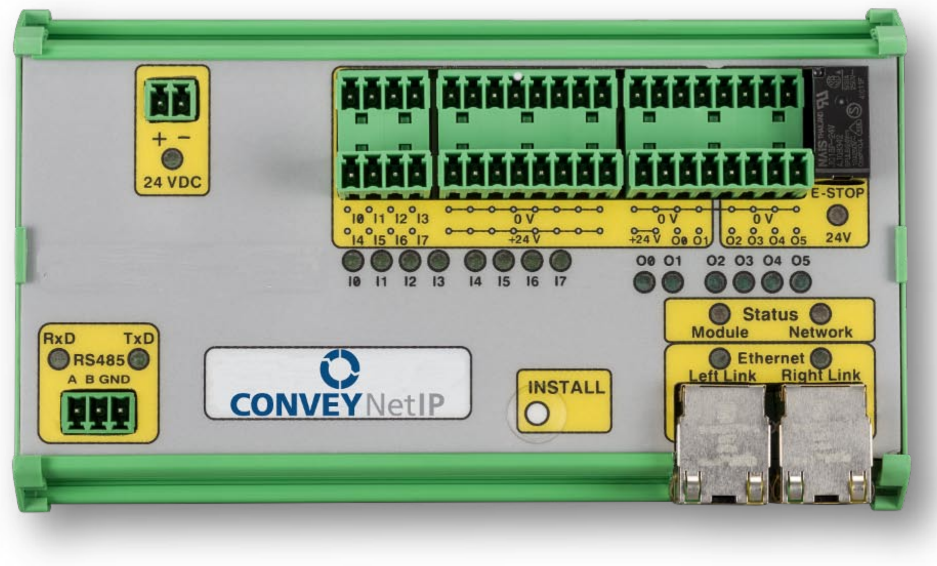
PULSEROLLER

CONVEY NetIP

User's Guide



Version 1.4

January 2016

Publication CNIP-1000

ConveyLinx module firmware and functionality is protected by U.S. and international patents. For complete patent information visit www.pulseroller.com/patents

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

ConveyNet IP (CNIP) 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

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	July 2013	Initial Release
1.1	April 2014	Updated Contact Information
1.2	June 2014	Updated formatting
1.3	August 2014	Modified registers in PLC Instances
1.4	January 2016	Updated registers for FW 4.25 and 5.2

Global Contact Information

PULSE ROLLER
WWW.PULSEROLLER.COM
SALES@PULSEROLLER.COM
SUPPORT@PULSEROLLER.COM

TABLE OF CONTENTS

Preface

Who Should Use This Manual?

- This manual is intended for users who need basic product information and simple application procedures to implement *ConveyNet IP* modules to perform simple remote I/O and RS-485 Modbus RTU device control via PLC's capable of either Modbus TCP, Ethernet I/P, or Profinet I/O protocols.
- You should have a basic understanding of electrical circuitry and familiarity with relay logic, conveyor equipment, photo-sensors, etc. If you do not, obtain the proper training before using this product.
- You should also be very familiar with PLC programming techniques including specifically Allen-Bradley Logix platform hardware and RSLogix 5000 programming software.
- This manual also assumes the reader is familiar with the *ConveyLinx* family of control modules.
- This manual also assumes familiarity with the use of *EasyRoll* software.

Purpose of This Manual

The purpose of this manual is to:

- Identify the components, termination points, and ports available on a module
- Provide guidelines for proper installation and wiring
- Provide PLC Instance Assembly information to allow PLC programmers to read/write I/O and RS-485 data.

Not Included in This Manual

Because system applications vary; this manual assumes users and application engineers have properly designed the emergency stop and other safety systems in accordance with all applicable codes, standards, and site requirements. From a PLC/System perspective, ConveyNet I/P modules operate the same as any generic remote I/O device and the same programming practices apply.

Introduction to ConveyNet I/P®

*ConveyNet IP (**CNIP**)* modules are designed to work in concert with the *ConveyLinx* and *ConveyBlox* control system architecture. *CNIP* modules are designed to be installed and configured along with *ConveyLinx ERSC* and *ConveyBlox* modules in the same Ethernet network. While *ConveyLinx ERSC* and *ConveyBlox* are specialized modules for direct conveyor equipment control, *CNIP* is a general purpose I/O module that requires either a remote PLC on an on-board *ConveyLogix* program in order to provide any function. *CNIP* modules require the same auto-configuration installation procedures as *ConveyLinx ERSC* modules. Each *CNIP* also includes convenient connectivity ports for upstream and downstream Ethernet network cabling as well as removable terminal block connection points for discrete I/O signal wiring. Each *CNIP* also includes terminal block wiring connection point for RS-485 serial communications connection.

CNIP Module Features

Each individual *CNIP* module has the following features:

- ✓ Built-in 3-port Ethernet switch
- ✓ Removable Terminal Block plugs for convenient wiring
- ✓ Dual power connection point to separate module power from digital output bus power
- ✓ 24VDC / DC Common bus terminals for convenient connection of two and 3 wire devices without the need for external terminal blocks
- ✓ On-Board contact relay for physical disconnect of output power bus for 4 of the discrete outputs
- ✓ On board RS-485 Modbus RTU port

TYPICAL ARCHITECTURE

The typical architecture for *CNIP* modules is to be distributed I/O connected to and controlled by a central PLC. The RS-485 port is most often used to connect to a Modbus-RTU capable VFD making the *CNIP* module a convenient gateway for an Ethernet based PLC to have dynamic control over VFD parameters over the network. *Figure 1* shows a line diagram of a typical architecture using *CNIP* modules with a PLC.

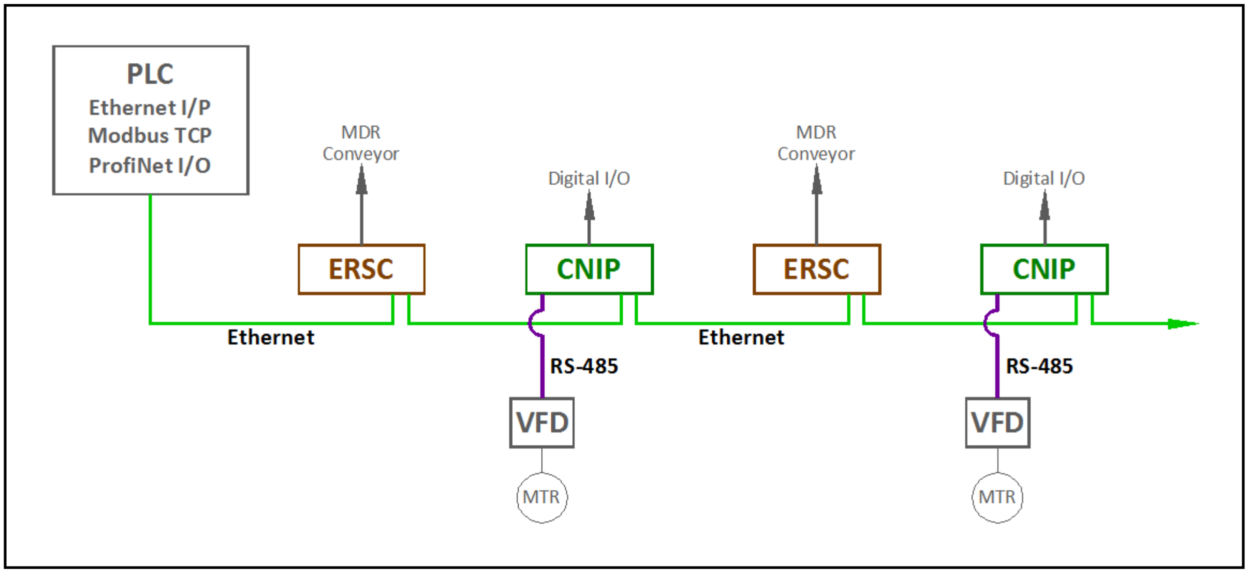


FIGURE 1 - TYPICAL *CNIP* ARCHITECTURE

The *CNIP* module can also accept a logic program download created from *ConveyLogix* programming software. When running its own logic program, the *CNIP* can still be incorporated within a *ConveyLinx* network and automatically exchange status data with neighbouring *ConveyLinx* modules.

CNIP Module Hardware Overview

CNIP modules are designed to be installed and integrated into the conveyor's mechanical side frame assembly. Please refer to *Appendix A – Mounting* Dimensions on page 55 for module dimensions.

