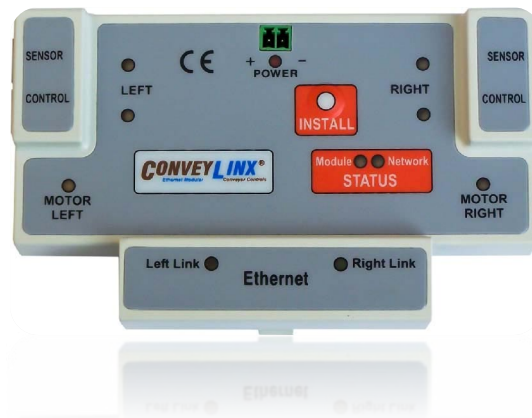
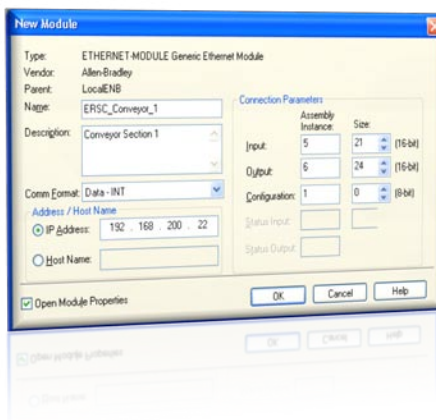
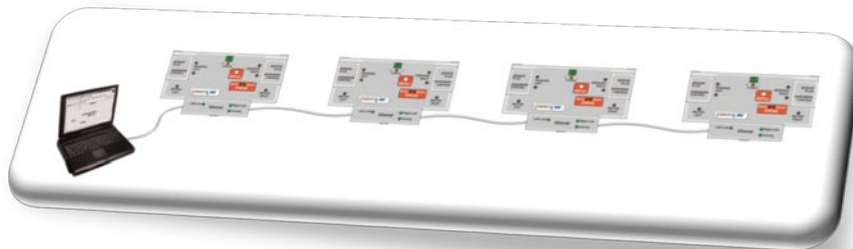
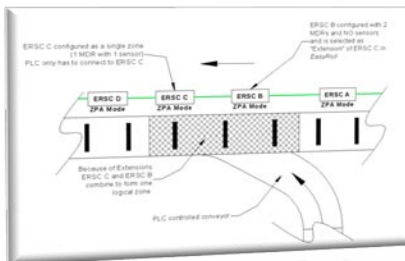


PLC Developer's Guide

Version 4.7

September 2014





GLOSSARY OF TERMS

ConveyLinx	Conveyor controls architecture based upon modular distributed devices connected via Ethernet network.
ConveyStop	Optional functionality available for <i>ConveyLinx</i> systems to provide controlled stop and reset of MDR function. Requires software package to configure and enable.
ERSC	E thernet R oller S peed C ontrol module - Conveyor control module that is part of the <i>ConveyLinx</i> family. Each ERSC Can accommodate up to 2 MDR conveyor zones
Ethernet I/P	<p>Ethernet/IP (Ethernet Industrial Protocol) is a network communication standard capable of handling large amounts of data at speeds of 10 Mbps or 100 Mbps, and at up to 1500 bytes per packet. The specification uses an open protocol at the Application layer. It is especially popular for control applications.</p> <p>Ethernet/IP typically employs active star network technology. This type of network is easy to set up, operate, maintain, and expand. It allows mixing of 10 Mbps and 100 Mbps products, and is compatible with most Ethernet switches. Ethernet/IP is used with personal computers, mainframes, robots, input/output (I/O) devices and adapters, programmable logic controllers (PLCs), and other devices. The specification is supported by the Industrial Ethernet Association (IEA), ControlNet International (CI), and the Open DeviceNet Vendor Association (ODVA).</p>
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.
Carton	A separate (usually wrapped or boxed) object to be transported by the conveyor. The terms tray , tote , or load may also be used interchangeably in this document.
MDR	M otorized D rive R oller or M otor D riven R oller - Brushless DC motor and gearbox assembly integrated into a single conveyor roller.
Modbus TCP	<p>Application layer messaging protocol at Level 7 of the OSI Model that provides client/server communication between devices connected on different types of buses or networks. The Modbus messaging structure was developed by Modicon in 1979. Different versions of Modbus used today include Modbus RTU (based on serial communication like RS485 and RS232), Modbus ASCII and Modbus TCP, which is the Modbus RTU protocol embedded into TCP/IP packets. It's an open protocol, meaning the specification is available free of charge for download, and there are no licensing fees required for using Modbus or Modbus TCP/IP protocols.</p> <p>Modbus TCP/IP specification was developed in 1999 to combining a ubiquitous physical network (Ethernet) with a universal networking standard (TCP/IP) and a vendor-neutral data representation. Modbus TCP/IP used the Modbus instruction set and wraps TCP/IP around it.</p>
Photo-sensor	A device, mounted near the end of the conveyor zone to sense the presence of a Carton on the zone
PLC	P rogrammable L ogic C ontroller – A wide variety of industrial computing devices that control automatic equipment

Profinet I/O	PROFINET IO uses traditional Ethernet hardware and software to define a network that structures the task of exchanging data, alarms and diagnostics with Programmable Controllers and other automation controllers.
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
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 Cartons to queue on a conveyor in discrete zones such that Cartons do not touch each other

SYMBOL CONVENTIONS



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



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



IMPORTANT USER INFORMATION

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



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

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



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



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



SUMMARY OF CHANGES

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

Revision	Date	Change / Update
1.0	October 2009	Initial Release
3.0	July 2010	Supersedes Version 1.0 – VALID FOR FIRMWARE 3.0 AND HIGHER - Added Ethernet I/P Functionality with examples
4.0	June 2011	Added functionality to Ethernet I/P Instances.
4.1	February 2012	Added more example applications and Instance items for optional functionality
4.2	October 2012	General format revisions, changed title to PLC Developer's Guide, added Assemblies for Modbus TCP, added examples
4.3	November 2013	Added Servo control registers to Instances 7/8 and Servo Control example
4.4	January 2014	Added Global Contact Information
4.5	April 2014	Updated Global Contact Information
4.6	June 2014	Added ConveyStop registers to instances
4.7	September 2014	Added Profinet I/O to charts, removed Rockwell PLC connect instructions

GLOBAL CONTACT INFORMATION



PULSE ROLLER

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



TABLE OF CONTENTS

Glossary of Terms.....	3
Symbol Conventions	4
Important User Information.....	5
Summary of Changes.....	6
Global Contact Information.....	6
Table of Contents.....	7
Preface	9
Who Should Use This Manual?.....	9
Not Included in This Manual	9
Introduction to <i>ConveyLinx</i> ®.....	11
ConveyLinx® Concept.....	11
ConveyLinx ERSC Operational Modes	13
ZPA Mode	13
PLC I/O Mode	13
<i>ConveyLinx</i> Network Architecture	15
Modbus Assembly Instance Structure.....	15
Ethernet I/P Assembly Instance Structure.....	17
Profinet IO Assembly Instance Structure.....	18
PLC Controller with ZPA Mode.....	19
Notes About Assemblies for ERSC in ZPA Mode	19
Register Chart Legend	19
PLC Inputs for ZPA Mode (Assembly #5).....	20
Assembly #5 Inputs Note 2.....	23
Assembly #5 Inputs Note 1.....	23
PLC Outputs for ZPA Mode (Assembly #6).....	24
Assembly #6 Outputs Note 1.....	27
Assembly #6 Outputs Note 2.....	27
Assembly #6 Outputs Note 3.....	28
Assembly #6 Outputs Note 4.....	29
Assembly #6 Outputs Note 5.....	29
Assembly #6 Outputs Note 6.....	29
ZPA Mode Examples	31
Example 1 – Basic Accumulate and Release with Tracking Data	31
Example 2 – Basic Bar Code Reader	35
Example 3 – Upstream Accept Interface	36
Example 4 – Downstream Discharge Interface	38
Example 5 – Simple Divert Example.....	40
Example 6 – Merge onto ZPA Main Line	42
More Examples.....	43
Ethernet I/P ZPA Mode with PLC Enable	45
Assembly #25 Inputs for ZPA Mode with PLC Enable	46
Assembly #26 – Outputs for ZPA Mode with PLC Enable	47
PLC Enable Usage	48
PLC Controller with PLC I/O Mode.....	49
Setting PLC I/O Mode in EasyRoll.....	50
Optional “Clear Connections” Choice.....	51

Configuring Action for Loss of Communication	52
Inputs for PLC I/O Mode (Assembly #7)	53
Assembly #7 Inputs Note #1.....	56
Assembly #7 Inputs Note #2.....	56
Assembly #7 Inputs Note #3.....	56
Assembly #7 Inputs Note #4.....	56
Assembly #7 Inputs Note #5.....	56
Outputs for PLC I/O Mode (Assembly #8)	57
Assembly #8 Outputs Note #1.....	60
Assembly #8 Outputs Note #2.....	60
Servo Motor Control Example.....	63
PLC Ladder Diagram	65
Appendix A - Motor Port as Digital I/O	67
Appendix B – ConveyLinx Connections	69
ZPA Mode Register Data Flow	69
External Controller as ConveyLinx Bridge	70
Notes:	73



PREFACE

WHO SHOULD USE THIS MANUAL?

This manual is intended for users who need to utilize a PLC or PC to connect to *ConveyLinx* Ethernet network to access module status and control conveyor operation.

You should have an intermediate to advanced level of understanding of PLC logic and network structures. Familiarity with at least one of Modbus TCP, Ethernet I/P, or Profinet I/O protocols is also essential.

You should have a basic understanding of electrical circuitry and familiarity with relay logic, conveyor equipment, photoelectric sensors, etc. in order to follow example scenarios and sample programs included herein. If you do not, obtain the proper training before using this product.

For basic understanding of *ERSC* module hardware and simple application and installation guidelines, please refer to Insight Automation publication *ConveyLinx User's Guide* (publication *ERSC-1000*)

NOT INCLUDED IN THIS MANUAL



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

INTRODUCTION TO CONVEYLINX®

CONVEYLINX® CONCEPT

ConveyLinx control system as applied to conveyor control is a series of individual *ConveyLinx ERSC* modules interconnected via standard Ethernet cabling to form an integrated solution for MDR (Motorized Drive Roller) conveyor functionality. Each *ConveyLinx ERSC* module can accommodate up to 2 MDR's and 2 photo-sensors to provide control for up to 2 conveyor zones. Each *ERSC* Also includes convenient connectivity ports for upstream and downstream Ethernet network cabling as well as connectivity ports for discrete I/O signals with non-networked controls for local interlock interface functions.

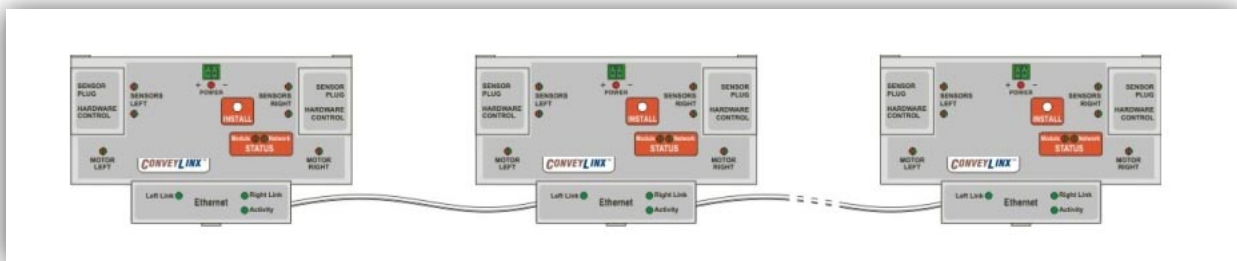


FIGURE 1 - CONVEYLINX® CONCEPT WITH ERSC MODULES

ConveyLinx ERSC modules can be easily automatically configured to operate multiple zones of linear conveyor “right out of the box” with the push of a button without any special tools or PC software required. However, with the *ConveyLinx Easy Roll* software tool and a PC; each *ERSC* module’s default configuration can be modified to customize functionality for specific applications.



Please refer to Insight Automation publication *ConveyLinx User’s Guide for ConveyLinx ERSC* module hardware details and basic function. The remaining sections of this document assumes reader if familiar with *ConveyLinx ERSC* module and *EasyRoll* software operation.



CONVEYLINX ERSC OPERATIONAL MODES

Each and every *ERSC* module on a conveyor system functions in **only one** of two possible operational modes:

- **Zero Pressure Accumulation (ZPA) Mode**
- **PLC I/O Mode**

These operational modes dictate which of the modules data registers are valid for use by a networked external controller.

ZPA MODE

ZPA Mode is the default mode of any *ERSC* that has been configured by the *Auto-Configuration Procedure*. In this mode, each *ERSC* has established logical connections to its neighboring *ERSC* modules in order to operate conveyor with Zero Pressure Accumulation (ZPA) functionality. No external controller is required for the conveyor to function and operation is as described in *ConveyLinx User's Guide* (publication *ERSC-1000*).

PLC I/O MODE

In PLC I/O Mode, the *ERSC* suspends all automatic ZPA functionality and its input, output and motor control functions are explicitly controlled by a networked external logic controller. The external controller reads from and writes to the *ERSC*'s internal data registers over the Ethernet network using Ethernet I/P or Modbus TCP protocol in order to initiate all *ERSC* functionality.



***ERSC* module can only be placed in PLC I/O Mode with *EasyRoll* software tool and only after it has been through an *Auto-Configuration Procedure*.**



IMPORTANT CONCEPT TO REMEMBER: A PLC or external controller can connect and interact with any *ERSC* regardless of whether it is in ZPA or PLC I/O Mode. PLC I/O Mode requires a PLC or external controller for operation. ZPA Mode *ERSC* can interface with a PLC or external controller to report local module status and provide local zone interaction.

