream Zone	E: I.Data [5]	M: 4:3605
4:0040	Set Left MDR Speed	E: I.Data [6]	M: 4:3606
4:0064	Set Right MDR Speed	E: I.Data [7]	M: 4:3607
4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	E: I.Data [8]	M: 4:3608
4:0185	Release and Accumulate on Next Arrival for Local Upstream Zone	E: I.Data [9]	M: 4:3609
4:0134	Set Status for Upstream Induct	E: I.Data [10]	M: 4:3610
4:0232	Set Status for Downstream Discharge	E: I.Data [11]	M: 4:3611
4:0139	Set Induct Tracking Forward Direction Word 1	E: I.Data [12]	M: 4:3612
4:0140	Set Induct Tracking Forward Direction Word 2	E: I.Data [13]	M: 4:3613
4:0237	Set Induct Tracking Reverse Direction Word 1	E: I.Data [14]	M: 4:3614
4:0238	Set Induct Tracking Reverse Direction Word 2	E: I.Data [15]	M: 4:3615
4:0022	Clear Motor Error	E: I.Data [16]	M: 4:3616

4:0063	Set Control Port Outputs	E: I.Data [17]	M: 4:3617
N/A	Set Module Reset Count	E: I.Data [18]	M: 4:3618
4:0020	ConveyStop Command	E: I.Data [19]	M: 4:3619
4:0109	Clear Sensor Jam Command for Local Upstream Zone	E: I.Data [20]	M: 4:3620
4:0189	Clear Sensor Jam Command for Local Downstream Zone	E: I.Data [21]	M: 4:3621
4:0365	Direction & Accumulation Mode Control for Local Upstream Zone	E: I.Data [22]	M: 4:3622
4:0375	Direction & Accumulation Mode Control for Local Downstream Zone	E: I.Data [23]	M: 4:3623
4:0387	ConveyMerge Interface	E: I.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 Module 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 Module detects that the value in *Set Module Reset Counter* register is equal to the value in the *Current Module Reset Counter* register; the ConveyLinx 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 not equal to the value in the *Set Module Reset Counter* register; the values in the Input registers will updated by the ConveyLinx Module and will be valid in your PLC program. In this state, even though the PLC may be writing data to Output registers, the ConveyLinx Module will ignore it

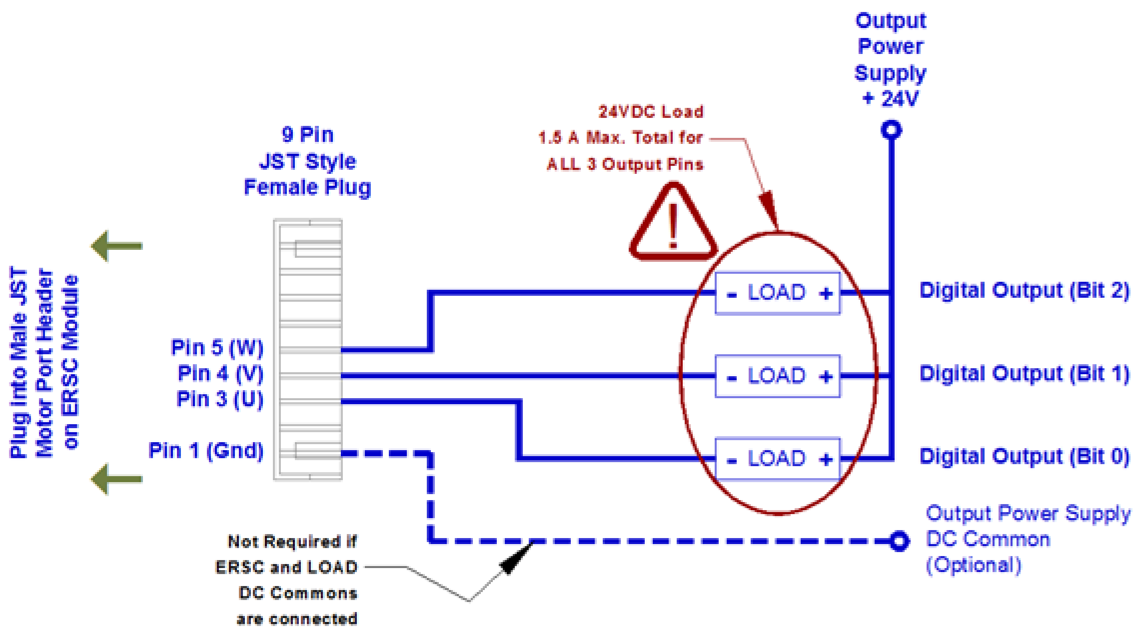
> To make sure a given ZPA zone accumulates upon power up, use *EasyRoll* configuration tool software to set the zone to “Accumulate”. When set from *EasyRoll*, this setting will be retained in the flash memory of the ConveyLinx Module so that the zone will initially accumulate if a load happens to be in the zone at the time of power up. Learn about Accumulate function in EasyRoll

> When a ZPA zone is commanded to accumulate with a PLC and you happen to connect to that ConveyLinx 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 PLC command to accumulate IS NOT retained upon power loss to the ConveyLinx Module. Only if the “Accumulate” switch is toggled “ON” with the 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 an ERSC in PLC I/O Mode

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 ERSC 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. Bit 0, bit 1, and bit 2 can be independently set and reset by the PLC and all 3 digital outputs can be energized simultaneously.



Motor Phases as Digital Outputs

Please note that the ERSC switches to GND to complete the circuit

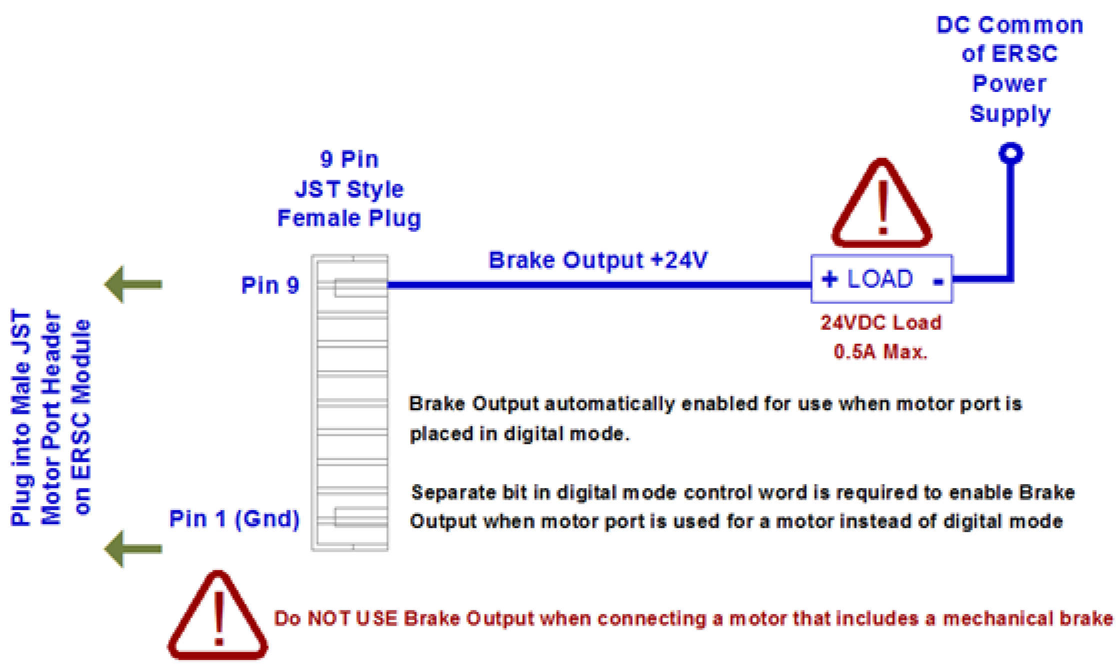
! Short Circuit Error on bit 12 is classified as a “fatal” error that will require either a cycle of power on the ERSC or an explicit Motor Fault Reset command from external controller the same as if the port was being used as a motor port. External controller must continuously write “1” to the Motor Fault Reset register for at least 500 msec for reset to occur

Using the Brake Output Pin on the Motor Port

You can also utilize the Brake Output (Pin 9 on the JST connector) as a +24V output signal controlled from your PLC program when the ERSC is in PLC I/O mode. This signal sources +24V and is limited to 0.5A load.

> ✻ Unlike the motor phases (Pins 3, 4, and 5), the Brake Output is available when the motor port is not in digital mode and is actually running a motor.

When using the Brake Output when the port is actually running a motor, you must set bit 7 in the appropriate left or right “Set Motor Port Digital Control” output word in order to enable the Brake Output functionality. Once you do this, then bit 6 in the same word controls the on/off state of the digital signal. If you have enabled the motor port for digital control by setting bit 15, you do not have to also set bit 7 to enable the Brake Output. It is already enabled when you set bit 15



Brake Output Pin as Digital Output

11.8. ODVA Compliant Cross-Reference

Firmware 5.2 was developed to be 100% compliant with the ODVA specification for Ethernet I/P. In generating this firmware, we found that the numbering system we used to identify our Ethernet I/P assembly instances encroached on reserved values in the ODVA specification. In order for Firmware 5.2 to be ODVA compliant requires that our previous assembly instance identification numbers be reassigned. In so doing, this also had an effect on our Modbus addressing for the contiguous blocks of registers used for accessing assembled data over Modbus TCP

11.8.1. ZPA Mode Assembly Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0116	Local Status Upstream Zone	4:11500	4:1500
4:0196	Local Status Downstream Zone	4:11501	4:1501
4:0106	Arrival Count Local Upstream Zone	4:11502	4:1502
4:0107	Departure Count Local Upstream Zone	4:11503	4:1503
4:0186	Arrival Count Local Downstream Zone	4:11504	4:1504
4:0187	Departure Count Local Downstream Zone	4:11505	4:1505
4:0088	Module Status Word 1	4:11506	4:1506
4:0089	Module Status Word 2	4:11507	4:1507
4:0119	Current Upstream Zone Tracking Word 1	4:11508	4:1508
4:0120	Current Upstream Zone Tracking Word 2	4:11509	4:1509
4:0199	Current Downstream Zone Tracking Word 1	4:11510	4:1510
4:0200	Current Downstream Zone Tracking Word 2	4:11511	4:1511
4:0105	Current Release Count for Upstream Zone	4:11512	4:1512
4:0185	Current Release Count for Downstream Zone	4:11513	4:1513
4:0201	Get Tracking Forward Direction Word 1	4:11514	4:1514
4:0202	Get Tracking Forward Direction Word 2	4:11515	4:1515
4:0237	Get Tracking Reverse Direction Word 1	4:11516	4:1516
4:0238	Get Tracking Reverse Direction Word 2	4:11517	4:1517
4:0035	Sensor & Control Port Inputs	4:11518	4:1518
N/A	Reserved	4:11519	4:1519
4:0019	ConveyStop Status	4:11520	4:1520

11.8.2. ZPA Mode Assembly Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0132	Set Local Upstream Zone Tracking Word 1	4:11600	4:1600
4:0133	Set Local Upstream Zone Tracking Word 2	4:11601	4:1601
4:0212	Set Local Downstream Zone Tracking Word 1	4:11602	4:1602
4:0213	Set Local Downstream Zone Tracking Word 2	4:11603	4:1603
4:0104	Accumulation Control for Local Upstream Zone	4:11604	4:1604
4:0184	Accumulation Control for Local Downstream Zone	4:11605	4:1605
4:0040	Set Left MDR Speed	4:11606	4:1606
4:0064	Set Right MDR Speed	4:11607	4:1607
4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	4:11608	4:1608
4:0185	Release and Accumulate on Next Arrival for Local Downstream Zone	4:11609	4:1609
4:0134	Set Status for Upstream Induct	4:11610	4:1610
4:0232	Set Status for Downstream Discharge	4:11611	4:1611
4:0139	Set Induct Tracking Forward Direction Word 1	4:11612	4:1612
4:0140	Set Induct Tracking Forward Direction Word 2	4:11613	4:1613
4:0237	Set Induct Tracking Reverse Direction Word 1	4:11614	4:1614
4:0238	Set Induct Tracking Forward Direction Word 2	4:11615	4:1615
4:0022	Clear Motor Error	4:11616	4:1616
4:0063	Set Control Port Outputs	4:11617	4:1617
N/A	Reserved	4:11618	4:1618
4:0020	ConveyStop Command	4:11619	4:1619
4:0109	Clear Sensor Jam Command for Local Upstream Zone	4:11620	4:1620

4:0189	Clear Sensor Jam Command for Local Downstream Zone	4:11621	4:1621
4:0365	Direction & Accumulation Mode Control for Local Upstream Zone	4:11622	4:1622
4:0375	Direction & Accumulation Mode Control for Local Downstream Zone	4:11623	4:1623
4:0387	ConveyMerge Interface	4:11624	4:1624

11.8.3. ZPA Mode Reduced Size

ZPA Mode Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0116	Local Status Upstream Zone	4:12900	4:2900
4:0196	Local Status Downstream Zone	4:12901	4:2901
4:0106	Arrival Count Local Upstream Zone	4:12902	4:2902
4:0107	Departure Count Local Upstream Zone	4:12903	4:2903
4:0186	Arrival Count Local Downstream Zone	4:12904	4:2904
4:0187	Departure Count Local Downstream Zone	4:12905	4:2905
4:0088	Module Status Word 1	4:12906	4:2906
4:0089	Module Status Word 2	4:12907	4:2907
4:0105	Current Release Count for Upstream Zone	4:12908	4:2908
4:0185	Current Release Count for Downstream Zone	4:12909	4:2909
4:0035	Sensor & Control Port Inputs	4:12910	4:2910
N/A	Reserved	4:12911	4:2911

ZPA Mode Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0104	Accumulation Control for Local Upstream Zone	4:13000	4:3000
4:0184	Accumulation Control for Local Downstream Zone	4:13001	4:3001
4:0040	Set Left MDR Speed	4:13002	4:3002
4:0064	Set Right MDR Speed	4:13003	4:3003
4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	4:13004	4:3004

4:0185	Release and Accumulate on Next Arrival for Local Downstream Zone	4:13005	4:3005
4:0134	Set Status for Upstream Induct	4:13006	4:3006
4:0232	Set Status for Downstream Discharge	4:13007	4:3007
4:0022	Clear Motor Error	4:13008	4:3008
4:0063	Set Control Port Outputs	4:13009	4:3009
N/A	Reserved	4:13010	4:3010
4:0109	Clear Sensor Jam Command for Local Upstream Zone	4:13011	4:3011
4:0189	Clear Sensor Jam Command for Local Downstream Zone	4:13012	4:3012
4:0365	Direction & Accumulation Mode Control for Local Upstream Zone	4:13013	4:3013
4:0375	Direction & Accumulation Mode Control for Local Upstream Zone	4:13014	4:3014

11.8.4. PLC I/O Mode Assembly Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0019	ConveyStop Status	4:11700	4:1700
4:0035	Sensor & Control Port Inputs	4:11701	4:1701
4:0036	Sensor Detect	4:11702	4:1702
4:0024	Module Voltage	4:11703	4:1703
4:0055	Left Motor Current	4:11704	4:1704
4:0056	Left Motor Frequency	4:11705	4:1705
4:0057	Left Motor Temperature	4:11706	4:1706
4:0058	Left Motor Status	4:11707	4:1707
4:0079	Right Motor Current	4:11708	4:1708
4:0080	Right Motor Frequency	4:11709	4:1709
4:0081	Right Motor Temperature	4:11710	4:1710
4:0082	Right Motor Status	4:11711	4:1711
4:0060	Left Motor Port Digital I/O Status	4:11712	4:1712
4:0084	Right Motor Port Digital I/O Status	4:11713	4:1713
4:0134	Upstream Module Status	4:11714	4:1714
4:0232	Downstream Module Status	4:11715	4:1715
4:0139	Current Tracking Word 1 for Adjacent Upstream Module	4:11716	4:1716
4:0140	Current Tracking Word 2 for Adjacent Upstream Module	4:11717	4:1717
N/A	Reserved	4:11718	4:1718
4:0062	Left Motor Servo Position	4:11719	4:1719
4:0086	Right Motor Servo Position	4:11720	4:1720
4:0011	Left Motor Servo Status	4:11721	4:1721
4:0016	Right Motor Servo Status	4:11722	4:1722

11.8.5. PLC I/O Mode Assembly Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0020	ConveyStop Command	4:11800	4:1800
4:0060	Set Left Motor Port Digital Control	4:11801	4:1801
4:0084	Set Right Motor Port Digital Control	4:11802	4:1802
4:0037	Control Port Digital Output Control	4:11803	4:1803
4:0260	Left Motor Run / Reverse	4:11804	4:1804
4:0261	Left Motor Brake Method	4:11805	4:1805
4:0262	Left Motor Speed Control Method	4:11806	4:1806
4:0270	Right Motor Run / Reverse	4:11807	4:1807
4:0271	Right Motor Brake Method	4:11808	4:1808
4:0272	Right Motor Speed Control Method	4:11809	4:1809
4:0040	Left Motor Speed Reference	4:11810	4:1810
4:0064	Right Motor Speed Reference	4:11811	4:1811
4:0043	Left Motor Acceleration Ramp	4:11812	4:1812
4:0044	Left Motor Deceleration Ramp	4:11813	4:1813
4:0067	Right Motor Acceleration Ramp	4:11814	4:1814
4:0068	Right Motor Deceleration Ramp	4:11815	4:1815
4:0022	Clear Motor Error	4:11816	4:1816
4:0196	Set Status to Downstream Module	4:11817	4:1817
4:0116	Set Status to Upstream Module	4:11818	4:1818
4:0034	Sensor Port Input Signal Condition Mask	4:11819	4:1819
4:0201	Set Discharge Tracking Word 1	4:11820	4:1820
4:0202	Set Discharge Tracking Word 2	4:11821	4:1821
N/A	Reserved	4:11822	4:1822
4:0008	Left Motor Servo Command Pulses	4:11823	4:1823

4:0009	Left Motor Servo Command Word	4:11824	4:1824
4:0013	Right Motor Servo Command Pulses	4:11825	4:1825
4:0014	Right Motor Servo Command Word	4:11826	4:1826

11.8.6. PLC I/O Mode Reduced Size

PLC I/O Mode Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0035	Sensor & Control Port Inputs	4:12700	4:2700
4:0036	Sensor Detect	4:12701	4:2701
4:0057	Left Motor Temperature	4:12702	4:2702
4:0058	Left Motor Status	4:12703	4:2703
4:0081	Right Motor Temperature	4:12704	4:2704
4:0082	Right Motor Status	4:12705	4:2705
4:0060	Left Motor Port Digital I/O Status	4:12706	4:2706
4:0084	Right Motor Port Digital I/O Status	4:12707	4:2707
N/A	Reserved	4:12708	4:2708

PLC I/O Mode Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0060	Set Left Motor Port Digital Control	4:12800	4:2800
4:0084	Set Right Motor Port Digital Control	4:12801	4:2801
4:0037	Control Port Digital Output Control	4:12802	4:2802
4:0260	Left Motor Run / Reverse	4:12803	4:2803
4:0270	Right Motor Run / Reverse	4:12804	4:2804
4:0040	Left Motor Speed Reference	4:12805	4:2805
4:0064	Right Motor Speed Reference	4:12806	4:2806
4:0022	Clear Motor Error	4:12807	4:2807
N/A	Reserved	4:12808	4:2808

11.8.7. Reset Protection for ZPA Mode Assembly Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0116	Local Status Upstream Zone	4:31500	4:3500
4:0196	Local Status Downstream Zone	4:31501	4:3501
4:0106	Arrival Count Local Upstream Zone	4:31502	4:3502
4:0107	Departure Count Local Upstream Zone	4:31503	4:3503
4:0186	Arrival Count Local Downstream Zone	4:31504	4:3504
4:0187	Departure Count Local Downstream Zone	4:31505	4:3505
4:0088	Module Status Word 1	4:31506	4:3506
4:0089	Module Status Word 2	4:31507	4:3507
4:0119	Current Upstream Zone Tracking Word 1	4:31508	4:3508
4:0120	Current Upstream Zone Tracking Word 2	4:31509	4:3509
4:0199	Current Downstream Zone Tracking Word 1	4:31510	4:3510
4:0200	Current Downstream Zone Tracking Word 2	4:31511	4:3511
4:0105	Current Release Count for Upstream Zone	4:31512	4:3512
4:0185	Current Release Count for Downstream Zone	4:31513	4:3513
4:0201	Get Tracking Forward Direction Word 1	4:31514	4:3514
4:0202	Get Tracking Forward Direction Word 2	4:31515	4:3515
4:0237	Get Tracking Reverse Direction Word 1	4:31516	4:3516
4:0238	Get Tracking Reverse Direction Word 2	4:31517	4:3517
4:0035	Sensor & Control Port Inputs	4:31518	4:3518
N/A	Current Module Reset Count	4:31519	4:3519
4:0019	ConveyStop Status	4:31520	4:3520

11.8.8. Reset Protection for ZPA Mode Assembly Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0132	Set Local Upstream Zone Tracking Word 1	4:31600	4:3600
4:0133	Set Local Upstream Zone Tracking Word 2	4:31601	4:3601
4:0212	Set Local Downstream Zone Tracking Word 1	4:31602	4:3602
4:0213	Set Local Downstream Zone Tracking Word 2	4:31603	4:3603
4:0104	Accumulation Control for Local Upstream Zone	4:31604	4:3604
4:0184	Accumulation Control for Local Downstream Zone	4:31605	4:3605
4:0040	Set Left MDR Speed	4:31606	4:3606
4:0064	Set Right MDR Speed	4:31607	4:3607
4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	4:31608	4:3608
4:0185	Release and Accumulate on Next Arrival for Local Downstream Zone	4:31609	4:3609
4:0134	Set Status for Upstream Induct	4:31610	4:3610
4:0232	Set Status for Downstream Discharge	4:31611	4:3611
4:0139	Set Induct Tracking Forward Direction Word 1	4:31612	4:3612
4:0140	Set Induct Tracking Forward Direction Word 2	4:31613	4:3613
4:0237	Set Induct Tracking Reverse Direction Word 1	4:31614	4:3614
4:0238	Set Induct Tracking Forward Direction Word 2	4:31615	4:3615
4:0022	Clear Motor Error	4:31616	4:3616
4:0063	Set Control Port Outputs	4:31617	4:3617
N/A	Set Module Reset Counter	4:31618	4:3618
4:0020	ConveyStop Command	4:31619	4:3619
4:0109	Clear Sensor Jam Command for Local	4:31620	4:3620

	Upstream Zone		
4:0189	Clear Sensor Jam Command for Local Downstream Zone	4:31621	4:3621
4:0365	Direction & Accumulation Mode Control for Local Upstream Zone	4:31622	4:3622
4:0375	Direction & Accumulation Mode Control for Local Downstream Zone	4:31623	4:3623
4:0387	ConveyMerge Interface	4:31624	4:3624

11.8.9. Reset Protection for Reduced Size ZPA Assemblies

ZPA Mode Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0116	Local Status Upstream Zone Forward Direction	4:32900	4:4900
4:0196	Local Status Downstream Zone Forward Direction	4:32901	4:4901
4:0106	Arrival Count Local Upstream Zone	4:32902	4:4902
4:0107	Departure Count Local Upstream Zone	4:32903	4:4903
4:0186	Arrival Count Local Downstream Zone	4:32904	4:4904
4:0187	Departure Count Local Downstream Zone	4:32905	4:4905
4:0088	Module Status Word 1	4:32906	4:4906
4:0089	Module Status Word 2	4:32907	4:4907
4:0105	Current Release Count for Upstream Zone	4:32908	4:4908
4:0185	Current Release Count for Downstream Zone	4:32909	4:4909
4:0035	Sensor & Control Port Inputs	4:32910	4:4910
N/A	Current Module Reset Count	4:32911	4:4911

ZPA Mode Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0104	Accumulation Control for Local Upstream Zone	4:33000	4:5000
4:0184	Accumulation Control for Local Downstream Zone	4:33001	4:5001
4:0040	Set Left MDR Speed	4:33002	4:5002
4:0064	Set Right MDR Speed	4:33003	4:5003

4:0105	Release and Accumulate on Next Arrival for Local Upstream Zone	4:33004	4:5004
4:0185	Release and Accumulate on Next Arrival for Local Downstream Zone	4:33005	4:5005
4:0134	Set Status for Upstream Induct	4:33006	4:5006
4:0232	Set Status for Downstream Discharge	4:33007	4:5007
4:0022	Clear Motor Error	4:33008	4:5008
4:0063	Set Control Port Outputs	4:33009	4:5009
N/A	Set Module Reset Counter	4:33010	4:5010
4:0109	Clear Sensor Jam Command for Local Upstream Zone	4:33011	4:5011
4:0189	Clear Sensor Jam Command for Local Downstream Zone	4:33012	4:5012
4:0365	Direction & Accumulation Mode Control for Local Upstream Zone	4:33013	4:5013
4:0375	Direction & Accumulation Mode Control for Local Upstream Zone	4:33014	4:5014

11.8.10. Reset Protection for PLC I/O Mode Assembly Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0019	ConveyStop Status	4:31700	4:3700
4:0035	Sensor & Control Port Inputs	4:31701	4:3701
4:0036	Sensor Detect	4:31702	4:3702
4:0024	Module Voltage	4:31703	4:3703
4:0055	Left Motor Current	4:31704	4:3704
4:0056	Left Motor Frequency	4:31705	4:3705
4:0057	Left Motor Temperature	4:31706	4:3706
4:0058	Left Motor Status	4:31707	4:3707
4:0079	Right Motor Current	4:31708	4:3708
4:0080	Right Motor Frequency	4:31709	4:3709
4:0081	Right Motor Temperature	4:31710	4:3710
4:0082	Right Motor Status	4:31711	4:3711
4:0060	Left Motor Port Digital I/O Status	4:31712	4:3712
4:0084	Right Motor Port Digital I/O Status	4:31713	4:3713
4:0134	Upstream Module Status	4:31714	4:3714
4:0232	Downstream Module Status	4:31715	4:3715
4:0139	Current Tracking Word 1 for Adjacent Upstream Module	4:31716	4:3716
4:0140	Current Tracking Word 2 for Adjacent Upstream Module	4:31717	4:3717
N/A	Current Module Reset Counter	4:31718	4:3718
4:0062	Left Motor Servo Position	4:31719	4:3719
4:0086	Right Motor Servo Position	4:31720	4:3720
4:0011	Left Motor Servo Status	4:31721	4:3721

4:0016	Right Motor Servo Status	4:31722	4:3722

11.8.11. Reset Protection for PLC I/O Mode Assembly Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0020	ConveyStop Command	4:31800	4:3800
4:0060	Set Left Motor Port Digital Control	4:31801	4:3801
4:0084	Set Right Motor Port Digital Control	4:31802	4:3802
4:0037	Control Port Digital Output Control	4:31803	4:3803
4:0260	Left Motor Run / Reverse	4:31804	4:3804
4:0261	Left Motor Brake Method	4:31805	4:3805
4:0262	Left Motor Speed Control Method	4:31806	4:3806
4:0270	Right Motor Run / Reverse	4:31807	4:3807
4:0271	Right Motor Brake Method	4:31808	4:3808
4:0272	Right Motor Speed Control Method	4:31809	4:3809
4:0040	Left Motor Speed Reference	4:31810	4:3810
4:0064	Right Motor Speed Reference	4:31811	4:3811
4:0043	Left Motor Acceleration Ramp	4:31812	4:3812
4:0044	Left Motor Deceleration Ramp	4:31813	4:3813
4:0067	Right Motor Acceleration Ramp	4:31814	4:3814
4:0068	Right Motor Deceleration Ramp	4:31815	4:3815
4:0022	Clear Motor Error	4:31816	4:3816
4:0196	Set Status to Downstream Module	4:31817	4:3817
4:0116	Set Status to Upstream Module	4:31818	4:3818
4:0034	Sensor Port Input Signal Condition Mask	4:31819	4:3819
4:0201	Set Discharge Tracking Word 1	4:31820	4:3820
4:0202	Set Discharge Tracking Word 2	4:31821	4:3821
N/A	Set Module Reset Count	4:31822	4:3822

4:0008	Left Motor Servo Command Pulses	4:31823	4:3823
4:0009	Left Motor Servo Command Word	4:31824	4:3824
4:0013	Right Motor Servo Command Pulses	4:31825	4:3825
4:0014	Right Motor Servo Command Word	4:31826	4:3826

11.8.12. Reset Protection for PLC I/O Mode Reduced Size

PLC I/O Mode Inputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0035	Sensor & Control Port Inputs	4:32700	4:4700
4:0036	Sensor Detect	4:32701	4:4701
4:0057	Left Motor Temperature	4:32702	4:4702
4:0058	Left Motor Status	4:32703	4:4703
4:0081	Right Motor Temperature	4:32704	4:4704
4:0082	Right Motor Status	4:32705	4:4705
4:0060	Left Motor Port Digital I/O Status	4:32706	4:4706
4:0084	Right Motor Port Digital I/O Status	4:32707	4:4707
N/A	Current Module Reset Count	4:32708	4:4708

PLC I/O Mode Outputs

Module Address	Register Name	Assembled Address for FW 5.2	Assembled Address for FW 4.25 and earlier
4:0060	Set Left Motor Port Digital Control	4:32800	4:4800
4:0084	Set Right Motor Port Digital Control	4:32801	4:4801
4:0037	Control Port Digital Output Control	4:32802	4:4802
4:0260	Left Motor Run / Reverse	4:32803	4:4803
4:0270	Right Motor Run / Reverse	4:32804	4:4804
4:0040	Left Motor Speed Reference	4:32805	4:4805
4:0064	Right Motor Speed Reference	4:32806	4:4806
4:0022	Clear Motor Error	4:32807	4:4807
N/A	Set Module Reset Count	4:32808	4:4808

12. Setting Up Dual Motor Zones

! THIS SECTION ONLY APPLIES TO FIRMWARE 4.25 AND OLDER AND 5.02 AND OLDER. FOR FIRMWARE 4.27 / 5.07 AND NEWER THIS FUNCTIONALITY IS PROVIDED AUTOMATICALLY THROUGH SET UP IN EASYROLL

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.