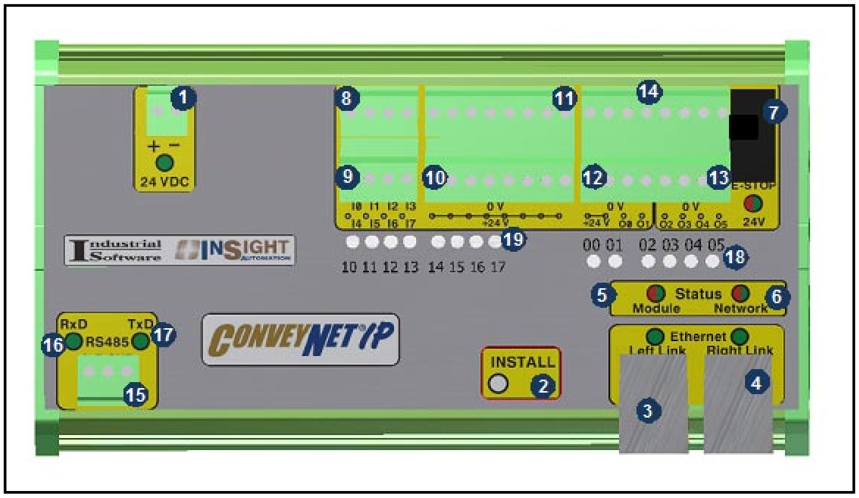


FIGURE 2 - *CNIP* MODULE HARDWARE

Item	Description
1	Module 24VDC Power Connector
2	Install Button – Used for Auto Configuration and module Auto Replacement functions
3 & 4	*Link Left* and *Link Right* – RJ-45 style Ethernet network communication connection between modules
5 & 6	Module Status and Network LED Indicators
7	Isolation Relay with LED Indicator
8 & 9	Terminals for the 8 Digital Inputs
10 & 11	+24V and 0V convenience power source terminals for input devices
12	Output Bus Terminal with extra terminals to provide jumpers to Module 24V Power terminals
13	Terminals for the 6 Digital Outputs
14	Terminals for 0V Common for Digital Outputs
15	RS-485 Connection
16 & 17	RS-485 Transmit and Receive Activity LEDs
18	Digital Output status LEDs
19	Digital Input status LEDs

ETHERNET PORTS

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

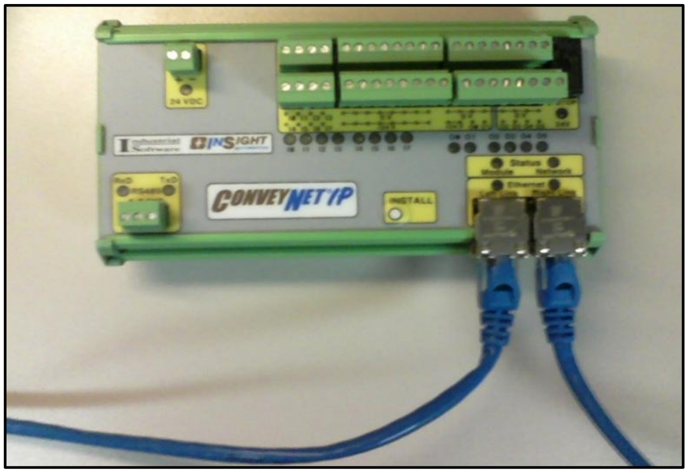


FIGURE 3 - *CNIP* WITH LEFT & RIGHT ETHERNET CABLES

All Ethernet cables for connections between modules are <u>recommended to be shielded</u>. Failure to use shielded cables could result in data loss and unexpected results.

Status Indicators

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

Communications

Indicator	Item	LED State	Description
Ethernet Left & Right Link	3 & 4	OFF	No link established
		Solid Green	Link is established
		Blinking Green	When data transmission activity is occurring
RxD RS-485 Link	16	OFF	No Data being transferred
		Blinking Green	When data is being received
TxD RS-485 Link	17	OFF	No data being transferred
		Blinking Green	When data is being sent

Network & Module Function

Indicator	Item	LED State	Description
Module Status	13	*Blinking* Red	*CNIP* is configured for RS-485 communications but there is RS-485 communications error
		Blinking Green	*CNIP* is ready. If RS-485 communications has been configured then RS-485 port is operational
		Flashing Green & Blinking Red	Module is in Auto-Replace Mode
		Flashing Red	Auto Configure Mode is active
		Solid Amber	Firmware upgrade in progress
Network Status	14	Blinking Red	No Ethernet I/P connection is established
		Blinking Green	Ethernet I/P connection is established
Isolation Relay	7	OFF	No 24V Power on Output Bus
		Solid Red	24V Power connected to Output Bus – Relay De-Energized
		Solid Green	24V Power connected to Output Bus – Relay Energized

CNIP WIRING

POWER WIRING

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

SEPARATELY POWERED OUTPUT BUS AND MODULE POWER

The *CNIP* module is designed such that all 6 digital outputs receive their power from the supply connected to the Output Bus (Item 12 from *Figure 2*). This is done so that the Output Bus Power Supply can be separately switched off so that the integrator is assured that all digital outputs can de-energized regardless of the logical state of the digital output being remotely controlled by the PLC. Because the *CNIP* Module power can be fed separately from the output bus power and can remain on when the output bus power is off; the *CNIP* module's digital inputs and communications can remain active. *Figure 4* shows the terminal connections on a *CNIP* module for powering the Output Bus separately from the Module Power Supply.

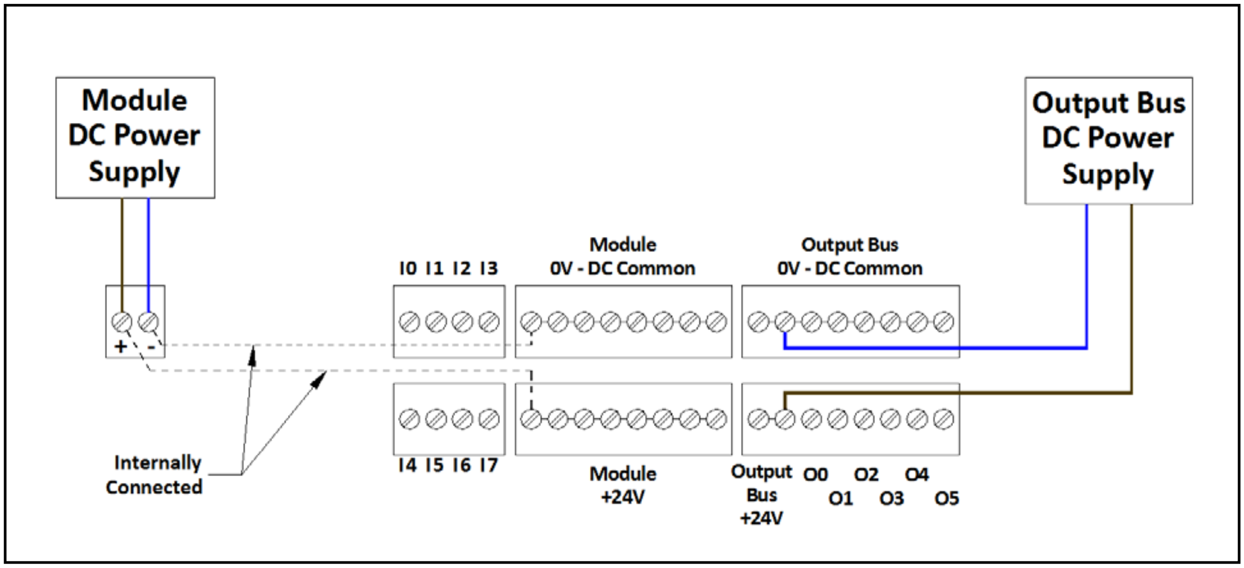


FIGURE 4 - SEPARATELY POWERED OUTPUT BUS EXAMPLE

OUTPUT BUS AND MODULE POWERED TOGETHER

For applications where separately disconnecting the Output Bus from the Module Power Supply is not required, the *CNIP* module provides convenience terminals such that a single power feed can connect to the Output Bus terminals (Item 12 in *Figure 2*) point and jumper wires can be conveniently installed to connect this power feed to the Module Power Feed terminals (Item 1 in *Figure 2*). This alleviates the need to provide 2 external sets of power feed wires for each *CNIP* module. *Figure 5* shows and example of a single power feed for both the Output Bus and the Module Power Supply.

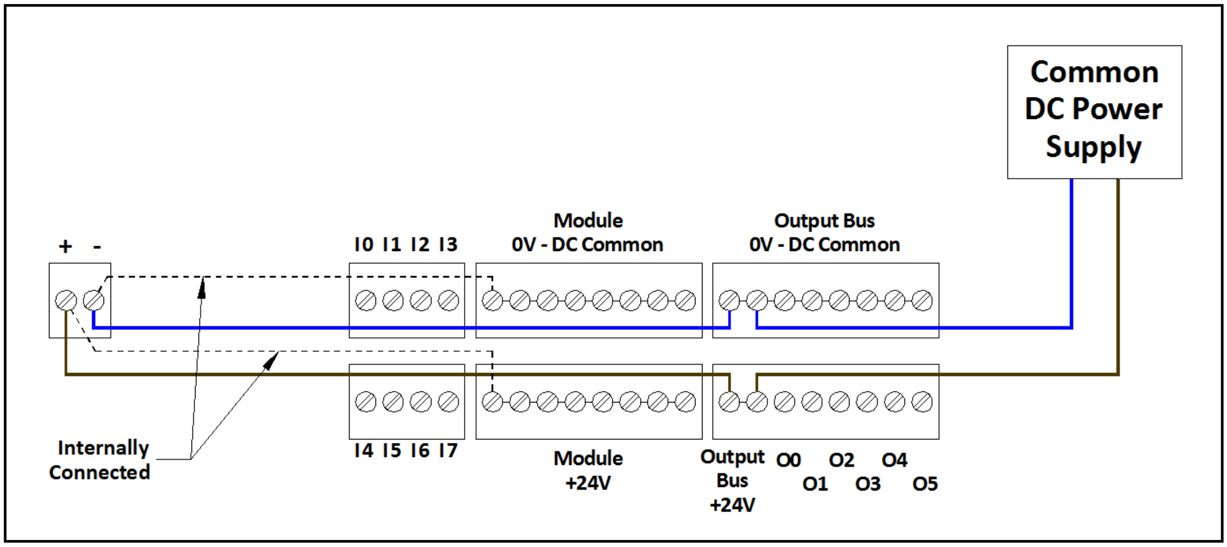


FIGURE 5 - SINGLE POWER FEED EXAMPLE

Please note that the *Module 0V DC Common* and the *Output Bus 0v DC Common* terminals are <u>not internally connected</u>.