The following chart provides a Quick Reference for the operational modes:

ERSC Control Strategy Quick Reference		
Any Mode	Read Status	<ul style="list-style-type: none"> Available to any Modbus TCP or Ethernet I/P networked device regardless of Module Mode Read Sensor & Control port inputs Read MDR diagnostics
ZPA Mode	Default	<ul style="list-style-type: none"> Out of the Box No PLC required Hard-wired connections available to control zone stop/release Some electrical connections may require RJ-12 to terminal breakout module for signal compatibility See User's Guide for connection details
	Interface PLC	<ul style="list-style-type: none"> Ethernet I/P or Modbus TCP Networks supported PLC can control over the network: zone accumulate/release, tracking data read/write, zone speed, monitor sensor & control port inputs See page 19 for I/O register details See page 31 for Examples
	Interface PLC + Port Outputs	<ul style="list-style-type: none"> All functions available with Default configuration All functions available with Interface PLC Using <i>EasyRoll</i>, allow PLC access to Control Port digital output signal See page 31 for Examples
PLC I/O Mode	Basic PLC	<ul style="list-style-type: none"> Requires Mode change in <i>EasyRoll</i> Suspends ALL ZPA functionality PLC has complete access and control of Sensor Port Inputs and Control Port Inputs and Output PLC has complete control of both MDR ports (speed, accel/decel, open/closed loop, braking method, etc.) See page 50 for setting PLC I/O Mode See page 53 Input register descriptions See page 57 for Output register descriptions See page 63 for servo movement example
	Full PLC	<ul style="list-style-type: none"> All functions with Basic PLC Requires PLC to configure One or both MDR ports can be changed to provide up to 3 high powered digital outputs (1A each) per MDR port for a total of 6 digital outputs available See page 67 for details



CONVEYLINX NETWORK ARCHITECTURE

Each *ERSC* Communicates to its adjacent modules and to any connected PC or PLC via Ethernet physical media. *ConveyLinx* modules recognize (2) TCP/IP based protocols: Modbus TCP 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 also recognized by *ConveyLinx* modules and any given *ERSC* Can be attached to an Ethernet I/P capable PLC (Allen-Bradley ControlLogix or CompactLogix platforms) and be recognized as a “generic Ethernet I/P device”. Both protocols access the same internal data locations on a given *ERSC*.

The logical addressing of these internal data locations is identical for each *ERSC*. The on-board *ERSC* Communication and control processes attach logical meanings to each data location and read and write data to specific locations to initiate and/or react to events. Certain data locations contain information as to how the *ERSC* is configured (MDR type, speed, direction, etc.) for its local controls. Other data locations are used for inter-module communications for conveyor operation. For example, when an upstream *ERSC* has a Carton ready to discharge to its neighboring downstream *ERSC*, the upstream *ERSC* will write a specific value to a specific address in the downstream *ERSC*'s internal data memory. The downstream *ERSC*'s on board logic monitors this internal memory location 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 accessible memory locations, external control devices (PC's and PLC's) can easily interact with *ERSC* modules to monitor and control various points along the conveyor path.

MODBUS ASSEMBLY INSTANCE STRUCTURE

Each *ERSC* 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 *ERSC*'s memory structure contains a fixed array of internal data locations that are constructed as Modbus *Holding Registers*. Each *ERSC* 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 *ERSC*'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.

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 4 - Read Holding Register (up to 40 registers per instruction)
- Service Code 6 - Write Single Register
- Service Code 16 - Write Multiple Registers (up to 40 registers per instruction)
- Service Code 23 - Read/Write Multiple Registers (up to 40 registers per instruction)



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

For example, for Assembly #5, if you only need to read register 4:1504; your instruction in the 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 *ERSC* will return an error.



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



ETHERNET I/P ASSEMBLY INSTANCE STRUCTURE

When an *ERSC* 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 Insight Automation.

For this manual Ethernet I/P Instances are described using the Generic Ethernet I/O method of connection. Consult pulseroller.com for more information on obtaining EDS file(s) for incorporation into your RSLogix5000 development software.

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. The *ERSC* recognizes 3 specific instance types; one for each of the *ERSC* operational modes.

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

[ModuleName]:O.Data[Index].Bit

[ModuleName]:I.Data[Index].Bit



Where:

- **ModuleName** is the user-defined name of the device
- **“O.Data”** indicates the output image of the device
- **“I.Data”** indicates the input image of 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 instances essentially group the appropriate Modbus registers into contiguous Input and Output array image that fit into the Allen-Bradley Logix 5000 controller tags.

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

PROFINET IO ASSEMBLY INSTANCE STRUCTURE

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

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

Please refer to [Connecting ConveyLinx to Profinet PLCs – Insight Automation publication ERSC-1525](#) for details on how to connect ConveyLinx to Profinet I/O capable PLCs.

PLC CONTROLLER WITH ZPA MODE

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

NOTES ABOUT ASSEMBLIES FOR ERSC 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 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 address are different depending on which zone (or both) that need to be controlled

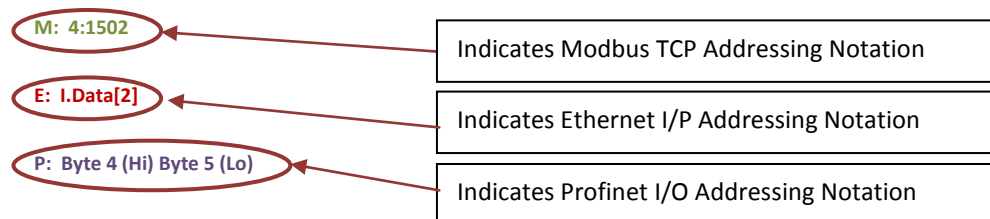


In general, when utilizing ZPA Mode registers; “upstream” and “downstream” registers are logically determined by conveyor flow after the system has been Auto-Configured and will not necessarily be associated with the *ERSC*’s physical “left” or “right” side’s connections. For motor and port specific 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.

REGISTER CHART LEGEND



PLC INPUTS FOR ZPA MODE (ASSEMBLY #5)

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

No.	Register Name / Module Address	Assembled Address for PLC	Description
5-0	Local Status Upstream Zone Forward Direction 4:0116	M: 4:1500 (Low Byte) E: I.Data[0] (Low Byte) P: Byte 1	Unsigned Value of Low Byte 0xXX01 = Zone sensor clear and motor stopped 0xXX02 = Zone sensor clear, motor running, accepting from upstream zone 0xXX04 = Zone sensor blocked, motor running, discharging to downstream zone 0xXX05 = Zone sensor blocked and motor stopped See <i>Assembly #5 Inputs Note 1</i> on page 23 for further details
	Local Status Upstream Zone Reverse Direction 4:0116	M: 4:1500 (High Byte) E: I.Data[0] (High Byte) P: Byte 0	
5-1	Local Status Downstream Zone Forward Direction 4:0196	M: 4:1501 (Low Byte) E: I.Data[1] (Low Byte) P: Byte 3	
	Local Status Downstream Zone Reverse Direction 4:0196	M: 4:1501 (High Byte) E: I.Data[1] (High Byte) P: Byte 2	
5-2	Arrival Count Local Upstream Zone 4:0106	M: 4:1502 E: I.Data[2] P: Byte 4 (Hi) Byte 5 (Lo)	Unsigned Integer Value <ul style="list-style-type: none"> • Increments by 1 each time a Carton arrives in the zone • Value rolls over from 65,535 back to 0
5-3	Departure Count Local Upstream Zone 4:0107	M: 4:1503 E: I.Data[3] P: Byte 6 (Hi) Byte 7 (Lo)	Unsigned Integer Value <ul style="list-style-type: none"> • Increments by 1 each time a Carton departs the zone • Value rolls over from 65,535 back to 0
5-4	Arrival Count Local Downstream Zone 4:0186	M: 4:1504 E: I.Data[4] P: Byte 8 (Hi) Byte 9 (Lo)	Unsigned Integer Value <ul style="list-style-type: none"> • Increments by 1 each time a Carton arrives in the zone • Value rolls over from 65,535 back to 0
5-5	Departure Count Local Downstream Zone 4:0187	M: 4:1505 E: I.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	Unsigned Integer Value <ul style="list-style-type: none"> • Increments by 1 each time a Carton departs the zone • Value rolls over from 65,535 back to 0

No.	Register Name / Module Address	Assembled Address for PLC	Description
5-6	Module Status Word 1 4:0088	M: 4:1506 E: I.Data[6] P: Byte 12 (Hi) Byte 13 (Lo)	<u>Bitwise Value - Read only</u> bit 0 = Module Reset Flag – 1 when module resets, 0 when PLC connected bit 1 = Reserved bit 2 = Over-Voltage – Module power supply has exceeded 30V bit 3 = Left Motor Error – bits 7 thru 15 indicate specific error bit 4 = Ethernet Connections NOT OK bit 5 = Upstream Jam Error bit 6 = Left Sensor Error bit 7 = Low Voltage Error – Module Power Supply less than 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 – MDR slower than 10% of selected speed bit 13 = Left Motor Stalled bit 14 = Left Motor Hall Sensor Error bit 15 = Left Motor Not Used
5-7	Module Status Word 2 4:0089	M: 4:1507 E: I.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	<u>Bitwise Value - Read only</u> bit 0 = Reserved bit 1 = Reserved bit 2 = Over-Voltage – Module power supply has exceeded 30V bit 3 = Right Motor Error bit 4 = Reserved bit 5 = Downstream Jam Error bit 6 = Right Sensor Error bit 7 = Low Voltage Error - Module Power Supply less than 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 – MDR slower than 10% of selected speed bit 13 = Right Motor Stalled bit 14 = Right Motor Hall Sensor Error bit 15 = Right Motor Not Used
5-8	Current Upstream Zone Tracking Word 1 4:0119	M: 4:1508 E: I.Data[8] P: Byte 16 (Hi) Byte 17 (Lo)	<u>When Carton is accumulated in UPSTREAM zone:</u> Value = Tracking data word #1 (16 bit integer) for the Carton currently accumulated and stopped in the module's Upstream zone.
5-9	Current Upstream Zone Tracking Word 2 4:0120	M: 4:1509 E: I.Data[9] P: Byte 18 (Hi) Byte 19 (Lo)	<u>When Carton is accumulated in UPSTREAM zone:</u> Value = Tracking data word #2 (16 bit integer) for the Carton currently accumulated and stopped in the module's Upstream zone.
5-10	Current Downstream Zone Tracking Word 1 4:0199	M: 4:1510 E: I.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	<u>When Carton is accumulated in DOWNSTREAM zone:</u> Value = Tracking data word #1 (16 bit integer) for the Carton currently accumulated and stopped in the module's Downstream zone.
5-11	Current Downstream Zone Tracking Word 2 4:0200	M: 4:1511 E: I.Data[11] P: Byte 22 (Hi) Byte 23 (Lo)	<u>When Carton is accumulated in DOWNSTREAM zone:</u> Value = Tracking data word #2 (16 bit integer) for the Carton currently accumulated and stopped in the module's Downstream zone.
5-12	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 <i>Instance 6 - Release Upstream</i> which can be used by PLC logic to confirm release count prior to writing new data to the <i>Instance 6 - Release Upstream</i> register

No.	Register Name / Module Address	Assembled Address for PLC	Description
5-13	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 <i>Instance 6 - Release Downstream</i> which can be used by PLC logic to confirm release count prior to writing new data to the <i>Instance 6 - Release Downstream</i> register
5-14	Get Tracking Forward Direction Word 1 4:0201	M: 4:1514 E: I.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	<u>When module is discharging to Non-ConveyLinX controlled conveyor:</u> 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.
5-15	Get Tracking Forward Direction Word 2 4:0202	M: 4:1515 E: I.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	<u>When module is discharging to Non-ConveyLinX controlled conveyor:</u> 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.
5-16	Get Tracking Reverse Direction Word 1 4:0237	M: 4:1516 E: I.Data[16] P: Byte 32 (Hi) Byte 33 (Lo)	<u>When module is discharging to Non-ConveyLinX controlled conveyor:</u> Value = Tracking data word #1 (16 bit integer) for the load that has just discharged from the local downstream zone when local conveyor is operating in opposite of default or "reverse" direction.
5-17	Get Tracking Reverse Direction Word 2 4:0238	M: 4:1517 E: I.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)	<u>When module is discharging to Non-ConveyLinX controlled conveyor:</u> 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.
5-18	Sensor 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 00 = Left Sensor Port - Pin 3 bit 01 = Left Hardware Port - Pin 3 bit 02 = Right Sensor Port - Pin 3 bit 03 = Right Hardware Port - Pin 3 bit 04 = Left Sensor Port - Pin 4 bit 05 = Left Hardware Port - Pin 4 bit 06 = Right Sensor Port - Pin 4 bit 07 = Right Hardware Port - Pin 4 bit 08 thru bit 14 = Reserved bit 15 = 1 sec on / 1 sec off heartbeat See <i>Assembly #5 Inputs Note 2</i> on page 23
5-19	Module Reset Count	M: 4:1519 E: I.Data[19] P: Byte 38 (Hi) Byte 38 (Lo)	Reserved for use in Instance #25. Only applicable to EIP Installations
5-20	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 00 – bit04 = Reserved bit 05 = Stop active on other module in Stop Group bit 06 = Stop active due to lost communication connection bit 07 = Stop active due to lost PLC connection bit 08 = Stop active on local module's Left Control Port bit 09 = Stop active on local module's Right Control Port bit 10 = Stop active due to Stop Command from PLC bit 11- bit 15 = Reserved Refer to Insight Automation publication ERSC-1800 <i>ConveyStop User's Guide</i> for details on using <i>ConveyStop</i> .



ASSEMBLY #5 INPUTS NOTE 2

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.

ASSEMBLY #5 INPUTS NOTE 1

The values 0xXX01 thru 0xXX05 are shown because these are the possible logical values used for inter-module communication. External networked devices (PLC or PC) utilizing either Modbus TCP or Ethernet I/P to monitor 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 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.

PLC OUTPUTS FOR ZPA MODE (ASSEMBLY #6)

These registers are written by the PLC to the *Module* when the module is in ZPA mode. All registers are 16-bit Integer data type.