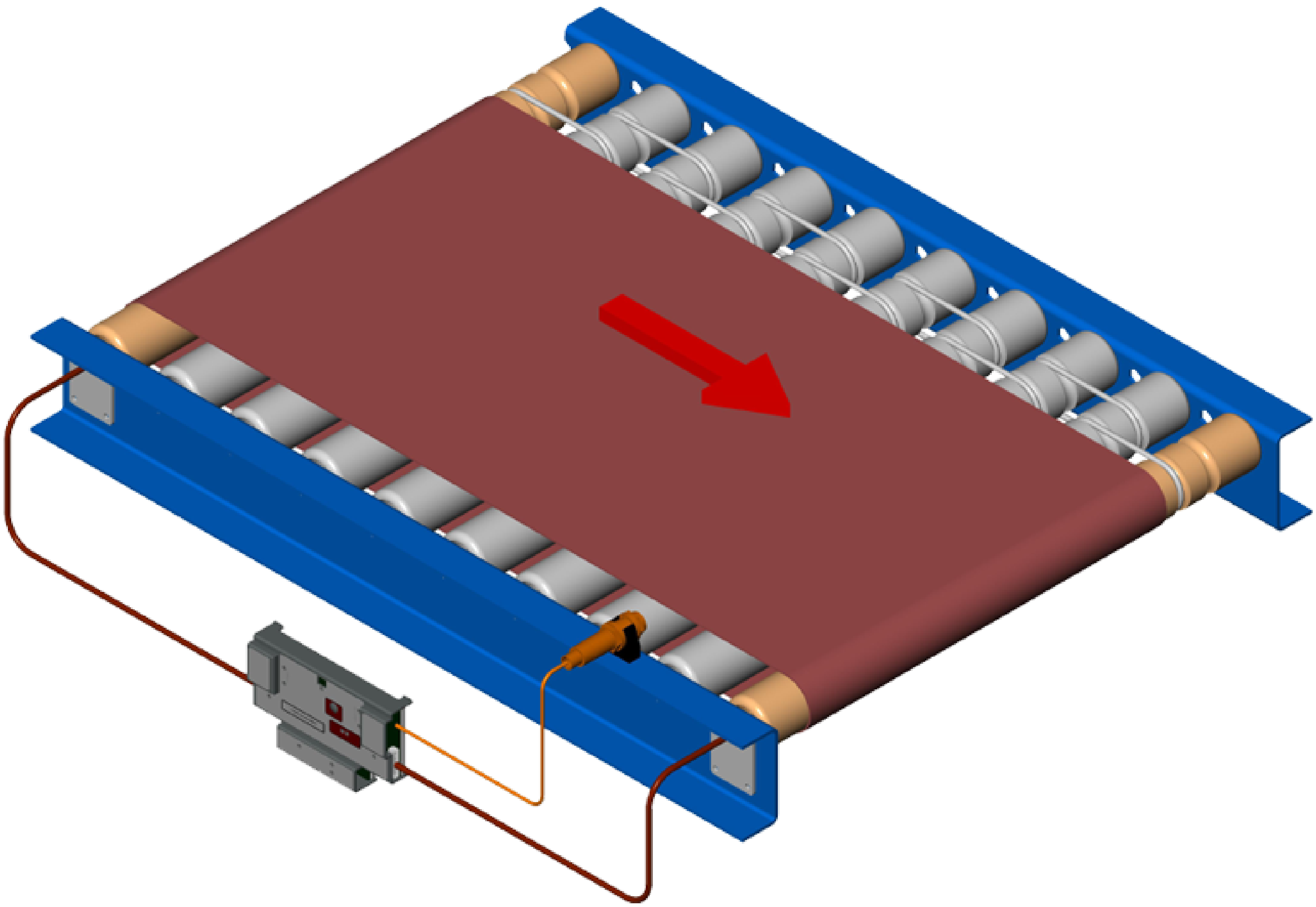
Solution Matrix for ConveyLinx-ERSC Family Modules

Zone Type	Operational Mode	Solution
2 MDRs – Coupled	ZPA	Requires set-up in EasyRoll
	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	Not required to be set-up in Easy-Roll, but can be
	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	Requires set-up in EasyRoll
	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. 2 MDRs Mechanically Coupled

> ! THIS SECTION ONLY APPLIES TO FIRMWARE 4.25 AND OLDER AND 5.02 AND OLDER. FOR FIRMWARE 4.27 / 5.07 AND NEWER THIS FUNCTIONALITY IS PROVIDED AUTOMATICALLY THROUGH SET UP IN EASYROLL

This scenario is perhaps the most common in that this would be the case for a two-motor roller belted zone used in an incline or decline application as shown. Please note that mechanically the two motor rollers can be coupled with just O-bands, Poly-V belts, etc. 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 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. If the modules are at factory default, both motors will be at factory default settings. Otherwise, the motor settings will be the ones last set by EasyRoll.

> ! Auto-Configuration of a two motor roller logical zone with an ERSC **does not** automatically cause both motors to adopt the same motor settings.

For Firmware 4.25 or ODVA Firmware 5.02 and EasyRoll version 4.19

For Firmware 4.27 and later or ODVA Firmware 5.07 and later and EasyRoll version 4.21

12.1.1.1. For Firmware 4.25/5.02

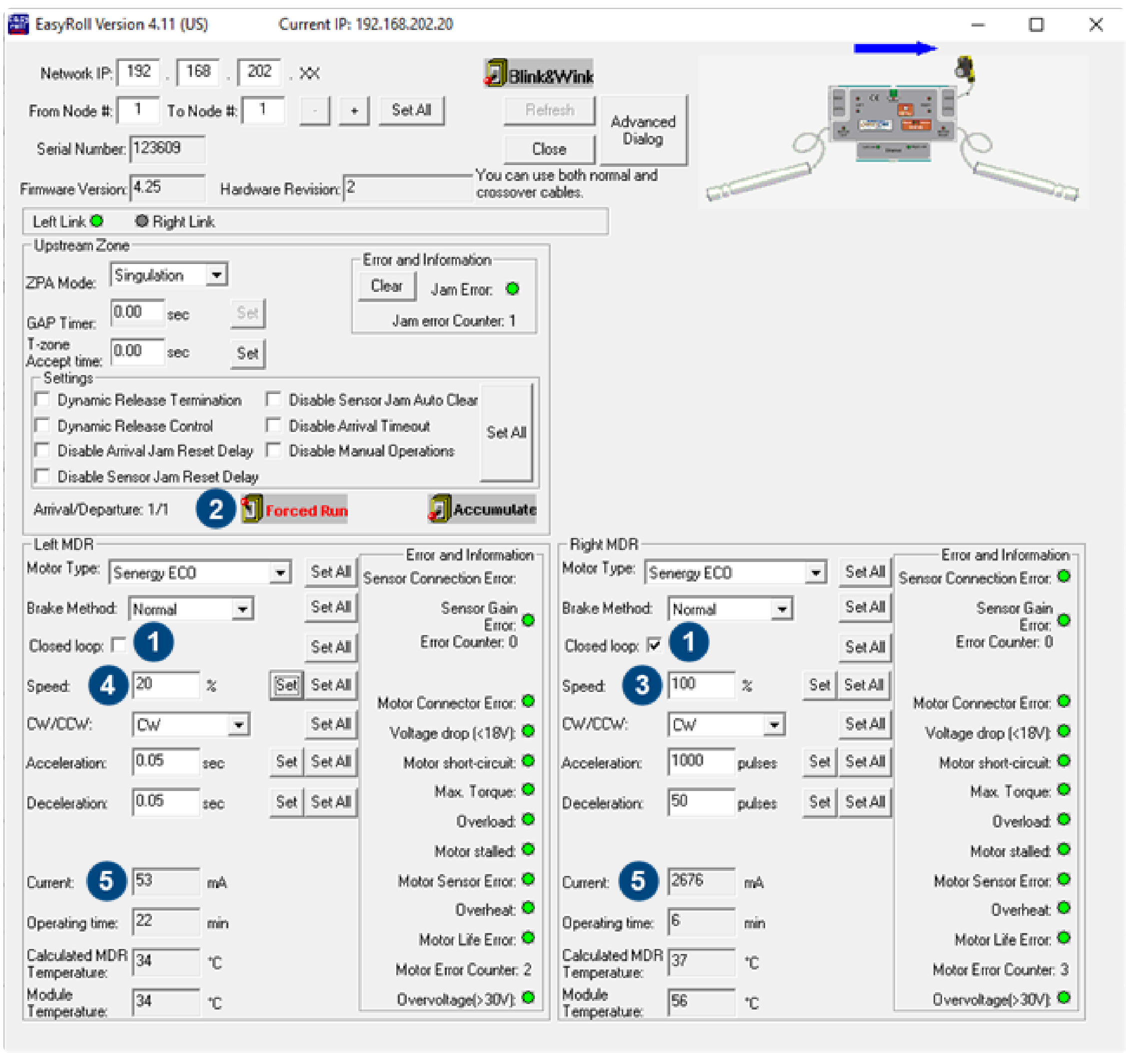
Open Loop

If open loop speed control is desired; then set both motors to the same settings in EasyRoll

Closed Loop

When using closed loop control, it is important to *never place both motors into closed loop* control because they will "fight" each other particularly when accelerating and decelerating. Perform the following basic steps in EasyRoll for the selected ERSC module:

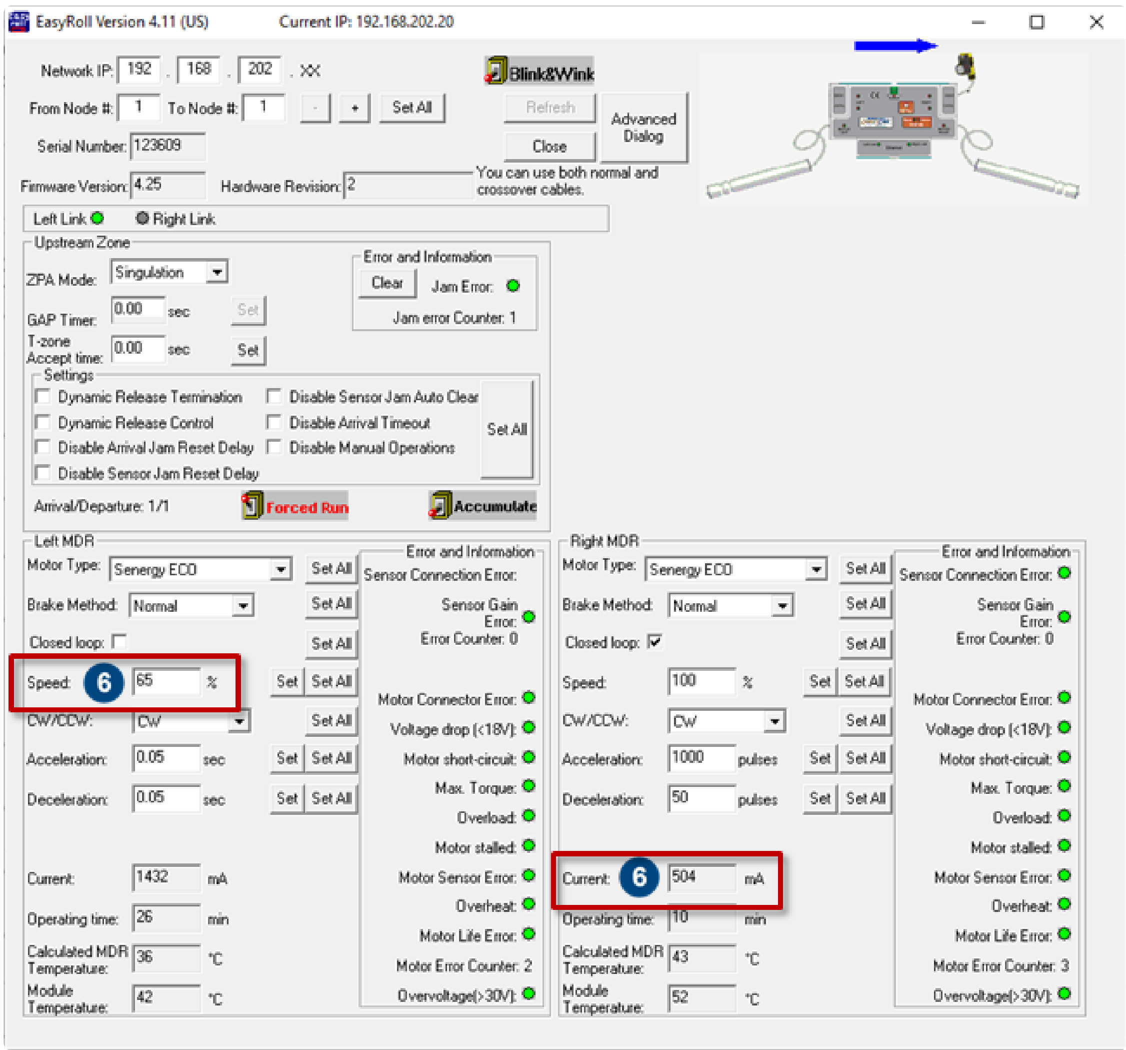
1. Place discharge end motor in closed loop and infeed end motor in open loop
2. Force Run and both motors should run
3. Enter desired speed for the zone for the closed loop motor
4. Enter slow speed in open loop motor
5. Verify that the open loop motor



current is non zero and that closed loop current is in the range of the rated current

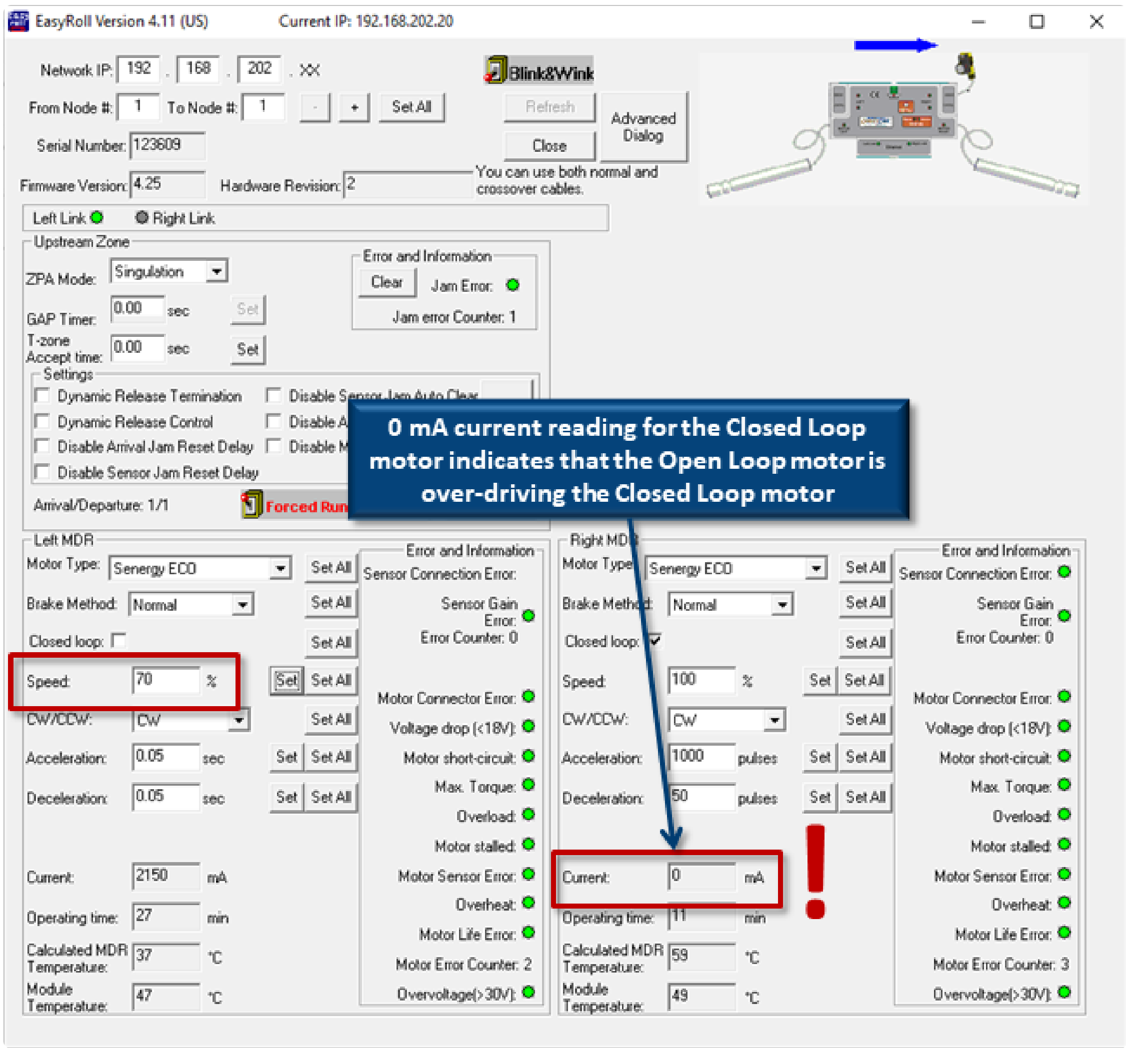
6 Adjust the open loop motor’s speed until the closed loop motor’s current is approximately 500 to 800 mA

Please Note that the speed values shown are examples only. The actual speed and current values you see may be different



It is important to not set the Open Loop motor's speed so high as to make the Closed Loop motor's current draw be 0 mA

In our example, previously our setting of 65% speed on the Open Loop motor resulted in @500 mA draw on the Closed Loop motor. But an adjustment to 70% speed on the Open Loop motor results in 0 mA current draw on the Closed Loop motor. This will eventually cause an overheat error on the



Closed Loop motor.

12.1.1.2. For Firmware 4.27/5.07 and EasyRoll 4.21

This is automatically handled when you utilize the *Slave Function* in EasyRoll

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 either control each motor independently with your logic or utilize some specific specialized programming to couple the motors together so that the on-board firmware manages the master/slave control of the motors.

For Firmware 4.25 or ODVA Firmware 5.02

For Firmware 4.27 and later or ODVA Firmware 5.07 and later and EasyRoll 4.21 and later

12.1.2.1. For Firmware 4.25/5.02

Open Loop

If open loop control is acceptable for your application, simply set both motors to open loop and use the same settings in EasyRoll and write your PLC logic to run both motors at the same time and in the same direction.

Closed Loop

You have two choices if you wish to used closed loop speed control: ***Independent*** and ***Master/ Slave***.

Independent

With this method you set up closed loop as described above for *ZPA mode*. In your PLC logic, you must run both motors at the same time and in the same direction.

Master / Slave

The ConveyLinx Module firmware has the capability when in PLC I/O mode to operate both motors at the same time by setting one motor as a master and its settings will be used for both motors. With this method, your PLC logic first configures the functionality and then only logically controls the master motor.

Unfortunately, the register to configure this functionality (4:0015) is not included in any of the PLC I/O mode assemblies. Your PLC code must utilize a MSG instruction to write to this configuration register.

The data is to be written to Register 4:0015 based on the following:

- Bit 0: Set for Left motor to be Master
- Bit 1: Set to make Right motor rotate opposite direction when Left is Master
- Bit 2: Set Right motor to be Master
- Bit 3: Set to make Left motor rotate opposite direction when Right is Master
- Bit 4: Always set to enable the functionality

Programming Tip: You can set up your MSG instruction to continually write to Register 15, even though the ConveyLinx Module only needs to get written to once. If continually sending a MSG instruction is of concern for conservation of PLC resources; you can monitor the module's Reset Counter register or AOI Module Enable bit to trigger the MSG write to register 15 only when you detect the module has power cycled or re-booted

You should use these bits as pairs. Either use bits 0 and 1 or use bits 2 and 3. Setting both bits 0 and 2 will not work.
For example, if you set bit 0, then your program logic only needs to control the Left motor in the PLC I/O instance assembly and the Right motor will have its settings internally controlled by the module.

> ! Please note when using Master/Slave motor control, the ConveyLinx Module ignores the EasyRoll settings for the slave motor

> ! In 2 motor coupled conveyor application, always remember to never set bits 1 or 3

12.1.2.2. For Firmware 4.27/5.07 and EasyRoll 4.21

Functionality is provided within the defined *PLC I/O Assembly*. See the Left Motor Control topic to configure the Left Motor as a slave and see the Right Motor Control topic to configure the Right Motor as a slave.

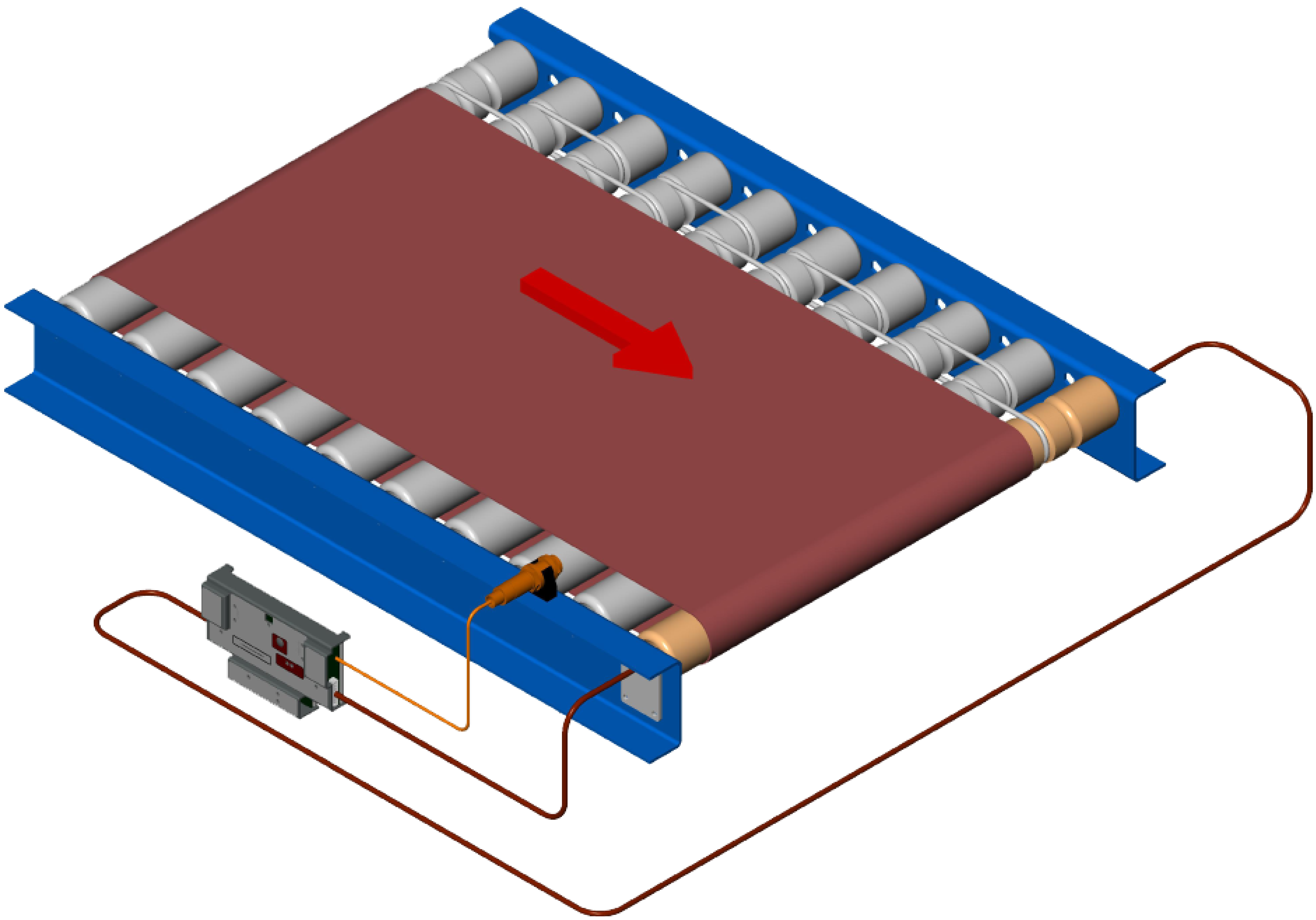
12.1.3. ConveyLogix Program

You follow the same procedure as described for PLC I/O mode. You use the *FirstLadderExec* controller tag Boolean in a rung that executes a *WRC* block that writes the desired data to Register 4:0015. By using the FirstLadderExec in this rung, it makes sure the data is written upon any power up or re-boot of the ConveyLinx Module and that it is not continuously writing the data every program scan and using up unnecessary module resources.

12.2. Two Motors in One Roller Tube

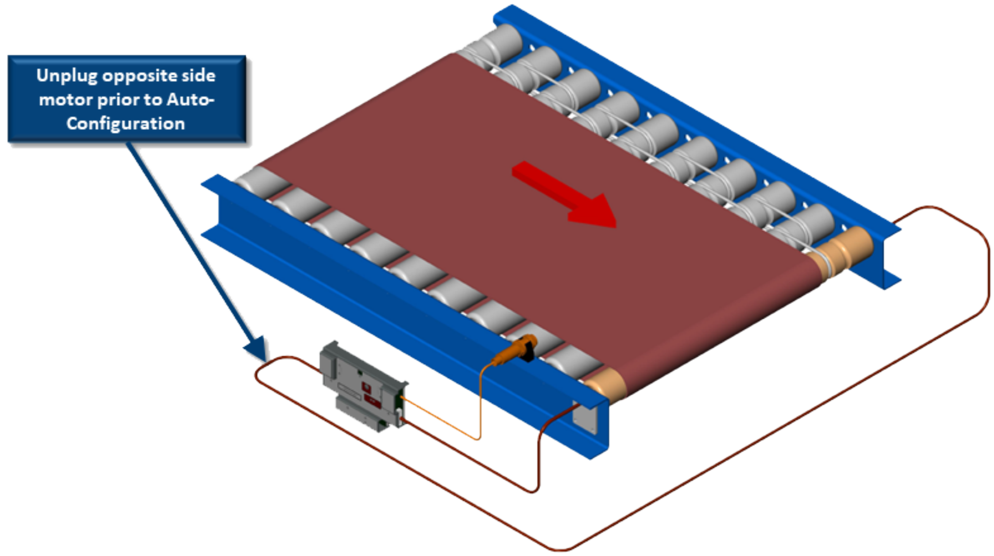
> ! THIS SECTION ONLY APPLIES TO FIRMWARE 4.25 AND OLDER AND 5.02 AND OLDER. FOR FIRMWARE 4.27 / 5.07 AND NEWER THIS FUNCTIONALITY IS PROVIDED AUTOMATICALLY THROUGH SET UP IN EASYROLL

Some Pulseroller models are available with two motors in one tube as shown.



> ! Regardless of which mode you will use, prior to performing the ***Auto-Configuration Procedure***, unplug the motor cable opposite the ConveyLinx Module

During the *Auto-Configure Procedure*, all zones become *ZPA mode* and motors will run. The ConveyLinx Module 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



12.2.1. ZPA Mode

Once *Auto-Configure* is complete with the opposite motor cable unplugged, you can reconnect this motor cable.

For Firmware 4.25 or ODVA Firmware 5.02

You follow the same procedures as described for ZPA Mode in section Mechanically Coupled 2 Motor Roller – ZPA Mode for both Open Loop and Closed Loop control.

> ! Be sure to change the rotation direction in EasyRoll for the opposite motor

For Firmware 4.27 and later or ODVA Firmware 5.07 and later and EasyRoll 4.21

This is handled by the *Motor Slave Function* in EasyRoll

> ! Be sure to select the appropriate "opposite dir." selection from the Motor Slave drop down selector

12.2.2. PLC I/O Mode

After *Auto-Configuration* is complete, you can reconnect the opposite motor cable and place the ConveyLinx Module into *PLC I/O* mode with *EasyRoll*.

For Firmware 4.25 or ODVA Firmware 5.02

You have the same options as described in section *Mechanically Coupled 2 Motor Roller – PLC I/O Mode*. For Open Loop control and Closed Loop – Independent control, remember to change the direction of the opposite motor in *EasyRoll*. For Master/Slave control, be sure to remember to set either bit 1 or bit 3 in Register 4:0015 to change the direction of the slave motor.

> ! Be sure to change the opposite motor's direction either in EasyRoll or Register 15 depending on the method you choose

For Firmware 4.27 and later or ODVA Firmware 5.07 and later and EasyRoll 4.21

Functionality is provided within the defined *PLC I/O Assembly*. See the Left Motor Control topic to configure the Left Motor as a slave and see the Right Motor Control topic to configure the Right Motor as a slave.