I/O Wiring

Typical Input Wiring

All 8 *CNIP* digital inputs expect a 24V sourcing (PNP) signal from a given input device. The *CNIP* module provides convenient +24 and 0V common power terminals so that 3 wire devices such as photo sensors can be easily connected at the *CNIP* without the need for separate external terminal blocks for device power. *Figure 6* shows a typical 3-wire and 2-wire device example.

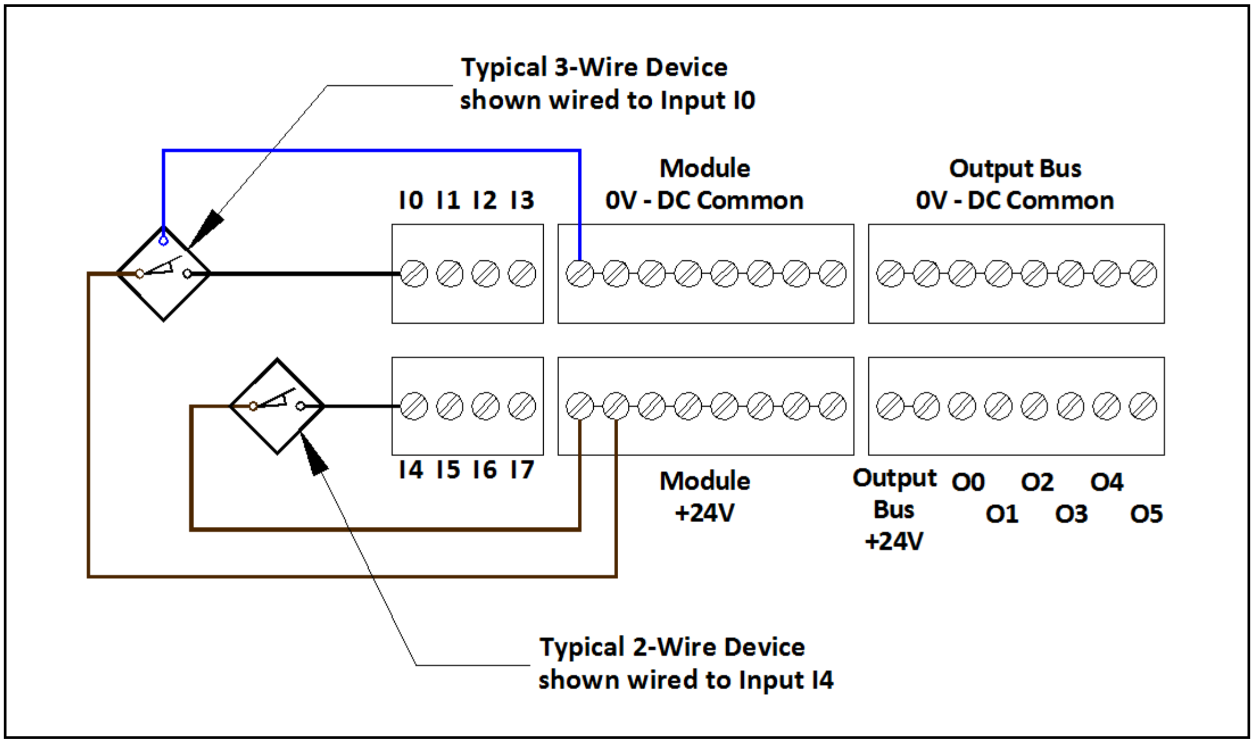


FIGURE 6 - TYPICAL 3-WIRE AND 2-WIRE INPUT DEVICE EXAMPLE

Typical Output Wiring

All 6 of the outputs on the *CNIP* provide a sourcing 24V signal only. As a convenience, the *CNIP* includes a 0V DC common terminal for each output so that a typical 2-wire device can connect at the *CNIP* without the need for separate external terminal blocks for device 0V common connection. *Figure 7* shows a typical 2-wire output device connection example.

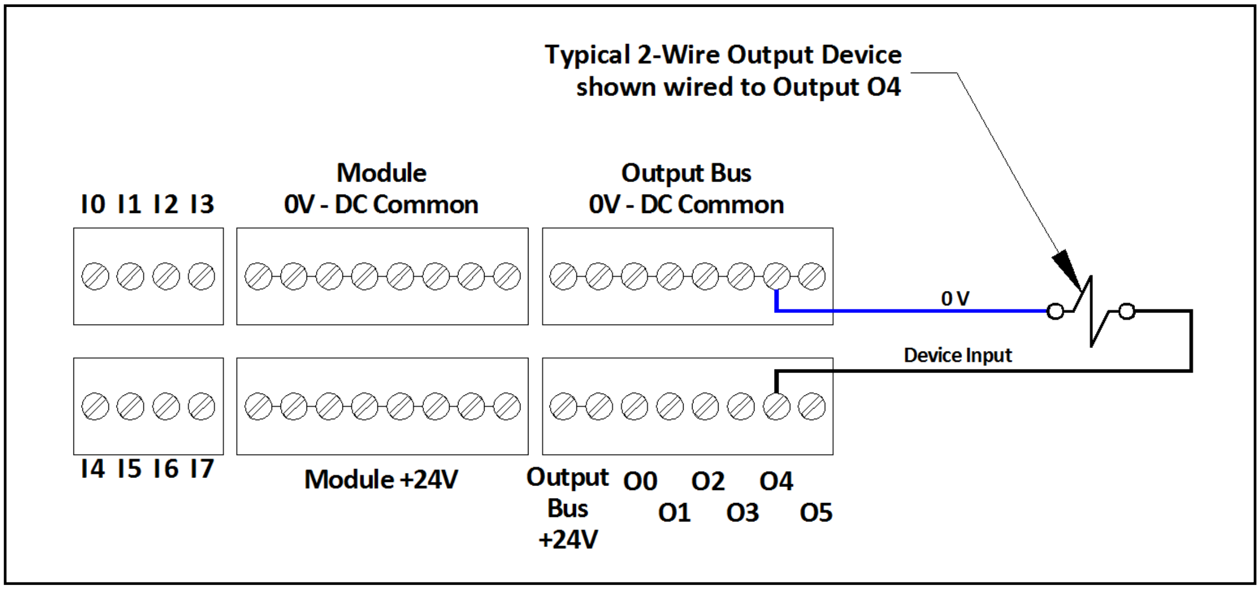


FIGURE 7 - TYPICAL 2-WIRE OUTPUT DEVICE EXAMPLE

Isolation Relay

The Isolation Relay (Item 7 in *Figure 2*) is used as an electro-mechanical means for the on board processor to disconnect control power from outputs O2 thru O5 regardless of the logical state of these outputs. The on-board processor must be functioning for the Isolation Relay to be energized so that its contacts can supply power from the Output Bus terminal (item 12 from *Figure 2*) to outputs O2 thru O5. Also, the functioning on-board processor can de-energize the Isolation Relay based upon its own internal logic criteria. This functionality is reserved for future network-based stop/disconnect implementation(s). *Figure 8* shows the logical internal circuit for the Isolation Relay contact and its connection to digital outputs O2 thru O5.