No.	Register Name	PLC Address	Description
6-0	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. See <i>Assembly #6 Outputs Note 1</i> on page 27 for special reserved values
6-1	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. See <i>Assembly #6 Outputs Note 1</i> on page 27 for special reserved values
6-2	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. See <i>Assembly #6 Outputs Note 1</i> on page 27 for special reserved values
6-3	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. See <i>Assembly #6 Outputs Note 1</i> on page 27 for special reserved values
6-4	Accumulation Control for Local Upstream Zone 4:0104	M: 4:1604 E: O.Data[4] P: Byte 8 (Hi) Byte 9 (Lo)	<u>Bitwise Values</u> bit 0 = Set/Clear Accumulation Mode for Local Zone bit 8 = Accumulate adjacent upstream zone bit 9 = Set Arrival Confirmation for adjacent downstream zone bit 10 = Jog zone in default direction bit 11 = Jog zone in opposite of default direction bit 12 = Wake up Local Zone All other bits reserved See <i>Assembly #6 Outputs Note 3</i> on page 28 for further details
6-5	Accumulation Control for Local Downstream Zone 4:0184	M: 4:1605 E: O.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	<u>Bitwise Values</u> bit 0 = Set/Clear Accumulation Mode for Local Zone bit 8 = Accumulate adjacent upstream zone bit 9 = Set Arrival Confirmation for adjacent downstream zone bit 10 = Jog zone in default direction bit 11 = Jog zone in opposite of default direction bit 12 = Wake up Local Zone All other bits reserved See <i>Assembly #6 Outputs Note 3</i> on page 28 for further details
6-6	Set Left MDR 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%



No.	Register Name	PLC Address	Description
6-7	Set Right MDR Speed 4:0064	M: 4:1607 E: O.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	0 = Run at Configured Speed See <i>Assembly #6 Outputs Note 2</i> on page 27 for further details
6-8	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)	<u>When bit 0 of Accumulation Control for Local Upstream Zone is set:</u> 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
6-9	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)	<u>When bit 0 of Accumulation Control for Local Downstream Zone is set:</u> 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
6-10	Set Status for Upstream Induct 4:0134	M: 4:1610 E: O.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinX controlled conveyor:</u> 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 <i>Set Induct Tracking Word 1 / Word 2</i>
6-11	Set Status for Downstream Discharge 4:0232	M: 4:1611 E: O.Data[11] P: Byte 22 (Hi) Byte 23 (Lo)	<u>Only used when local module is discharging loads to Non-ConveyLinX controlled conveyor:</u> <ul style="list-style-type: none"> 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
6-12	Set Induct Tracking Forward Direction Word 1 4:0139	M: 4:1612 E: O.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinX controlled conveyor:</u> Set value for 16 bit integer tracking word #1 for Carton leaving non-ConveyLinX controlled conveyor that is in transit to arrive on the local upstream zone when conveyor is running in default or "forward" direction.
6-13	Set Induct Tracking Forward Direction Word 2 4:0140	M: 4:1613 E: O.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinX controlled conveyor:</u> Set value for 16 bit integer tracking word #2 for Carton leaving non-ConveyLinX controlled conveyor that is in transit to arrive on the local upstream zone when conveyor is running in default or "forward" direction.
6-14	Set Induct Tracking Reverse Direction Word 1 4:0237	M: 4:1614 E: O.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	<u>Only used when local module is accepting loads from Non-ConveyLinX controlled conveyor:</u> Set value for 16 bit integer tracking word #1 for Carton leaving non-ConveyLinX controlled conveyor that is in transit to arrive on the local upstream zone when conveyor is running in opposite of default or "reverse" direction.

No.	Register Name	PLC Address	Description
6-15	Set Induct Tracking Forward Direction Word 2 4:0238	M: 4:1615 E: O.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	<p><u>Only used when local module is accepting loads from Non-ConveyLinX controlled conveyor:</u></p> <p>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.</p>
6-16	Clear Motor Error 4:0022	M: 4:1616 E: O.Data[16] P: Byte 32 (Hi) Byte 33 (Lo)	<p>Logical 0 or 1</p> <p>0 = Stop Reset 1 = Send Reset</p> <p>See <i>Assembly #6 Outputs Note 4</i> on page 29 for further details</p>
6-17	Set Control Port Outputs 4:0063	M: 4:1617 E: O.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)	<p><u>Bitwise Value – "1" Energizes Output</u></p> <p>bit 1 = Left Control Port bit 3 = Right Control Port</p> <p>See <i>Assembly #6 Outputs Note 5</i> on page 29 for further details</p>
6-18	Module Reset Counter	M: 4:1618 E: O.Data[18] P: Byte 36 (Hi) Byte 37 (Lo)	Reserved for Set Module Reset Counter for Assembly #26
6-19	ConveyStop Command 4:0020	M: 4:1619 E: O.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)	<p>Integer Value</p> <p>1 = Command local module's Stop Group to go to Stopped State 2 = Command local module's Stop Group to Clear Stopped State</p> <p>Refer to Insight Automation publication ERSC-1800 <i>ConveyStop User's Guide</i> for details on using <i>ConveyStop</i>.</p>
6-20	Clear Sensor Jam Command for Local Upstream Zone 4:0109	M: 4:1620 E: O.Data[20] P: Byte 40 (Hi) Byte 41 (Lo)	<p><u>Use when PLC has detected a local Upstream jam in Module Status Word #1 Bit 5:</u></p> <p>PLC creates transition from 0 to 1 to send command to local upstream zone to clear the jam condition.</p> <p>See <i>Assembly #6 Outputs Note 6</i> on page 29</p>
6-21	Clear Sensor Jam Command for Local Downstream Zone 4:0189	M: 4:1621 E: O.Data[21] P: Byte 42 (Hi) Byte 43 (Lo)	<p><u>Use when PLC has detected a local Downstream jam in Module Status Word #2 Bit 5:</u></p> <p>PLC creates transition from 0 to 1 to send command to local downstream zone to clear the jam condition.</p> <p>See <i>Assembly #6 Outputs Note 6</i> on page 29</p>
6-22	Direction & Accumulation Mode Control for Local Upstream Zone 4:0365	M: 4:1622 E: O.Data[22] P: Byte 44 (Hi) Byte 45 (Lo)	<p>Used to change direction of flow or accumulation mode for a contiguous group of zones beginning with the local upstream / downstream zone</p> <p>Value for Low Byte of Register</p> <p>0 = Normal Function 1 = Accumulate Zones</p>

No.	Register Name	PLC Address	Description
6-23	Direction & Accumulation Mode Control for Local Upstream Zone 4:0375	M: 4:1623 E: O.Data[23] P: Byte 46 (Hi) Byte 47 (Lo)	2 = Accumulate Zones 3 = Change Accumulation Release mode 4 = Return Release Mode to Configured Default 10 = Set Direction to Configured Default Direction (Forward) 11 = Set Direction to opposite of Configured Direction (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".

ASSEMBLY #6 OUTPUTS NOTE 1

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 ERSC's Assembly #6 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.

ASSEMBLY #6 OUTPUTS NOTE 2

Leaving these registers at "0" will instruct the ERSC to use its configured speed. Any non-zero value will instruct the ERSC 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 ERSC will use its configured speed as reference.

ASSEMBLY #6 OUTPUTS NOTE 3



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 repositioned 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.



Bit 12 – Wake up Local Zone functionality available in ERSC firmware versions 4.0 and higher.

ASSEMBLY #6 OUTPUTS NOTE 4

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.



External controller 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.

ASSEMBLY #6 OUTPUTS NOTE 5

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.



Please refer to the *ConveyLinX User’s Guide* for electrical connection details before using PLC control for Control Port outputs.

ASSEMBLY #6 OUTPUTS NOTE 6

Refer to *ConveyLinX User’s Guide* and/or *EasyRoll* software on screen help for description of various jam conditions. 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

ZPA MODE EXAMPLES

EXAMPLE 1 – BASIC ACCUMULATE AND RELEASE WITH TRACKING DATA

Figure 2 shows a typical arrangement of an upstream or downstream zone on a *Module* 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.

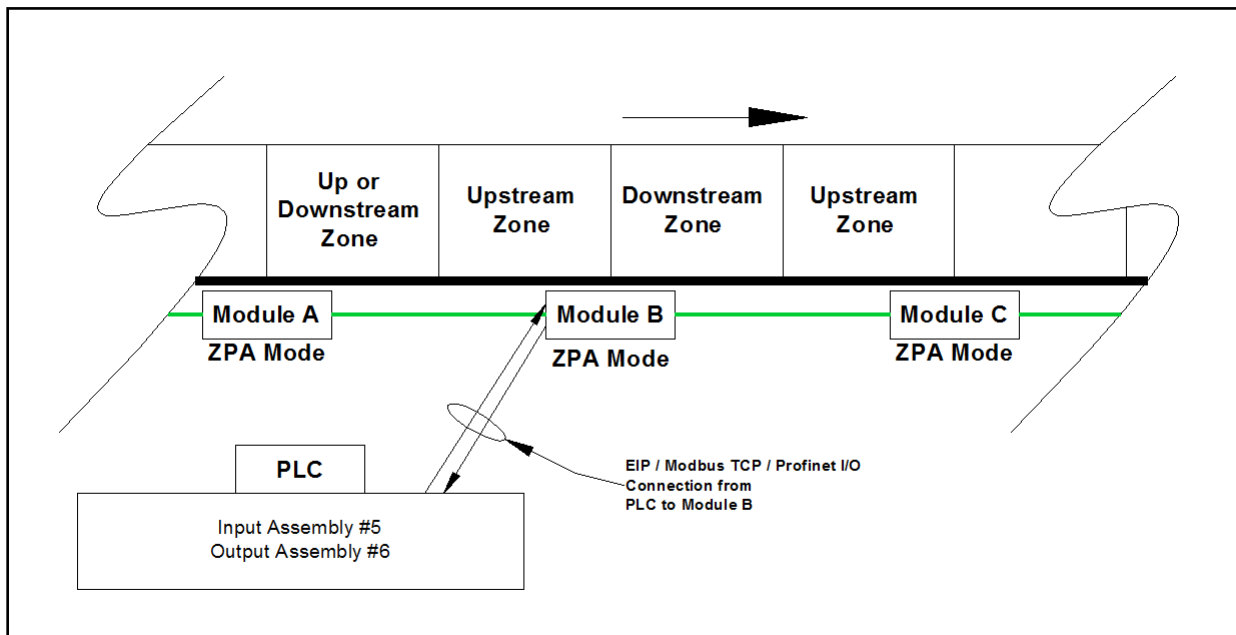


FIGURE 2 - BASIC ACCUMULATE / RELEASE CONTROL EXAMPLE

For this example, the PLC must establish an appropriate connection to *Module B* utilizing Input Assembly #5 and Output Assembly #6 as shown in Figure 2.

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 (6-4) to instruct this zone to accumulate any Carton that arrives
2. Monitor *Arrival Count Local Upstream Zone* (5-2) and *Departure Count Local Upstream Zone* (5-3) 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* (5-8) and *Current Upstream Zone Tracking Word 2* (5-9) 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* (6-0) and *Set Local Upstream Zone Tracking Word 2* (6-1)
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 (5-12), add 1 to this value, and then write this new value to *Release and Accumulate on Next Arrival for Local Upstream Zone* register (6-8). 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* (5-2) and *Departure Count Local Upstream Zone* (5-3) 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 (6-4). 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

Downstream Zone example is identical to Upstream Zone example shown above except the registers used are different. It is repeated below to explicitly show which registers to use.

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

1. Set bit 0 in *Accumulation Control for Local Downstream Zone* register (6-5) to instruct this zone to accumulate any Carton that arrives.
2. Monitor *Arrival Count Local Downstream Zone* (5-4) and *Departure Count Local Downstream Zone* (5-5) 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 Downstream Zone Tracking Word 1* (5-10) and *Current Downstream Zone Tracking Word 2* (5-11) 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* (6-2) and *Set Local Downstream Zone Tracking Word 2* (6-3)
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 Downstream Zone* register (5-13), add 1 to this value, and then write this new value to *Release and Accumulate on Next Arrival for Local Downstream Zone* register (6-9). When *Module B* sees this new value in this register, it will release the Carton in the downstream 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 Downstream Zone* (5-4) and *Departure Count Local Downstream Zone* (5-5) 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 (6-5). 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, 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.

EXAMPLE 2 – BASIC BAR CODE READER

This example shows how to easily set up the conveyor control to easily singulate cartons through a bar code scanning region. *Figure 3* shows a conveyor and *Module* configuration for a simple barcode scanning example.

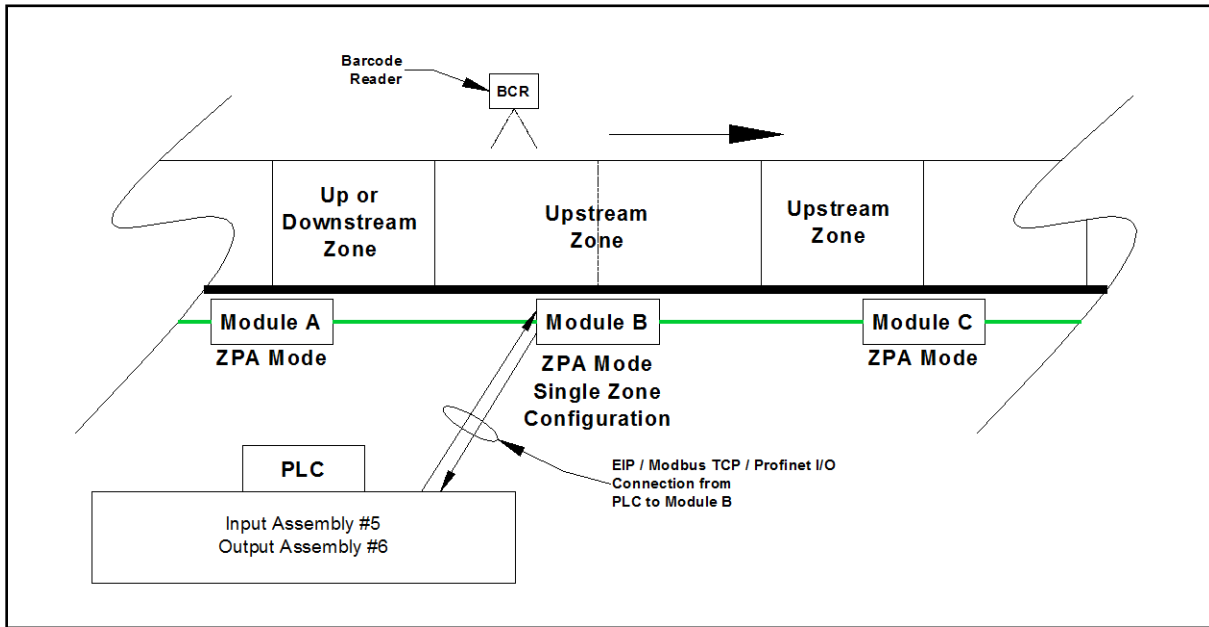


FIGURE 3 - SIMPLE BARCODE SCANNER APPLICATION

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. Please refer to the *ConveyLinx User's Guide* for details on single zone operation.

Also for this example to work properly; the discharging zone of *Module A* and the single zone of *Module B* must be in the default singulation release mode (not Train mode). *Module A* must also not have its Arrival Jam disabled from *EasyRoll* main screen. Please refer to the *ConveyLinx User's Guide* and/or *EasyRoll* software on screen help for details on these items.

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.

EXAMPLE 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. *Figure 4* shows a typical example configuration.