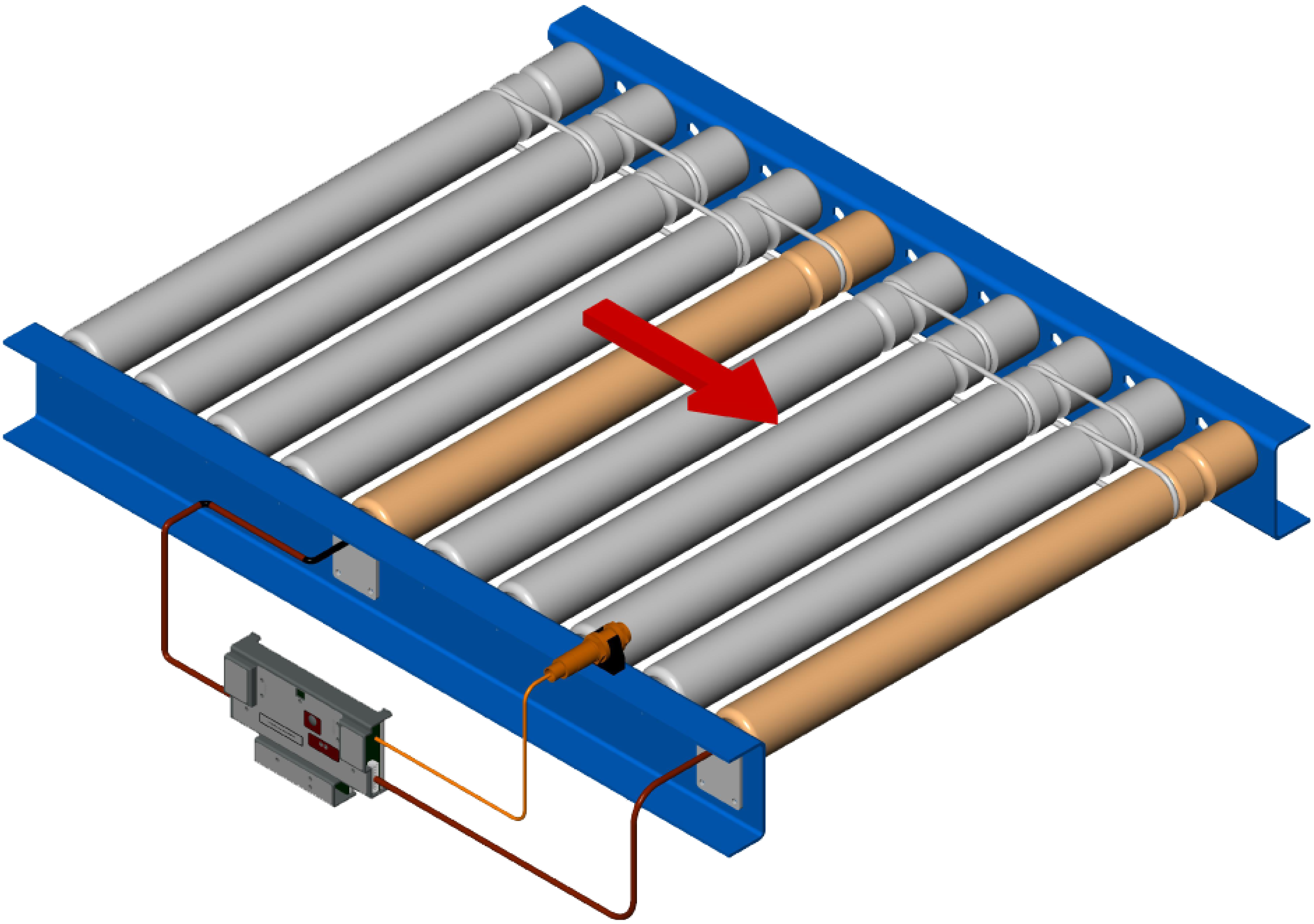
> ! Be sure to use the appropriate "opposite direction" value for the Hi Byte in the register

12.2.3. ConveyLogix Program

Follow the exact procedure as described in section *Mechanically Coupled 2 Motor Roller – ConveyLogix Program*

12.3. Two Motor Rollers in One Logical Zone Not Coupled

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



✻ Because the motors are not mechanically coupled, set the motor settings the same in *EasyRoll* and there are no other special things to do regardless of ZPA or PLC I/O mode

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 our PLC Developer’s Guide, the data that you exchange with your PLC and a given ConveyLinx module 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 module for each set of Input and Output Assemblies described in the PLC Developer's Guide.

ODVA Compliant Firmware 5.xx for ERSC

Pulseroller has been granted a Certificate of Conformity from ODVA for ConveyLinx ERSC firmware version 5.02. The main difference between firmware 5.xx and previous versions (4.xx, 3.xx) as it pertains to connecting to ODVA compliant Ethernet I/P (EIP) PLC devices is that these previous versions utilized Instance Identifiers that were classified as "reserved" by the ODVA specification.

All Firmware 5.xx versions have re-assigned these identifiers into the allowable range for ODVA compliance. The actual data registers and functionality of all EIP assemblies has remained unchanged from the published assemblies in our PLC Developer's Guide. The only thing that has changed in version 5.xx is the value used for the Instance Identifiers when connecting to the PLC. Firmware 4.25/4.27 recognizes both the previous and ODVA values for backward compatibility if you happen to upgrade ERSC firmware from 4.24 (or earlier) to 4.25 or 4.27. The following chart is a reference showing all the available assemblies and their respective Instance Values used when connecting as a Generic Ethernet Device.

Assembly	Recognized Instance Values		
	Pre 4.25	4.25/4.27	5.xx
ZPA Mode Assembly Inputs	5	5 & 105	105
ZPA Mode Assembly Inputs	5	5 & 105	105
ZPA Mode Assembly Outputs	6	6 & 106	106
ZPA Mode Assembly Inputs with Reset Protection	25	25 & 305	305
ZPA Mode Assembly Outputs with Reset Protection	26	26 & 306	306
Reduced Size ZPA Mode Assembly Inputs	19	19 & 119	119
Reduced Size ZPA Mode Assembly Outputs	20	20 & 120	120
Reduced Size ZPA Mode Assembly Inputs with Reset Protection	39	39 & 319	319

Reduced Size ZPA Mode Assembly Outputs with Reset Protection	40	40 & 320	320
PLC I/O Mode Assembly Inputs	7	7 & 107	107
PLC I/O Mode Assembly Outputs	8	8 & 108	108
PLC I/O Mode Assembly Inputs with Reset Protection	27	27 & 307	307
PLC I/O Mode Assembly Outputs with Reset Protection	28	28 & 308	308
Reduced Size PLC I/O Mode Assembly Inputs	17	17 & 117	117
Reduced Size PLC I/O Mode Assembly Outputs	18	18 & 118	118
Reduced Size PLC I/O Mode Assembly Inputs with Reset Protection	37	37 & 317	317
Reduced Size PLC I/O Mode Assembly Outputs with Reset Protection	38	38 & 318	318
ConveyLogix Assembly Inputs	Not Available	121	121
ConveyLogix Assembly Outputs	Not Available	122	122

13.3. Procedure for Connecting using Generic Ethernet Module

All assembly pairs can be connected to a single ERSC 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 ERSC modules 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 ERSC.

https://www.youtube.com/embed/21vxtPUUWv8?rel=0

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"

For Firmware 4.24 and Earlier

Assembly	Type	Instance Value	Size Value
ZPA Mode Assembly	Input	5	21
	Output	6	24
ZPA Mode Assembly with Reset Protection	Input	25	21
	Output	26	24
PLC I/O Mode Assembly	Input	7	23
	Output	8	27
PLC I/O Mode Assembly with Reset Protection	Input	27	23
	Output	28	27

For Firmware 4.25 and 4.27

Assembly	Type	Instance Value	Size Value
ZPA Mode Assembly	Input	5	21
	Output	6	25
ZPA Mode Assembly with Reset Protection	Input	25	21
	Output	26	25
Reduced Size ZPA Mode Assembly	Input	19	12
	Output	20	15
Reduced Size ZPA Mode Assembly with Reset Protection	Input	39	12
	Output	40	15
PLC I/O Mode Assembly	Input	7	23
	Output	8	27

PLC I/O Mode Assembly with Reset Protection	Input	27	23
	Output	28	27

Reduced Size PLC I/O Mode Assembly	Input	17	9
	Output	18	9
Reduced Size PLC I/O Mode Assembly with Reset Protection	Input	37	9
	Output	38	9
ConveyLogix Assembly	Input	121	16
	Output	122	16

For Firmware 5.02 and Higher

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	23
	Output	108	27
PLC I/O Mode Assembly with Reset Protection	Input	307	23
	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

Selecting the Proper EDS File

The first step is to select the proper EDS file based upon the firmware version of your ConveyLinx ERSC modules. Our Pulseroller.com website contains all EDS files for download including older versions. The following chart lists firmware version, operation mode, and EDS file cross-reference information:

ERSC Firmware	ERSC Mode	EDS File
4.24 and Earlier	ZPA Mode Only	ConveyLinx_ZPA_Instance_1.eds
4.24 and Earlier	PLC I/O Mode Only	ConveyLinx_PLC_IO_Instance_1.eds
4.25 and 4.27	ZPA & PLC I/O Mode	ConveyLinx_V5_6.eds
5.02 and Later	ZPA & PLC I/O Mode	ConveyLinx_V5_6.eds

! For best results, you should remove any previous ERSC EDS file(s) you may have installed in your RSLogix 5000 environment before installing a version from this list

! 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 ERSC 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 an ERSC, the data is arranged in assembled registers with the data appearing in your Controller Tags similarly to how the data appears when you connect to an ERSC as a Generic Ethernet Module.

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

13.5. Using ERSC 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 an ERSC module regardless of mode; your PLC program needs to directly access the register data array tags created when you created the ERSC instance. The AOIs attach to created ERSC'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 ERSC you want to connect: a ZPA mode AOI and a PLC I/O AOI

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

Selecting the Proper AOI Instruction

AOI 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 ERSC firmware version and mode of operation

ERSC Firmware	ERSC Mode	EDS File	AOI File
4.24 and Earlier	ZPA Mode	ConveyLinx_ZPA_Instance_1.eds	ERSC_ZPA_424.L5X
4.24 and Earlier	PLC I/O Mode	ConveyLinx_PLC_IO_Instance_1.eds	ERSC_PLCIO_424.L5X
4.25	ZPA Mode	ConveyLinx_V5_6.eds	ERSC_ZPA_425_5xx.L5X
4.25	PLC I/O Mode	ConveyLinx_V5_6.eds	ERSC_PLCIO_425_5xx.L5X
5.02	ZPA Mode	ConveyLinx_V5_6.eds	ERSC_ZPA_425_5xx.L5X
5.02	PLC I/O Mode	ConveyLinx_V5_6.eds	ERSC_PLCIO_425_5xx.L5X
4.27/5.07	ZPA Mode	ConveyLinx_V5_6.eds	ERSC_ZPA_427_507.L5X
4.27/5.07	PLC I/O Mode	ConveyLinx_V5_6.eds	ERSC_PLCIO_425_507.L5X

> Please refer to the downloads section of Pulseroller.com for the latest updates to

EDS and/or AOI files

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

These tags are for AOI file ERSC_ZPA_427_507.L5X

Tag Name	Data Type	Module Register	Bit	Description
I_ArrivalUPZn	BOOL	4:0116	AOI Logic	Arrival at Local Upstream Zone – Only active if zone becomes occupied and it has been set to accumulate
I_ArrivalDNZn	BOOL	4:0196	AOI Logic	Arrival at Local Downstream Zone – Only active if zone becomes occupied and it has been set to accumulate
I_ControlPortPin3_Left	BOOL	4:0035	1	Left Control Port Pin 3 Energized
I_ControlPortPin3_Right	BOOL	4:0035	3	Right Control Port Pin 3 Energized
I_ControlPortPin4_Left	BOOL	4:0035	5	Left Control Port Pin 4 Energized
I_ControlPortPin4_Right	BOOL	4:0035	7	Right Control Port Pin 4 Energized
I_ConveyStopByLeftControlPort	BOOL	4:0020	8	ConveyStop Activated at Local Left Control Port
I_ConveyStopByLostConnection	BOOL	4:0020	6	ConveyStop Activated because of Lost Connection
I_ConveyStopByPLCCmd	BOOL	4:0020	7	ConveyStop Activated because of PLC Command
I_ConveyStopByPLCDisconnect	BOOL	4:0020	10	ConveyStop Activated because of Lost PLC Connection
I_ConveyStopByRemoteModule	BOOL	4:0020	5	ConveyStop Activated by another module in Stop Group
I_ConveyStopByRightControlPort	BOOL	4:0020	9	ConveyStop Activated at Local Right Control Port
I_GetForwardTracking	DINT	4:0201 (MSW) 4:0202 (LSW)	-	Current Forward Tracking Value at Induct to Local Upstream Zone

I_Heartbeat	BOOL	4:0035	15	Module Heartbeat
I_JamAtUPZn	BOOL	4:0088	5	Sensor Jam at Local Upstream Zone
I_JamAtDNZn	BOOL	4:0089	5	Sensor Jam at Local Downstream Zone
I_ModuleEnabled	BOOL	-	-	Local ERSC Module Output are Enabled
I_ModuleFault	BOOL	4:0088 4:0089	AOI Logic	Module Fault Active (Logical OR of bits 2,4,and 7 from 5-6 and bits 2 and 7 from 5-7
I_ModuleStatus	DINT	4:0088 (MSW) 4:0089 (LSW)		Modules Status Words 1 and 2
I_MtrError_Left	BOOL	4:0088	3	Left Motor Error is Active
I_MtrError_Right	BOOL	4:0089	3	Right Motor Error is Active
I_SensorPortPin3_Left	BOOL	4:0035	0	Left Sensor Port Pin 3 Energized
I_SensorPortPin3_Right	BOOL	4:0035	2	Right Sensor Port Pin 3 Energized
I_SensorPortPin4_Left	BOOL	4:0035	4	Left Sensor Port Pin 4 Energized
I_SensorPortPin4_Right	BOOL	4:0035	6	Right Sensor Port Pin 4 Energized
I_TrackingDNZn	DINT	4:0199 (MSW) 4:0200 (LSW)	-	Current Tracking Value for Arrival at Local Downstream Zone
I_TrackingUPZn	DINT	4:0119 (MSW) 4:0120 (LSW)	-	Current Tracking Value for Arrival at Local Upstream Zone
I_ZoneStatusDnZn	SINT	4:0196	Lo Byte	Zone Status Local Downstream Zone Forward Direction
I_ZoneStatusUpZn	SINT	4:0116	Lo Byte	Zone Status Local Uptream Zone Forward Direction

13.5.1.2. ZPA Mode Outputs

✻ These tags are for AOI file ERSC_ZPA_427_507.L5X

Tag Name	Data Type	Module Register	Bit	Description
O_AccForArrivalDNZn	BOOL	4:0184	0	Set Local Downstream Zone to Accumulate
O_AccForArrivalUPZn	BOOL	4:0104	0	Set Local Upstream Zone to Accumulate
O_AccumAdjUpstreamToDNZn	BOOL	4:0184	8	Accumulate Adjacent Upstream to Local Downstream Zone
O_AccumAdjUpstreamToUPZn	BOOL	4:0104	8	Accumulate Adjacent Upstream to Local Upstream Zone
O_ClearJamDNZn	BOOL	4:0189	0	Clear Jam at Local Upstream Zone
O_ClearJamUPZn	BOOL	4:0109	0	Clear Jam at Local Downstream Zone
O_ClearMotorError	BOOL	4:0022	0	Clear Motor Error Left & Right
O_ConfArrivalAdjDownstreamToDNZn	BOOL	4:0184	9	Confirm Downstream Arrival for Local Downstream Zone
O_ConfArrivalAdjDownstreamToUPZn	BOOL	4:0104	9	Confirm Downstream Arrival for Local Upstream Zone
O_ControlPortOutputLeft	BOOL	4:0063	1	Set Left Control Port Output
O_ControlPortOutputRight	BOOL	4:0063	3	Set Right COntrol Port Output
O_ConveyMerge_DisableCenter	BOOL	4:0387	4	Set to disable center release
O_ConveyMerge_DisableLeft	BOOL	4:0387	5	Set to disable left release
O_ConveyMerge_DisableRight	BOOL	4:0387	6	Set to disable right release
O_ConveyMerge_EnablePLCCtrl	BOOL	4:0387	15	Set to enable PLC over-ride of configured ConveyMerge
O_ConveyMergePriority	SINT	4:0387	-	Numerical value to set merge priority

O_ConveyStopCommand	INT	4:0020	-	Set Local ConveyStop Command Word
O_DAModeCmdDNZn	SINT	4:0375	Lo Byte	Direction & Accumulation Mode Command Byte for Downstream Zone
O_DAModeCmdUPZn	SINT	4:0365	Lo Byte	Direction & Accumulation Mode Command Byte for Upstream Zone
O_DAModeValueDNZn	SINT	4:0375	Hi Byte	Direction & Accumulation Mode Data Byte for Downstream Zone
O_DAModeValueUPZn	SINT	4:0365	Hi Byte	Direction & Accumulation Mode Data Byte for Upstream Zone
O_EnableModule	BOOL	-	AOI Logic	Enable Output to ERSC
O_JogFwdDNZn	BOOL	4:0184	10	Jog Forward for Local Downstream Zone
O_JogFwdUpZn	BOOL	4:0104	10	Jog Forward for Local Upstream Zone
O_JogRevDNZn	BOOL	4:0184	11	Jog Reverse for Local Downstream Zone
O_JogRevUPZn	BOOL	4:0104	11	Jog Reverse for Local Upstream Zone
O_ReleaseDNZn	BOOL	4:0105	AOI Logic	Release and Accumulate on Next at Downstream Zone – Automatically increments release counter
O_ReleaseUPZn	BOOL	4:0185	AOI Logic	Release and Accumulate on Next at Upstream Zone – Automatically increments release counter
O_SpeedLeftMtr	INT	4:0040	-	Set Left Motor Speed Reference
O_SpeedRightMtr	INT	4:0064	-	Set Right Motor Speed Reference
O_StatusDownstreamDischarge	INT	4:0232	-	Set Downstream Discharge Zone Status Value

O_StatusUpstreamInduct	INT	4:0134	-	Set Upstream Induct Zone Status Value
O_TrackingDNZn	DINT	4:0212 (MSW) 4:0213 (LSW)	-	Set Tracking Value for Local Downstream Zone
O_TrackingInductFwd	DINT	4:0139 (MSW) 4:0140 (LSW)	-	Set Forward Induct Tracking Value
O_TrackingUPZn	DINT	4:0132 (MSW) 4:0133 (LSW)	-	Set Tracking Value for Local Upstream Zone
O_WakeUpDNZn	BOOL	4:0184	12	Wakeup Local Downstream Zone
O_WakeUpUPZn	BOOL	4:0104	12	Wakeup Local Upstream Zone

13.5.1.3. PLC I/O Mode Inputs

✻ These tags are for AOI file ERSC_PLCIO_427_507.L5X

Tag Name	Data Type	Module Register	Bit	Description
I_ControlPortPin3_Left	BOOL	4:0019	1	Port Inputs
I_ControlPortPin3_Right	BOOL	4:0035	3	Port Inputs
I_ControlPortPin4_Left	BOOL	4:0035	5	Port Inputs
I_ControlPortPin4_Right	BOOL	4:0035	7	Port Inputs
I_ConveyStopByLeftControlPort	BOOL	4:0019	8	ConveyStop
I_ConveyStopByLostConnection	BOOL	4:0019	6	ConveyStop
I_ConveyStopByPLCDisconnect	BOOL	4:0019	7	ConveyStop
I_ConveyStopByPLCCmd	BOOL	4:0019	10	ConveyStop
I_ConveyStopByRemoteModule	BOOL	4:0019	5	ConveyStop
I_ConveyStopByRightControlPort	BOOL	4:0019	9	ConveyStop
I_DigitalMtrOverCurrent_Left	BOOL	4:0060	14	Left Motor Port as Digital
I_DigitalMtrOverCurrent_Right	BOOL	4:0084	14	Right Motor Port as Digital
I_DigitalMtrShortCkt_Left	BOOL	4:0060	12	Left Motor Port as Digital
I_DigitalMtrtShortCkt_Right	BOOL	4:0084	12	Right Motor Port as Digital
I_DownstreamModuleStatus	SINT	4:0232	Lo Byte	Module Status
I_Heartbeat	BOOL	4:0035	15	Port Inputs
I_ModuleEnabled	BOOL	-	-	Module Status
I_ModuleVoltage	REAL	4:0024	-	Module Status
I_MtrCurrent_Left	REAL	4:0055	-	Left Motor Status
I_MtrCurrent_Right	REAL	4:0079	-	Right Motor Status
I_MtrFreq_Left	INT	4:0056	-	Left Motor Status
I_MtrFreq_Right	INT	4:0080	-	Right Motor Status

I_MtrRunningCCW_Left	BOOL	4:0058	AOI Logic	Left Motor Status
I_MtrRunningCCW_Right	BOOL	4:0082	AOI Logic	Right Motor Status
I_MtrRunningCW_Left	BOOL	4:0058	AOI Logic	Left Motor Status
I_MtrRunningCW_Right	BOOL	4:0082	AOI Logic	Right Motor Status
I_MtrStatus_Left	INT	4:0058	-	Left Motor Status
I_MtrStatus_Right	INT	4:0082	-	Right Motor Status
I_SensorDetectLeftPort	BOOL	4:0036	1	Sensor Port Status
I_SensorDetectRightPort	BOOL	4:0036	0	Sensor Port Status
I_SensorPortPin3_Left	BOOL	4:0035	0	Port Inputs
I_SensorPortPin3_Right	BOOL	4:0035	2	Port Inputs
I_SensorPortPin4_Left	BOOL	4:0035	4	Port Inputs
I_SensorPortPin4_Right	BOOL	4:0035	6	Port Inputs
I_ServoCmdStatus_Left	BOOL	4:0011	2	Left Servo Function
I_ServoCmdStatus_Right	BOOL	4:0016	2	Right Servo Function
I_ServoLastCmdComplete_Left	BOOL	4:0011	0	Left Servo Function
I_ServoLastCmdComplete_Right	BOOL	4:0016	0	Right Servo Function
I_ServoPosition_Left	INT	4:0062	-	Left Servo Function
I_ServoPosition_Right	INT	4:0086	-	Right Servo Function
I_ServoResetStatus_Left	BOOL	4:0011	1	Left Servo Function
I_ServoResetStatus_Right	BOOL	4:0016	1	Right Servo Function
I_TemperatureCalculated_Left	SINT	4:0057	Hi Byte	Left Motor Status
I_TemperatureCalculated_Right	SINT	4:0081	Hi Byte	Right Motor Status
I_TemperatureOnBoard_Left	SINT	4:0057	Lo Byte	Left Motor Status
I_TemperatureOnBoard_Right	SINT	4:0081	Lo Byte	Right Motor Status
I_UpstreamModuleStatus	SINT	4:0134	Lo Byte	ZPA Status

I_UpstreamTracking	DINT	4:0139 (MSW) 4:0140 (LSW)	-	ZPA Tracking

13.5.1.4. PLC I/O Mode Outputs

✻ These tags are for AOI file ERSC_PLCIO_427_507.L5X

Tag Name	Data Type	Module Register	Bit	Description
O_BrakeMethod_Left	SINT	4:0261	Lo Byte	Left Motor Control
O_BrakeMethod_Right	SINT	4:0271	Lo Byte	Right Motor Control
O_ClearMotorError	BOOL	4:0022		Motor Control
O_ControlPortOutput_Left	BOOL	4:0037	1	Left Motor Control
O_ControlPortOutput_Right	BOOL	4:0037	3	Right Motor Control
O_ControlPortPin3Mask_Left	BOOL	4:0034	1	Sensor/Control Port Configuration
O_ControlPortPin3Mask_Right	BOOL	4:0034	3	Sensor/Control Port Configuration
O_ControlPortPin4Mask_Left	BOOL	4:0034	5	Sensor/Control Port Configuration
O_ControlPortPin4Mask_Right	BOOL	4:0034	7	Sensor/Control Port Configuration
O_ConveyStopCommand	INT	4:0020		ConveyStop
O_DischargeTracking	DINT	4:0201 (MSW) 4:0202 (LSW)		ZPA Tracking
O_DownstreamStatus	SINT	4:0196	Lo Byte	ZPA Status
O_EnableModule	BOOL	-	-	Module Control
O_LeftMtrDigitalPin3	BOOL	4:0060	0	Left Motor Port Digital Control
O_LeftMtrDigitalPin4	BOOL	4:0060	1	Left Motor Port Digital Control
O_LeftMtrDigitalPin5	BOOL	4:0060	2	Left Motor Port Digital Control

O_MtrAccel_Left	INT	4:0043		Left Motor Control
O_MtrAccel_Right	INT	4:0067		Right Motor Control
O_MtrDecel_Left	INT	4:0044		Left Motor Control
O_MtrDecel_Right	INT	4:0068		Right Motor Control
O_RightMtrDigitalPin3	BOOL	4:0084	0	Right Motor Port Digital Control
O_RightMtrDigitalPin4	BOOL	4:0084	1	Right Motor Port Digital Control
O_RightMtrDigitalPin5	BOOL	4:0084	2	Right Motor Port Digital Control
O_Mtr_Run_Left	BOOL	4:0260	0	Left Motor Control
O_Mtr_Run_Right	BOOL	4:0270	0	Right Motor Control
O_Mtr_Dir_Left	BOOL	4:0260	8	Left Motor Control
O_Mtr_Dir_Right	BOOL	4:0270	8	Right Motor Control
O_Mtr_Slave_Mode_Left	SINT	4:0260	Hi Byte	Left Motor Control
O_Mtr_Slave_Mode_Right	SINT	4:0260	Hi Byte	Right Motor Control
O_SensorPortPin3Mask_Left	BOOL	4:0034	0	Sensor/Control Port Configuration
O_SensorPortPin3Mask_Right	BOOL	4:0034	2	Sensor/Control Port Configuration
O_SensorPortPin4Mask_Left	BOOL	4:0034	4	Sensor/Control Port Configuration
O_SensorPortPin4Mask_Right	BOOL	4:0034	6	Sensor/Control Port Configuration
O_ServoCmdPulses_Left	INT	4:0008		Left Servo Function
O_ServoCmdPulses_Right	INT	4:0013		Right Servo Function
O_ServoGoCmd_Left	BOOL	4:0009	1	Left Servo Function
O_ServoGoCmd_Right	BOOL	4:0014	1	Right Servo Function
O_ServoZero_Left	BOOL	4:0009	0	Left Servo Function

O_ServoZero_Right	BOOL	4:0014	0	Right Servo Function
O_LeftMtrDigital_Enable	BOOL	4:0060	15	Left Motor Port Digital Control
O_RightMtrDigital_Enable	BOOL	4:0084	15	Right Motor Port Digital Control
O_SpeedMethod_Left	SINT	4:0262	Lo Byte	Left Motor Control
O_SpeedMethod_Right	SINT	4:0272	Lo Byte	Right Motor Control
O_SpeedReference_Left	INT	4:0040		Left Motor Control
O_SpeedReference_Right	INT	4:0064		Right Motor Control
O_UpstreamStatus	SINT	4:0116	Lo Byte	ZPA Status
O_LeftMtrDigital_BrakePinEnable_NoD	BOOL	4:0060	7	Enable the Left Motor Port Brake pin digital output
O_LeftMtrDigital_BrakePin	BOOL	4:0060	6	Left Motor Brake Pin Digital Control
O_LeftMtrDigital_ClearOC	BOOL	4:0060	8	Left Motor Digital Clear Over-current Error
O_RightMtrDigital_BrakePinEnable_NoD	BOOL	4:0084	7	Enable the Right Motor Port Brake pin digital output
O_RightMtrDigital_BrakePin	BOOL	4:0084	6	Right Motor Brake Pin Digital Control
O_RightMtrDigital_ClearOC	BOOL	4:0084	8	Right Motor Digital Clear Over-current Error

13.6. Using Logix 5000 MSG Instruction

Access to ConveyLinx ERSC modules 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 not be used for dedicated "real time" control of equipment**. For ConveyLinx ERSC modules, *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 ERSC modules.