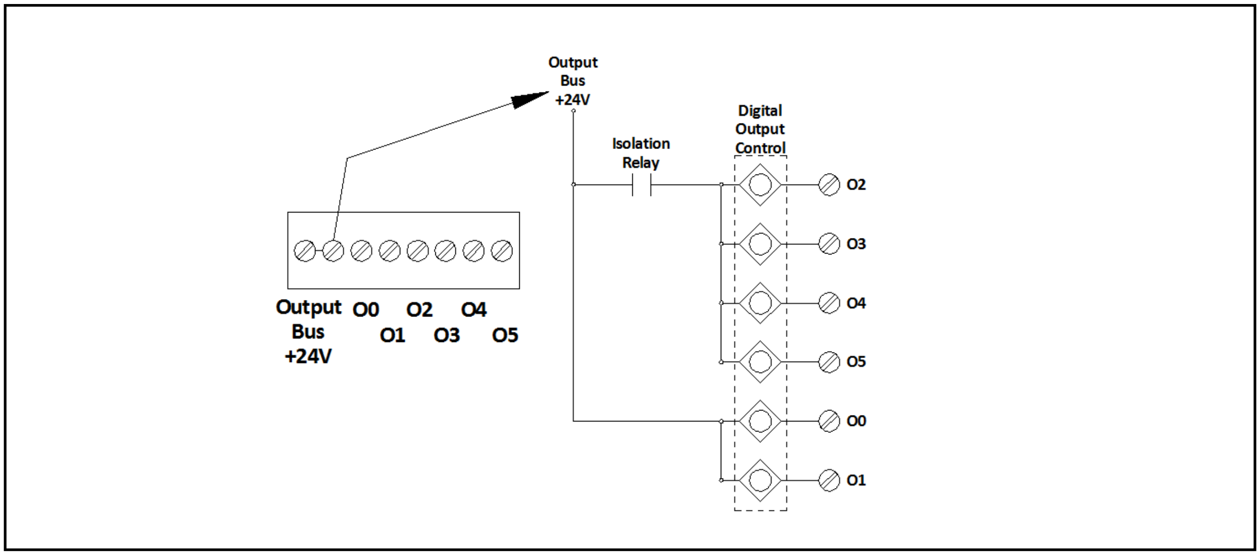


FIGURE 8 - ISOLATION RELAY INTERNAL CIRCUIT

RS-485 Wiring

RS-485 utilizes 2 wires for communication (plus a GND). The *CNIP* module terminals for these two wires (*Figure 2* Item 15) are designated “A” and “B”. In most device’s documentation these two wires are indicated generically as “TxRx+” and “TxRx-“ and are shown for illustration purposes only. Note that there is no standard for these “+” and “-” wires, and consequently Modbus device manufacturers interpret them differently.

Because there is no standard for the “+” and “-“ connections for RS-485, if you have problems with initially establishing communications, try swapping the two network wires at either the *CNIP* or the device.

EasyRoll Software Configuration Tool

Introduction

The *EasyRoll* Software Configuration Tool is a PC based application that provides a means to configure and examine the status the entire *ConveyLinx* family of modules. The majority of the functions included in *EasyRoll* pertain to the *ERSC* module for ZPA control of MDR conveyor. Because *CNIP* modules are designed to integrate seamlessly with *ERSC* modules on the same network, some of the basic functions included with *EasyRoll* are applicable to the *CNIP* module. Here are the basic functions available for *CNIP* with *EasyRoll*:

- Discover and identify individual modules on a network
- View and set I.P. Address settings for individual modules on a network
- Ability to download firmware updates to individual or groups of modules on a network
- View the I/O status and RS-485 Communication status of individual modules
- Establish or remove data connections between adjacent *ConveyLinx* family modules

Please refer to Insight Automation publication *ConveyLinx* User's Guide for a more complete description of the common *ConveyLinx* family operations with *EasyRoll* including how to modify I.P. address settings, data connections, firmware updates, etc. as these are identical for both *ERSC* and *CNIP* modules.

EasyRoll Main Screen for CNIP

The main difference between connecting *EasyRoll* with *CNIP* versus *ERSC* is that the main screen depicting the individual connected module is different for the two module types. The main screen for a *CNIP* module will show:

- Whether the *CNIP* is connected to a PLC
- If connected, the on/off state of each individual I/O point
- If enabled, the read/write data registers of the RS-485 port

Figure 9 shows a typical *EasyRoll* main screen for a *CNIP* module when connected to a PLC and when the RS-485 port is not used and *Figure 10* shows a typical main screen when a *CNIP* is connected to a PLC and the RS-485 port is being used.

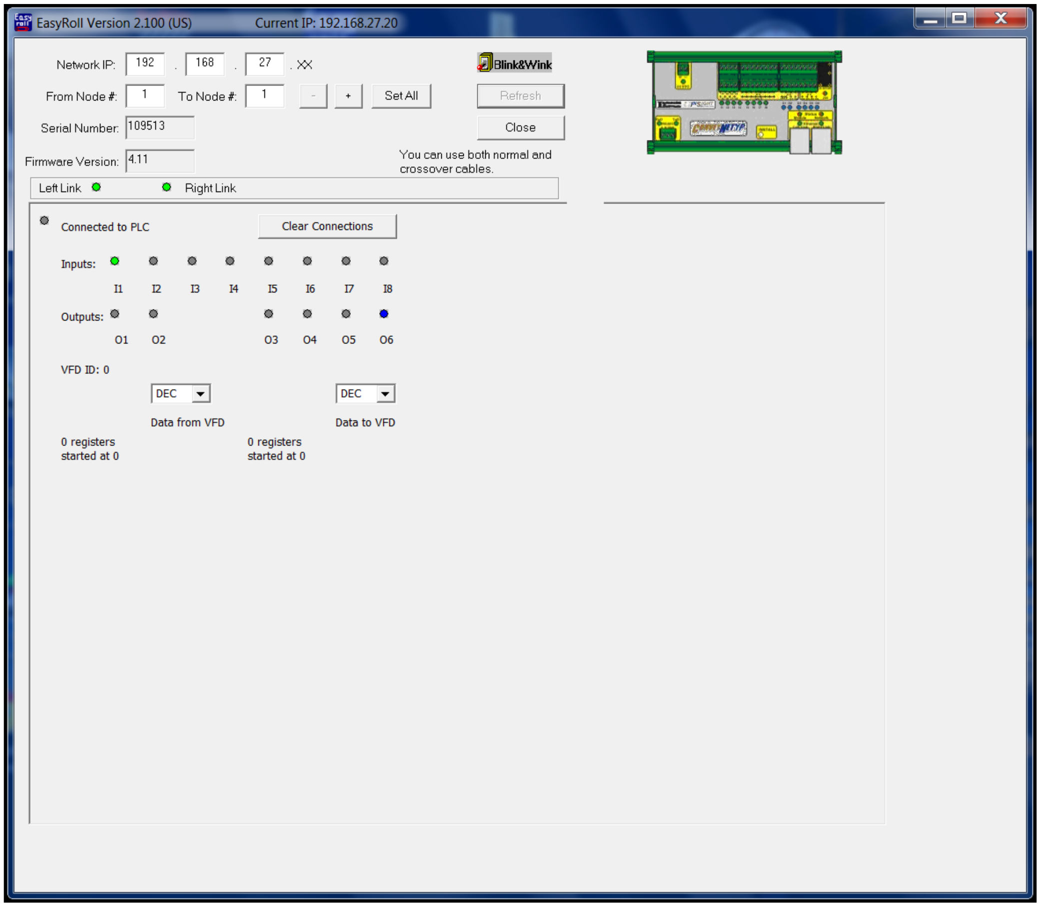


FIGURE 9 - *EASYROLL CNIP* MAIN SCREEN WHEN RS-485 PORT IS NOT USED

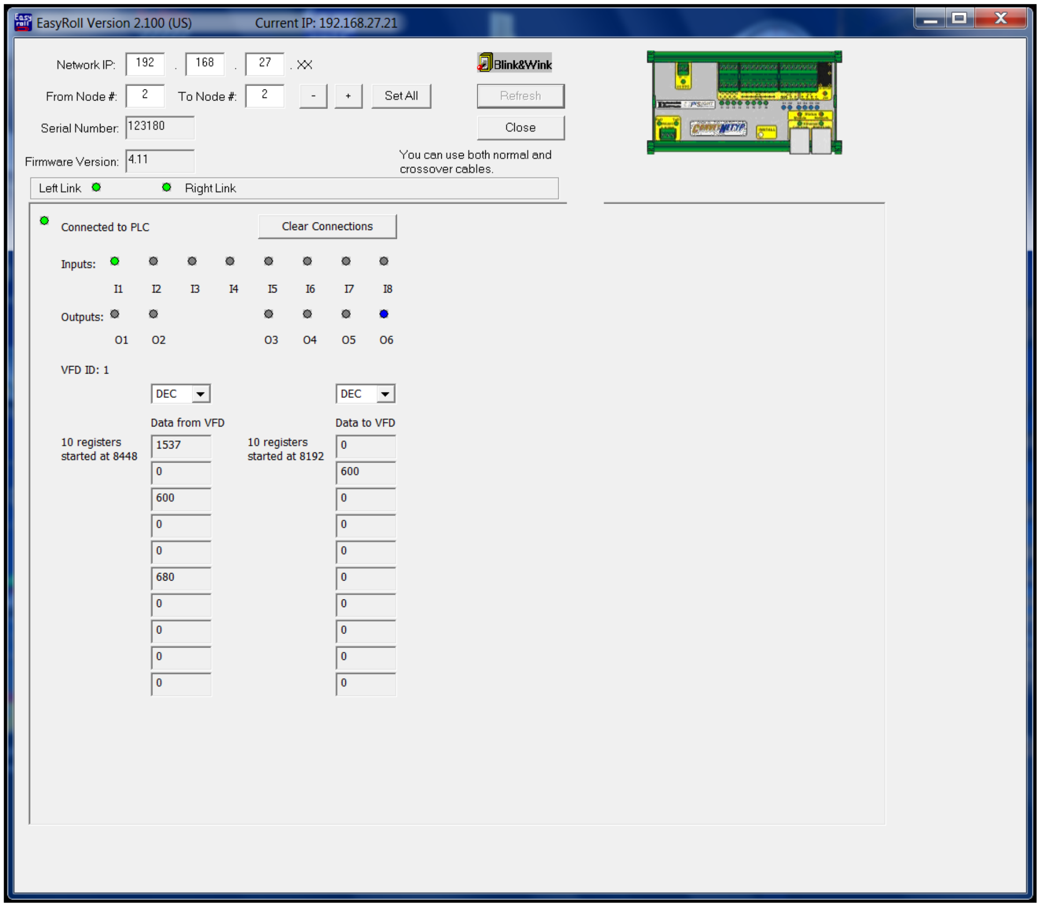


FIGURE 10 - *EASYROLL CNIP* MAIN SCREEN WHEN RS-485 PORT IS USED

Please note that some of the detailed information shown in these figures may be different for your particular system and that most of these fields will be blank until you actually initiate communications with a PLC and/or your RS-485 device.