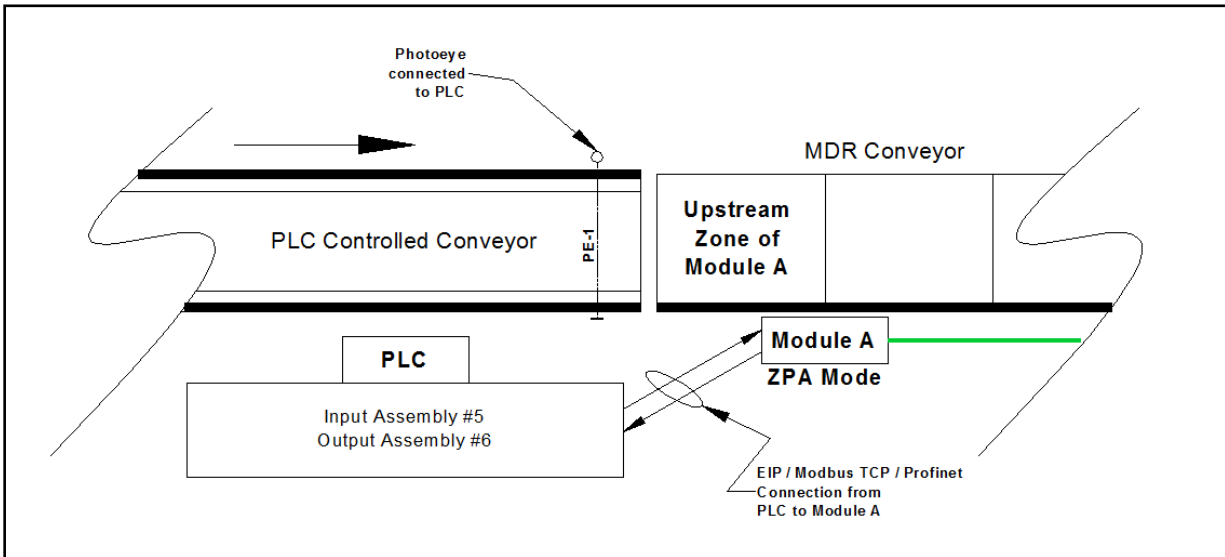


FIGURE 4 - UPSTREAM ACCEPT INTERLOCK EXAMPLE

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 (6-10). 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 / 2* (6-12 / 6-13).
3. When Carton clears *PE-1*, write a “1” to *Set Status for Upstream Induct* register (6-10). When *Module A* sees this register change to “1”, it knows that the tracking data written to *Set Induct Tracking Forward Word 1 / 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 than 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 insure single Carton induction to the local upstream zone, the PLC programmer must insure proper use of the *Set Status for Upstream Induct* register.

EXAMPLE 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. *Figure 5* shows a typical example.

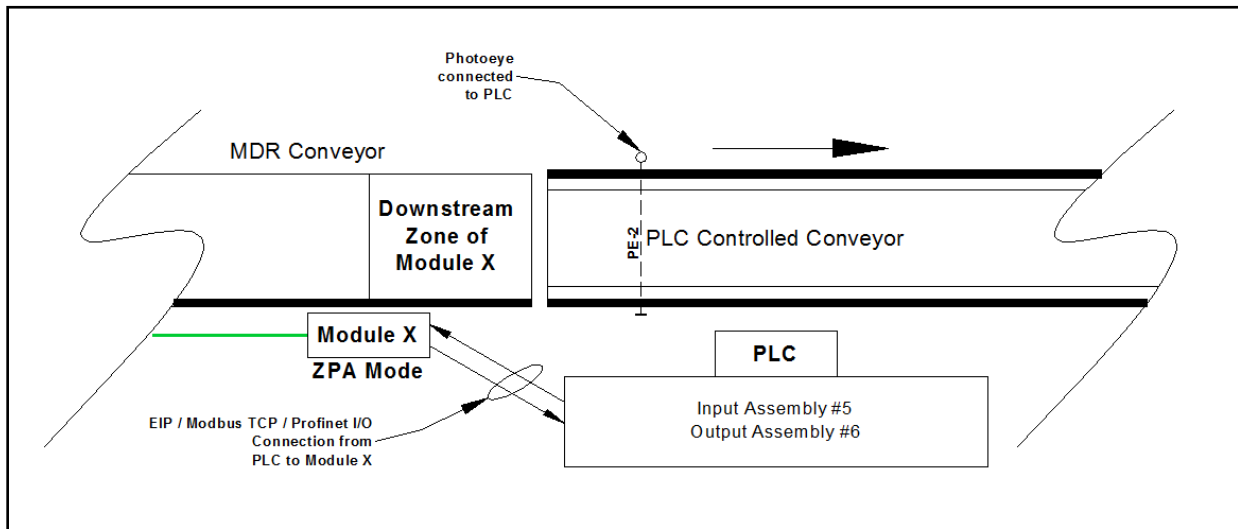


FIGURE 5 - DOWNSTREAM DISCHARGE INTERFACE EXAMPLE

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 (6-11). 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 (6-11). 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 1/2* registers (5-14 and/or 5-15).
4. When Carton arrives at *PE-2*, the PLC writes a “5” to *Set Status for Downstream Discharge* register (6-11). 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* (6-5) and this will cause the adjacent upstream zone to accumulate (whether this zone is on *Module X* or next adjacent upstream ConveyLinx module).



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 (6-4).

To return the accumulated upstream zone to normal operation; the PLC simply resets bit 8 in this register



Functionality for Accumulate Upstream using bit 8 is only available in ConveyLinx-Ai firmware versions 3.45 and higher.

EXAMPLE 5 – SIMPLE DIVERT EXAMPLE

This example shows how to use a PLC to control a simple divert mechanism to divert a Carton from a ZPA zone and capture its tracking. This example also illustrates how to use the set downstream arrival function with the PLC to keep the discharging Module from generating a jam condition when the Carton is moved from the ZPA zone to the divert lane. *Figure 6* shows the layout for this example.

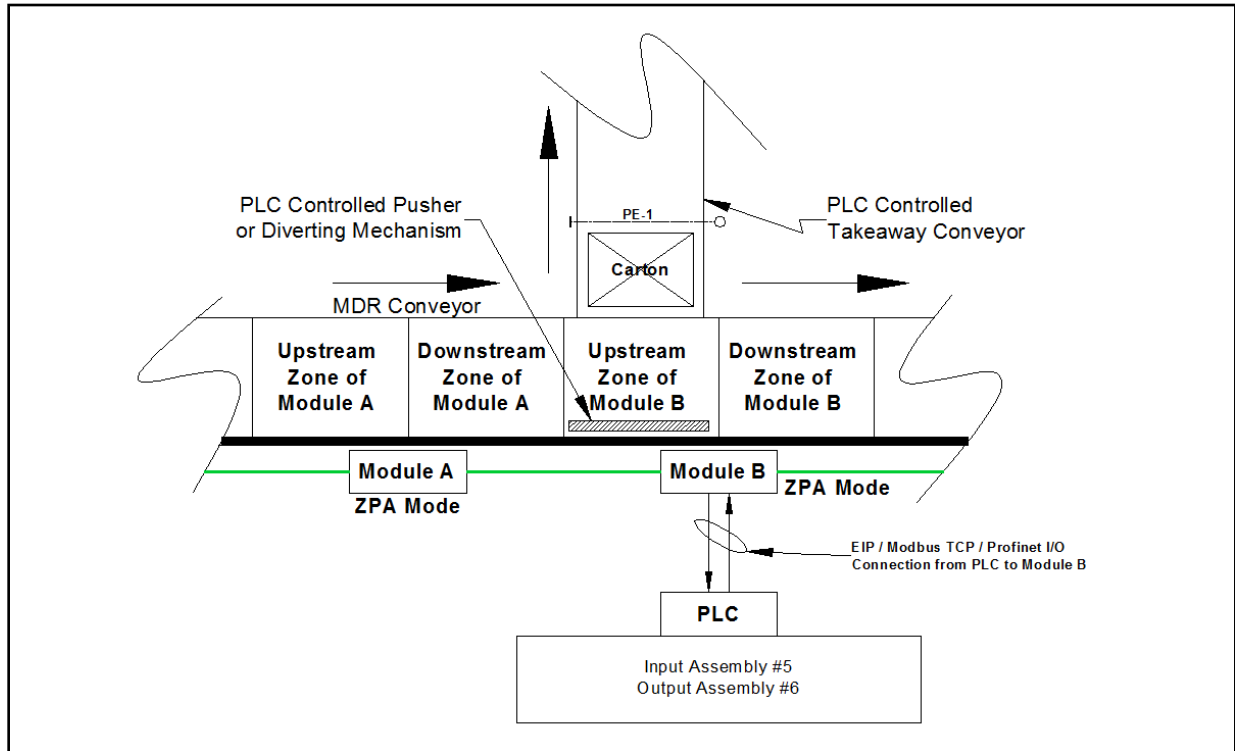


FIGURE 6 - SIMPLE DIVERT EXAMPLE

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. It is not required that the PLC use Carton tracking data be used to determine when divert a Carton.

1. Set bit 0 in *Accumulation Control for Local Upstream Zone* register (6-4) to instruct this zone to accumulate any Carton that arrives.
2. Monitor *Arrival Count Local Upstream Zone* (5-2) and *Departure Count Local Upstream Zone* (5-3) 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* (5-8) and *Current Upstream Zone Tracking Word 2* (5-9) will be valid for the newly arrived Carton.

Assuming the PLC determines that the newly arrived Carton needs to divert:



- Set bit 8 of *Accumulation Control for Local Upstream Zone* register (6-4). This will keep *Module A* from releasing a Carton to *Module B* while the divert operation is in progress.
- PLC will then initiate its divert mechanism.
- When carton reaches PE-1, the PLC will then set bit 9 of the *Accumulation Control for Local Upstream Zone* register (6-4). This will tell *Module B* that the carton “successfully arrived at its downstream position”.
- Once carton has cleared PE-1, then the PLC should clear bit 9 of the *Accumulation Control for Local Upstream Zone* register (6-4).
- PLC then can clear bit 8 of the *Accumulation Control for Local Upstream Zone* register (6-4) to instruct *Module A* that it is OK to allow the next carton to enter *Module B*'s upstream zone.



Please note that in this example, the setting of bit 9 in the *Accumulation Control for Local Upstream Zone* register can simply be written in the PLC logic based upon block and clear of PE-1. Care must be taken so that *Module B*'s Upstream zone's sensor is not blocked at the same time this bit 9 is set. If this zone's sensor is blocked at the same time bit 9 is set; *Module B* will assume that the carton is more than one zone in length and will automatically initiate Flex Zone operation which may produce unexpected results in carton flow. Please refer to User's Guide for complete description of Flex Zone function and how to disable if needed.

If PLC determines that the Carton needs to go straight and not divert:

The PLC can simply modify Carton's tracking data if desired and release the Carton as described in section *ZPA Mode Examples* on page 31.

EXAMPLE 6 – MERGE ONTO ZPA MAIN LINE

This example shows how to perform a simple merge with only a single *Module* connection to the PLC. Figure 7 depicts the conveyor layout and *Module* configuration for this example. In the example, *Module B* could be eliminated by simply making the shaded merging area be 2 extended zones on *Module C*. This example shows how easy it is to put pieces together with *EasyRoll* and using the *Extensions* functionality. Please refer to *ConveyLinX User's Guide* and *EasyRoll* software on screen directions for more details on creating extension zones.

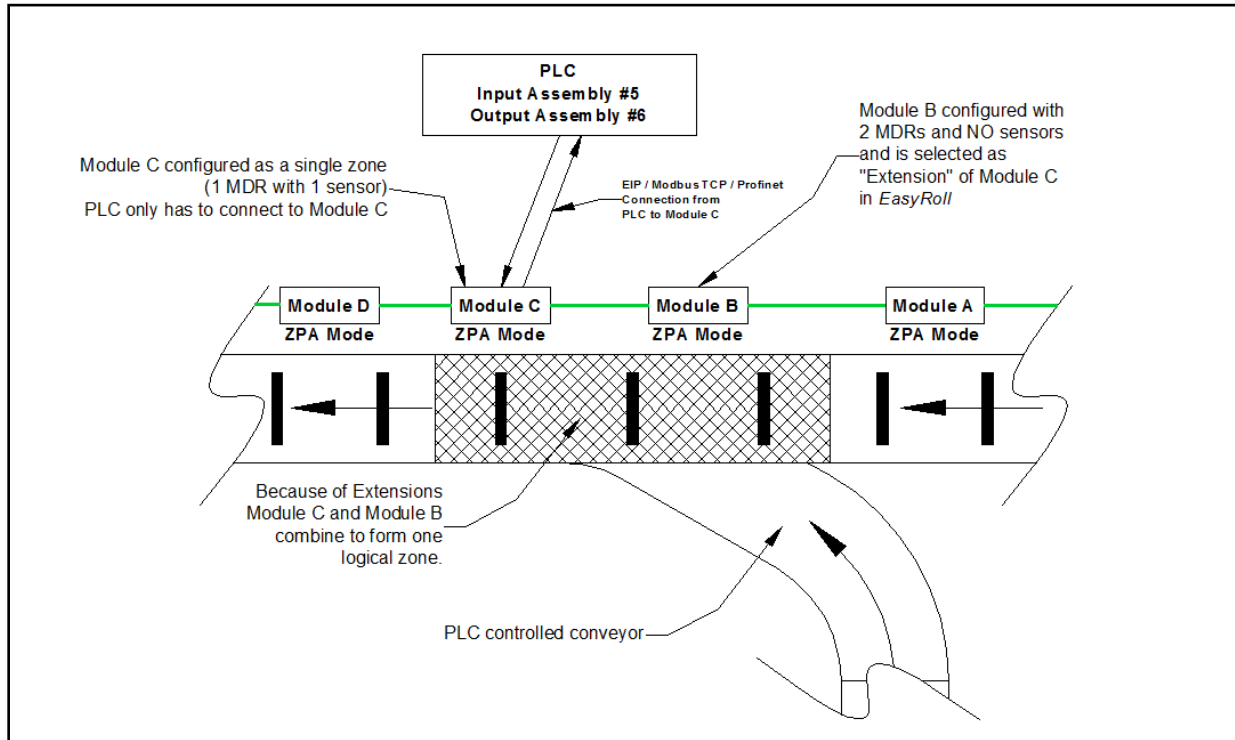


FIGURE 7 - MERGE ONTO ZPA MAIN LINE EXAMPLE

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 *ConveyLinX-Ai 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* register (5-0). When 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 (6-4) 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 merging curve conveyor, the PLC sets bit 12 in the *Accumulation Control for Local Upstream Zone* register (6-4) 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 *Local Status Upstream Zone* register (5-0) 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 (6-4). Either of these values will indicate a successful arrival at *Module C*.
6. Once bit 12 of the *Accumulation Control for Local Upstream Zone* register (6-4) has been reset, the PLC can reset bit 8 of the same register to indicate to *Module A* that it is OK to continue normal ZPA function.