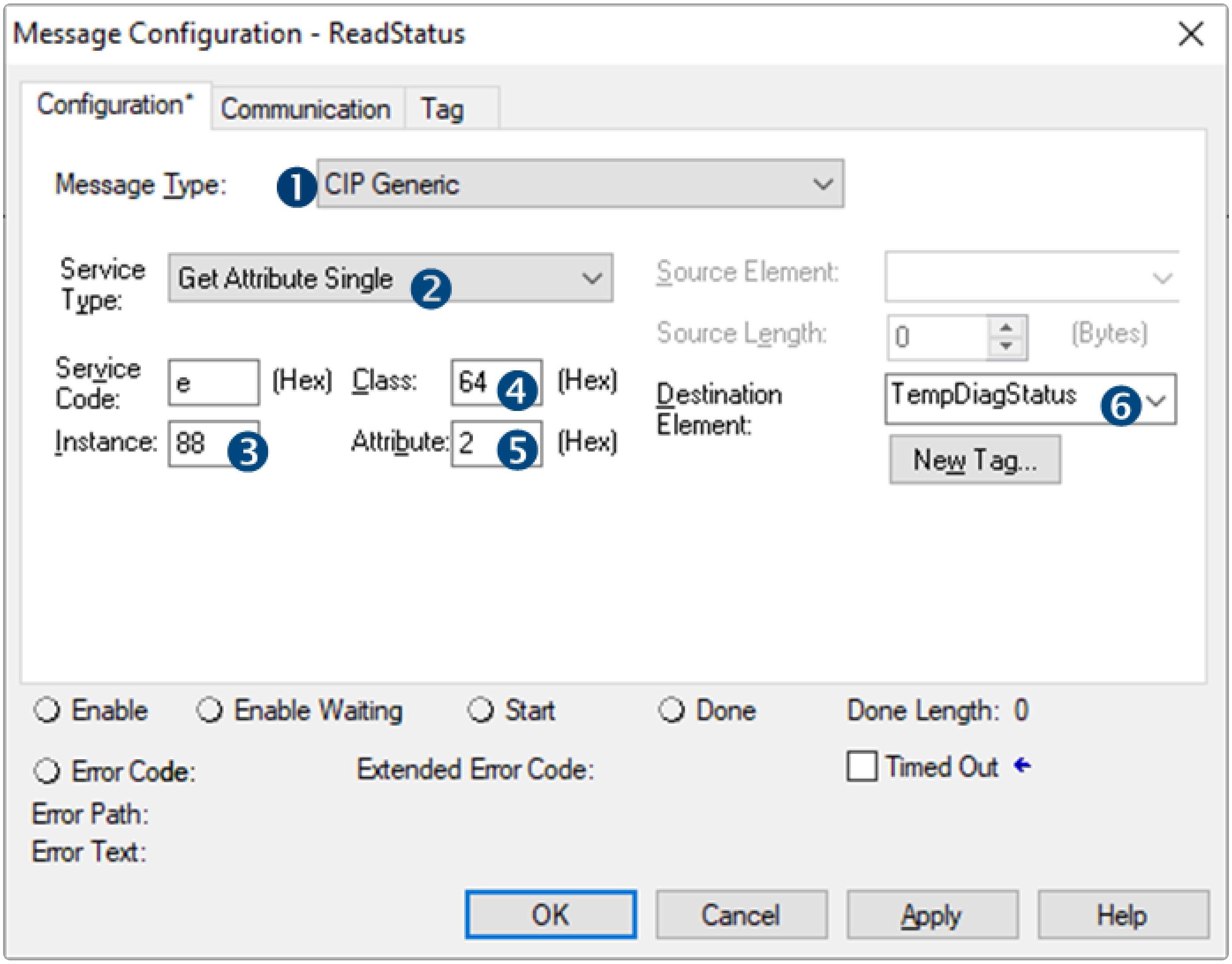
Module VS. Assembly Addresses with MSG Instruction

Module Addresses are the 512 "actual" registers in the module and 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 Module registers
- You can use a single MSG instruction to **write** to one (an only one) of the Module registers
- You can use a single MSG instruction to **read** any of the available *Input Assembly* registers in their entirety
- You **CANNOT** use any MSG instruction to **write** to any *Output Assembly* registers.

13.6.1. Read MSG Instruction

For this example, we have created an ERSC module 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.

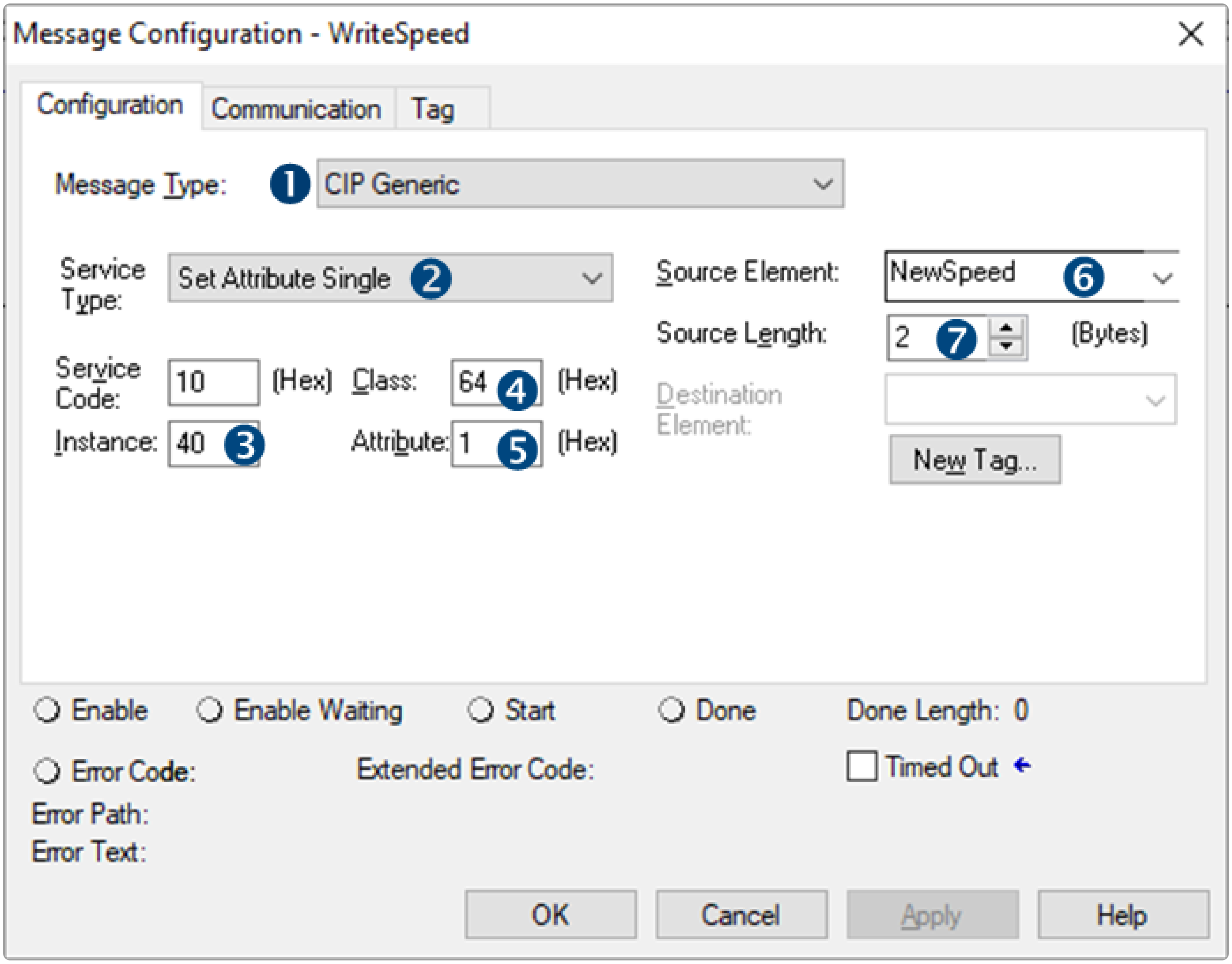


1	From the Message Type drop down, select *CIP Generic*
2	From the Service Type drop down, select *Get Attribute Single*
3	For the Instance Field, enter “88”

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 an ERSC module 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.



1	From the *Message Type* drop down, select *CIP Generic*
2	From the *Service Type* drop down, select *Set Attribute Single*
3	For the *Instance* Field, enter “40”

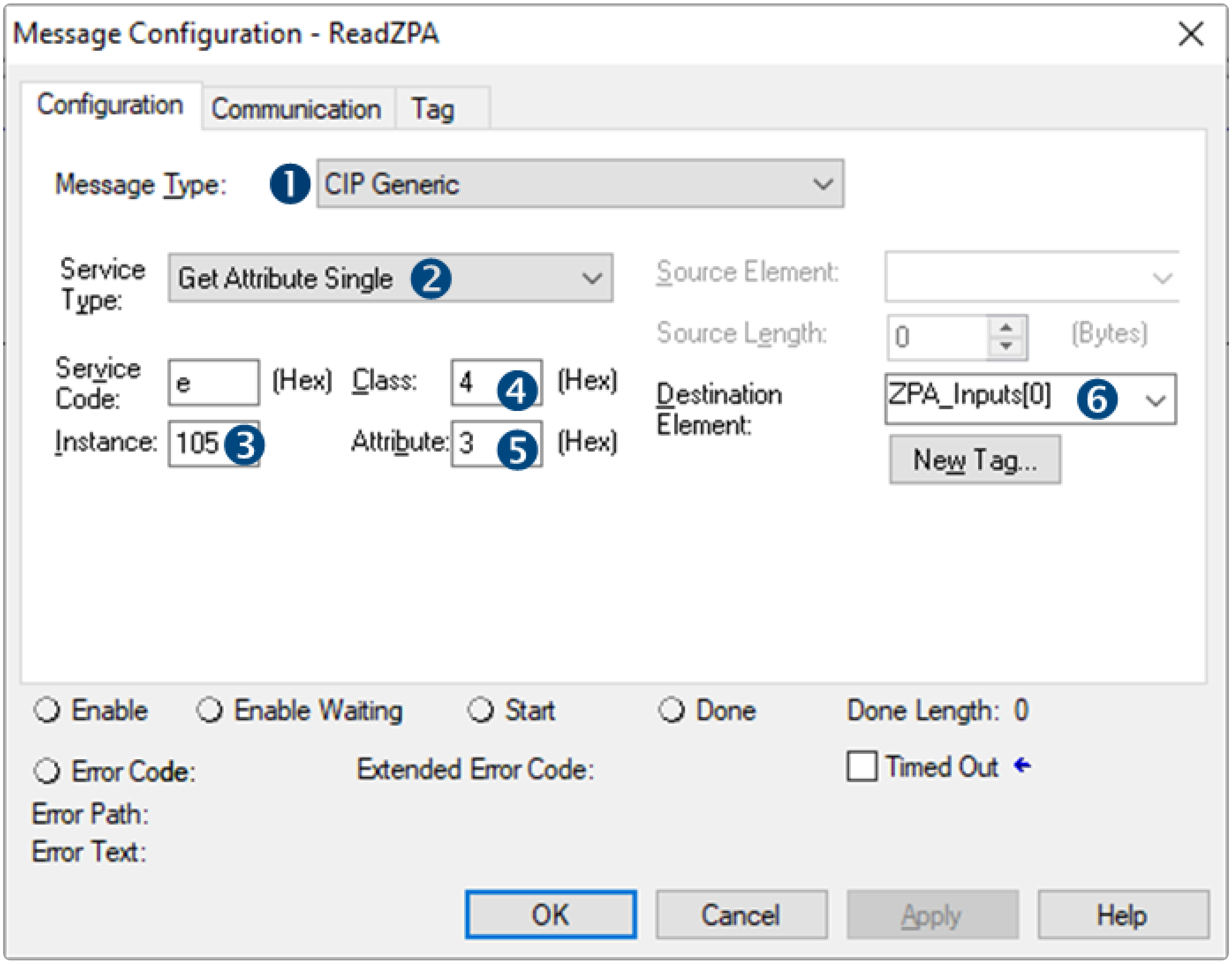
4	For the *Class* Field, enter “64”
5	For the *Attribute* Field, enter “1”
6	For the *Source Element*, use the Tag Browser drop down to select *NewSpeed*
7	For the *Source Length*, enter “2” (2 bytes = 1 INT)

13.6.3. Reading ERSC Input Assembly with MSG Instruction

! This functionality is only available in firmware versions 4.25 and higher

For this example, we have created an ERSC module 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	Firmware	Module Data Type
ConveyLinx_ZPA_Instance_1.eds	ERSC_ZPA_424.L5X	4.24 and older	_055C:ERSC_ZPA_84EED4D3:I
			_055C:ERSC_ZPA_FB496954:O
ConveyLinx_PLC_IO_Instance_1.eds	ERSC_PLCIO_424.L5X	4.24 and older	_055C:ERSC_PLCIO_C92675D8:
			_055C:ERSC_PLCIO_7DDD1BFA:
ConveyLinx_V5_4.eds	ERSC_ZPA_5_2.L5X	4.25 / 5.02 and newer	_055C:ERSC_84EED4D3:I:0
			_055C:ERSC_3015BAF1:O:0
	ERSC_PLCIO_5_2.L5X	4.25 / 5.02 and newer	_055C:ERSC_C92675D8:I:0
			_055C:ERSC_7DDD1BFA:O:0
ConveyLinx_V5_6.eds	ERSC_ZPA_425_5xx.L5X	4.25 / 5.02 and newer	_055C:ERSC_84EED4D3:I:0
			_055C:ERSC_3015BAF1:O:0
	ERSC_PLCIO_425_5xx.L5X	4.25 / 5.02 and newer	_055C:ERSC_C92675D8:I:0
			_055C:ERSC_7DDD1BFA:O:0
ConveyLinx_ZPA_Instance.eds	ERSC_ZPA_424.L5X	4.24 and older	_055C:ERSC_ZPA_84EED4D3:I
			_055C:ERSC_ZPA_FB496954:O
ConveyLinx_PLC_IO_Instance.eds	ERSC_PLCIO_424.L5X	4.24 and older	_055C:ERSC_PLCIO_C92675D8:

			_055C:ERSC_PLCIO_7DDD1BFA:

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 Device Access Point (DAP). For a given *ConveyLinx* module type or family, your PLC programming environment needs a GSDML file 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, 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.

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 cannot be modified
* Your PLC must use the exact syntax 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

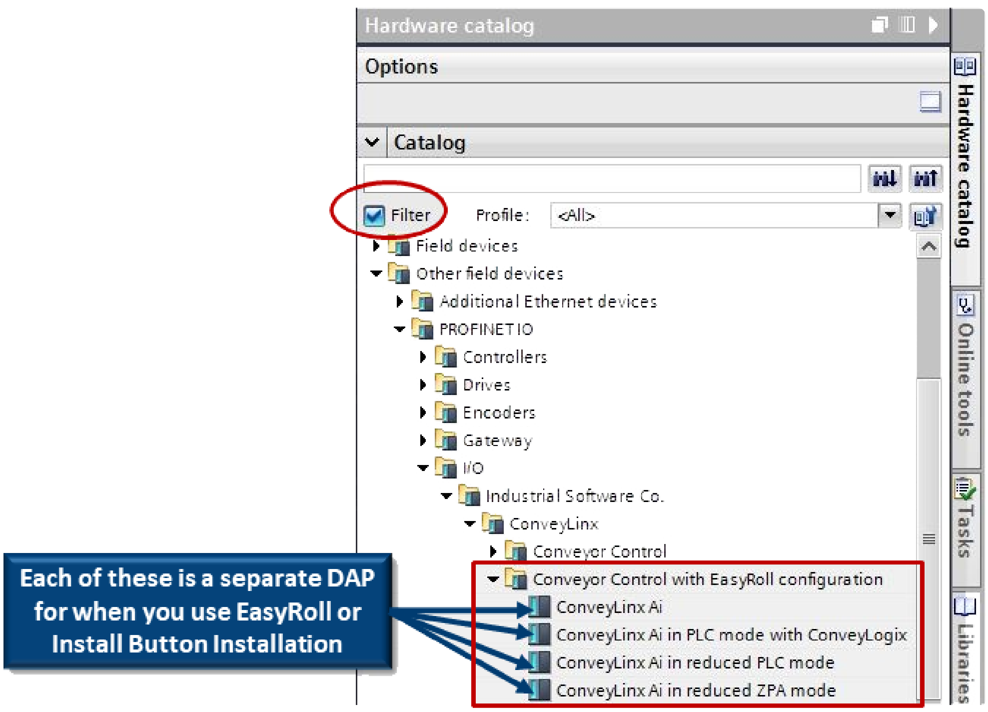
14.4. Separate ConveyLinx Auto-Configuration

! 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 exact syntax 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:

<table>
<tr><th>Mode</th><th>Syntax</th><th>Remarks</th></tr>
<tr><td>Full ZPA</td><td rowspan="2">conveylinxzpa-xxx-yyy</td><td rowspan="4">xxx = 3rd Octet of I.P. Address

yyy = 4th Octet of I.P. Address</td></tr>
<tr><td>Reduced ZPA</td></tr>
<tr><td>Full PLC Controlled</td><td rowspan="2">conveylinxplc-xxx-yyy</td></tr>
<tr><td>Reduced PLC Controlled</td></tr>
</table>

With ConveyLinx Auto-Configured modules, you cannot change the Profinet Name in the module and you MUST use the EXACT Profinet Name 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.

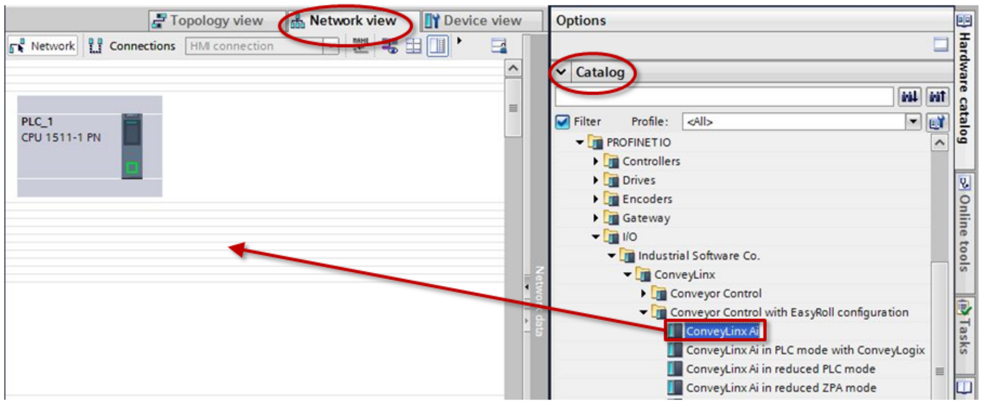
Module Mode	I.P. Address	DAP from Hardware Catalog	Profinet Name
Full ZPA	192.168.0.20	ConveyLinx-Ai	conveylinxzpa-0-20
Full PLC Controlled	192.168.101.21	ConveyLinx-Ai	*conveylinxplc-101-21*
Reduced ZPA	192.168.101.22	ConveyLinx-Ai in reduced ZPA mode	*conveylinxzpa-101-22*
Reduced PLC Controlled	192.168.20.123	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

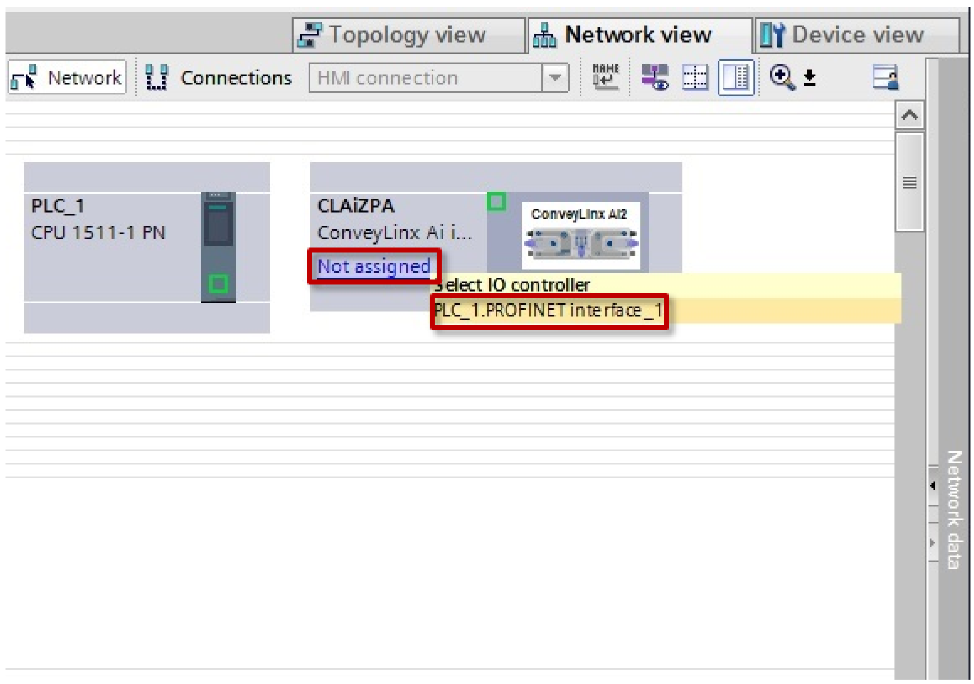
14.4.2.1. Full ZPA

STEP 1: Add the Module to the PLC Network

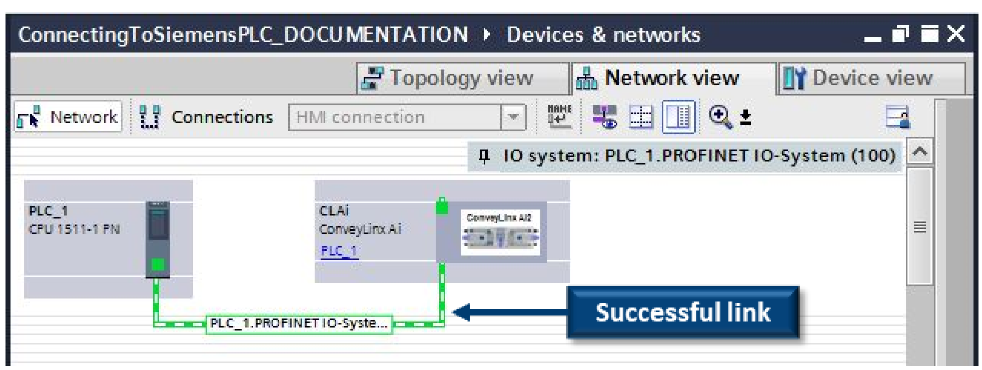
From the *Catalog* window, drag and drop the *ConveyLinx-Ai* DAP into the *Network View* window



To attach the module to the PLC, click on the Not Assigned link and select the desired Profinet Interface of the PLC

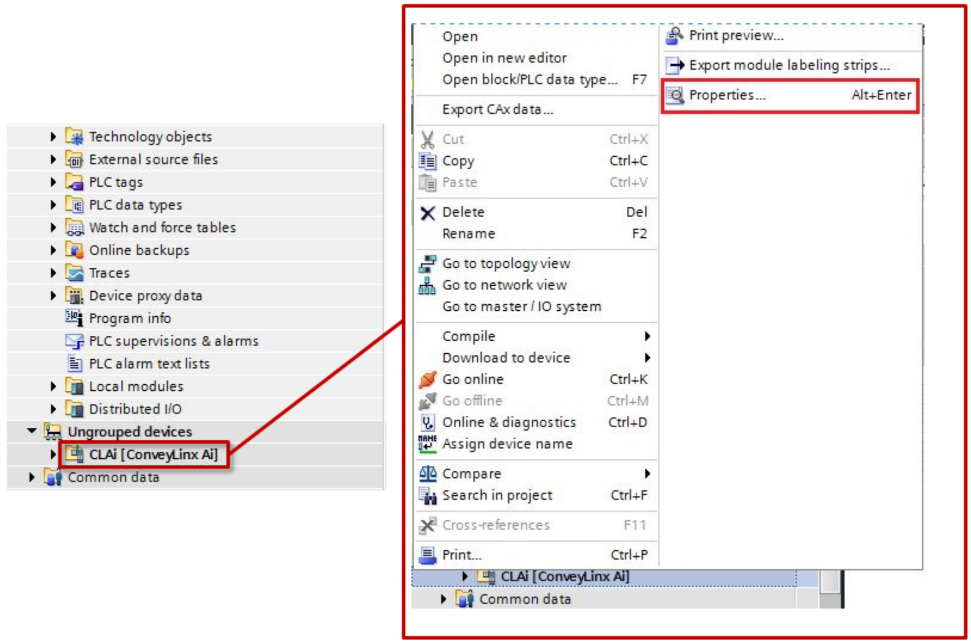


When attached, you will see the green graphic linking the PLC and the module together

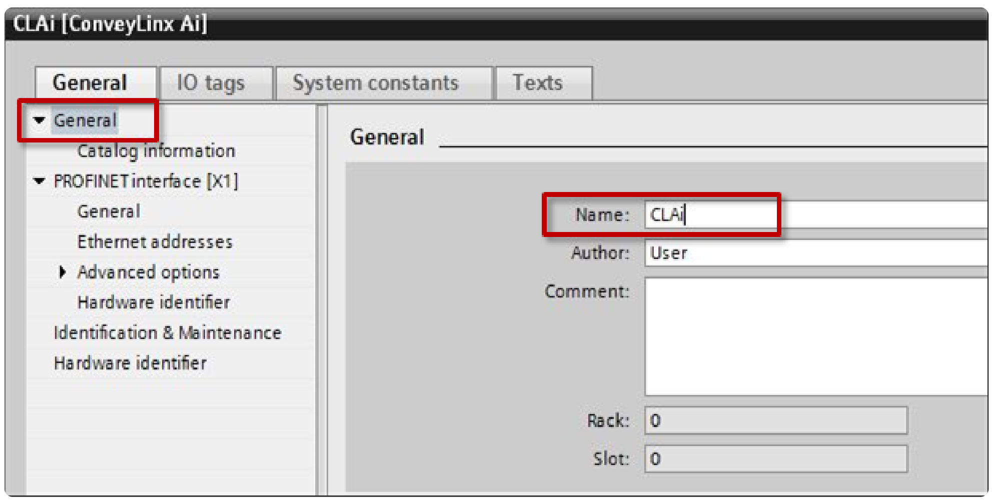


STEP 2: Assign the EXACT Profinet Name to the Module

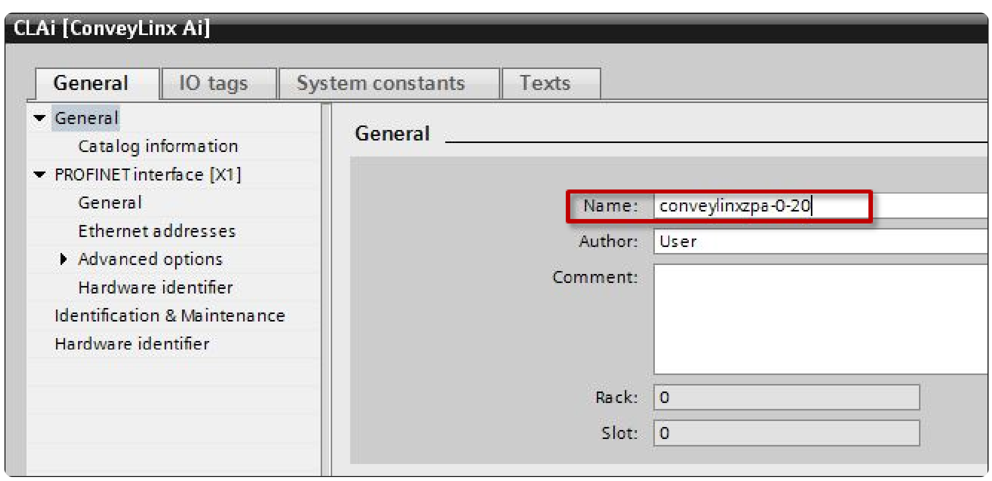
In the *Project Tree* locate your newly created module under the *Ungrouped devices* folder. Right Click on your module to show a pop-up menu and select *Properties*



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](#)

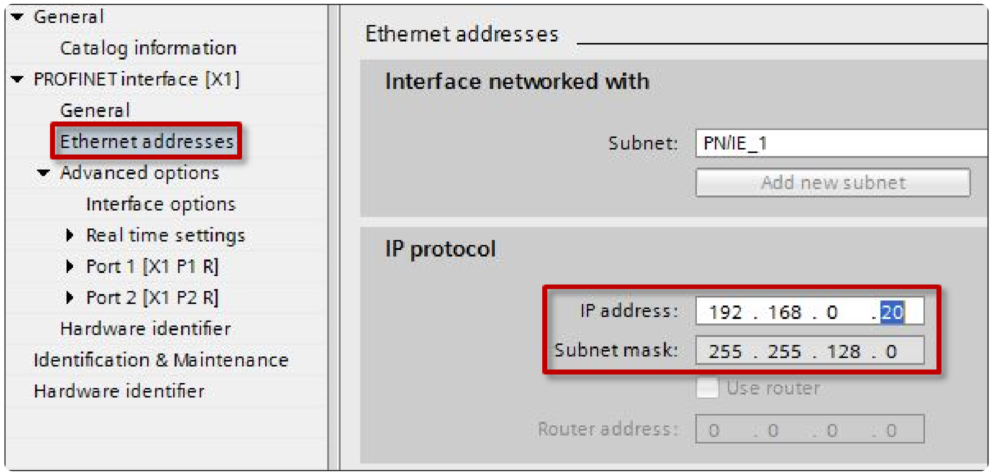


Type in the EXACT *Profinet Name* and press enter to make the change



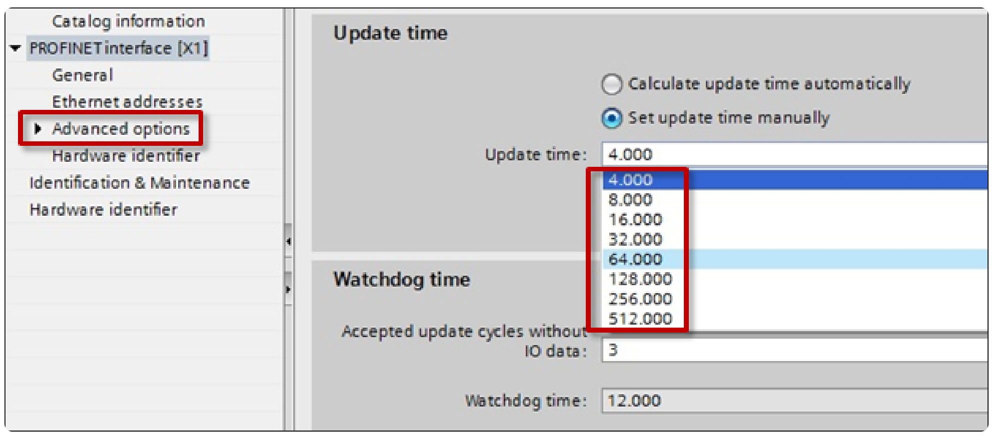
STEP 3: Enter the I.P. Address

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

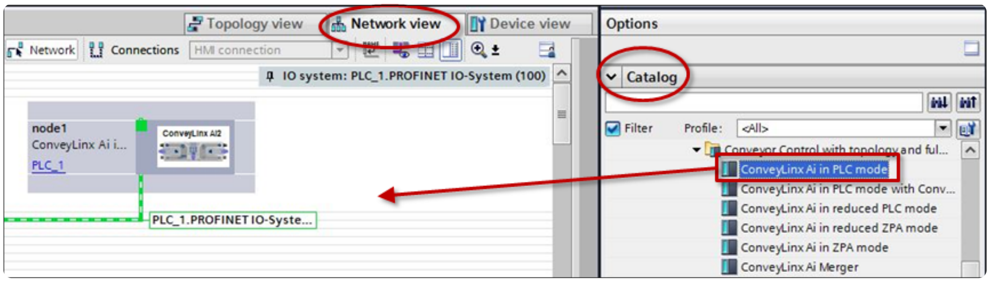
In the same properties area that you changed the *Profinet Name*, click on *Advanced options* and select a proper Update Time. For a ConveyLinx module in ZPA mode, the recommended Update time should be between 32ms and 512ms.