CNIP Network Architecture

The *CNIP* module is part of the *ConveyLinx* family of hardware modules. All *ConveyLinx* family hardware is designed to reside on the same Ethernet network and exchange data in the same format so that systems can seamlessly use all *ConveyLinx* family hardware types for an integrated solution.

Each *CNIP C*ommunicates to its adjacent modules and to any connected PC or PLC via Ethernet physical media. All *ConveyLinx* family modules recognize (3) TCP/IP based protocols:

- Modbus TCP
- Ethernet I/P
- Profinet I/O (not covered in this manual – contact Insight Automation for details)

Modbus TCP is the "native" protocol for communications between *ConveyLinx* modules and the *EasyRoll* PC software. Regardless of whether Ethernet I/P and/or Profinet are being used simultaneously; *ConveyLinx* modules utilize Modbus TCP for inter-module communication. Unlike the other devices in the *ConveyLinx* family that provide pre-programmed functions "out of the box"; the *CNIP* module does not contain any pre-programmed functionality.

To utilize a CNIP module you must either:

- Use a remote PLC to connect (using one of the 3 aforementioned protocols) to the CNIP to read/write data to control its I/O and/or use the RS-485 port.
- Create your own user logic program with ConveyLogix programming software and download the program to the CNIP

Each *CNIP's* internal data structure is arranged and addressed as Modbus Holding registers. The on-board *CNIP* communication and control processes attach logical meanings to each holding register and read and write data to specific registers to initiate and/or react to events. Certain registers contain information as to how the *CNIP's* RS-485 port is configured; others are for the data used by the port as well as the status of the connected digital I/O.

Assemblies for Remote PLC Connection

Understanding Assemblies

For the holding registers described above, we will define these as ***Module Register Addresses*** and they are used for the function of the *CNIP* regardless of if any remote PLC or PC is connected. There are many *Module Register Addresses* that are used for the operation of the *CNIP* that are not applicable, required, or should even be made available to a remote PLC. When connecting to a PLC, the *CNIP* needs to gather certain specific *Module Register Addresses* located scattered throughout the entire listing of *Module Register Addresses* into a concise contiguous group or ***Assembly*** that the PLC can efficiently read from and write to. Figure 11 illustrates this concept.

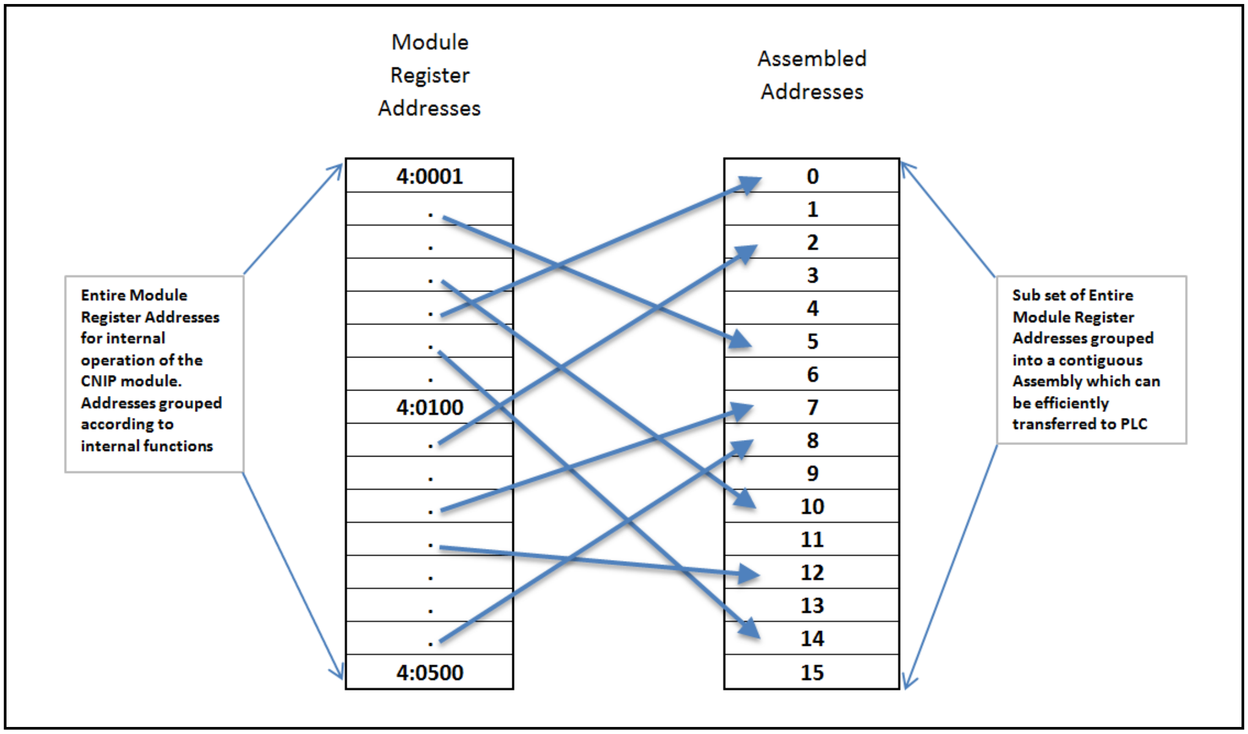


FIGURE 11 - PLC ASSEMBLY CONCEPT

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

Modbus Assembly Instance Structure

Each *CNIP* 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 *CNIP*'s memory structure contains a fixed array of internal data locations that are constructed as Modbus *Holding Registers*. Each *CNIP* 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 *CNIP*'s to read from and write to specific register address locations to achieve inter-module communications. Certain registers are read from and written to by the *EasyRoll* software in order to monitor and/or change default configuration values.

Modbus TCP addressing convention utilizes a "4:xxxx" notation. The "4:" in Modbus protocol designates that the address is a *Holding Register* and the xxxx is a numerical value representing the offset or index for a specific register 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 *CNIP* visible on its network and access Input and Output *Holding Register Assemblies*. The *CNIP* supports the following Modbus TCP Service Codes:

- Service Code 4 - Read Holding Register (up to 45 registers per instruction)
- Service Code 6 - Write Single Register
- Service Code 16 - Write Multiple Registers (up to 45 registers per instruction)
- Service Code 23 - Read/Write Multiple Registers (up to 45 registers per instruction)

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

For example, for the Input Assembly, if you only need to read register 4:2104; your instruction in the PLC must use the starting address of 4:2100 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 4:2104, the *CNIP* will return an error.

Also, you can only read from (and not write to) the 4:21xx registers of Assembly #11 and you can only write to (and not read) the 4:22xx registers of Assembly #12.

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

ETHERNET I/P ASSEMBLY INSTANCE STRUCTURE

When a *CNIP* is attached to an external Ethernet I/P controller (Logix 5000 based PLC), it is done so as a Generic Ethernet I/O device. 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 *CNIP* recognizes only 1 specific input instance type and one specific output instance type.

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:

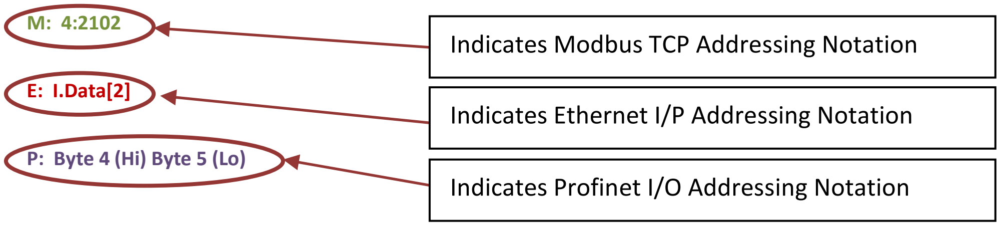
- ***ModuleName* is the user-defined name of the device**
- **"O.Data" indicates the output image of the device**
- **"I.Data" indicates the input image of the device**
- **"[*Index*].*Bit*" indicates the word and bit within the image. If the bit notation is absent the notation refers to the entire word 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.

This manual assumes any reader interested in utilizing Ethernet I/P interface to *CNIP* is experienced with Allen-Bradley Logix 5000 programming software and is familiar with User Defined Data Type structures and attaching Ethernet I/P Generic I/O device instances to a PLC program project.

CNIP ASSEMBLIES FOR PLCS

REGISTER CHART LEGEND



INPUT ASSEMBLY

The following chart lists the contents of the input data image coming from the CNIP to the PLC.