MORE EXAMPLES

Visit our website at www.pulseroller.com for additional examples including sample PLC programs.



ETHERNET I/P ZPA MODE WITH PLC ENABLE



Please note that this functionality is only applicable for Ethernet I/P PLC. Function is not applicable or used for Modbus TCP nor Profinet I/O PLCs.

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 in ZPA mode 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* 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.

Ethernet I/P Input Instance 25 and Output Instance 26 are used for applications where the *ERSC* remains in a "hold" state until the PLC has established communications. Otherwise, the register mapping for Input Instance 25 is identical to Input Instance 5 and Output Instance 2 is identical to Output Instance 6.

ASSEMBLY #25 INPUTS FOR ZPA MODE WITH PLC ENABLE

You use Ethernet I/P Instance 25 to access the INPUT data from a given ERSC module when it is in ZPA Mode

No.	Register Name	Logix Tag Address	Description
25-0	Local Status Upstream Zone	[Name]:I.Data[0]	Same as Instance 5
25-1	Local Status Downstream Zone	[Name]:I.Data[1]	Same as Instance 5
25-2	Arrival Count Local Upstream Zone	[Name]:I.Data[2]	Same as Instance 5
25-3	Departure Count Local Upstream Zone	[Name]:I.Data[3]	Same as Instance 5
25-4	Arrival Count Local Downstream Zone	[Name]:I.Data[4]	Same as Instance 5
25-5	Departure Count Local Downstream Zone	[Name]:I.Data[5]	Same as Instance 5
25-6	Module Status Word 1	[Name]:I.Data[6]	Same as Instance 5
25-7	Module Status Word 2	[Name]:I.Data[7]	Same as Instance 5
25-8	Current Upstream Zone Tracking Word 1	[Name]:I.Data[8]	Same as Instance 5
25-9	Current Upstream Zone Tracking Word 2	[Name]:I.Data[9]	Same as Instance 5
25-10	Current Downstream Zone Tracking Word 1	[Name]:I.Data[10]	Same as Instance 5
25-11	Current Downstream Zone Tracking Word 2	[Name]:I.Data[11]	Same as Instance 5
25-12	Current Release Count for Upstream Zone	[Name]:I.Data[12]	Same as Instance 5
25-13	Current Release Count for Downstream Zone	[Name]:I.Data[13]	Same as Instance 5
25-14	Get Tracking Forward Direction Word 1	[Name]:I.Data[14]	Same as Instance 5
25-15	Get Tracking Forward Direction Word 2	[Name]:I.Data[15]	Same as Instance 5
25-16	Get Tracking Reverse Direction Word 1	[Name]:I.Data[16]	Same as Instance 5
25-17	Get Tracking Reverse Direction Word 2	[Name]:I.Data[17]	Same as Instance 5
25-18	Sensor & Control Port Inputs	[Name]:I.Data[18]	Same as Instance 5
25-19	Current Module Reset Counter	[Name]:I.Data[19]	Value in this register must be written to Set Module Reset Counter to allow the ConveyLinX-Ai to process Output Data from PLC
25-20	ConveyStop Status	[Name]:I.Data[20]	Same as Instance 5



ASSEMBLY #26 – OUTPUTS FOR ZPA MODE WITH PLC ENABLE

You use Ethernet I/P Instance Assembly 26 to write OUTPUT data to a given ERSC module when it is in ZPA Mode.

No.	Register Name	Logix Tag Address	Description
26-0	Set Local Upstream Zone Tracking Word 1	[Name]:O.Data[0]	Same as Instance 6
26-1	Set Local Upstream Zone Tracking Word 2	[Name]:O.Data[1]	Same as Instance 6
26-2	Set Local Downstream Zone Tracking Word 1	[Name]:O.Data[2]	Same as Instance 6
26-3	Set Local Downstream Zone Tracking Word 2	[Name]:O.Data[3]	Same as Instance 6
26-4	Accumulation Control for Local Upstream Zone	[Name]:O.Data[4]	Same as Instance 6
26-5	Accumulation Control for Local Downstream Zone	[Name]:O.Data[5]	Same as Instance 6
26-6	Set Left MDR Speed	[Name]:O.Data[6]	Same as Instance 6
26-7	Set Right MDR Speed	[Name]:O.Data[7]	Same as Instance 6
26-8	Release and Accumulate on Next Arrival for Local Upstream Zone	[Name]:O.Data[8]	Same as Instance 6
26-9	Release and Accumulate on Next Arrival for Local Downstream Zone	[Name]:O.Data[9]	Same as Instance 6
26-10	Set Status for Upstream Induct	[Name]:O.Data[10]	Same as Instance 6
26-11	Set Status for Downstream Discharge	[Name]:O.Data[11]	Same as Instance 6
26-12	Set Induct Tracking Forward Direction Word 1	[Name]:O.Data[12]	Same as Instance 6
26-13	Set Induct Tracking Forward Direction Word 2	[Name]:O.Data[13]	Same as Instance 6
26-14	Set Induct Tracking Reverse Direction Word 1	[Name]:O.Data[14]	Same as Instance 6
26-15	Set Induct Tracking Forward Direction Word 2	[Name]:O.Data[15]	Same as Instance 6
26-16	Clear Motor Error	[Name]:O.Data[16]	Same as Instance 6
26-17	Set Control Port Outputs	[Name]:O.Data[17]	Same as Instance 6
26-18	Set Module Reset Counter	[Name]:O.Data[18]	When ConveyLinX-Ai detects that the value in this register equals the value in <i>Current Module Reset Counter</i> , the ConveyLinX-Ai will accept updates sent by the PLC for Output Instance 26
26-19	ConveyStop Command	[Name]:O.Data[19]	Same as Instance 6
26-20	Clear Sensor Jam Command for Local Upstream Zone	[Name]:O.Data[20]	Same as Instance 6
26-21	Clear Sensor Jam Command for Local Downstream Zone	[Name]:O.Data[21]	Same as Instance 6
26-22	Direction & Accumulation Mode Control for Local Upstream Zone	[Name]:O.Data[22]	Same as Instance 6
26-23	Direction & Accumulation Mode Control for Local Upstream Zone	[Name]:O.Data[23]	Same as Instance 6

PLC ENABLE USAGE

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

- PLC moves the value in the *Current Module Reset Counter* register ([ModuleName]:I.Data[19]) to the *Set Module Reset Counter* register ([ModuleName]:O.Data[18]).
- When the ERSC detects that the value in *Set Module Reset Counter* register ([ModuleName]:O.Data[18]) is equal to the value in the *Current Module Reset Counter* register ([ModuleName]:I.Data[19]) ; the ERSC will respond to data being written by the PLC to the remaining output instance 26 registers.

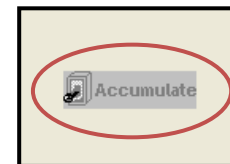


Please note that when the value in *Current Module Reset Counter* register is not equal to the value in the *Set Module Reset Counter* register; the values in the Input Instance 25 registers will updated by the ERSC and will be valid. In this state even though the PLC may be writing data to Output Instance 26, the ERSC will ignore it.



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

Be sure to click the “Accumulate” switch on the main *EasyRoll* screen to cause the zone to accumulate upon power up



When a zone is commanded to accumulate with a PLC and you happen to connect to that ERSC 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 ERSC. 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.



PLC CONTROLLER WITH PLC I/O MODE

When an *ERSC* is in PLC I/O mode, all automatic functions of detecting loads and running motors are suspended by the local *ERSC* 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 *ERSC* 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 E-Stop motor outputs



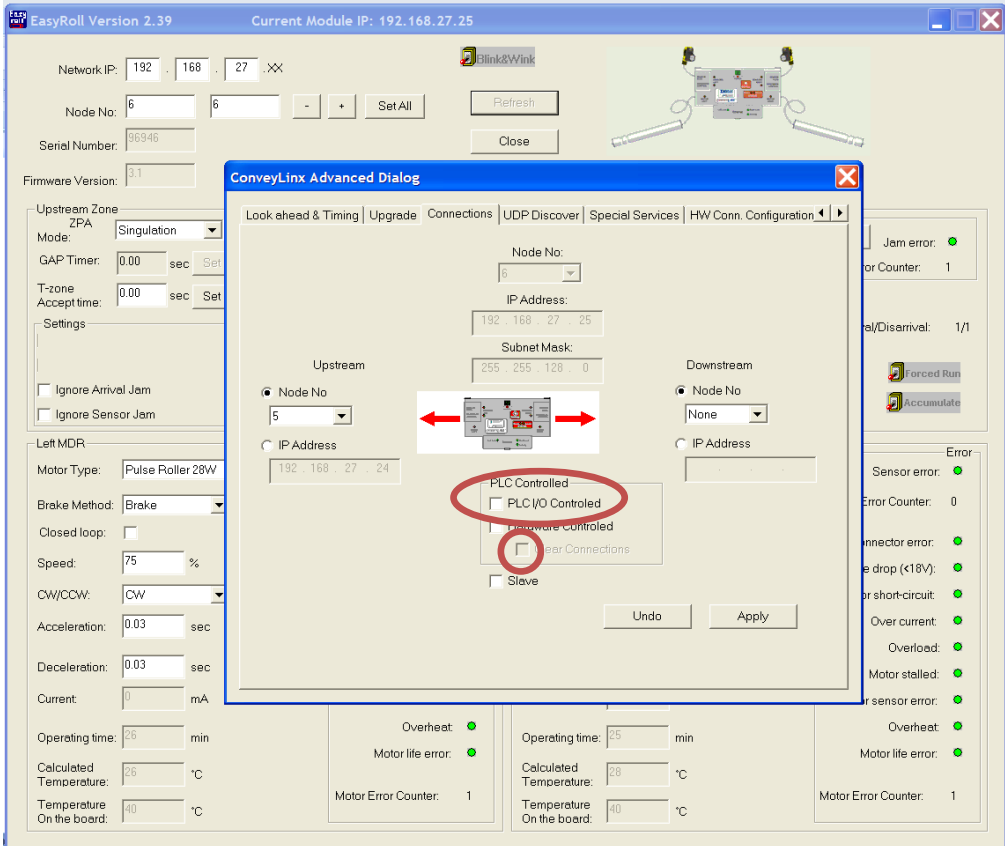
Refer to *ConveyLinx User's Guide* (publication *ERSC-1000*) for connection details for Sensor and Control Port input and output signals.

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.

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.

Configuring *ERSC* for PLC I/O Mode



The screenshot displays the EasyRoll software interface. The main window shows network configuration for a module with IP 192.168.27.25. A 'ConveyLinX Advanced Dialog' is open, showing the 'Connections' tab. In this dialog, the 'Node No.' is set to 6 and the 'IP Address' is 192.168.27.25. The 'Subnet Mask' is 255.255.128.0. Under the 'Upstream' section, 'Node No.' is set to 5. Under the 'Downstream' section, 'Node No.' is set to 'None'. The 'PLC I/O Controlled' checkbox is checked and circled in red. The 'Clear Connections' checkbox is also checked. The background shows the main EasyRoll interface with various motor settings and error indicators.

From the main screen, first enter the correct *Subnet* into the “Network IP” boxes and the correct *Node* you want to connect. Invoke the *ConveyLinX Advanced Dialog* and select the *Connections* tab.

Note that the *Node* is being viewed is in the center and it is greyed out. Select the “PLC I/O Controlled” checkbox. With this checked the “Clear Connections” checkbox becomes enabled. **Check or Uncheck the “Clear Connections” checkbox depending upon your application.** Click “Apply” to initiate the change. The *ERSC* will restart and this may take several seconds to complete.



OPTIONAL “CLEAR CONNECTIONS” CHOICE



Refer to [Appendix B – ConveyLinx Connections](#) for further details on ConveyLinx Connections.

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.

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.

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.

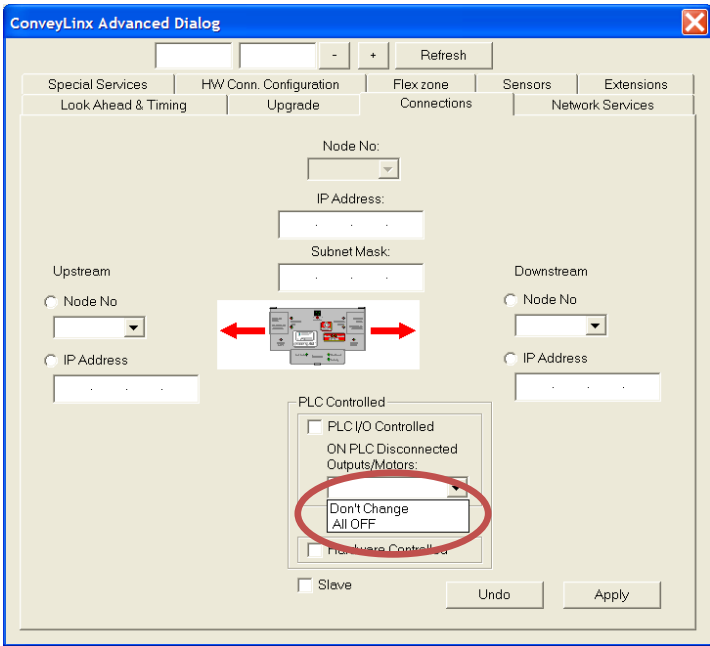


IMPORTANT NOTE: Once a given *ERSC* has been placed in PLC I/O mode, the ONLY way to return it to ZPA mode is to perform an Auto-Configuration procedure or Restore from a backup file with EasyRoll. There is no “undo” or “reset” function for this action.

CONFIGURING ACTION FOR LOSS OF COMMUNICATION

When changing the mode of a given *ERSC* to PLC I/O mode in *EastRoll*, you are given the option to select the behavior of the *ERSC*'s outputs upon loss of communications with the PLC.