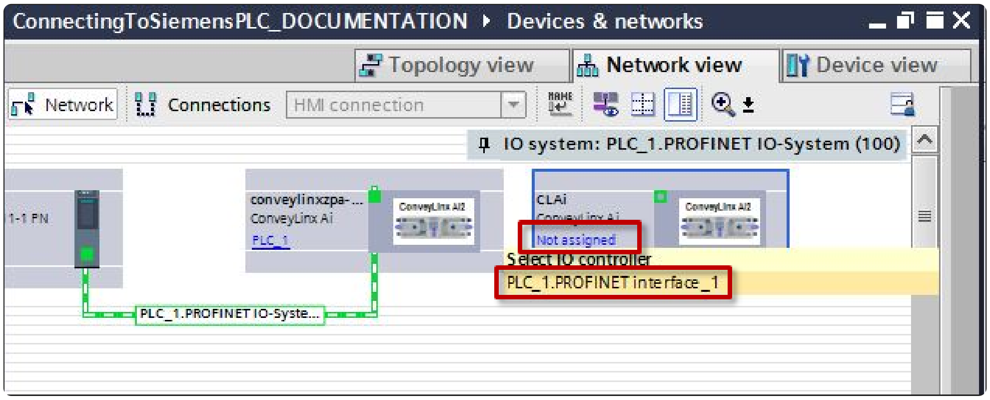
14.4.2.2. Full PLC Controlled

STEP 1: Add the Module to the PLC Network

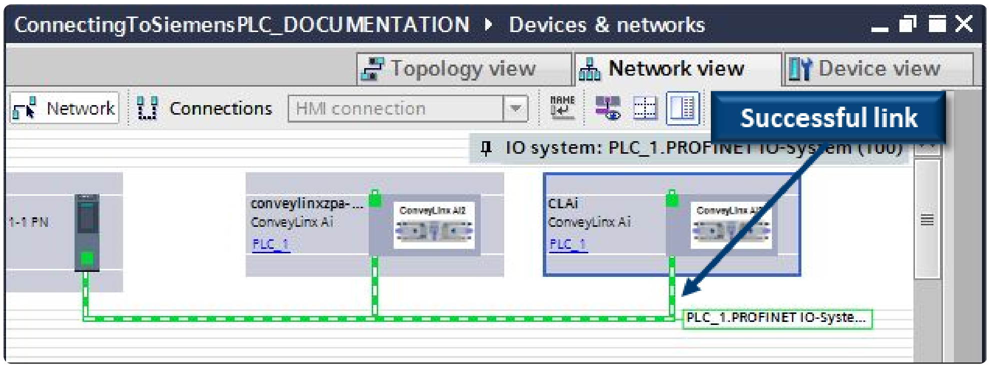
From the *Catalog* window, drag and drop the *ConveyLinx-Ai* DAP into the *Network View* window



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*

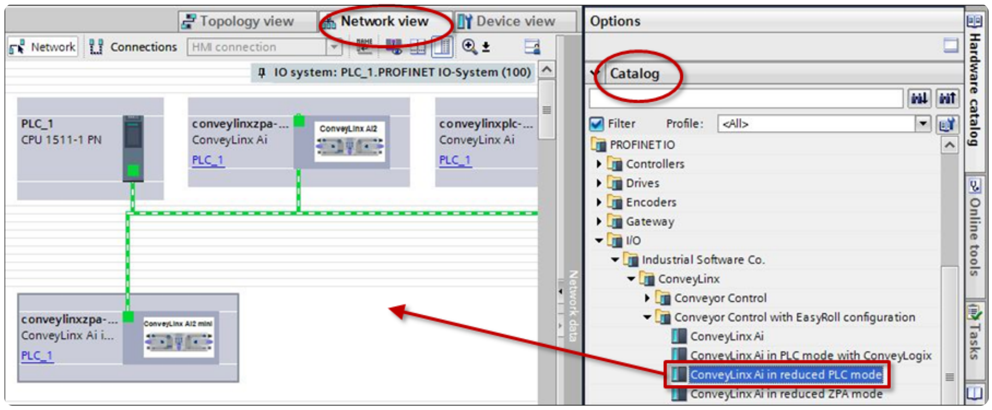
STEP 4: Select Update Time Interval

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.

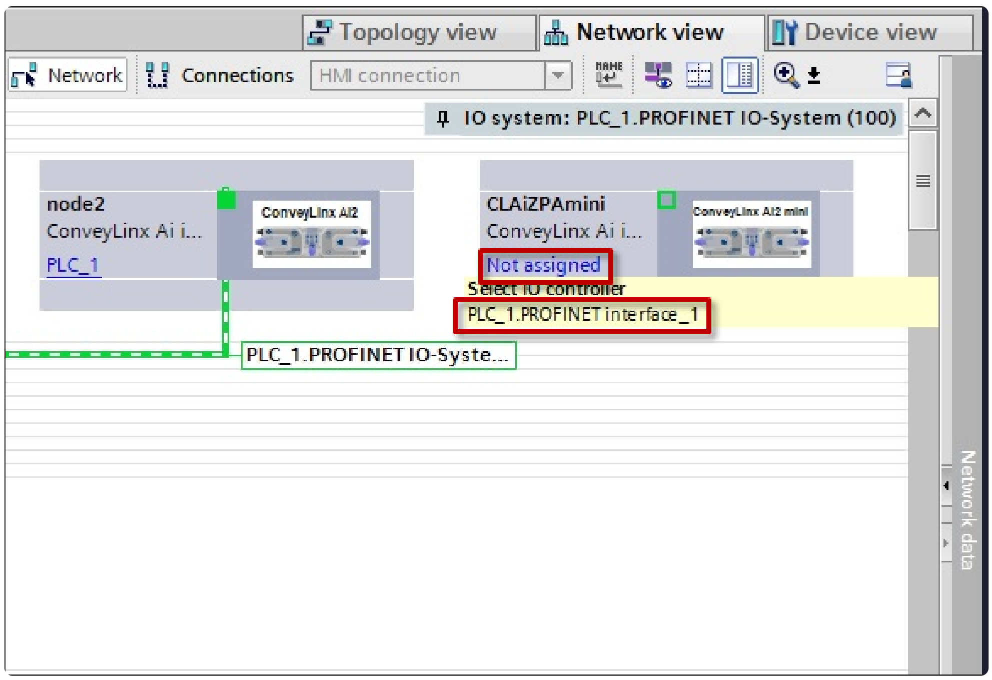
14.4.2.3. Reduced ZPA

STEP 1: Add the Module to the PLC Network

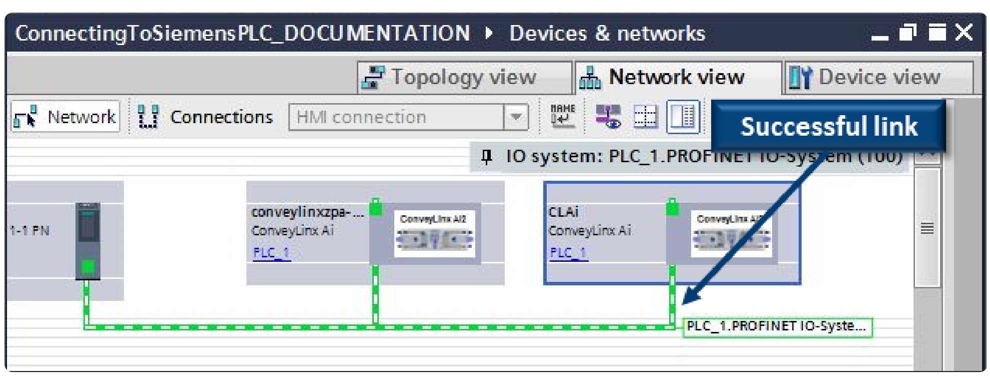
From the *Catalog* window, drag and drop the *ConveyLinx-Ai in reduced ZPA mode* DAP into the *Network View* window



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*

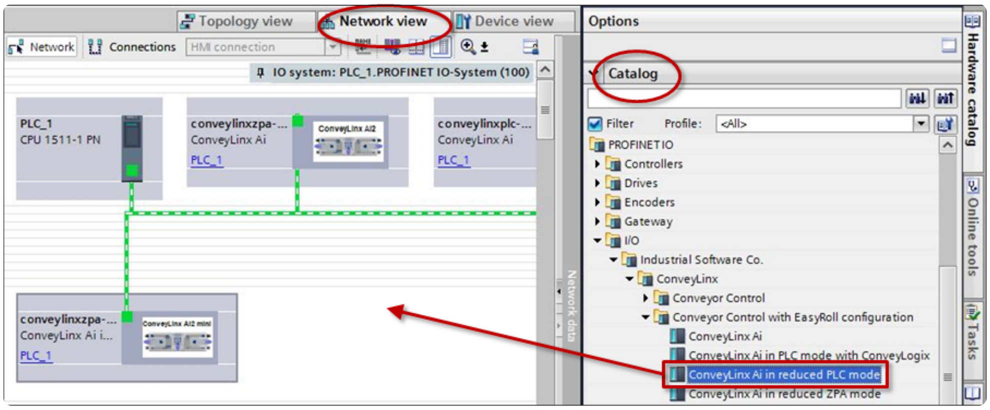
STEP 4: Select Update Time Interval

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.

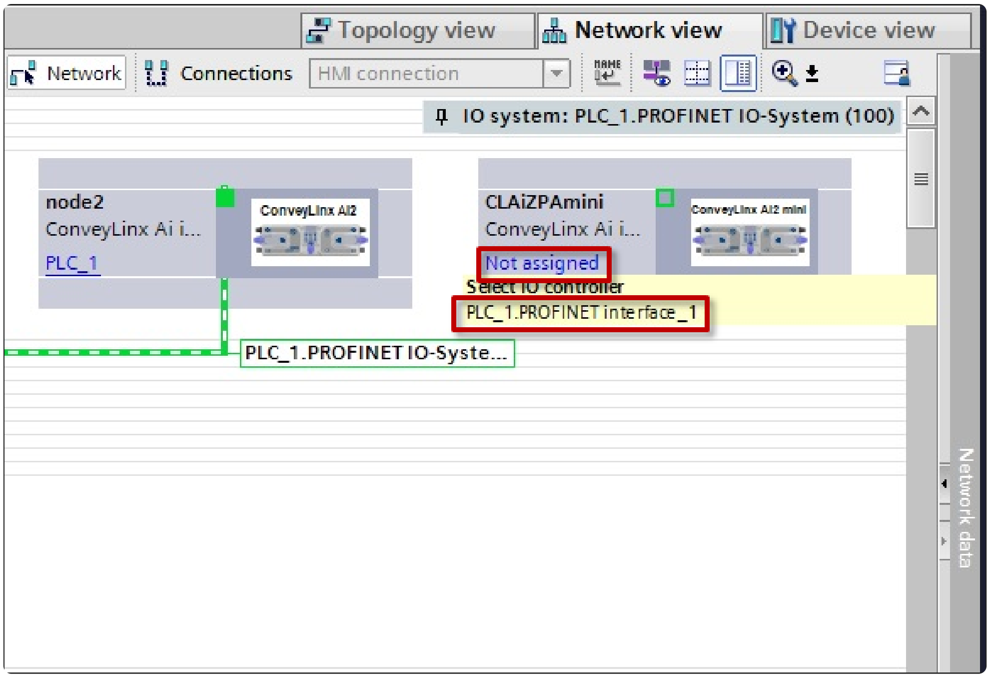
14.4.2.4. Reduced PLC

STEP 1: Add the Module to the PLC Network

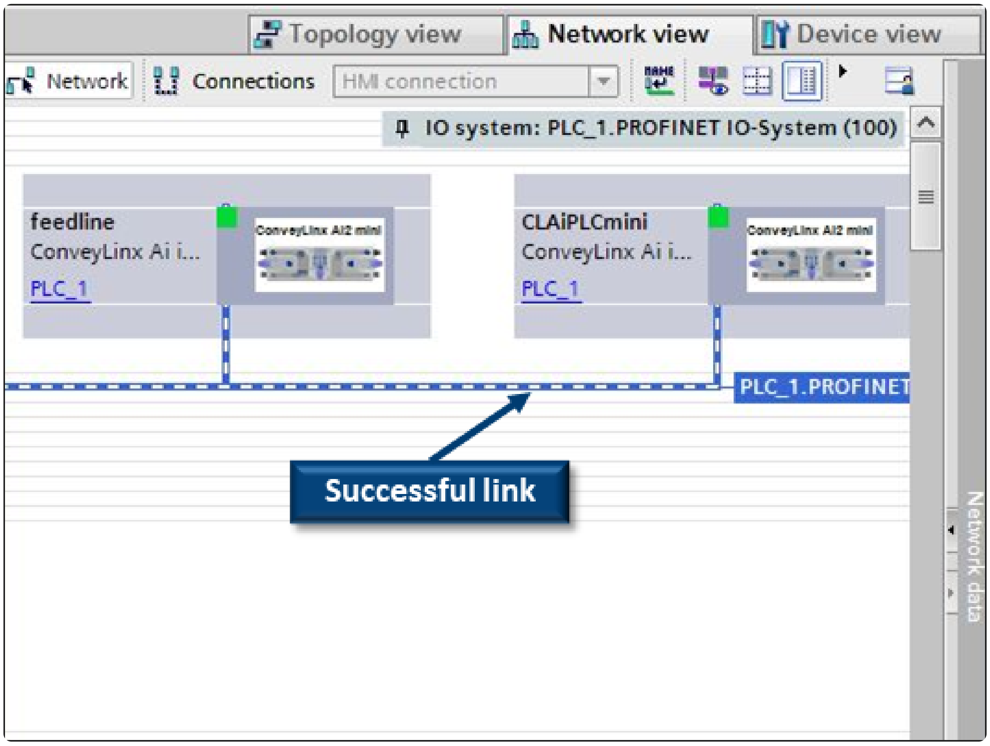
From the *Catalog* window, drag and drop the *ConveyLinx-Ai in reduced ZPA mode* DAP into the *Network View* window



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.

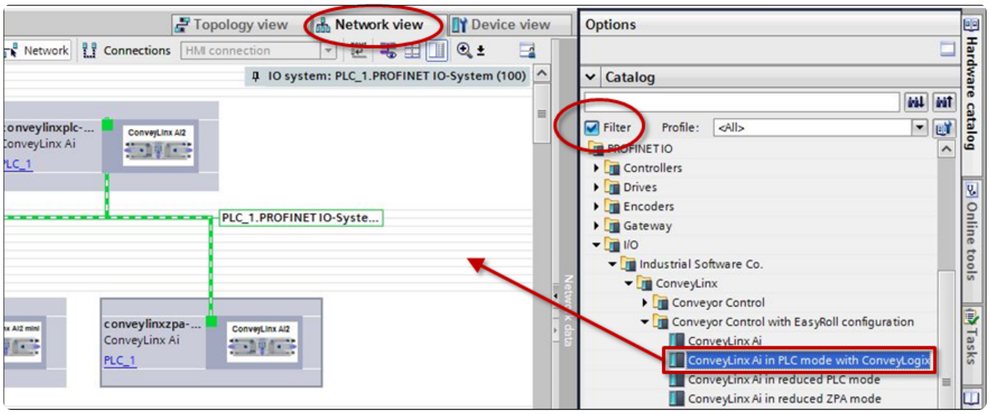
14.4.2.5. Merger Mode

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*

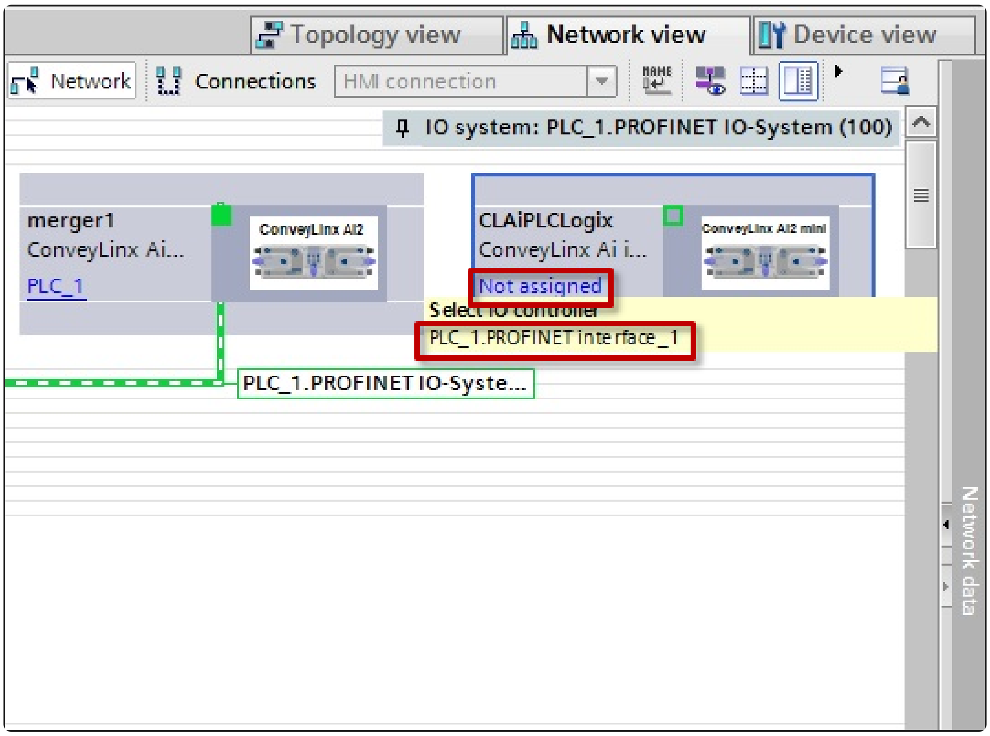
14.4.2.6. PLC Controlled with ConveyLogix Interface

STEP 1: Add the Module to the PLC Network

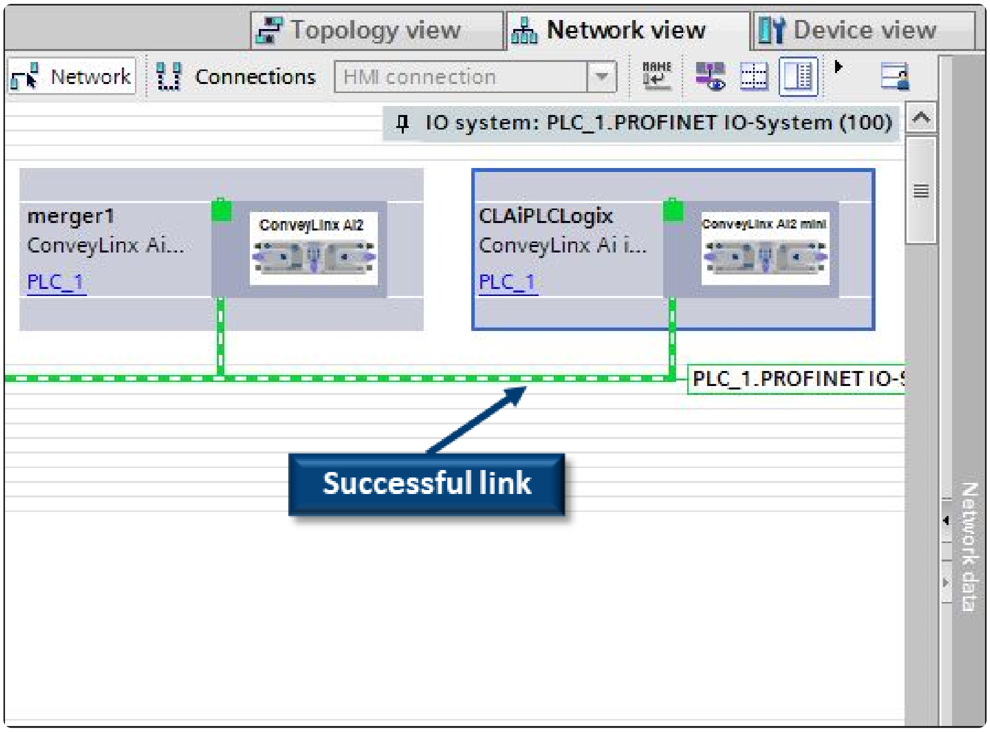
From the *Catalog* window, drag and drop the *ConveyLinx-Ai in PLC mode with ConveyLogix* DAP into the *Network View* window



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

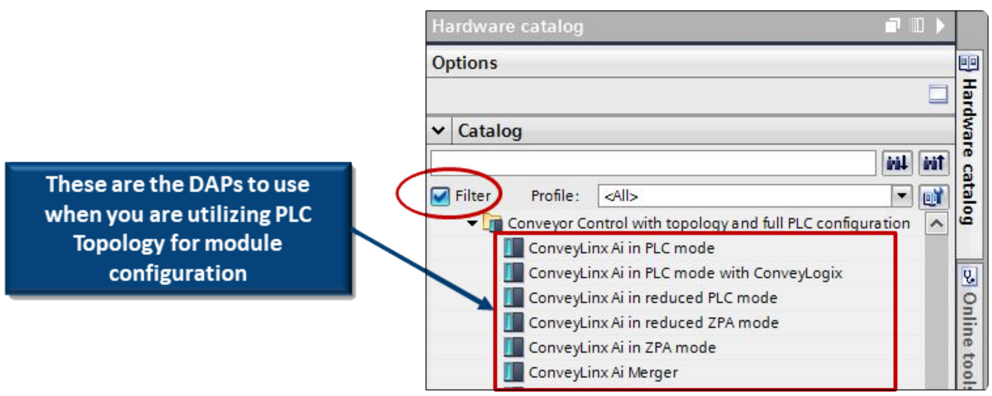
> ! This section applies ONLY when you use TIA Portal or STEP 7 Manager Topology to configure your ConveyLinx modules

> ! 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



> ✻ 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

14.5.1. 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.

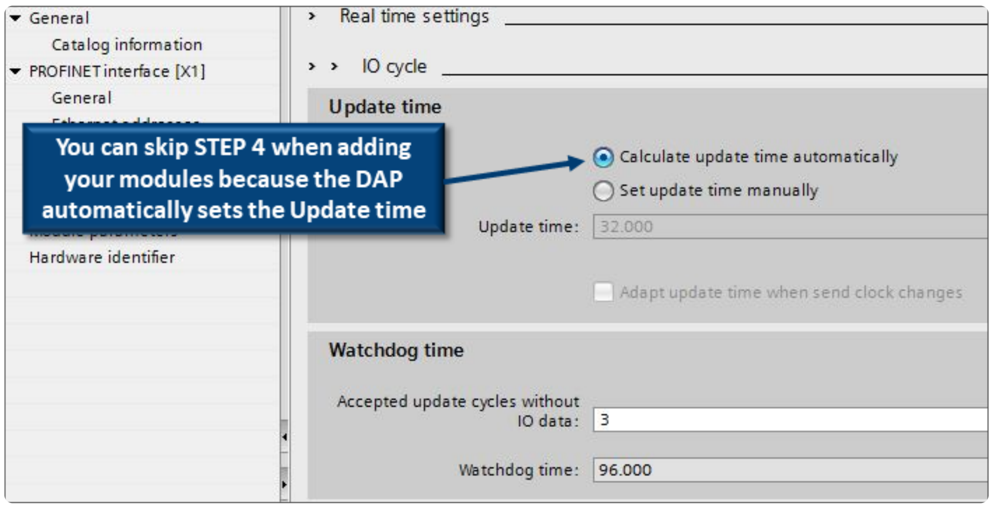
14.5.2. Adding Modules

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

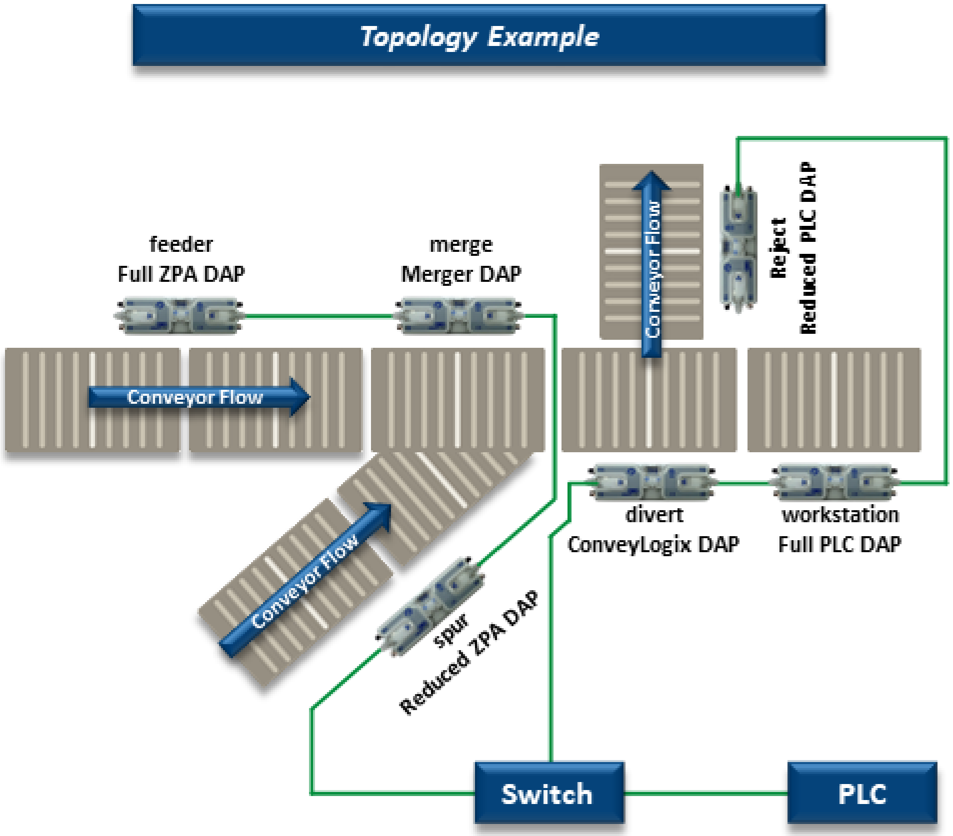
The DAPs for Integrated PLC Topology Configuration are already pre-configured with the lowest recommended Update time for the given functionality. This means that you can skip STEP 4 when adding your modules.