Register Name / Module Address	*Assembled Address for PLC*	*Description*
RS-485 Input 0 **4:0040**	**M: 4:2100** **E: I.Data[0]** **P: Byte 0 (Hi) Byte 1 (Lo)**	Register 1 of 10 register input block read from RS-485 connected device
RS-485 Input 1 **4:0041**	**M: 4:2101** **E: I.Data[1]** **P: Byte 2 (Hi) Byte 3 (Lo)**	Register 2 of 10 register input block read from RS-485 connected device
RS-485 Input 2 **4:0042**	**M: 4:2102** **E: I.Data[2]** **P: Byte 4 (Hi) Byte 5 (Lo)**	Register 3 of 10 register input block read from RS-485 connected device
RS-485 Input 3 **4:0043**	**M: 4:2103** **E: I.Data[3]** **P: Byte 6 (Hi) Byte 7 (Lo)**	Register 4 of 10 register input block read from RS-485 connected device
RS-485 Input 4 **4:0044**	**M: 4:2104** **E: I.Data[4]** **P: Byte 8 (Hi) Byte 9 (Lo)**	Register 5 of 10 register input block read from RS-485 connected device
RS-485 Input 5 **4:0045**	**M: 4:2105** **E: I.Data[5]** **P: Byte 10 (Hi) Byte 11 (Lo)**	Register 6 of 10 register input block read from RS-485 connected device
RS-485 Input 6 **4:0046**	**M: 4:2106** **E: I.Data[6]** **P: Byte 12 (Hi) Byte 13 (Lo)**	Register 7 of 10 register input block read from RS-485 connected device
RS-485 Input 7 **4:0047**	**M: 4:2107** **E: I.Data[7]** **P: Byte 14 (Hi) Byte 15 (Lo)**	Register 8 of 10 register input block read from RS-485 connected device
RS-485 Input 8 **4:0048**	**M: 4:2108** **E: I.Data[8]** **P: Byte 16 (Hi) Byte 17 (Lo)**	Register 9 of 10 register input block read from RS-485 connected device
RS-485 Input 9 **4:0049**	**M: 4:2109** **E: I.Data[9]** **P: Byte 18 (Hi) Byte 19 (Lo)**	Register 10 of 10 register input block read from RS-485 connected device

Register Name / Module Address	*Assembled Address for PLC*	*Description*
RS-485 Errors **4:0079**	**M: 4:2110** **E: I.Data[10]** **P: Byte 20 (Hi) Byte 21 (Lo)**	Bitwise Value - Read only Bit 00 = Reserved Bit 01 = Communication Error Bit 02 = Illegal Function Exception Bit 03 = Illegal Data Address Exception Bit 04 = Illegal Data Value Exception Bit 05 = Slave Device Failure Exception Bit 06 = Acknowledge Exception Bit 07 = Slave Device Busy Exception Bit 08 = Baud rate not supported Bit 09 = Bad RS485 Settings Bit 10 = Number of read registers out of range Bit 11 = Number of write registers out of range Bit 12 = Reserved Bit 13 = Reserved Bit 14 = Reserved Bit 15 = Reserved
Digital Inputs **4:0080**	**M: 4:2111** **E: I.Data[11]** **P: Byte 22 (Hi) Byte 23 (Lo)**	Bitwise Value - Read only Bit 00 = Input terminal I0 Bit 01 = Input terminal I1 Bit 02 = Input terminal I2 Bit 03 = Input terminal I3 Bit 04 = Input terminal I4 Bit 05 = Input terminal I5 Bit 06 = Input terminal I6 Bit 07 = Input terminal I7 Bit 08 = Reserved Bit 09 = Reserved Bit 10 = Reserved Bit 11 = Reserved Bit 12 = Reserved Bit 13 = Reserved Bit 14 = Reserved Bit 15 = Module Heartbeat – 2 sec ON, 2 sec OFF
ConveyStop Status **4:0019**	**M: 4:2112** **E: I.Data[12]** **P: Byte 24 (Hi) Byte 25 (Lo)**	Bitwise Value - Read only 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 *ConveyStop User's Guide* for details on using *ConveyStop*.
Reserved	**M: 4:2113** **E: I.Data[13]** **P: Byte 26 (Hi) Byte 27 (Lo)**	

<table>
<tr><th>Register Name / Module Address</th><th>Assembled Address for PLC</th><th>Description</th></tr>
<tr><td>Upstream Module Status
4:0134</td><td>M: 4:2114
E: I.Data[14]
P: Byte 28 (Hi) Byte 29 (Lo)</td><td rowspan="2">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
0xXX05 = Sensor blocked and motor stopped

See Input Assembly Note on page 33 for more details</td></tr>
<tr><td>Downstream Module Status
4:0232</td><td>M: 4:2115
E: I.Data[15]
P: Byte 30 (Hi) Byte 31 (Lo)</td></tr>
<tr><td>From Upstream Tracking Word 1
4:0139</td><td>M: 4:2116
E: I.Data[16]
P: Byte 32 (Hi) Byte 33 (Lo)</td><td>Value = Tracking data word #1 (16 bit integer) for the Carton that has just discharged from the ERSC Adjacent to this local CNIP module</td></tr>
<tr><td>From Upstream Tracking Word 2
4:0140</td><td>M: 4:2117
E: I.Data[17]
P: Byte 34 (Hi) Byte 35 (Lo)</td><td>Value = Tracking data word #2 (16 bit integer) for the Carton that has just discharged from the ERSC Adjacent to this local CNIP module</td></tr>
</table>

Input Assembly Note ①

Upstream/Downstream Module Status is only valid if there are ConveyLinx modules installed and contain logical upstream and/or downstream connections to the CNIP module. The values shown are for the logical states for ConveyLinx ERSC modules in ZPA mode. If there are no upstream and/or downstream connections to your CNIP module, then any data in these registers is to be ignored.

Please refer to *ConveyLinx PLC Developers Guide* (Insight Automation publication ERSC-1500) for more details on ERSC ZPA mode and applications for utilizing ZPA zone status data.

Output Assembly

The following chart lists the contents of the output data image sent by the PLC to the *CNIP*.

Register Name / Module Address	Assembled Address for PLC	Description
RS-485 Output 0 4:0050	**M: 4:2200** **E: O.Data[0]** **P: Byte 0 (Hi) Byte 1 (Lo)**	Register 1 of 10 register output block written to RS-485 connected device
RS-485 Output 1 4:0051	**M: 4:2201** **E: O.Data[1]** **P: Byte 2 (Hi) Byte 3 (Lo)**	Register 2 of 10 register output block written to RS-485 connected device
RS-485 Output 2 4:0052	**M: 4:2202** **E: O.Data[2]** **P: Byte 4 (Hi) Byte 5 (Lo)**	Register 3 of 10 register output block written to RS-485 connected device
RS-485 Output 3 4:0053	**M: 4:2203** **E: O.Data[3]** **P: Byte 6 (Hi) Byte 7 (Lo)**	Register 4 of 10 register output block written to RS-485 connected device
RS-485 Output 4 4:0054	**M: 4:2204** **E: O.Data[4]** **P: Byte 8 (Hi) Byte 9 (Lo)**	Register 5 of 10 register output block written to RS-485 connected device
RS-485 Output 5 4:0055	**M: 4:2205** **E: O.Data[5]** **P: Byte 10 (Hi) Byte 11 (Lo)**	Register 6 of 10 register output block written to RS-485 connected device
RS-485 Output 6 4:0056	**M: 4:2206** **E: O.Data[6]** **P: Byte 12 (Hi) Byte 13 (Lo)**	Register 7 of 10 register output block written to RS-485 connected device
RS-485 Output 7 4:0057	**M: 4:2207** **E: O.Data[7]** **P: Byte 14 (Hi) Byte 15 (Lo)**	Register 8 of 10 register output block written to RS-485 connected device
RS-485 Output 8 4:0058	**M: 4:2208** **E: O.Data[8]** **P: Byte 16 (Hi) Byte 17 (Lo)**	Register 9 of 10 register output block written to RS-485 connected device
RS-485 Output 9 4:0059	**M: 4:2209** **E: O.Data[9]** **P: Byte 18 (Hi) Byte 19 (Lo)**	Register 10 of 10 register output block written to RS-485 connected device
Set Status to Upstream Module 4:0116	**M: 4:2210** **E: O.Data[10]** **P: Byte 20 (Hi) Byte 21 (Lo)**	Used to inform upstream ConveyLinx ZPA module to stop or release its downstream zone. See Input Assembly Note on page 33 for more details
Set Status to Downstream Module 4:0196	**M: 4:2211** **E: O.Data[11]** **P: Byte 22 (Hi) Byte 23 (Lo)**	Used to inform downstream ConveyLinx ZPA module to wake up its upstream zone See *Input Assembly Note* ① on page 33 for more details