Configuring Communication Loss Action for *ERSC* in PLC I/O Mode



The screenshot shows the 'ConveyLinX Advanced Dialog' window. It has several tabs: 'Special Services', 'HW Conn. Configuration', 'Flex zone', 'Sensors', and 'Extensions'. Under 'HW Conn. Configuration', there are sub-tabs for 'Look Ahead & Timing', 'Upgrade', 'Connections', and 'Network Services'. The 'Connections' sub-tab is active. The dialog contains fields for 'Node No.', 'IP Address', and 'Subnet Mask'. There are also sections for 'Upstream' and 'Downstream' with radio buttons for 'Node No' and 'IP Address'. A central graphic shows a PLC rack with red arrows pointing left and right. Below this is a 'PLC Controlled' section with a dropdown menu. The dropdown menu is open, showing 'Don't Change' and 'All OFF' options, which are circled in red. Other options include 'PLC I/O Controlled', 'ON PLC Disconnected Outputs/Motors', and 'Hardware Controlled'. There are 'Undo' and 'Apply' buttons at the bottom right.

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 *ERSC* 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 *ERSC* will automatically resume having its outputs and MDR's controlled by PLC command.



INPUTS FOR PLC I/O MODE (ASSEMBLY #7)

This chart shows the data sent by a Module in PLC I/O mode to the PLC when connected. All registers in the module are 16-bit integer data type.

No.	Register Name / Module Address	Address	Data Description
7-0	ConveyStop Status 4:0019	M: 4:1700 E: I.Data[0] P: Byte 0 (Hi) Byte 1 (Lo)	<p><u>Bitwise Value - Read only</u></p> <ul style="list-style-type: none"> bit 00 = Reserved bit 01 = Reserved bit 02 = Reserved bit 03 = Reserved bit 04 = Reserved bit 05 = Stop active on other module in Stop Group bit 06 = Stop active due to lost communication connection bit 07 = Stop active due to lost PLC connection bit 08 = Stop active on local module's Left Control Port bit 09 = Stop active on local module's Right Control Port bit 10 = Stop active due to Stop Command from PLC bit 11 = Reserved bit 12 = Reserved bit 13 = Reserved bit 14 = Reserved bit 15 = Reserved <p>Refer to Insight Automation publication ERSC-1800 <i>ConveyStop User's Guide</i> for details on using <i>ConveyStop</i>.</p>
7-1	Sensor Port Inputs 4:0035	M: 4:1701 E: I.Data[1] P: Byte 2 (Hi) Byte 3 (Lo)	<p><u>Bitwise Value - Read only</u></p> <ul style="list-style-type: none"> bit 00 = Left Sensor Port - Pin 3 bit 01 = Left Hardware Port - Pin 3 bit 02 = Right Sensor Port - Pin 3 bit 03 = Right Hardware Port - Pin 3 bit 04 = Left Sensor Port - Pin 4 bit 05 = Left Hardware Port - Pin 4 bit 06 = Right Sensor Port - Pin 4 bit 07 = Right Hardware Port - Pin 4 bit 08 thru bit 14 = Reserved bit 15 = 1 sec on / 1 sec off heartbeat <p>See on page 56</p>
7-2	Sensor Detect 4:0036	M: 4:1702 E: I.Data[2] P: Byte 4 (Hi) Byte 5 (Lo)	<p><u>Bitwise Value - Read only</u></p> <ul style="list-style-type: none"> bit 0 = Device is connected to Right Sensor Port bit 1 = Device is connected to Left Sensor Port
7-3	Module Voltage 4:0024	M: 4:1703 E: I.Data[3] P: Byte 6 (Hi) Byte 7 (Lo)	<p>Value in mV of Module Power Supply</p> <p>Range: 0 to 35000</p> <p><i>Example: 23500 = 23.5 Volts</i></p>
7-4	Left Motor Current 4:0055	M: 4:1704 E: I.Data[4] P: Byte 8 (Hi) Byte 9 (Lo)	<p>Integer Value in mA – Current that motor is currently drawing</p> <p><i>Example: 1900 = 1.9 Amps</i></p>
7-5	Left Motor Frequency 4:0056	M: 4:1705 E: I.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	<p>Integer Value in Hz – Current frequency that motor is currently running</p> <p><i>Example: 300 = 300 Hz</i></p> <p>See <i>Assembly #7 Inputs Note #2</i> on page 56</p>

No.	Register Name / Module Address	Address	Data Description		
7-6	Left Motor Temperature 4:0057	M: 4:1706 E: I.Data[6] P: Byte 12 (Hi) Byte 13 (Lo)	High Byte / Low Byte Value of temperatures in °C High Byte = Calculated motor temperature Low Byte = Temperature reading from on-board sensor		
7-7	Left Motor Status 4:0058	M: 4:1707 E: I.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	Bitwise Value – See <i>Assembly #7 Inputs Note #3</i> on page 56 <table border="1"> <tr> <td>bit 00 = Motor Status bit 01 = Motor Status bit 02 = Port in Digital Mode bit 03 = Reserved bit 04 = Reserved bit 05 = Reserved bit 06 = Over-Voltage bit 07 = Low Voltage</td> <td>bit 08 = Overheated bit 09 = Over Current bit 10 = Short Circuit bit 11 = Motor Not Connected bit 12 = Overloaded bit 13 = Motor Stalled bit 14 = Hall Sensor Error bit 15 = Motor Not Used</td> </tr> </table>	bit 00 = Motor Status bit 01 = Motor Status bit 02 = Port in Digital Mode bit 03 = Reserved bit 04 = Reserved bit 05 = Reserved bit 06 = Over-Voltage bit 07 = Low Voltage	bit 08 = Overheated bit 09 = Over Current bit 10 = Short Circuit bit 11 = Motor Not Connected bit 12 = Overloaded bit 13 = Motor Stalled bit 14 = Hall Sensor Error bit 15 = Motor Not Used
bit 00 = Motor Status bit 01 = Motor Status bit 02 = Port in Digital Mode bit 03 = Reserved bit 04 = Reserved bit 05 = Reserved bit 06 = Over-Voltage bit 07 = Low Voltage	bit 08 = Overheated bit 09 = Over Current bit 10 = Short Circuit bit 11 = Motor Not Connected bit 12 = Overloaded bit 13 = Motor Stalled bit 14 = Hall Sensor Error bit 15 = Motor Not Used				
7-8	Right Motor Current 4:0079	M: 4:1708 E: I.Data[8] P: Byte 16 (Hi) Byte 17 (Lo)	Integer Value in mA – Current that motor is currently drawing <i>Example: 1900 = 1.9 Amps</i>		
7-9	Right Motor Frequency 4:0080	M: 4:1709 E: I.Data[9] P: Byte 18 (Hi) Byte 19 (Lo)	Integer Value in Hz – Current frequency that motor is currently running <i>Example: 300 = 300 Hz</i> See <i>Assembly #7 Inputs Note #2</i> on page 56		
7-10	Right Motor Temperature 4:0081	M: 4:1710 E: I.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	High Byte / Low Byte Value of temperatures in °C High Byte = Calculated motor temperature Low Byte = Temperature reading from on-board sensor		
7-11	Right Motor Status 4:0082	M: 4:1711 E: I.Data[11] P: Byte 22 (Hi) Byte 23 (Lo)	Bitwise Value - See <i>Assembly #7 Inputs Note #3</i> on page 56 <table border="1"> <tr> <td>bit 00 = Motor Status bit 01 = Motor Status bit 02 = Port in Digital Mode bit 03 = Reserved bit 04 = Reserved bit 05 = Reserved bit 06 = Over-Voltage bit 07 = Low Voltage</td> <td>bit 08 = Overheated bit 09 = Over Current bit 10 = Short Circuit bit 11 = Motor Not Connected bit 12 = Overloaded bit 13 = Motor Stalled bit 14 = Hall Sensor Error bit 15 = Motor Not Used</td> </tr> </table>	bit 00 = Motor Status bit 01 = Motor Status bit 02 = Port in Digital Mode bit 03 = Reserved bit 04 = Reserved bit 05 = Reserved bit 06 = Over-Voltage bit 07 = Low Voltage	bit 08 = Overheated bit 09 = Over Current bit 10 = Short Circuit bit 11 = Motor Not Connected bit 12 = Overloaded bit 13 = Motor Stalled bit 14 = Hall Sensor Error bit 15 = Motor Not Used
bit 00 = Motor Status bit 01 = Motor Status bit 02 = Port in Digital Mode bit 03 = Reserved bit 04 = Reserved bit 05 = Reserved bit 06 = Over-Voltage bit 07 = Low Voltage	bit 08 = Overheated bit 09 = Over Current bit 10 = Short Circuit bit 11 = Motor Not Connected bit 12 = Overloaded bit 13 = Motor Stalled bit 14 = Hall Sensor Error bit 15 = Motor Not Used				
7-12	Left Motor Port Digital I/O Status 4:0060	M: 4:1712 E: I.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)	<u>Bitwise Value – Read Only</u> bit 12 = Short Circuit Error on one or more outputs bit 14 = Over Current – More than 1A detected on one or more outputs		
7-13	Right Motor Port Digital I/O Status 4:0084	M: 4:1713 E: I.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)	See <i>Appendix A - Motor Port as Digital I/O</i> on page 67 for usage details		
7-14	Upstream Module Status 4:0134 See <i>Assembly #7 Inputs Note #1</i> on page 56 for Details	M: 4:1714 E: I.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	Integer Value of Low Byte 0xXX01 = Sensor clear and motor stopped 0xXX02 = Sensor clear, motor running, accepting from upstream 0xXX03 = Reserved 0xXX04 = Sensor blocked, motor running, discharging to downstream		

No.	Register Name / Module Address	Address	Data Description
7-15	Downstream Module Status 4:0232 See Assembly #7 Inputs Note #1 on page 56 for Details	M: 4:1715 E: I.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	0xXX05 = Sensor blocked and motor stopped See Assembly #5 Inputs Note 1 on page 23 for further details
7-16	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. See Assembly #7 Inputs Note #1 on page 56 for Details
7-17	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. See Assembly #7 Inputs Note #1 on page 56 for Details
7-18	Reserved	M: 4:1718 E: I.Data[18] P: Byte 36 (Hi) Byte 37 (Lo)	
7-19	Left Motor Servo Position 4:0062	M: 4:1719 E: I.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)	Signed integer value that indicates the current position of the Left Motor in relation to its "0" position See section <i>Servo Motor Control Example</i> on page 63
7-20	Right Motor Servo Position 4:0086	M: 4:1720 E: I.Data[20] P: Byte 40 (Hi) Byte 41 (Lo)	Signed integer value that indicates the current position of the Right Motor in relation to its "0" position See section <i>Servo Motor Control Example</i> on page 63
7-21	Left Motor Servo Status 4:0011	M: 4:1721 E: I.Data[21] P: Byte 42 (Hi) Byte 43 (Lo)	Bit 0: Servo Command Status 1 = Last Servo Run Command Complete 0 = Servo Command in Process Bit 1: Servo Reset Status Echoes state of Left Motor Servo Command bit 0 Bit 2: Servo Command Status Echoes state of Left Motor Servo Command bit 1 See section <i>Servo Motor Control Example</i> on page 63
7-22	Right Motor Servo Status 4:0016	M: 4:1722 E: I.Data[22] P: Byte 44 (Hi) Byte 45 (Lo)	Bit 0: Servo Command Status 1 = Last Servo Run Command Complete 0 = Servo Command in Process Bit 1: Servo Reset Status Echoes state of Left Motor Servo Command bit 0 Bit 2: Servo Command Status Echoes state of Left Motor Servo Command bit 1 See section <i>Servo Motor Control Example</i> on page 63

ASSEMBLY #7 INPUTS NOTE #1

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. Refer to section *Optional "Clear Connections" Choice* on page 51 for further details.

ASSEMBLY #7 INPUTS NOTE #2

Motor frequency is determined by a combination of motor RPM and the number of poles the motor contains and these will be different depending on the particular MDR used. Please consult MDR documentation to determine meaningful values for motor frequency

ASSEMBLY #7 INPUTS NOTE #3

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

Motor Status bit 0 and bit 1		
Bit 1	Bit 0	Description
0	0	Motor not running, standard or servo braking applied
0	1	Motor running in CCW Direction
1	0	Motor running in CW Direction
1	1	Motor not running and no braking applied (free to spin)

ASSEMBLY #7 INPUTS NOTE #4

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 (8-19) to arrive at the bit values seen in the *Sensor & Control Port Inputs (7-1)* 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 (7-1)* register. This same relationship is also mirrored by the state of the input's corresponding LED indicator.

See *Assembly #8 Outputs Note #2* on page 60 for further details.

ASSEMBLY #7 INPUTS NOTE #5

The data in this register is only valid when a given motor port is placed into Digital I/O mode by the PLC. When a given motor port is being used as Digital I/O the only meaningful bits are bits 12 and 14 as shown. Other bits in this register may contain values and should be ignored as valid input data.

Refer to *Appendix A - Motor Port as Digital I/O* on page 67 for wiring details for using a motor port as digital I/O.



OUTPUTS FOR PLC I/O MODE (ASSEMBLY #8)

These registers are written by the PLC to the *Module* when the module is in PLC I/O mode. All registers are 16-bit Integer data type.