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

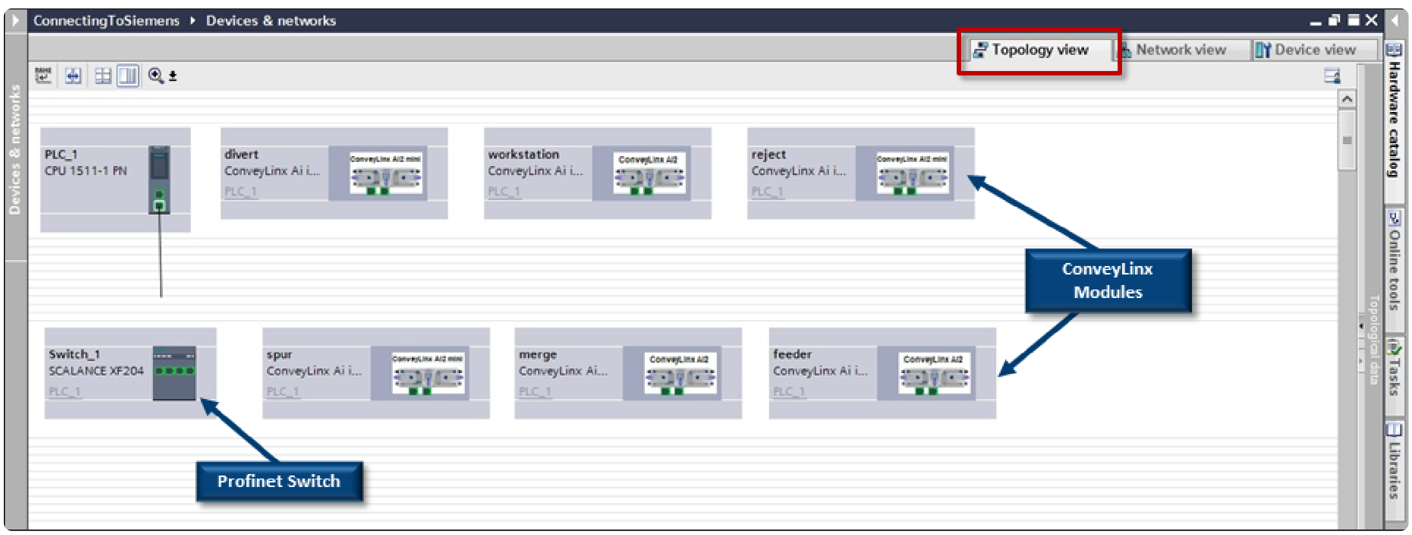
An example of a conveyor layout is a great way to illustrate the Integrated PLC Topology Configuration method.



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*

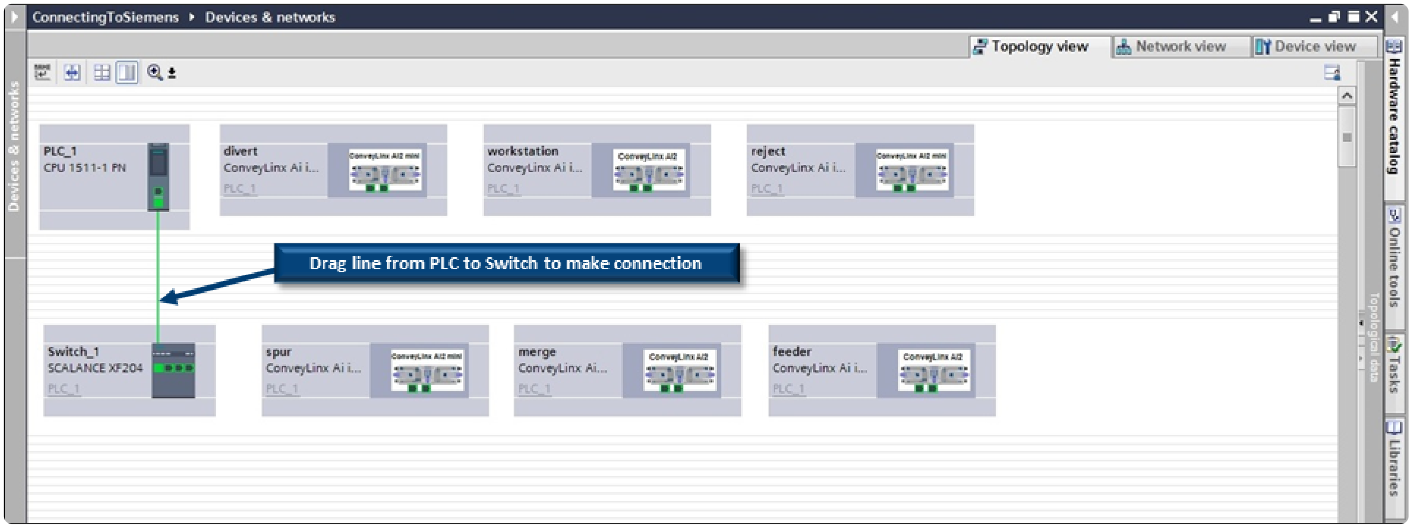
		mode	
Reduced ZPA	192.168.10.20	ConveyLinx-Ai in reduced ZPA mode	*spur*
Reduced PLC Controlled	192.168.15.20	ConveyLinx-Ai in reduced PLC mode	*reject*

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.



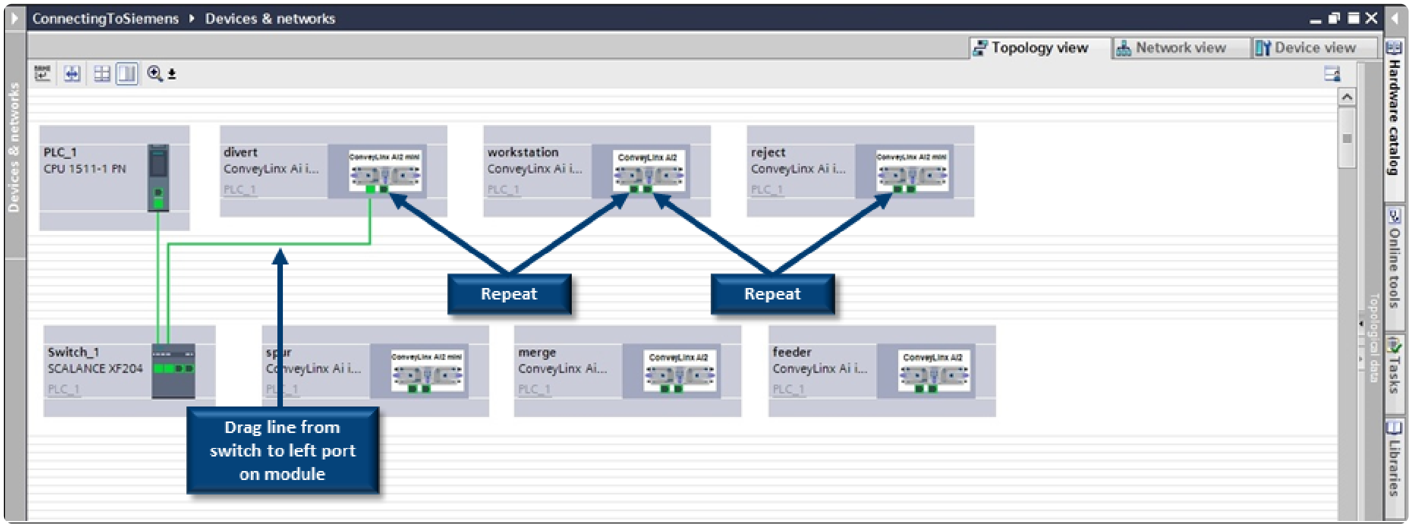
Connecting PLC and Switch

To make a topology connection, click and drag from the small connection square on the PLC to one of the ports on the switch.

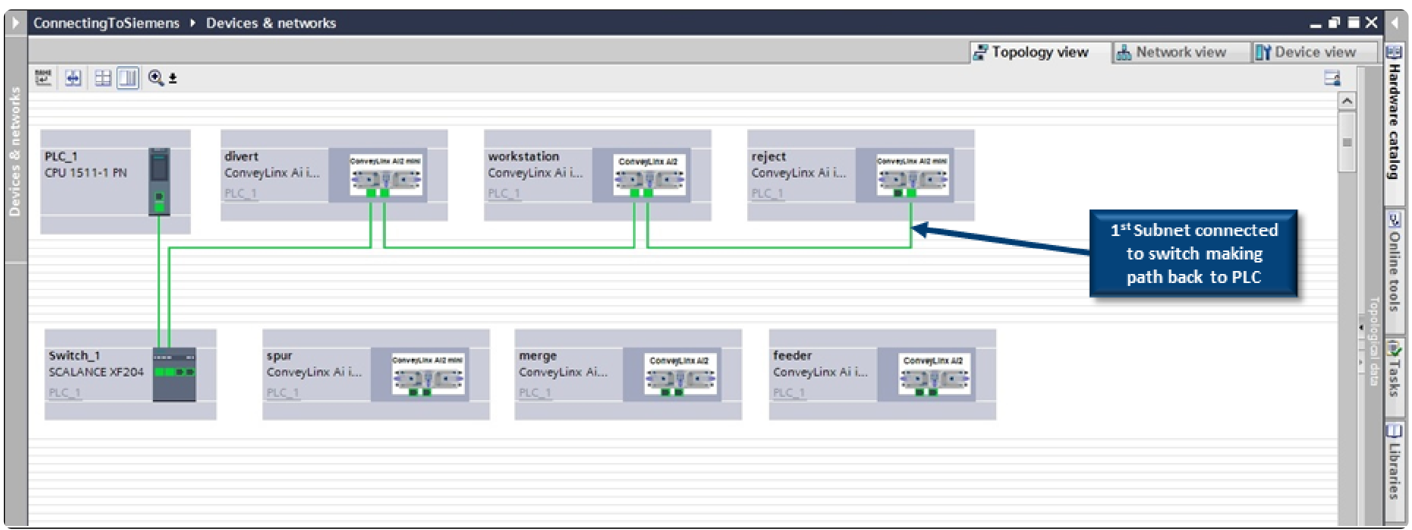

ConnectingToSiemens ▸ Devices & networks
Topology view
Network view
Device view
PLC_1
CPU 1511-1 PN
divert
ConveyLinx Ai i...
PLC_1
workstation
ConveyLinx Ai i...
PLC_1
reject
ConveyLinx Ai i...
PLC_1
Drag line from PLC to Switch to make connection
Switch_1
SCALANCE XF204
PLC_1
spur
ConveyLinx Ai i...
PLC_1
merge
ConveyLinx Ai...
PLC_1
feeder
ConveyLinx Ai i...
PLC_1
Devices & networks
Hardware catalog
Online tools
Tasks
Libraries

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.



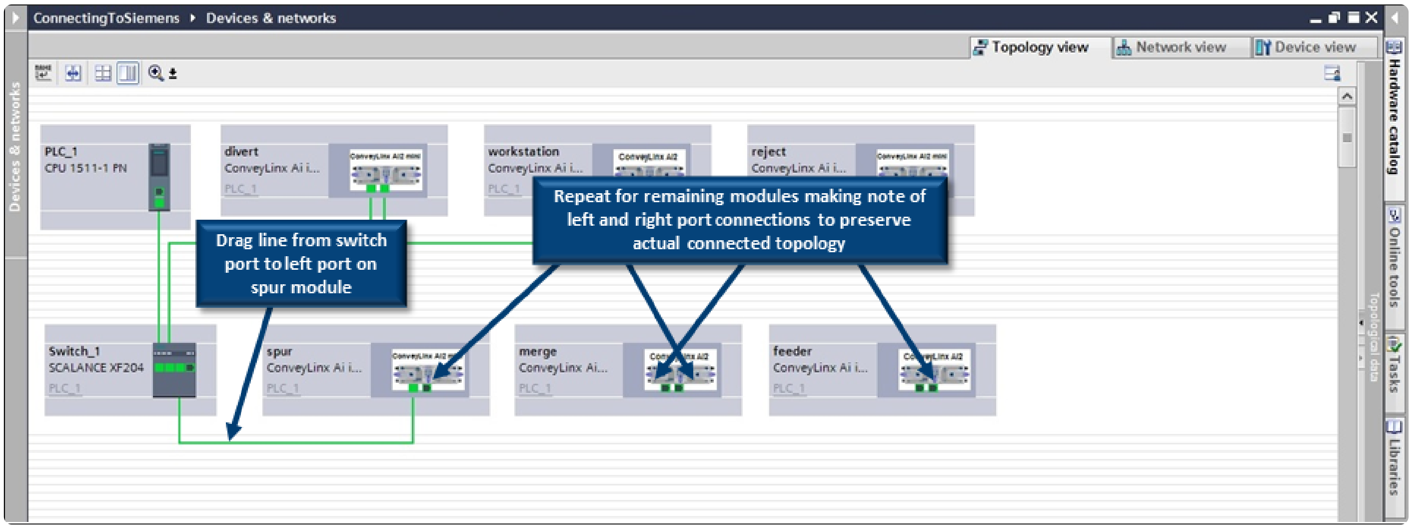
When you are done with the 1st subnet of modules, it should look like the following:



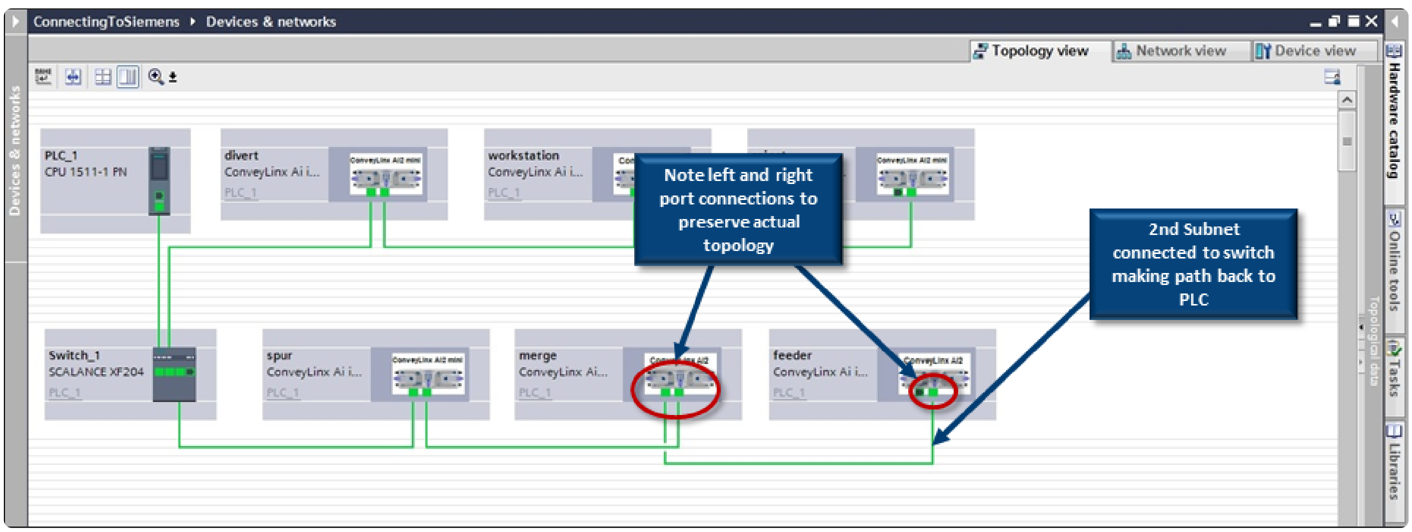
14.5.3.2. Connecting 2nd Subnet of Modules

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.

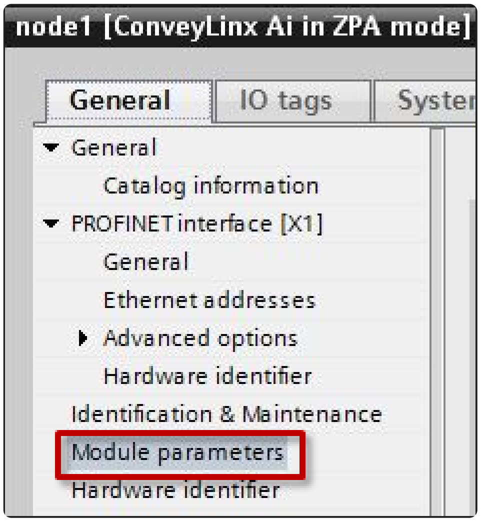


When you are done with the 2nd subnet of modules, it should look like the following:



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.

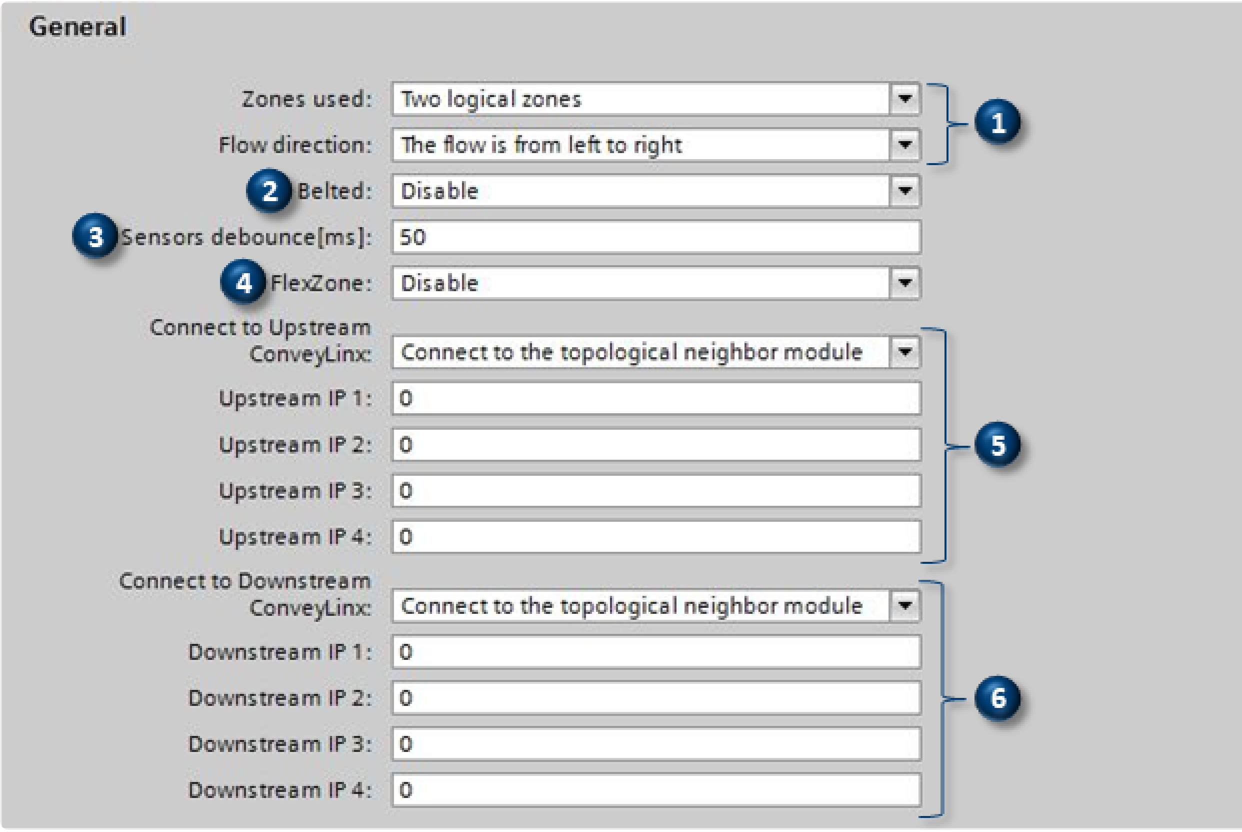


The following sections show the various parameters you have access to for each of the DAPs along with links to the corresponding EasyRoll explanations of the particular parameter:

ZPA/Reduced ZPA Modes
PLC/Reduced PLC Modes
Merger Mode
ConveyLogix Mode

14.5.4.1. ZPA/Reduced ZPA Modes

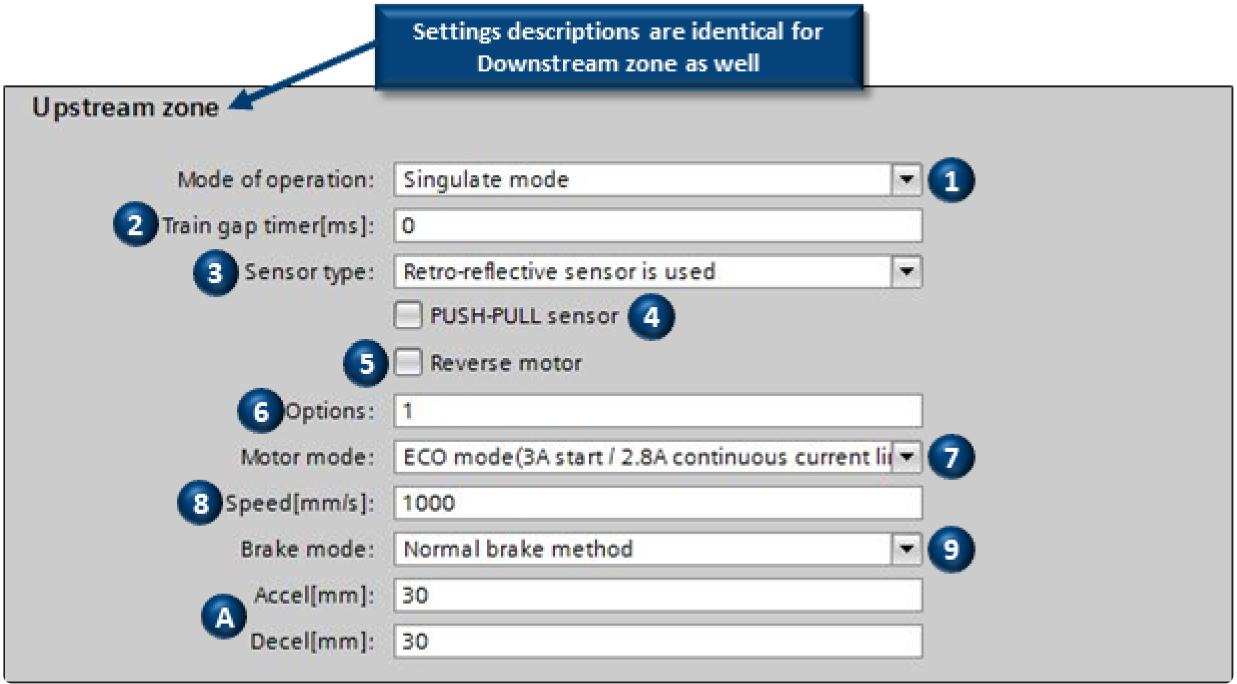
General



1. There is no equivalent setting in *EasyRoll* because these are determined by the *Auto-Configuration* procedure
2. Set up for master/slave motor rollers in a belted zone. There is no equivalent setting in *EasyRoll*
3. Sensor Debounce setting
4. Enable/Disable FlexZone operation
5. 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

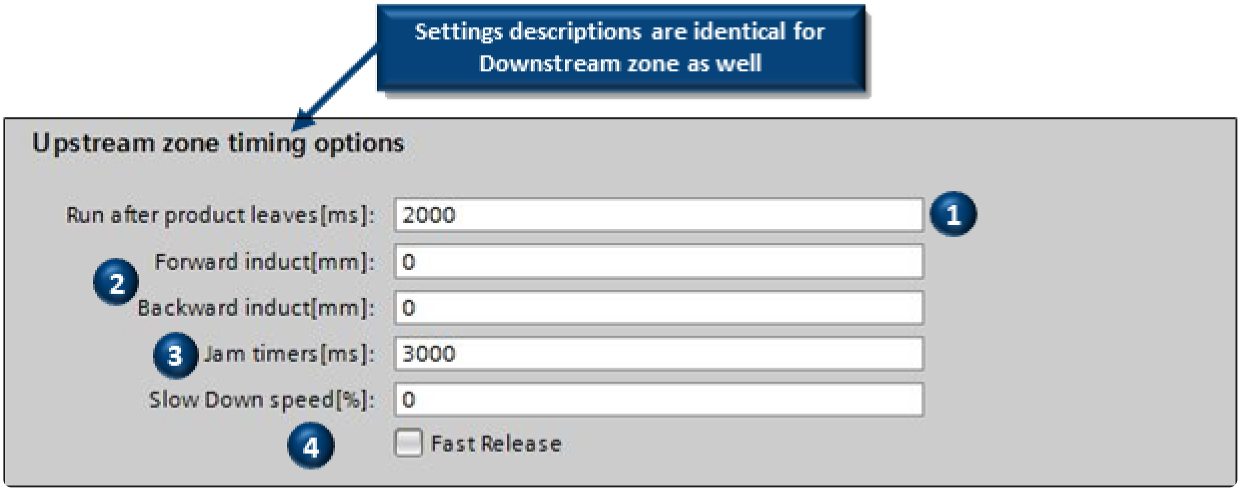
8. Enter desired Motor Speed

9. Change motor's Brake Method

A. Enter desired motor Accel/ Decel values

> ✱ These settings are identical for the Downstream Zone

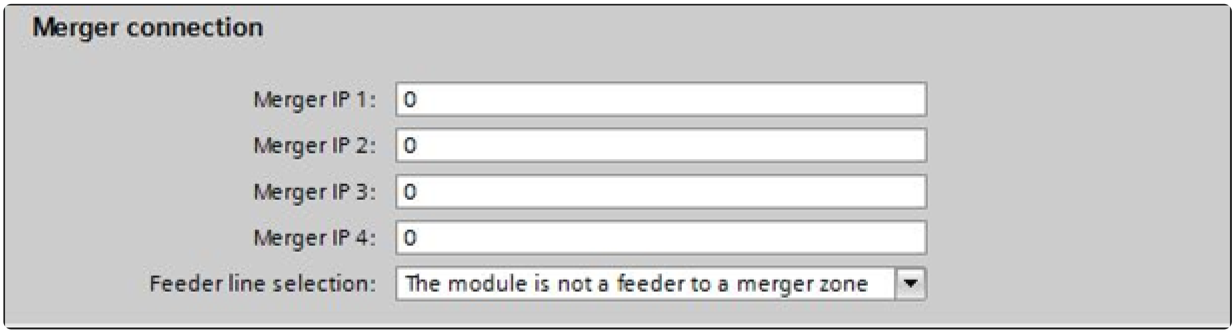
14.5.4.1.2. Upstream/Downstream Zone Timing



1. Change zone’s Run After Timer from its default value
2. Change zone’s Induct Forward and Reverse Timer values
3. Change zone’s Jam Timer from its default value
4. Configure zone’s Look Ahead Slowdown feature operation