Register Name / Module Address	*Assembled Address for PLC*	*Description*
Set Downstream Tracking Word 1 4:0201	**M: 4:2212 E: O.Data[12] P: Byte 24 (Hi) Byte 25 (Lo)**	Only used when local CNIP needs to pass tracking data to a downstream connected ConveyLinx ERSC in ZPA Mode. Used in conjunction with *Set Status to Downstream Module (O.Data[11] / 4:2111)* register
Set Downstream Tracking Word 2 4:0202	**M: 4:2213 E: O.Data[13] P: Byte 26 (Hi) Byte 27 (Lo)**	
Reserved	**M: 4:2214 E: O.Data[14] P: Byte 28 (Hi) Byte 29 (Lo)**	
Reserved	**M: 4:2215 E: O.Data[15] P: Byte 30 (Hi) Byte 31 (Lo)**	
Reserved	**M: 4:2216 E: O.Data[16] P: Byte 32 (Hi) Byte 33 (Lo)**	
Reserved	**M: 4:2217 E: O.Data[17] P: Byte 34 (Hi) Byte 35 (Lo)**	
Reserved	**M: 4:2218 E: O.Data[18] P: Byte 36 (Hi) Byte 37 (Lo)**	
Reserved	**M: 4:2219 E: O.Data[19] P: Byte 38 (Hi) Byte 39 (Lo)**	
RS-485 Slave ID 4:0070	**M: 4:2220 E: O.Data[20] P: Byte 40 (Hi) Byte 41 (Lo)**	RS-485 Slave ID of the remote RS-485 device
RS-485 Read Block Start Address 4:0071	**M: 4:2221 E: O.Data[21] P: Byte 42 (Hi) Byte 43 (Lo)**	Modbus address of first register of contiguous block of registers to be read from remote device
RS-485 Number or Registers to Read 4:0072	**M: 4:2222 E: O.Data[22] P: Byte 44 (Hi) Byte 45 (Lo)**	Quantity of contiguous Modbus registers to be read from remote device. Valid values are from 1 to 10
RS-485 Write Block Start Address 4:0073	**M: 4:2223 E: O.Data[23] P: Byte 46 (Hi) Byte 47 (Lo)**	Modbus address of first register of contiguous block of registers to be written to the remote device
RS-485 Number or Registers to Write 4:0074	**M: 4:2224 E: O.Data[24] P: Byte 48 (Hi) Byte 49 (Lo)**	Quantity of contiguous Modbus registers to be written to the remote device. Valid values are from 1 to 10
RS-485 Baud Rate 4:0075	**M: 4:2225 E: O.Data[25] P: Byte 50 (Hi) Byte 51 (Lo)**	Numerical value to set the baud rate 0 = 1200 1 = 2400 2 = 4800 3 = 9600 4 = 14400 5 = 19200 6 = 38400 7 = 56000 8 = 57600 9 = 115200

Register Name / Module Address	*Assembled Address for PLC*	*Description*
RS-485 Port Settings 4:0076	**M: 4:2226 E: O.Data[26] P: Byte 52 (Hi) Byte 53 (Lo)**	Bit 00 ➔ 0 = 1 Stop Bit – 1 = 2 Stop Bits Bit 01 ➔ 0 = 8 Data Bits – 1 = 7 Data Bits Bit 02 ➔ Parity – See *Output Assembly Note* ④ Bit 03 ➔ Parity – See *Output Assembly Note* ④ Bit 04 ➔ Reserved Bit 05 ➔ Reserved Bit 06 ➔ Reserved Bit 07 ➔ Reserved Bit 08 ➔ Modbus Write Mode – See *Output Assembly Note* ② Bit 09 ➔ Reserved Bit 10 ➔ Reserved Bit 11 ➔ Reserved Bit 12 ➔ Reserved Bit 13 ➔ Reserved Bit 14 ➔ Reserved Bit 15 ➔ Reserved
RS-485 Scan Rate 4:0077	**M: 4:2227 E: O.Data[27] P: Byte 54 (Hi) Byte 55 (Lo)**	The rate in milliseconds the *CNIP* performs read/write transactions with the RS-485 remote device
RS-485 Response Time Out 4:0078	**M: 4:2228 E: O.Data[28] P: Byte 56 (Hi) Byte 57 (Lo)**	The value in milliseconds
Digital Outputs 4:0081	**M: 4:2229 E: O.Data[29] P: Byte 58 (Hi) Byte 59 (Lo)**	Bit 0 ➔ Output terminal O0 Bit 1 ➔ Output terminal O1 Bit 2 ➔ Output terminal O2 Bit 3 ➔ Output terminal O3 Bit 4 ➔ Output terminal O4 Bit 5 ➔ Output terminal O5 Bit 6 ➔ Reserved Bit 7 ➔ Reserved Bit 08 ➔ Reserved Bit 09 ➔ Reserved Bit 10 ➔ Reserved Bit 11 ➔ Reserved Bit 12 ➔ Reserved Bit 13 ➔ Reserved Bit 14 ➔ Reserved Bit 15 ➔ Reserved
Digital Outputs Defaults 4:0082	**M: 4:2230 E: O.Data[30] P: Byte 60 (Hi) Byte 61 (Lo)**	Output State for PLC Communication Loss See *Output Assembly Note* ③ Output O0: Bit 0 ➔ 0 = Disable - 1 = Enable; Bit 1 ➔ 0 = Turn OFF – 1 = Turn ON Output O1: Bit 2 ➔ 0 = Disable - 1 = Enable; Bit 3 ➔ 0 = Turn OFF – 1 = Turn ON Output O2: Bit 4 ➔ 0 = Disable - 1 = Enable; Bit 5 ➔ 0 = Turn OFF – 1 = Turn ON Output O3: Bit 6 ➔ 0 = Disable - 1 = Enable; Bit 7 ➔ 0 = Turn OFF – 1 = Turn ON Output O4: Bit 8 ➔ 0 = Disable - 1 = Enable; Bit 9 ➔ 0 = Turn OFF – 1 = Turn ON Output O5: Bit 10 ➔ 0 = Disable - 1 = Enable; Bit 11 ➔ 0 = Turn OFF – 1 = Turn ON

Register Name / Module Address	*Assembled Address for PLC*	*Description*
Reserved	**M: 4:2231** **E: O.Data[31]** **P: Byte 62 (Hi) Byte 63 (Lo)**	
ConveyStop Command 4:0020	**M: 4:2232** **E: O.Data[32]** **P: Byte 64 (Hi) Byte 65 (Lo)**	Integer Value 0 = No Command 1 = Command local module's Stop Group to go to Stopped State 2 = Command local module's Stop Group to Clear Stopped State Refer to Insight Automation publication ERSC-1800 *ConveyStop User's Guide* for details on using *ConveyStop*.

Output Assembly Note ②

Some Modbus devices support writing of multiple registers in one Modbus command and others will only allow one register to be written per Modbus transaction. This bit allows you to set how the *CNIP* RS-485 port will handle register write function. When this bit is reset to 0, the *CNIP* will perform a multiple register write in a single transaction. When this bit is set to 1, the *CNIP* will only allow one register write per single transaction.

The transaction rate is controlled by the value in the *RS-485 Scan Rate* register ([*ModuleName*]:O.Data[27] / 4:2227). For example, let's assume the Scan Rate is set to 100 msec. and the number of registers to read is 3 and the number of registers to write is 3. If the Modbus Write Mode bit is 1 (i.e. single register write), the cycle of transactions would be like the following:

- Modbus Command 03 – Read Multiple Registers – Read all 3 Input Registers
- Scan Rate Delay (100 msec)
- Modbus Command 06 – Write Single Register – Write Output Register 1 of 3
- Scan Rate Delay (100 msec)
- Modbus Command 06 – Write Single Register – Write Output Register 2 of 3
- Scan Rate Delay (100 msec)
- Modbus Command 06 – Write Single Register – Write Output Register 3 of 3
- Scan Rate Delay (100 msec)

If the Modbus Write Mode bit is 0 (i.e. multiple register write), the cycle of transactions would be like the following:

- Modbus Command 03 – Read Multiple Registers – Read all 3 Input Regsiters
- Scan Rate Delay (100 msec)
- Modbus Command 16 – Write Multiple Registers – Write all 3 Output Registers
- Scan Rate Delay (100 msec)

OUTPUT ASSEMBLY NOTE ③

The *Digital Outputs Default* register is used to control the behavior of the digital Outputs upon detecting the loss of communications from the PLC. The possibilities are:

- Leave the output in its last state (No Change)
- Force Output ON
- Force Output OFF

This register contains 2 bit pairs that control this function for each of the digital Outputs. The first bit of the pair enables the function. If enabled (first bit in the pair set to 1), the output uses the value of the other bit in the pair to determine its state upon loss of communication. If the first bit is disabled (first bit in the pair reset to 0), the output will remain in its last state at the time of the communication loss and ignore the other bit in the pair.

IMPORTANT NOTE: By default, if no change is made to the *Digital Outputs Default* register, ALL outputs remain in their last state if communications is lost to the PLC. Be sure to review your application to insure that this behavior is acceptable!

OUTPUT ASSEMBLY NOTE ④

Bit 2 and Bit 3 of the *RS-485 Settings* register determine parity settings for the port. The following chart indicates the settings:

Bit 2	Bit 3	Parity
0	0	None
0	1	Odd
1	0	Even

Input and Output Assemblies with Reset Protection

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

Please note that this functionality is only applicable for Ethernet I/P and Modbus TCP PLCs. Functionality is not applicable or used with Profinet I/O PLCs.