No.	Register Name	PLC Address	Data Description
8-0	ConveyStop Command 4:0020	M: 4:1800 E: O.Data[0] P: Byte 0 (Hi) Byte 1 (Lo)	Integer Value 1 = Command local module's Stop Group to go to Stopped State 2 = Command local module's Stop Group to Clear Stopped State Refer to Insight Automation publication ERSC-1800 <i>ConveyStop User's Guide</i> for details on using <i>ConveyStop</i> .
8-1	Set Left Motor Port Digital Control 4:0060	M: 4:1801 E: O.Data[1] P: Byte 2 (Hi) Byte 3 (Lo)	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 14 = Reserved bit 15 = Digital Output Enable 0 = Use port as Motor Control 1 = Use port as Digital Output See Appendix A - <i>Motor Port as Digital I/O</i> on page 67 for connection details.
8-2	Set Right Motor Port Digital Control 4:0084	M: 4:1802 E: O.Data[2] P: Byte 4 (Hi) Byte 5 (Lo)	
8-3	Sensor & Control Port Digital Output Control 4:0037	M: 4:1803 E: O.Data[3] P: Byte 6 (Hi) Byte 7 (Lo)	Bitwise Value – "1" Energizes Output bit 0 = Reserved bit 1 = Left Control Port bit 2 = Reserved bit 3 = Right Control Port
8-4	Left Motor Run / Reverse 4:0260	M: 4:1804 E: O.Data[4] P: Byte 8 (Hi) Byte 9 (Lo)	Bit 0: 1 = Run Command 0 = Stop Command Bit 8: 0 = Run in Configured Direction 1 = Run opposite of Configured Direction
8-5	Left Motor Brake Method 4:0261	M: 4:1805 E: O.Data[5] P: Byte 10 (Hi) Byte 11 (Lo)	Integer Value 0 = Use Configured Brake Method 1 = Use Standard Brake Method 2 = Use Free Coast Brake Method 3 = Use Servo 1 Brake Method 4 = Use Servo 2 Brake Method
8-6	Left Motor Speed Control Method 4:0262	M: 4:1806 E: O.Data[6] P: Byte 12 (Hi) Byte 13 (Lo)	Integer Value 0 = Use Configured Speed Control Method 1 = Use Open Loop 2 = Use Closed Loop
8-7	Right Motor Run / Reverse 4:0270	M: 4:1807 E: O.Data[7] P: Byte 14 (Hi) Byte 15 (Lo)	Bit 0: 1 = Run Command 0 = Stop Command Bit 8: 0 = Run in Configured Direction 1 = Run opposite of Configured Direction

No.	Register Name	PLC Address	Data Description
8-8	Right Motor Brake Method 4:0271	M: 4:1808 E: O.Data[8] P: Byte 16 (Hi) Byte 17 (Lo)	Integer Value 0 = Use Configured Brake Method 1 = Use Standard Brake Method 2 = Use Free Coast Brake Method 3 = Use Servo 1 Brake Method 4 = Use Servo 2 Brake Method
8-9	Right Motor Speed Control Method 4:0272	M: 4:1809 E: O.Data[9] P: Byte 18 (Hi) Byte 19 (Lo)	Integer Value 0 = Use Configured Speed Control Method 1 = Use Open Loop 2 = Use Closed Loop
8-10	Left Motor Speed Reference 4:0040	M: 4:1810 E: O.Data[10] P: Byte 20 (Hi) Byte 21 (Lo)	Value in % PWM Range: 0 to 1000 Example: 400 = 40% 0 = Run at Configured Speed
8-11	Right Motor Speed Reference 4:0064	M: 4:1811 E: O.Data[11] P: Byte 22 (Hi) Byte 22 (Lo)	
8-12	Left Motor Acceleration Ramp 4:0043	M: 4:1812 E: O.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)	<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 Closed Loop:</u> Value in motor pulses Range: 0 to 10000 (Deceleration) Range: 30 to 10000 (Acceleration)
8-13	Left Motor Deceleration Ramp 4:0044	M: 4:1813 E: O.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)	
8-14	Right Motor Acceleration Ramp 4:0067	M: 4:1814 E: O.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)	<u>For Either Open or Closed Loop:</u> 0 = Use Configured Value
8-15	Right Motor Deceleration Ramp 4:0068	M: 4:1815 E: O.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)	
8-16	Clear Motor Error 4:0022	M: 4:1816 E: O.Data[16] P: Byte 32 (Hi) Byte 33 (Lo)	Logical 0 or 1 0 = Stop Reset 1 = Send Reset Same as Clear Motor Error in Assembly # 5
8-17	Set Status to Downstream Module 4:0196	M: 4:1817 E: O.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)	Used to write ERSC ZPA Status data to downstream ERSC: 4 = Instruct Downstream ERSC to "wake-up" and run its most upstream zone 1 = Instructs Downstream ERSC that carton has exited local zone and to accept any tracking data written in <i>Set Discharge Tracking Word 1 / Word 2</i> registers when carton arrives. See <i>Assembly #8 Outputs Note #1</i> on page 60



No.	Register Name	PLC Address	Data Description
8-18	Set Status to Upstream Module 4:0116	M: 4:1818 E: O.Data[18] P: Byte 36 (Hi) Byte 37 (Lo)	Used to write ERSC ZPA Status data to next upstream ERSC: 5 = Instructs Upstream ERSC's discharge zone to accumulate and hold any carton that arrives at its discharge zone. 1 = Instructs Upstream ERSC's discharge zone to release any carton that arrives at its discharge zone. See <i>Assembly #8 Outputs Note #1</i> on page 60
8-19	Sensor Port Input Signal Condition Mask 4:0034	M: 4:1819 E: O.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)	<u>Bitwise Value</u> bit 0 = Left Sensor Port - Red LED [Auxiliary (M8 Pin2)] bit 2 = Right Sensor Port - Red LED [Auxiliary (M8 Pin2)] bit 4 = Left Sensor Port - Green LED [Signal (M8 Pin4)] bit 6 = Right Sensor Port - Green LED [Signal (M8 Pin4)]
8-20	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 ERSC needs to pass tracking data to a downstream connected ERSC. Used in conjunction with <i>Set Status to Downstream Module (8-17)</i> register
8-21	Set Discharge Tracking Word 2 4:0202	M: 4:1821 E: O.Data[21] P: Byte 42 (Hi) Byte 42 (Lo)	See <i>Assembly #8 Outputs Note #1</i> on page 60
8-22	Reserved	M: 4:1822 E: O.Data[22] P: Byte 44 (Hi) Byte 44 (Lo)	
8-23	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 Left Motor Servo Run Command Valid values are from -32767 to +32767 See section <i>Servo Motor Control Example</i> on page 63
8-24	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 as "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 See section <i>Servo Motor Control Example</i> on page 63
8-25	Right Motor Servo Command Pulses 4:0013	M: 4:1825 E: O.Data[25] P: Byte 50 (Hi) Byte 51 (Lo)	Signed integer value in motor pulses of the position to move to on the next Left Motor Servo Run Command Valid values are from -32767 to +32767 See section <i>Servo Motor Control Example</i> on page 63
8-26	Right Motor Servo Command Word 4:0014	M: 4:1826 E: O.Data[26] P: Byte 52 (Hi) Byte 52 (Lo)	Bit 0: Reset Command 1 = Set Current Pulse Count as "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 See section <i>Servo Motor Control Example</i> on page 63

ASSEMBLY #8 OUTPUTS NOTE #1

For registers involving status and tracking to upstream or downstream ERSC's in ZPA mode, the connections must be preserved when placing the ERSC in question into PLC I/O mode from within *EasyRoll*. Refer to section *Optional "Clear Connections" Choice* on page 51 for further details.

ASSEMBLY #8 OUTPUTS NOTE #2

For an ERSC in standard ZPA mode, the Auto-Configuration procedure sets values in this register to allow the ERSC 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 ERSC's sensor input pin 4 when the zone is clear. The ERSC 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 (8-19) 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.

When an ERSC is placed in PLC I/O mode; the *Sensor & Control Port Input Signal Condition Mask* register (8-19) is cleared of the values set during the Auto-Configure procedure. The *Sensor & Control Port Input Signal Condition Mask* register (8-19) 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* (7-1) 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* (7-1) 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 ERSC's inputs. The same pattern applies to all 8 available ERSC inputs:

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

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.

Note this is applicable only to Ethernet I/P PLC and is not applicable to Modbus TCP PLC.



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; please note that when the green and red LEDs are illuminated at the same time, the color will be amber.

SERVO MOTOR CONTROL EXAMPLE



Servo Motor Control requires ConveyLinX firmware 4.14 or later

The Servo Motor Control function allows you to establish a zero or “home” rotational position of the MDR and then instruct the ERSC to rotate an assigned number of motor pulses in either direction from this zero or “home” position. Assemblies 7 & 8 provide the data registers to allow external PLC control of this function for both the Left and Right MDR ports.



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 (8-6 for the Left Motor or 8-9 for the Right Motor).

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.

For our example, we want to perform the following cycle using PLC control of the Left MDR on and ERSC:

- 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

Now let’s define some logic names that correspond to actual Assembly 7/8 tag values to make our example easier to follow:

Example Tag	Data Type	Assembly Register (/bit)
ZERO	Boolean	8-24 / bit 0
ZERO_ACK	Boolean	7-21 / bit 1
RUN	Boolean	8-24 / bit 1
READY	Boolean	7-21/bit 0
POSITION	Integer	8-23
SPEED	Integer	8-10
ACCEL	Integer	8-12
DECEL	Integer	8-13
MODE	Integer	8-6

Step #1: 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

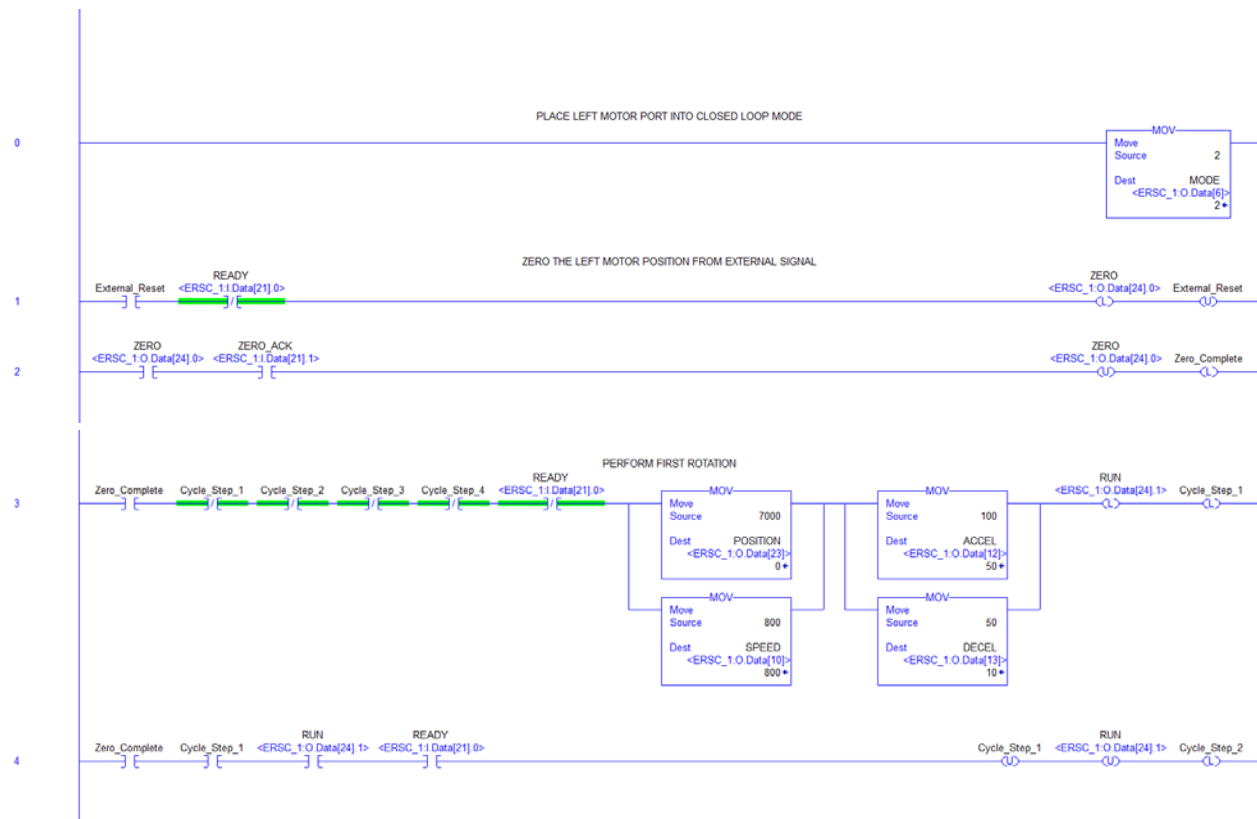


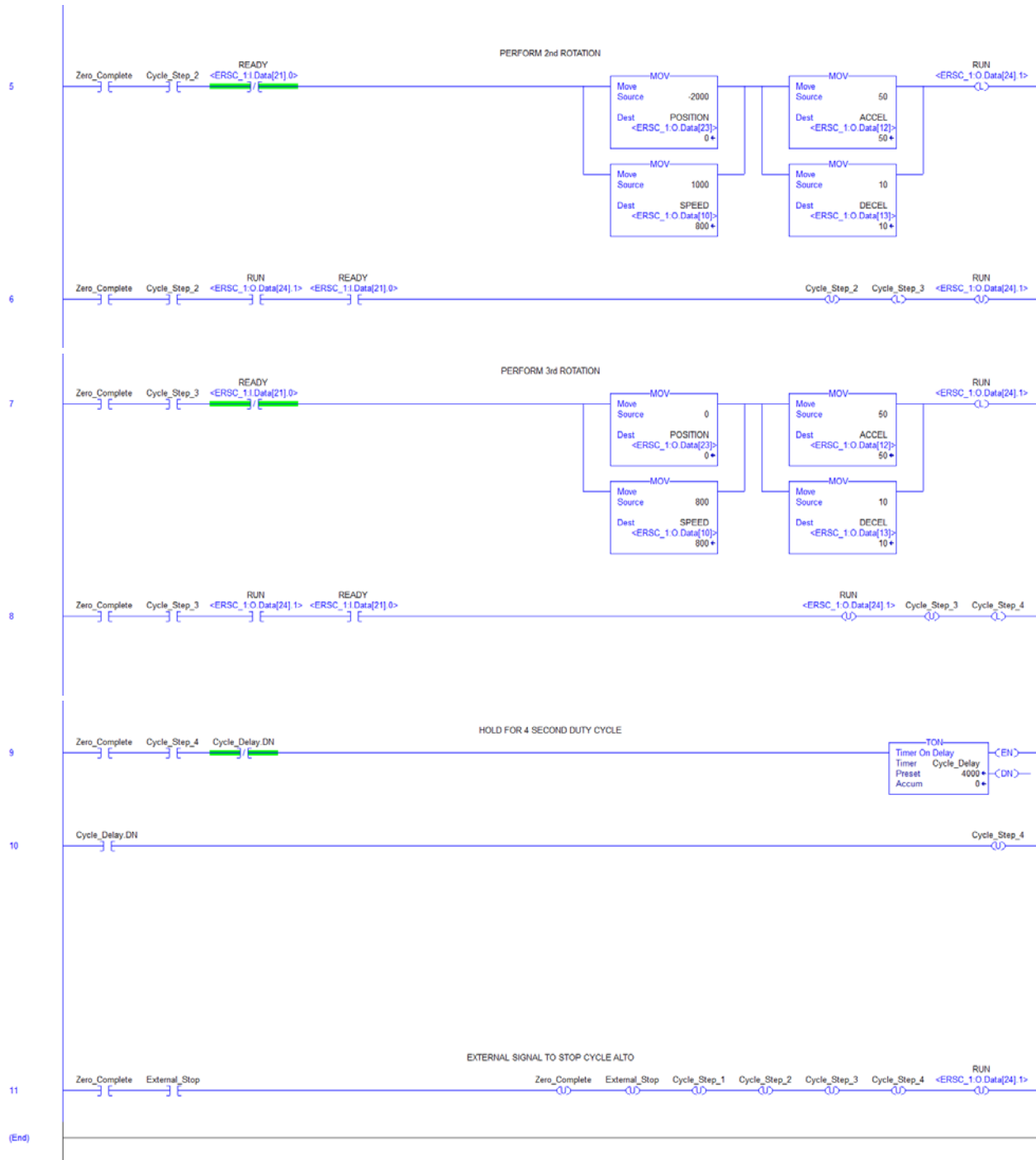
Rotation Note: Rotation from 0 to a positive value always causes rotation in CCW direction and rotation from 0 to a negative value always causes rotation in CW direction. Servo Motor Control ignores the CW/CCW direction setting in *EasyRoll* and/or any configured or default direction that may exist in the ERSC's configuration.