> These settings are identical for the Downstream Zone

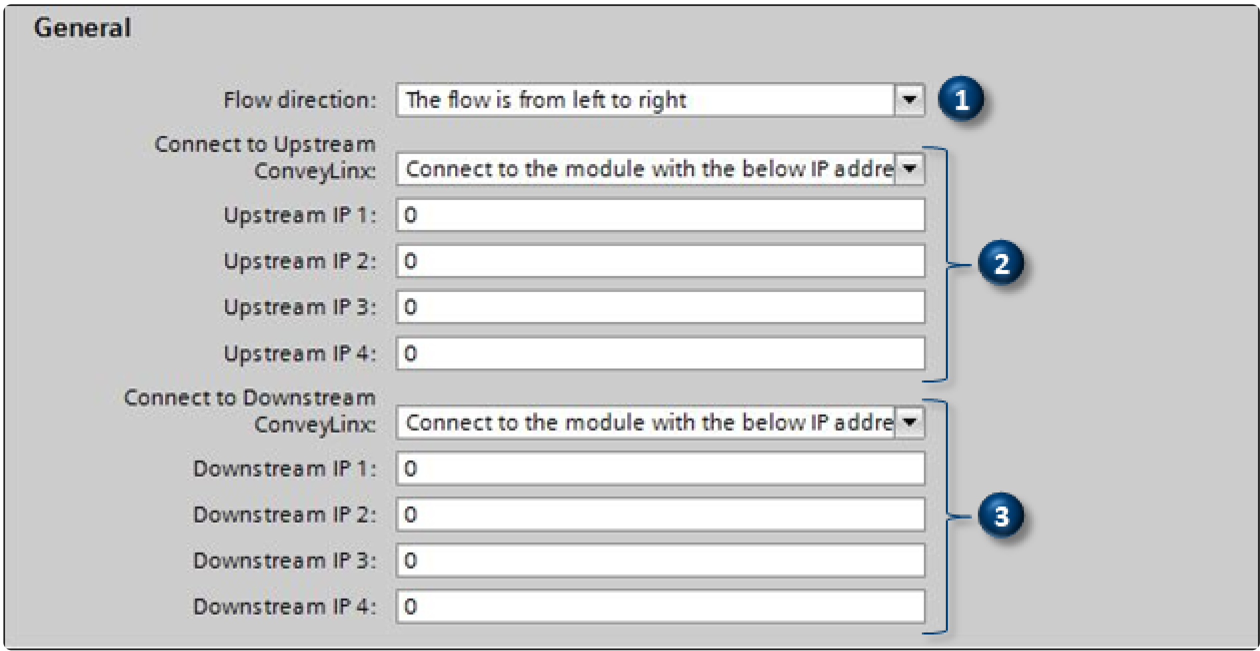
14.5.4.1.3. Connection to Merger Module



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

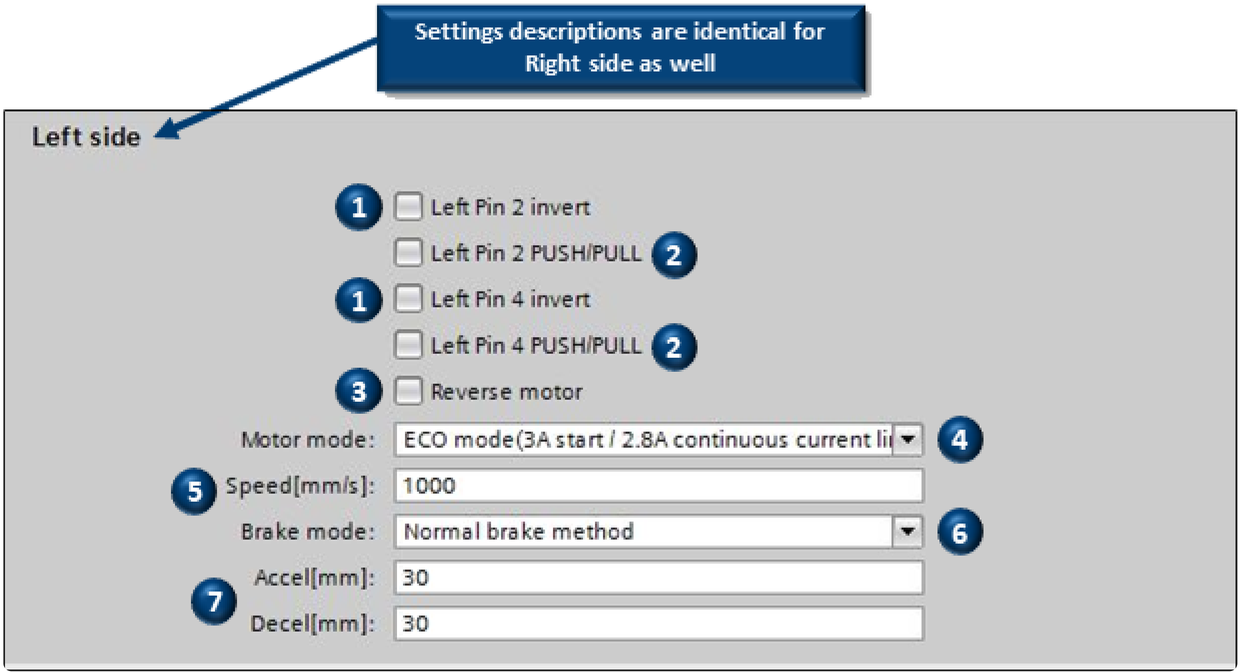


1 There is no equivalent setting in *EasyRoll* because these are determined by the *Auto-Configuration* procedure

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



1. 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

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

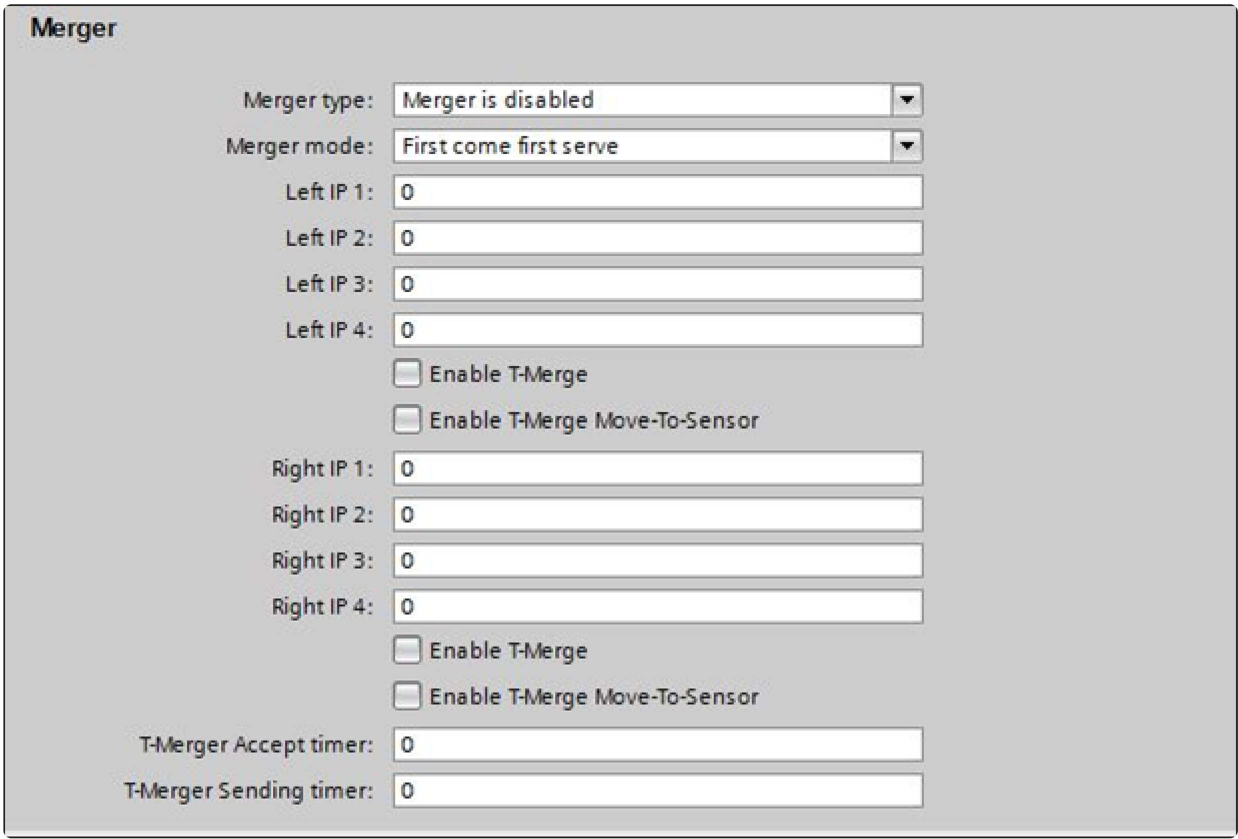
7. Enter desired motor Accel/Decel values

> 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, and Upstream/Downstream Zone Timing parameters are the same as previously described.

Merger Configuration



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

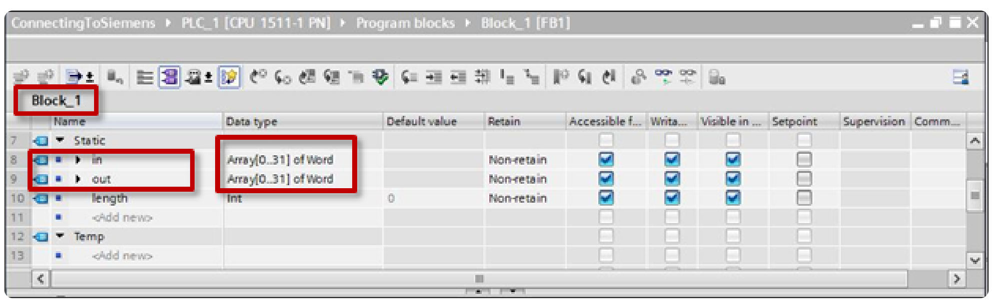
> This method is not recommended for new users or applications where multiple programmers have to edit and understand the same program

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.

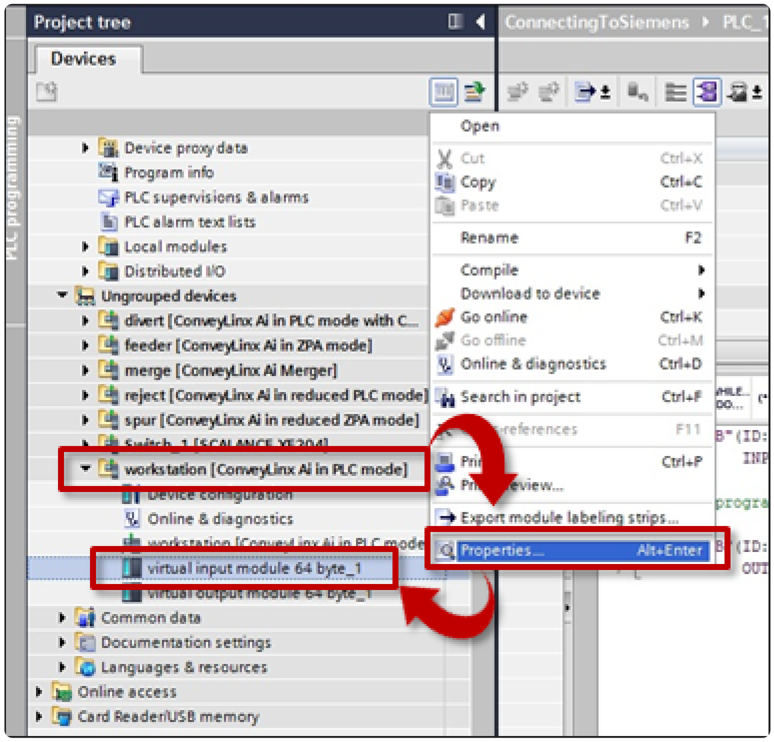
Example

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.

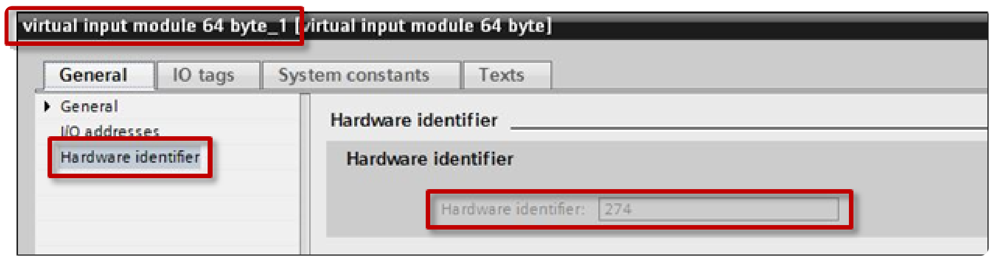
Create your *Block_1* and set the *Data type* for the *in* and *out* variables. We know the *workstation* module is in *Full PLC Mode* so it requires 64 Bytes (32 Words) of input and output data



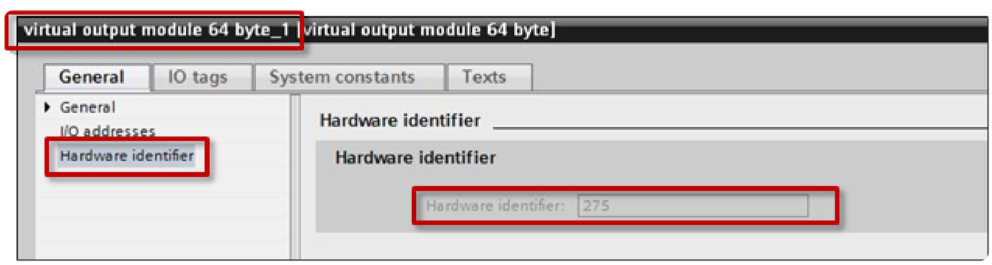
We need to find the *Hardware Identifier* for the inputs for the *workstation* module. Go to *Ungrouped devices* in the project tree, select the *workstation* module, select *virtual input module*, and select *Properties*



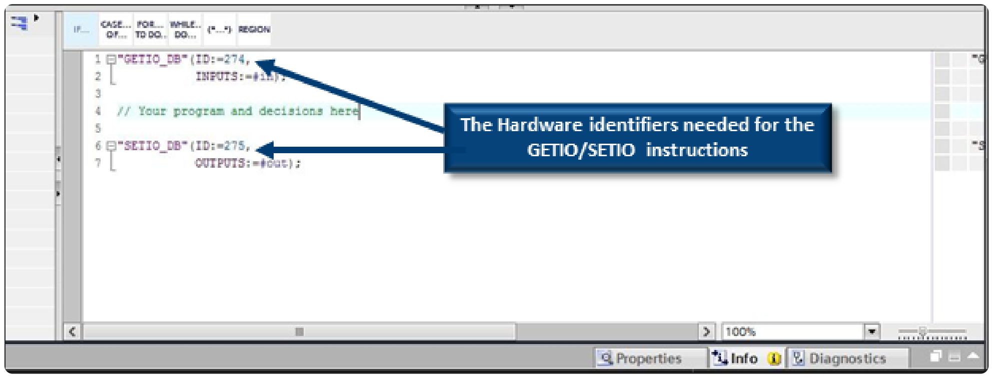
From here you can see that the *Hardware identifier* for the *inputs* for the *workstation* module is *274*. Note this value because you will need it later when you configure your *GETIO* function.



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

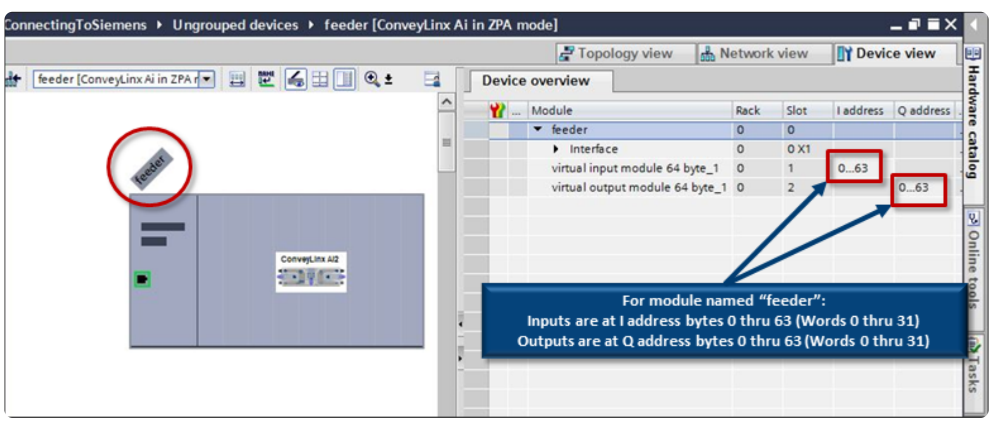
This method allows you to create your own *global tags* with meaningful names and map them to a specific *Word* address within a given module. To use this method you will have to be very familiar with the Instance Assembly structure for the ConveyLinx Module

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.

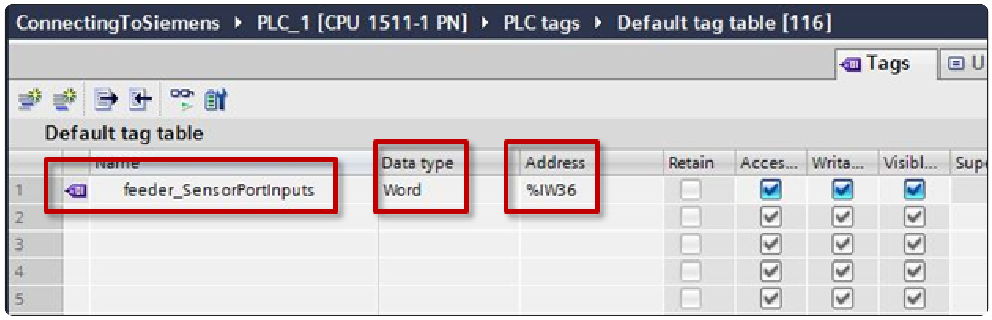
Create Tags for our *feeder* Module

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.



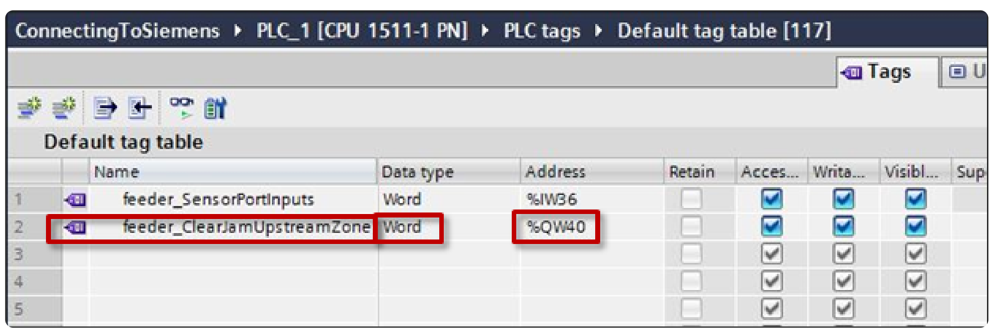
We want to read the *Sensor Port Inputs* register from the *feeder* module. As shown in the [Port Inputs and ConveyStop Status description from the PLC Developer's Guide section](), we can see that the *Sensor Port Inputs* data is at *word register offset 01* within the ZPA Mode assembly instance. To 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*.



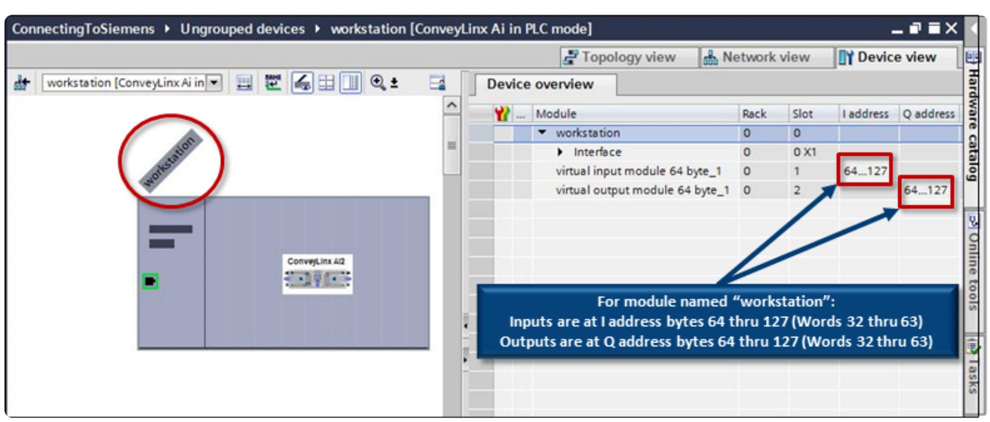
We want to write a command to the *Clear Jam for Local Upstream Zone* register to the *feeder* module. As shown in the ConveyStop and Clear Jams description from the PLC Developer's Guide section, we can see that the *Clear Jam for Local Upstream Zone* data is at ***word register offset 20*** within 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

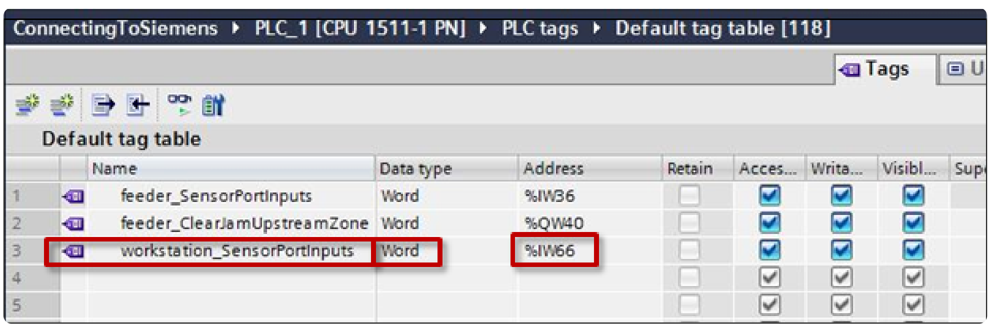
From *Network View*, locate the module *workstation*, right click and select *Device configuration* and from *Device view* expand the window from the right to display the *Device overview*. This will show you the I and Q address byte locations in memory that were assigned when



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.

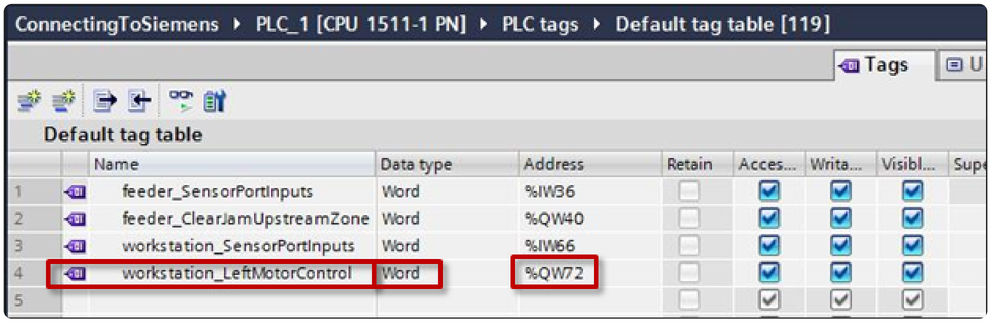
We want to read the *Sensor Port Inputs* register from the *workstation* module. As shown in the [Sensor Ports description from the PLC Developer's Guide section](), we can see that the *Sensor Port Inputs* data is at *word register offset 1* within the PLC I/O Mode assembly instance. To point our new 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)

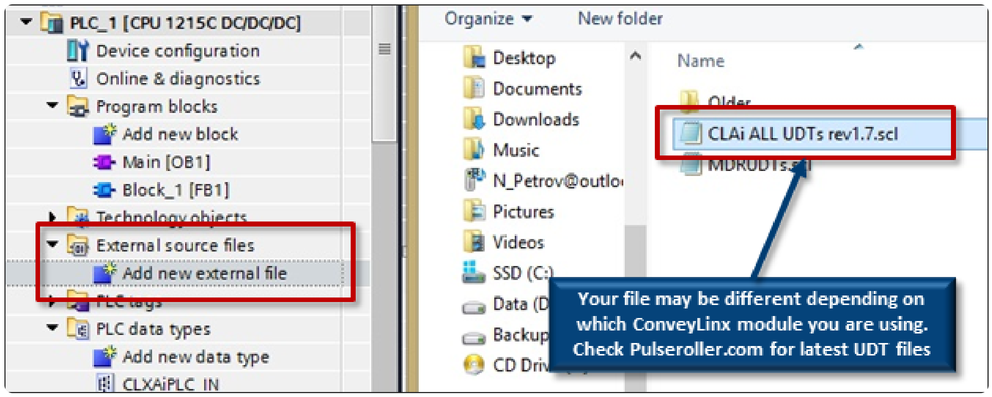
> Using the UDTs is the recommended method of data access with ConveyLinx modules because these have been tested and verified by Pulseroller.

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*.

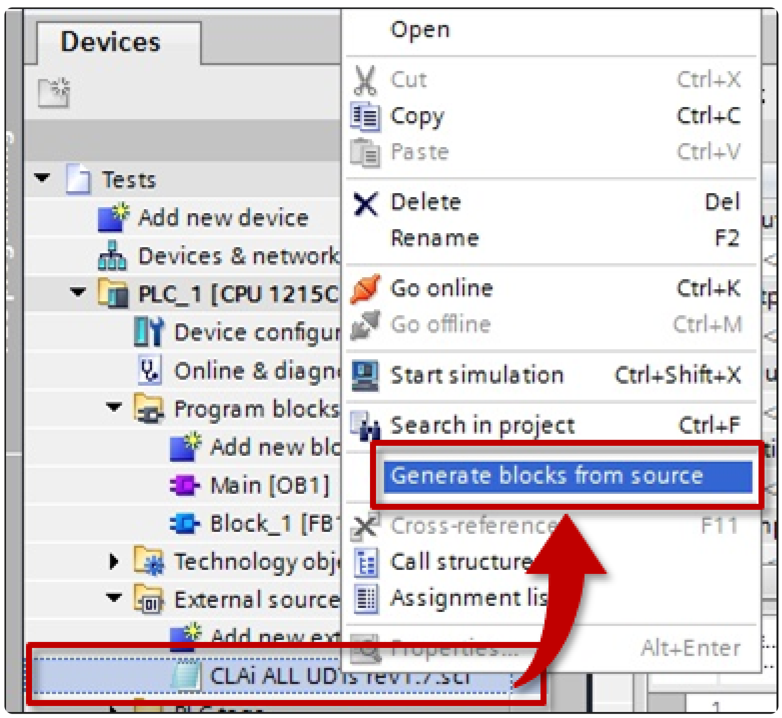
Each given *DAP* consumes some number of input memory bytes and output memory bytes. When you use *UDTs* and create tags with the UDT data types; the *PLC's I/O memory* is properly allocated and the memory boundaries between modules are easily established.

14.6.3.1. Installing UDTs into Programming Environment

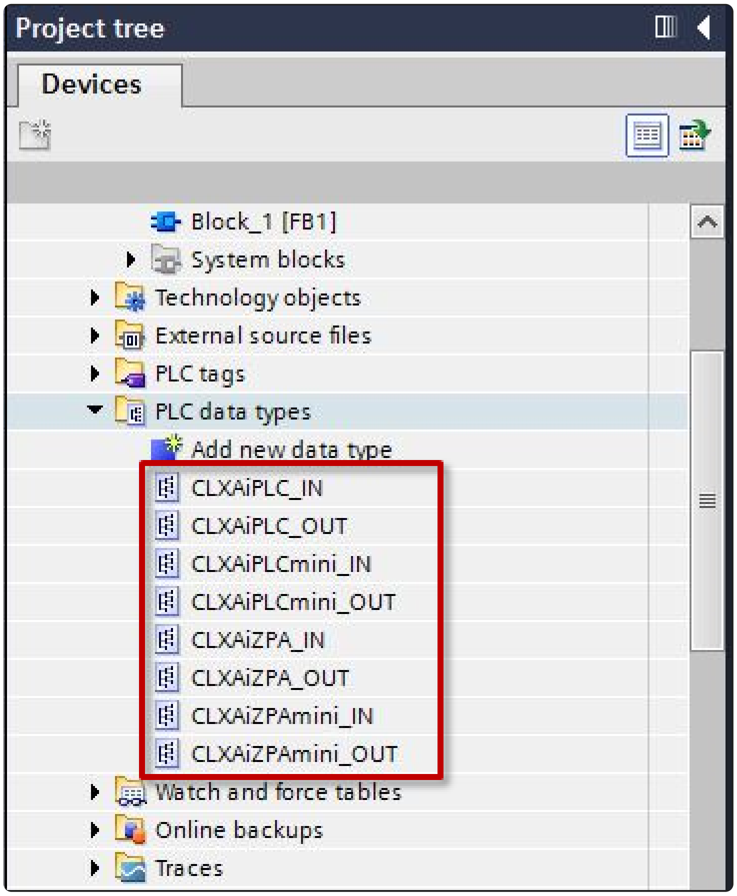
In your *Project tree*, expand *External source files* and select *Add new external file*. Browse to the location on your PC where you saved the UDT file you downloaded from Pulseroller.com. In this example we are using ConveyLinx-Ai Family modules so we select the corresponding file.



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

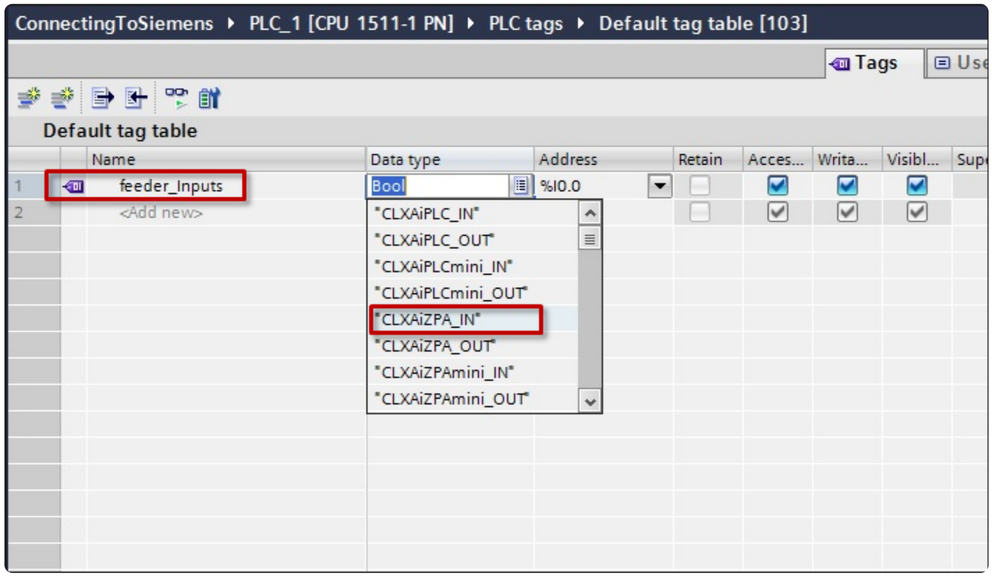
What about ConveyLogix Interface?

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.

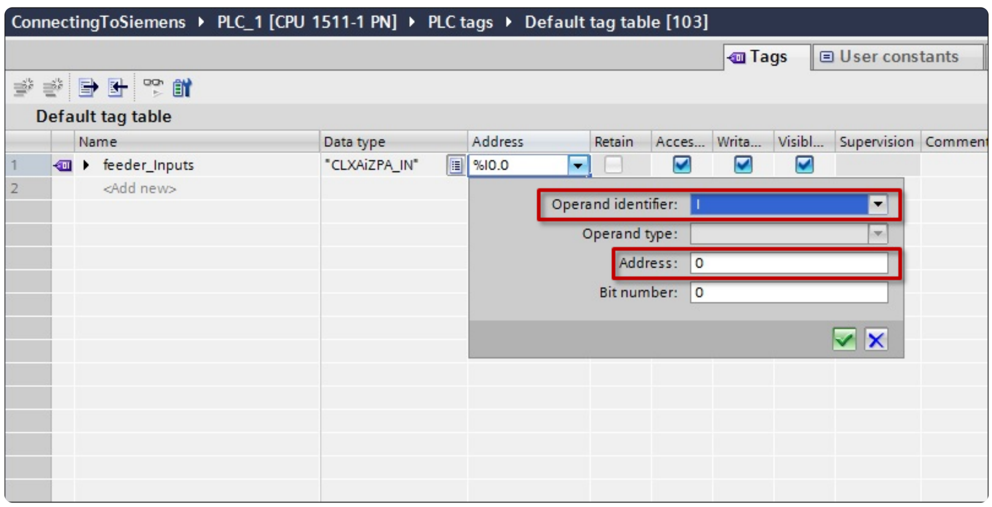
For this example, you can access the *divert* module by either the Raw Unmapped Data Direct from Module or Module Data Elements Mapped to Tags method previously described.

14.6.3.3.1. Add feeder Module

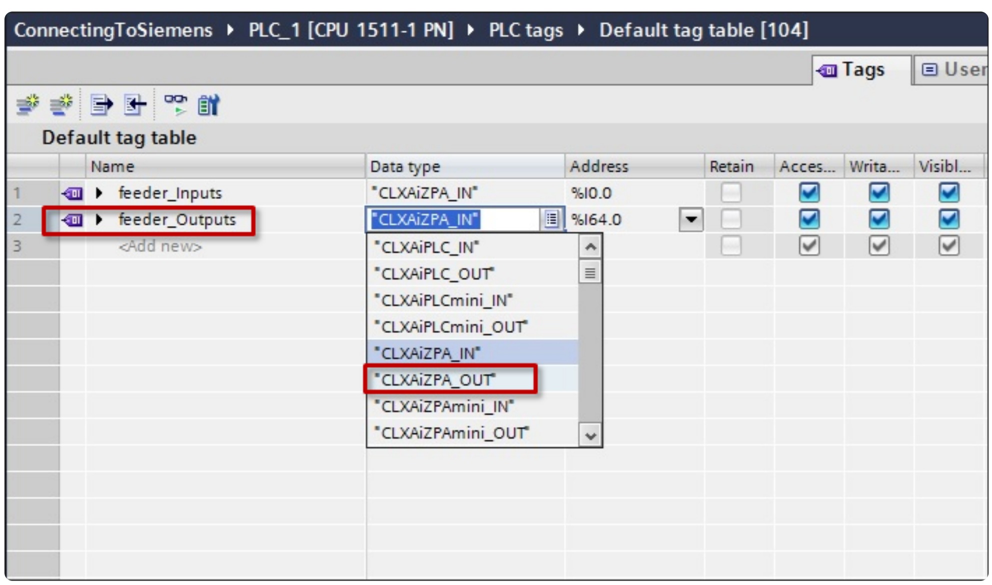
From our *Default tag table*, create a new tag and let's call it *feeder_Inputs*. Select *CLXAiZPA_IN* as the *Data type*.



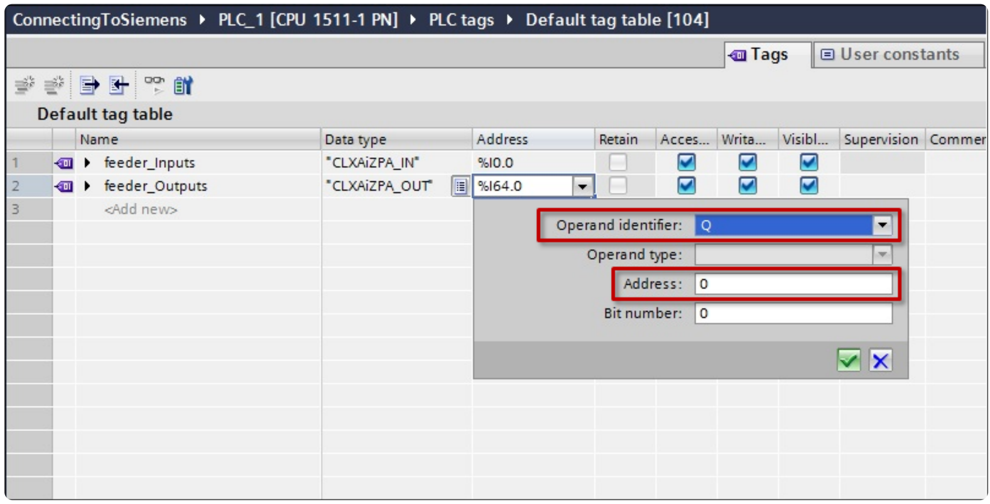
Because we know from [Module Data Elements Mapped to Tags section example]() that the *feeder* module starts at Input byte offset 0, we know what to enter for the address.



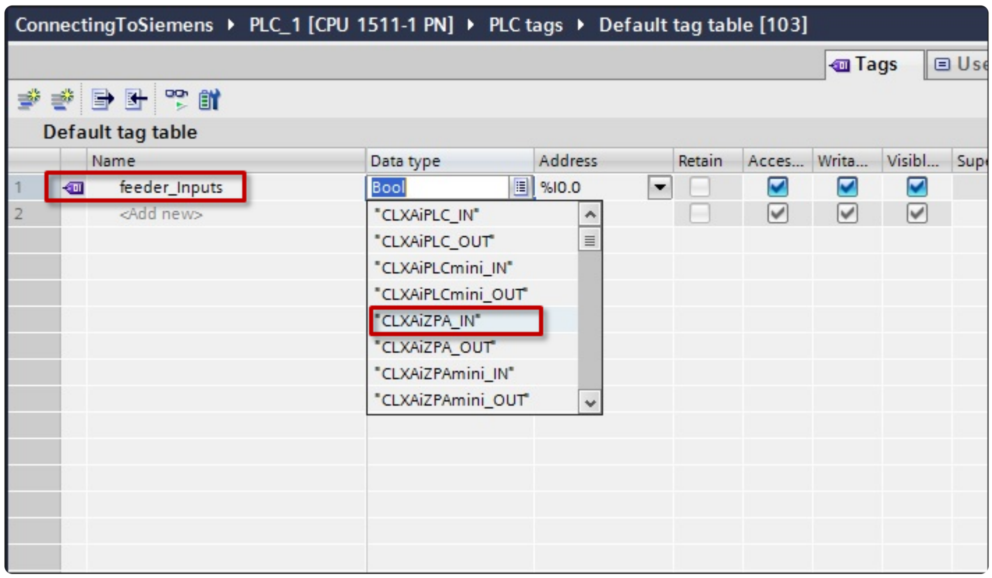
From our *Default tag table*, create a new tag for the outputs and let's call it *feeder_Inputs*. Select *CLXAiZPA_OUT* as the *Data type*.



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.

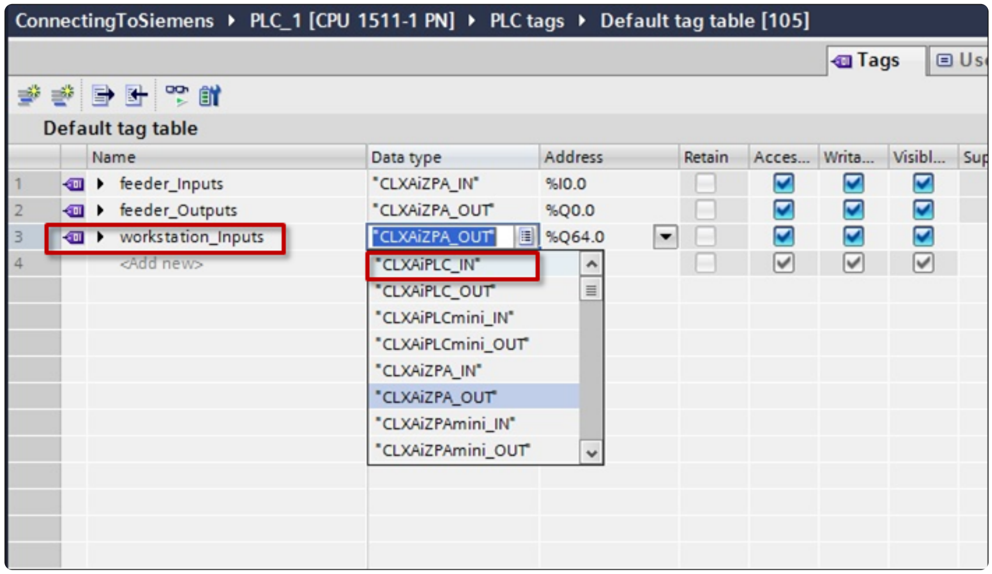


To continue and add the *workstation* module, from our *Default tag table*, create a new tag and let's call it *workstation_Inputs*. Select *CLXAiPLC_IN* as the *Data type*.

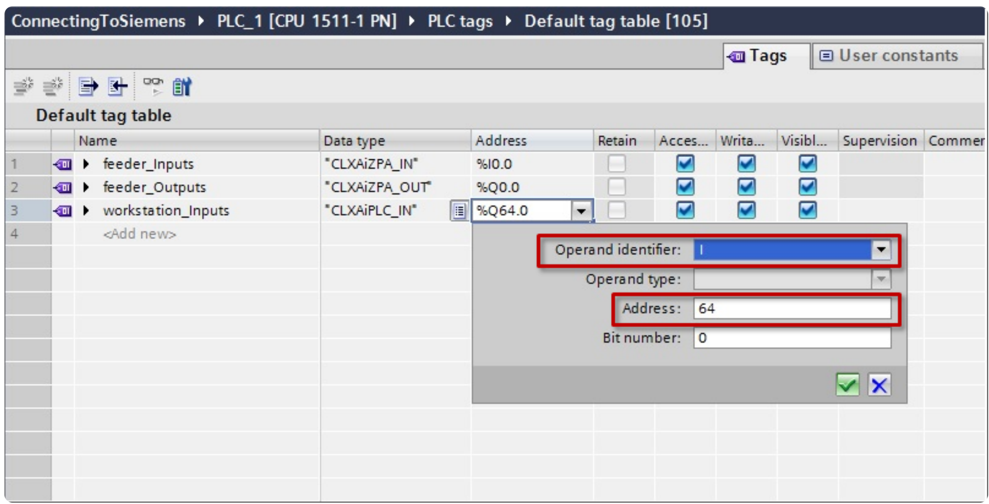


14.6.3.3.2. Add workstation Module

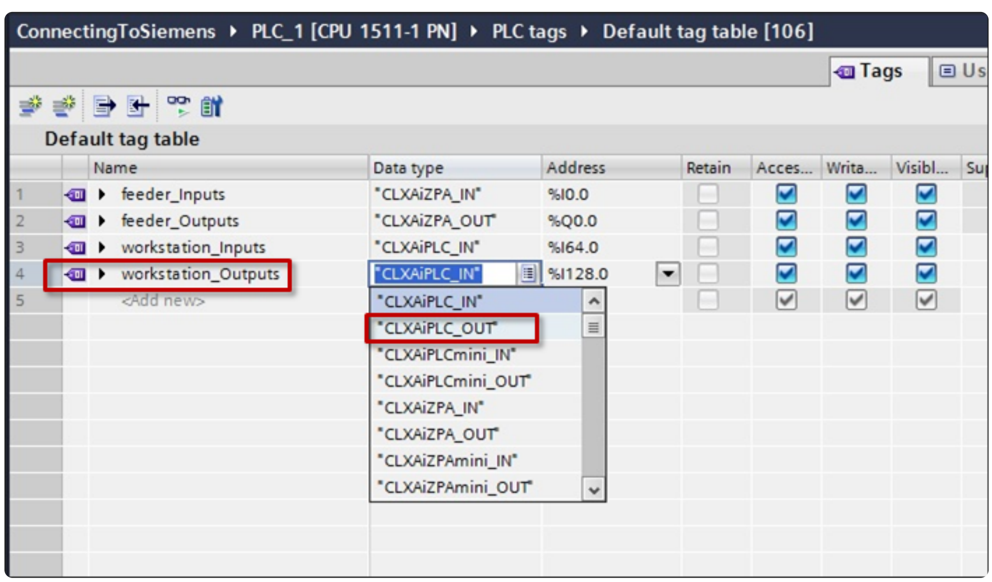
From our *Default tag table*, create a new tag and let's call it *workstation_Inputs*. From the chart in the [UDT Assignment Example topic](), we know that the *workstation* module is in Full PLC I/O mode. Select *CLXAiPLC_IN* as the *Data type*.



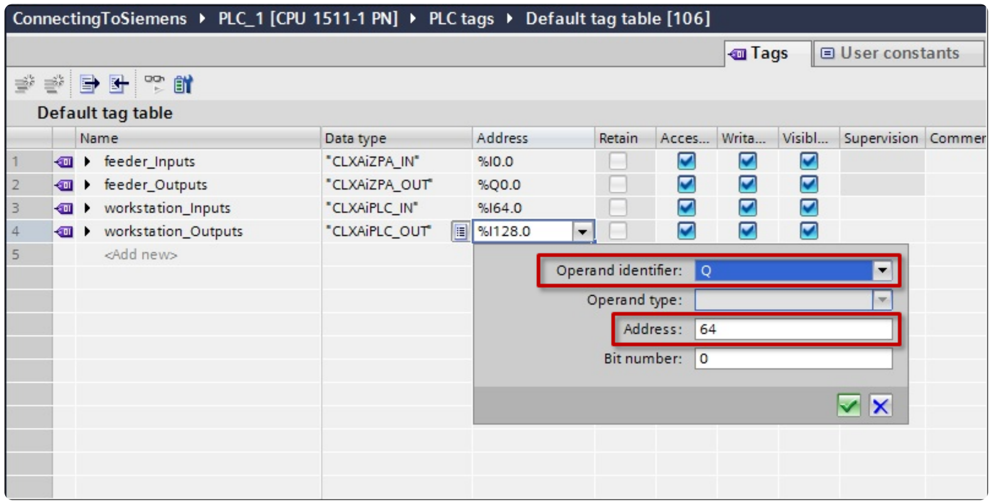
Because we know from [Module Data Elements Mapped to Tags section example]() that the *workstation* module starts at Input byte offset 64, we know what to enter for the address.



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*.



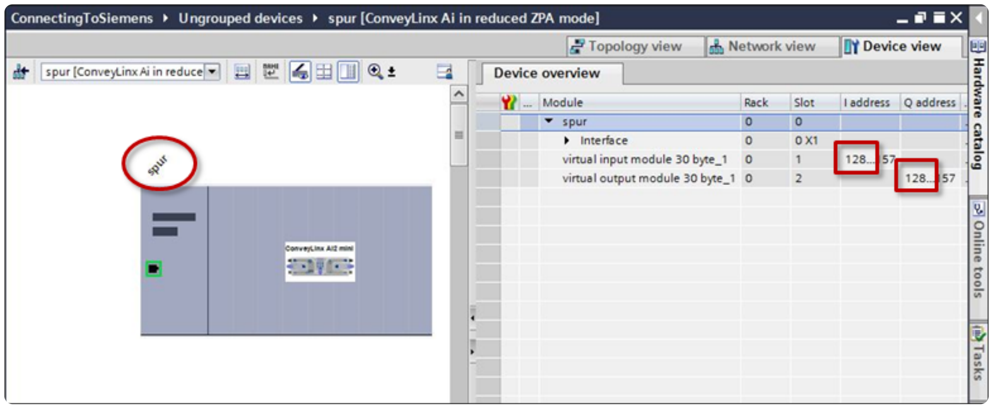
Because we know from Module Data Elements Mapped to Tags section example that the outputs for the *workstation* module starts at Output byte offset 64, we know what to enter for the address.



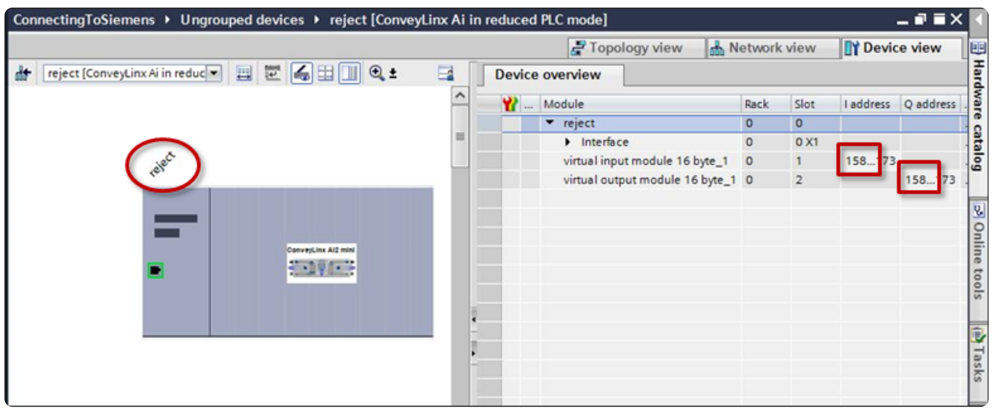
14.6.3.3.3. Add remaining Modules

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.

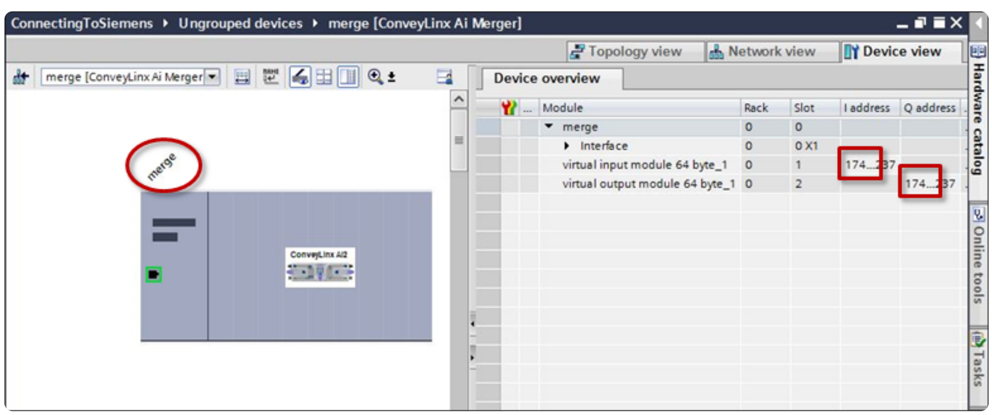
For the *spur* module, when we look at its *Device view* we see that its starting input and output byte offset is 128. From the chart in the [UDT Assignment Example topic](), we know that the *spur* module is in Reduced ZPA mode.



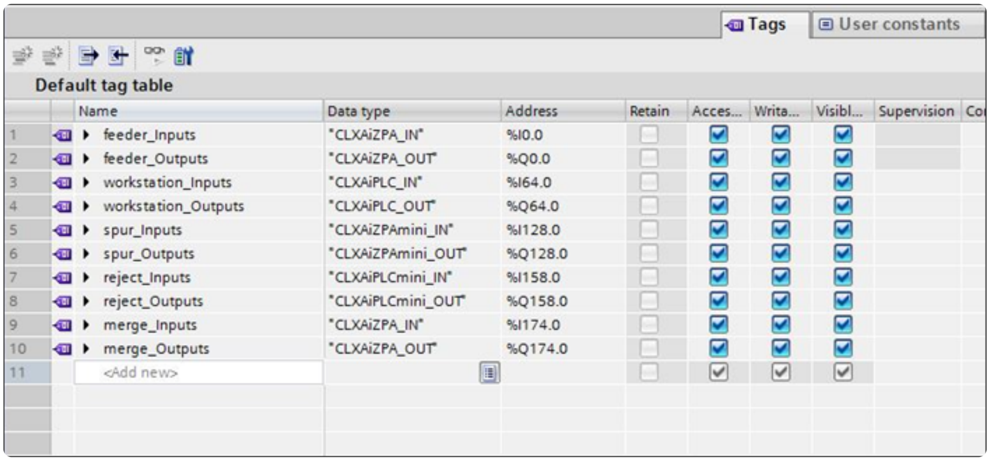
For the *reject* module, when we look at its *Device view* we see that its starting input and output byte offset is 158. From the chart in the [UDT Assignment Example topic](), we know that the *spur* module is in Reduced PLC mode.



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>				✔	✔	✔		

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 ConveyLinx-ERSC Family

Conveyor Control DAPs for Separate ConveyLinx Auto-Configuration

ConveyLinx Mode	DAP from Catalog	Input Bytes	Output Bytes
Full ZPA	*ConveyLinx in ZPA mode*	64	64
Full PLC	*ConveyLinx in PLC mode*	64	64
Full ZPA with Merger	*ConveyLinx in ZPA mode*	64	64
PLC with ConveyLogix	*ConveyLinx in PLC mode with ConveyLogix*	32	32
Reduced PLC	*ConveyLinx in reduced PLC mode*	16	16
Reduced ZPA	*ConveyLinx 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 in PLC Mode*	64	64
PLC with ConveyLogix	*ConveyLinx in PLC mode with ConveyLogix*	32	32
Reduced PLC	*ConveyLinx in reduced PLC*	16	16

	mode		
Reduced ZPA	*ConveyLinx in reduced ZPA mode*	30	30
Full ZPA	*ConveyLinx in ZPA mode*	64	64
ZPA with Merge	*ConveyLinx Merger*	64	64

14.7.1.1. Full ZPA Mode Inputs

CLX_ERSC_ZPA_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	More Info
ReleaseCounterDownstreamZone		Int	
GetDischargeTracking_Fwd		DWord	More Info
GetDischargeTracking_Rev		DWord	
PortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	

	Bit14	Bool	
	Heartbeat	Bool	
	LeftSensorPin3	Bool	
	LeftControlPin3	Bool	
	RightSensorPin3	Bool	
	RightControlPin3	Bool	
	LeftSensorPin4	Bool	
	LeftControlPin4	Bool	
	RightSensorPin4	Bool	
	RightControlPin4	Bool	
Index19		Word	
ConveyStopStatus	StopOnLeft	Bool	More Info
	StopOnRight	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

CLX_ERSC_ZPA_OUT

Element Name	Sub Element	Data Type	Link
InductTrackingOnUpstreamZone		DWord	More Info
InductTrackingOnDownstreamZone		DWord	
AccumulateControlUpstream	AccumUpstreamToThisZone	Bool	More Info
	SetArrival	Bool	
	JogZoneFwd	Bool	
	JogZoneRev	Bool	
	WakeUp	Bool	
	MaintEnable	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	
	JogZoneFwd	Bool	
	JogZoneRev	Bool	
	WakeUp	Bool	
	MaintEnable	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	More Info
SetInductTrackingRev		DWord	
ClearMotorError		Word	More Info
SetControlPortOutputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	

	Bit00	Bool	
	EnergizeLeft	Bool	
	Bit02	Bool	
	EnergizeRight	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
Index18		Word	
ConveyStopControl		Word	More Info
ClearJamUpstream		Word	More Info
ClearJamDownstream		Word	
SetDirectionControlUpstream		Word	More Info
SetDirectionControlDownstream		Word	
ConveyMerge		Word	More Info
Future		Array[25..31] of Word	

14.7.1.3. Reduced ZPA Mode Inputs

CLX_ERSC_ZPA_Reduced_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	
ReleaseCounterUpstreamZone		Int	More Info
ReleaseCounterDownstreamZone		Int	
PortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Heartbeat	Bool	
	LeftSensorPin3	Bool	
	LeftControlPin3	Bool	

	RightSensorPin3	Bool	
	RightControlPin3	Bool	
	LeftSensorPin4	Bool	
	LeftControlPin4	Bool	
	RightSensorPin4	Bool	
	RightControlPin4	Bool	
Future		Array[11..14] of Word	

14.7.1.4. Reduced ZPA Mode Outputs

CLX_ERSC_ZPA_Reduced_OUT

Element Name	Sub Element	Data Type	Link
AccumulateControlUpstream	AccumUpstreamToThisZone	Bool	More Info
	SetArrival	Bool	
	JogZoneFwd	Bool	
	JogZoneRev	Bool	
	WakeUp	Bool	
	MaintEnable	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	
	JogZoneFwd	Bool	
	JogZoneRev	Bool	
	WakeUp	Bool	
	MaintEnable	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
SetControlPortOutputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bit00	Bool	
	EnergizeLeft	Bool	
	Bit02	Bool	

	EnergizeRight	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
Index10		Word	
ClearJamUpstream		Word	More Info
ClearJamDownstream		Word	
SetDirectionControlUpstream		Word	More Info
SetDirectionControlDownstream		Word	

14.7.1.5. Full PLC I/O Mode Inputs

CLX_ERSC_PLC_IN

Element Name	Sub Element	Data Type	Link
ConveyStopStatus	StopOnLeft	Bool	More Info
	StopOnRight	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	
PortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Heartbeat	Bool	

	LeftSensorPin3	Bool	
	LeftControlPin3	Bool	
	RightSensorPin3	Bool	
	RightControlPin3	Bool	
	LeftSensorPin4	Bool	
	LeftControlPin4	Bool	
	RightSensorPin4	Bool	
	RightControlPin4	Bool	
SensorDetect	Bits08_15	Byte	More Info
	RightSensorDetect	Bool	
	LeftSensorDetect	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
ModuleVoltage		Int	More Info
LeftMotorCurrent		Int	More Info
LeftMotorFreq		Int	
LeftMotorCalcTemp		Byte	
LeftModuleTemp		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	
RightMotorCurrent		Int	More Info
RightMotorFreq		Int	
RightMotorCalcTemp		Byte	
RightModuleTemp		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	
	OverCurrent	Bool	
	Bit15	Bool	
	Bits00_07	Byte	
RightMotorDIOstatus	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	ShortCircuit	Bool	
	Bit13	Bool	
	OverCurrent	Bool	
	Bit15	Bool	
	Bits00_07	Byte	
UpstreamModuleStatus		Word	More Info
DownstreamModuleStatus		Word	
TrackingFromUpstream		DWord	
Index18		Word	
ServoLocationLeft		Int	More Info
ServoLocationRight		Int	
ServoStatusLeft		Word	

ServoStatusRight		Word	
Future		Array[23..31] of Word	

14.7.1.6. Full PLC I/O Mode Outputs

CLX_ERSC_PLC_OUT

Element Name	Sub Element	Data Type	Link
ConveyStopControl		Word	More Info
LeftMotorAsDIO	ClearPortError	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	EnableDIO	Bool	
	EnergizeJSTPin3	Bool	
	EnergizeJSTPin4	Bool	
	EnergizeJSTPin5	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	EnergizeJSTPin9	Bool	
	EnableJSTPin9	Bool	
RightMotorAsDIO	ClearPortError	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	

	Bit14	Bool	
	EnableMotDIO	Bool	
	EnergizeJSTPin3	Bool	
	EnergizeJSTPin4	Bool	
	EnergizeJSTPin5	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	EnergizeJSTPin9	Bool	
	EnableJSTPin9	Bool	
ControlPortsDO	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bit00	Bool	
	EnergizeLeftPort	Bool	
	Bit02	Bool	
	EnergizeRightPort	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
LeftMotorControl	Direction	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	
SetLeftMotorBrakeMode		Word	More Info
SetLeftMotorSpeedMode		Word	
RightMotorControl	Direction	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	
SetRightMotorBrakeMode		Word	More Info
SetRightMotorSpeedMode		Word	
SetSpeedLeftMotor		Int	More Info
SetSpeedRightMotor		Int	More Info
SetLeftMotorAccel		Int	More Info
SetLeftMotorDeccel		Int	
SetRightMotorAccel		Int	More Info
SetRightMotorDeccel		Int	
ClearMotorError		Word	More Info
SendStatusToDownstream		Word	More Info
SendStatusToUpstream		Word	
SetSensorPolarity	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	LeftSensorPin3	Bool	
	LeftControlPin3	Bool	
	RightSensorPin3	Bool	

	RightControlPin3	Bool	
	LeftSensorPin4	Bool	
	LeftControlPin4	Bool	
	RightSensorPin4	Bool	
	RightControlPin4	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 I/O Mode Inputs

CLX_ERSC_PLC_Reduced_IN

Element Name	Sub Element	Data Type	Link
PortInputs	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Heartbeat	Bool	
	LeftSensorPin3	Bool	
	LeftControlPin3	Bool	
	RightSensorPin3	Bool	
	RightControlPin3	Bool	
	LeftSensorPin4	Bool	
	LeftControlPin4	Bool	
	RightSensorPin4	Bool	
	RightControlPin4	Bool	
SensorDetect	Bits08_15	Byte	More Info
	RightSensorDetect	Bool	
	LeftSensorDetect	Bool	
	Bit02	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	

	Bit06	Bool	
	Bit07	Bool	
LeftMotorCalcTemp		Byte	More Info
LeftModuleTemp		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
RightModuleTemp		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	
	OverCurrent	Bool	
	Bit15	Bool	
	Bits00_07	Byte	
RightMotorDIOstatus	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	ShortCircuit	Bool	
	Bit13	Bool	
	OverCurrent	Bool	
	Bit15	Bool	
	Bits00_07	Byte	

14.7.1.8. Reduced PLC I/O Mode Outputs

CLX_ERSC_PLC_Reduced_OUT

Element Name	Sub Element	Data Type	Link
LeftMotorAsDIO	ClearPortError	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	EnableDIO	Bool	
	EnergizeJSTPin3	Bool	
	EnergizeJSTPin4	Bool	
	EnergizeJSTPin5	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	EnergizeJSTPin9	Bool	
	EnableJSTPin9	Bool	
RightMotorAsDIO	ClearPortError	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	

	EnableMotDIO	Bool	
	EnergizeJSTPin3	Bool	
	EnergizeJSTPin4	Bool	
	EnergizeJSTPin5	Bool	
	Bit03	Bool	
	Bit04	Bool	
	Bit05	Bool	
	EnergizeJSTPin9	Bool	
	EnableJSTPin9	Bool	
ControlPortsDO	Bit08	Bool	More Info
	Bit09	Bool	
	Bit10	Bool	
	Bit11	Bool	
	Bit12	Bool	
	Bit13	Bool	
	Bit14	Bool	
	Bit15	Bool	
	Bit00	Bool	
	EnergizeLeftPort	Bool	
	Bit02	Bool	
	EnergizeRightPort	Bool	
	Bit04	Bool	
	Bit05	Bool	
	Bit06	Bool	
	Bit07	Bool	
LeftMotorControl	Direction	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	Direction	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	
SetSpeedLeftMotor		Int	More Info
SetSpeedRightMotor		Int	More Info
ClearMotorError		Word	More Info