Values shown in this example are for demonstration purposes only. Actual position, speeds, acceleration, and deceleration values are highly dependent upon the mechanical application.

PLC LADDER DIAGRAM





APPENDIX A - 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.

Figure 8 shows the wiring diagram for a given motor port configured for digital output. Please note that the ERSC switches to GND to complete the circuit.

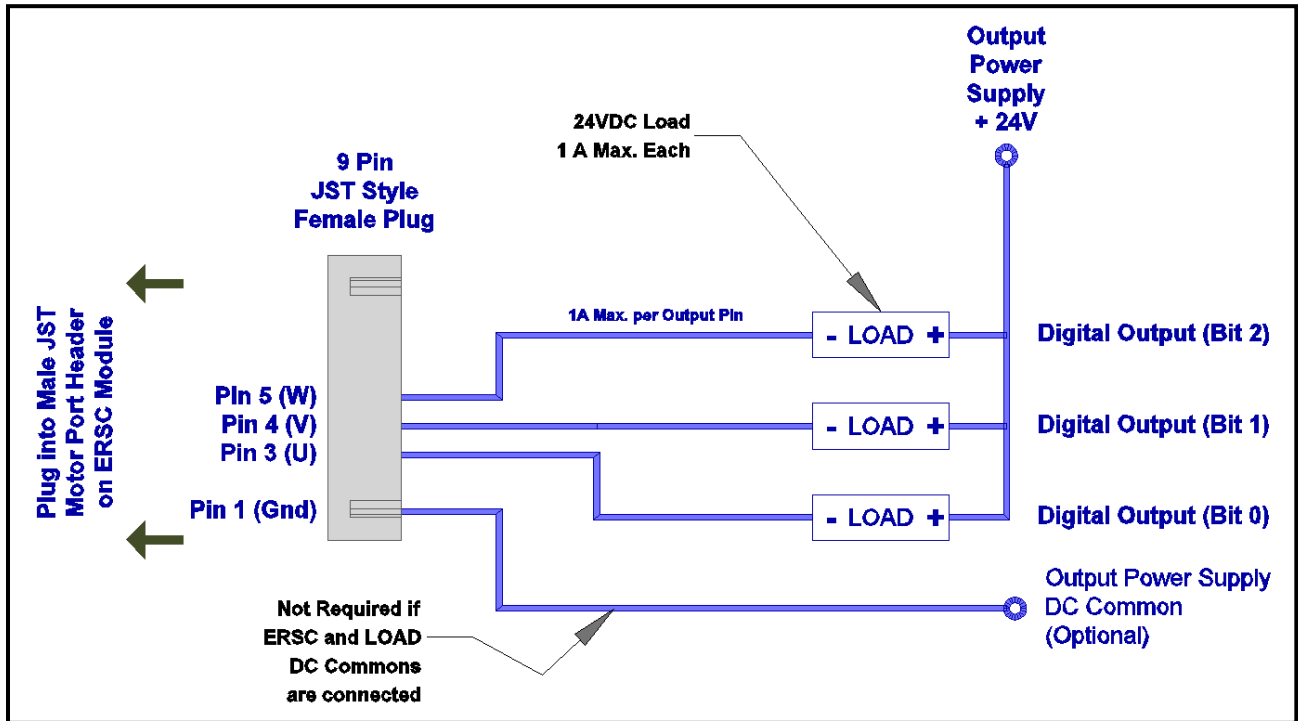


FIGURE 8 - WIRING DIAGRAM FOR MOTOR PORT CONFIGURED FOR DIGITAL OUTPUT



Connection to Pin 1 (GND) as shown is only required if the Output Power Supply is at different DC common than the ERSC's power supply. If both power supplies have the same DC common, then this connection is not required.



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.



APPENDIX B – CONVEYLINX CONNECTIONS

ZPA MODE REGISTER DATA FLOW

Because each *ERSC* has read/write access to its adjacent *ERSC*s and by virtue of the Auto-Configuration procedure that defines conveyor flow **connections**; each *ERSC* knows the I.P. address of its adjacent upstream and downstream *ERSC*. In the direction of flow, each *ERSC* writes its zone's status values to its next upstream and next downstream *ERSC*. The following figure graphically illustrates a simple configuration of three *ERSC*'s and the logical **connections** made by each *ERSC* to communicate with its adjacent modules and the local register used.

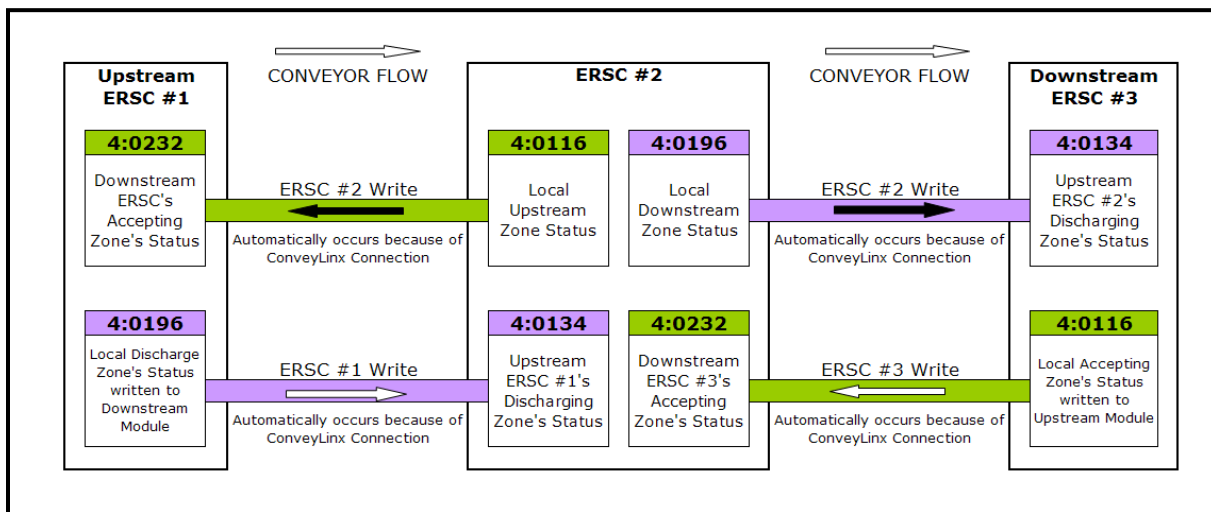


FIGURE 9 - ERSC REGISTER MAPPING FOR NORMAL FLOW



Please note that the following examples in this section assume the High Byte of all zone status registers have been masked such that only the Low Byte values are used.

From the preceding charts, we can see that all of the *Holding Registers* in Figure 9 contain integer values indicating a zone status. In essence, the ZPA control process in each *ERSC* writes its zone's status to its adjacent neighbors and monitors its local registers that are written to by its adjacent *ERSC* neighbors. The ZPA process acts upon what is written to it and updates its local status accordingly. For example, the following sequence would be typical:

- *ERSC* #1 continually reads register 4:0232 to know the status of *ERSC* #2's local upstream zone
- If a Carton arrives at *ERSC* #1 discharge zone and the value in its register 4:0232 is equal to 4 or 5, then it will stop the Carton at its downstream discharge zone
- *ERSC* #1 continues to read register 4:0232. Once the value in its register 4:0232 has changed to a value of 1 or 2; the Carton in its discharge zone can be conveyed to *ERSC* #2. To initiate this, *ERSC* #1 writes its status value of 4 (located in register 4:0196) to register 4:0134 on *ERSC* #2.

- When *ERSC #2* control logic sees its register 4:0134 change to a value of 4, it knows that a Carton is being discharged onto its upstream zone and thus will energize its upstream zone MDR to run and accept the Carton from *ERSC #1*.

There would be a similar sequence for *ERSC #2* to convey a Carton to *ERSC #3*.

EXTERNAL CONTROLLER AS CONVEYLINX BRIDGE

An external controller or PLC can also write valid zone status data to *ERSCs* that are adjacent to a PLC controlled non-*ConveyLinx* conveyor or machine. The PLC simply mimics the *ERSC* interface to control flow from upstream *ConveyLinx* controlled conveyor and to initiate Load transfer to downstream *ConveyLinx* controlled conveyor.

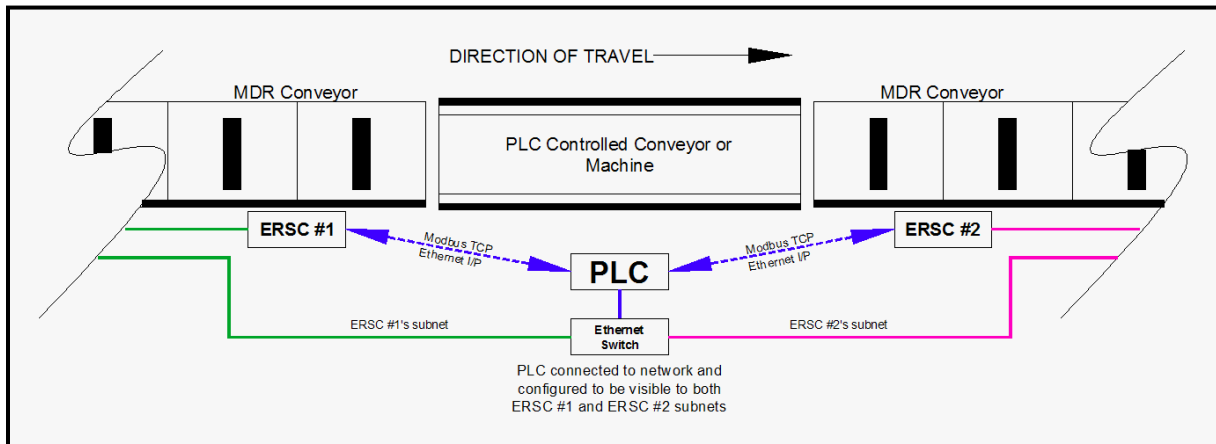


FIGURE 10 - PLC CONVEYLINX BRIDGE

Figure 10 shows a simple example where a PLC would act as a bridge between two separate *ConveyLinx* subnets. In this case, *ERSC #1* is at the end of its particular subnet, so when it was configured; it knows that it does not have a logical downstream connection. Similarly, *ERSC #2* is at the beginning of its particular subnet, so when it was configured it knows that it does not have a logical upstream connection. In order for Loads to flow from *ERSC #1* eventually to *ERSC #2*; the PLC must act as a bridge and “mimic” the writing of register data to *ERSC #1* and *ERSC #2* as if it was an *ERSC* module. This is illustrated in Figure 11.

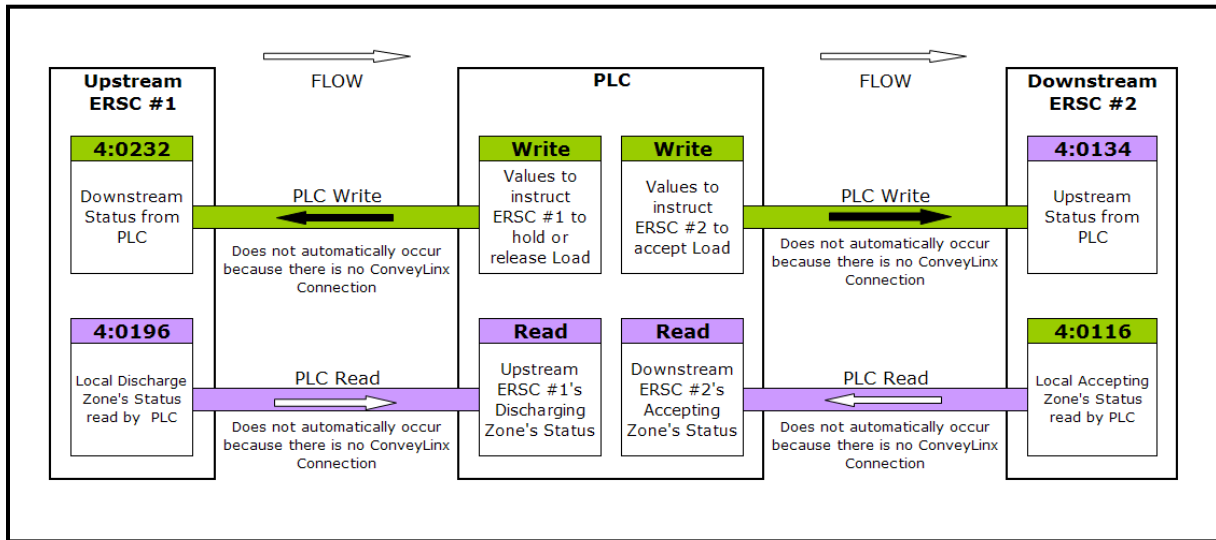


FIGURE 11 – DATA FLOW CHART FOR PLC CONVEYLINX BRIDGE EXAMPLE

For example, if the PLC wants to make sure that any Carton stops at the discharge of ERSC #1, it will write a value of 5 into ERSC #1 register 4:0232. A value of 5 means “zone sensor blocked and motor stopped” and will be interpreted by ERSC #1 as a reason to stop and hold any Carton that arrives at its discharge zone.

The PLC can monitor the status of ERSC #1 discharge zone by reading ERSC #1 register 4:0196. If this value is 5, the PLC knows a Carton is present and stopped in ERSC #1’s discharge zone. When the PLC is ready to accept the Carton, it simply writes a value of 1 to ERSC #1 register 4:0232.

For the discharge from the PLC controlled conveyor or machine, the PLC can monitor the status of the upstream zone of ERSC #2 by reading ERSC #2 register 4:0116. For example, if a value of 5 is read, the PLC will know that the ERSC #2 accepting zone is occupied and is not ready to accept a Carton. If the value read is a 1 or 2, then the PLC knows it can discharge a Carton when ready. In order to discharge a Carton, the PLC will need to write a value of 4 to ERSC #2 register 4:0134 to tell ERSC #2 to run its upstream accepting zone.

NOTES:



PULSE ROLLER

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