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

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

When using Assemblies with Reset Protection, if *Module Reset Count* (I.Data[13] / 4:2113) is NOT Equal to *Set Module Reset Count* (O.Data[31] / 4:2231), then the *CNIP* module does not respond to output coming from the PLC. The PLC can use this not equal condition to know that the *CNIP* module has been reset (power cycle or network reconnected after disconnect). When the PLC is ready to allow the *CNIP* to begin servicing the outputs it is sending to the *CNIP* module, the PLC logic simply copies the contents of the *Module Reset Count* register to the *Set Module Reset Count* register. When the *CNIP* module detects that these two values are equal, it will begin processing the PLC output data being written to it by the PLC.

INPUT ASSEMBLY WITH RESET PROTECTION

ERSC Internal Address	*Register Name*	*Assembled Ethernet I/P Address for PLC*	*Assembled Modbus Address for PLC*
4:0040	RS-485 Input 0	E: I.Data[0]	M: 4:4100
4:0041	RS-485 Input 1	E: I.Data[1]	M: 4:4101
4:0042	RS-485 Input 2	E: I.Data[2]	M: 4:4102
4:0043	RS-485 Input 3	E: I.Data[3]	M: 4:4103
4:0044	RS-485 Input 4	E: I.Data[4]	M: 4:4104
4:0045	RS-485 Input 5	E: I.Data[5]	M: 4:4105
4:0046	RS-485 Input 6	E: I.Data[6]	M: 4:4106
4:0047	RS-485 Input 7	E: I.Data[7]	M: 4:4107
4:0048	RS-485 Input 8	E: I.Data[8]	M: 4:4108
4:0049	RS-485 Input 9	E: I.Data[9]	M: 4:4109
4:0079	RS-485 Errors	E: I.Data[10]	M: 4:4110
4:0080	Digital Inputs	E: I.Data[11]	M: 4:4111
4:0019	ConveyStop Status	E: I.Data[12]	M: 4:4112
N/A	**Current Module Reset Count**	**E: I.Data[13]**	**M: 4:4113**
4:0134	Upstream Module Status	E: I.Data[14]	M: 4:4114
4:0232	Downstream Module Status	E: I.Data[15]	M: 4:4115
4:0139	From Upstream Tracking Word 1	E: I.Data[16]	M: 4:4116
4:0140	From Upstream Tracking Word 2	E: I.Data[17]	M: 4:4117

Output Assembly with Reset Protection

ERSC Internal Address	Register Name	Assembled Ethernet I/P Address for PLC	Assembled Modbus Address for PLC
4:0050	RS-485 Output 0	E: O.Data[0]	M: 4:4200
4:0051	RS-485 Output 1	E: O.Data[1]	M: 4:4201
4:0052	RS-485 Output 2	E: O.Data[2]	M: 4:4202
4:0053	RS-485 Output 3	E: O.Data[3]	M: 4:4203
4:0054	RS-485 Output 4	E: O.Data[4]	M: 4:4204
4:0055	RS-485 Output 5	E: O.Data[5]	M: 4:4205
4:0056	RS-485 Output 6	E: O.Data[6]	M: 4:4206
4:0057	RS-485 Output 7	E: O.Data[7]	M: 4:4207
4:0058	RS-485 Output 8	E: O.Data[8]	M: 4:4208
4:0059	RS-485 Output 9	E: O.Data[9]	M: 4:4209
4:0116	Set Status to Upstream Module	E: O.Data[10]	M: 4:4210
4:0196	Set Status to Downstream Module	E: O.Data[11]	M: 4:4211
4:0201	Set Downstream Tracking Word 1	E: O.Data[12]	M: 4:4212
4:0202	Set Downstream Tracking Word 2	E: O.Data[13]	M: 4:4213
N/A	Reserved	E: O.Data[14]	M: 4:4214
N/A	Reserved	E: O.Data[15]	M: 4:4215
N/A	Reserved	E: O.Data[16]	M: 4:4216
N/A	Reserved	E: O.Data[17]	M: 4:4217
N/A	Reserved	E: O.Data[18]	M: 4:4218
N/A	Reserved	E: O.Data[19]	M: 4:4219
4:0070	RS-485 Slave ID	E: O.Data[20]	M: 4:4220
4:0071	RS-485 Read Block Start Address	E: O.Data[21]	M: 4:4221
4:0072	RS-485 Number or Registers to Read	E: O.Data[22]	M: 4:4222
4:0073	RS-485 Write Block Start Address	E: O.Data[23]	M: 4:4223
4:0074	RS-485 Number or Registers to Write	E: O.Data[24]	M: 4:4224
4:0075	RS-485 Baud Rate	E: O.Data[25]	M: 4:4225
4:0076	RS-485 Port Settings	E: O.Data[26]	M: 4:4226
4:0077	RS-485 Scan Rate	E: O.Data[27]	M: 4:4227
4:0078	RS-485 Response Time Out	E: O.Data[28]	M: 4:4228
4:0081	Digital Outputs	E: O.Data[29]	M: 4:4229
4:0082	Digital Outputs Defaults	E: O.Data[30]	M: 4:4230
N/A	**Set Module Reset Count**	**E: O.Data[31]**	**M: 4:4231**
4:0020	ConveyStop Command	E: O.Data[32]	M: 4:4232

Ethernet I/P Instances for *CNIP*

This section provides instructions on how to attach a CNIP module to a Rockwell Logix-based platform PLC.

Ethernet I/P Guidelines

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

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

If the limit of either of these quantities is reached, the PLC processor will indicate I/O communications fault on one or more instances of device declaration.

For *CNIP* device declarations in general, the PLC limitation on TCP connections will be reached before I/O data table memory limit is realized.

For example, for a CompactLogix L3x series processor, the documented quantity of TCP connections available on its Ethernet Port is 32. The processor always keeps one TCP connection in reserve for programming terminal access, etc. An L3x series processor can accept 31 full-time *CNIP C*onnections as generic I/O modules.

When a *CNIP* is attached as a “full-time generic I/O module” to the PLC, the connection is continually maintained and data is exchanged at a minimum RPI value and if the PLC cannot communicate with the *CNIP* for any reason, the PLC’s I/O tree will register a fault.

For more information on determining the design and capacity of your Ethernet I/P network; please refer to Allen-Bradley document *EtherNet/IP Performance Application Solution* (publication ENET-AP001D-EN-P).

Instance IDs for CNIP Assemblies

Each Input and Output Assembly requires an Instance value that is used in the set-up of the connection from the PLC to a CNIP module. Along with this Instance value is a Size value which indicates the quantity of data elements in the assembly. These two values are used as input in the RSLogix 5000 software when you create a new connection for your CNIP module in your program.

The instance values you need to use can be dependent upon the version of firmware that is installed on your CNIP.

ODVA Compliant Firmware 5.2

Insight Automation has been granted a Certificate of Conformity from ODVA for ConveyLinx firmware version 5.2. The main difference between firmware 5.2 and previous versions (4.xx, 3.xx) as it pertains to connecting to ODVA compliant Ethernet I/P (EIP) PLC devices is that these previous versions utilized Instance Identifiers that were classified as “reserved” by the ODVA specification.

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

Parameters for Each Assembly

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

For Firmware 4.24 and Earlier

Assembly	Type	Instance Value	Size Value
Input & Output Assemblies	**Input**	**11**	**18**
	Output	**12**	**33**
Input & Output Assemblies with Reset Protection	**Input**	**31**	**18**
	Output	**32**	**33**

For Firmware 4.25

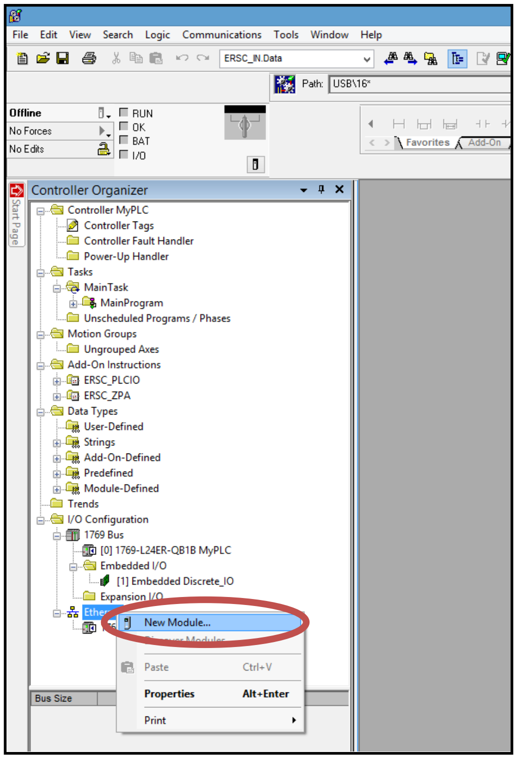
Assembly	Type	Instance Value	Size Value
Input & Output Assemblies	**Input**	**11 or 111**	**18**
	Output	**12 or 112**	**33**
Input & Output Assemblies with Reset Protection	**Input**	**31 or 311**	**18**
	Output	**32 or 312**	**33**

For Firmware 5.2

Assembly	Type	Instance Value	Size Value
Input & Output Assemblies	**Input**	**111**	**18**
	Output	**112**	**33**
Input & Output Assemblies with Reset Protection	**Input**	**311**	**18**
	Output	**312**	**33**

Connecting to Logix PLC

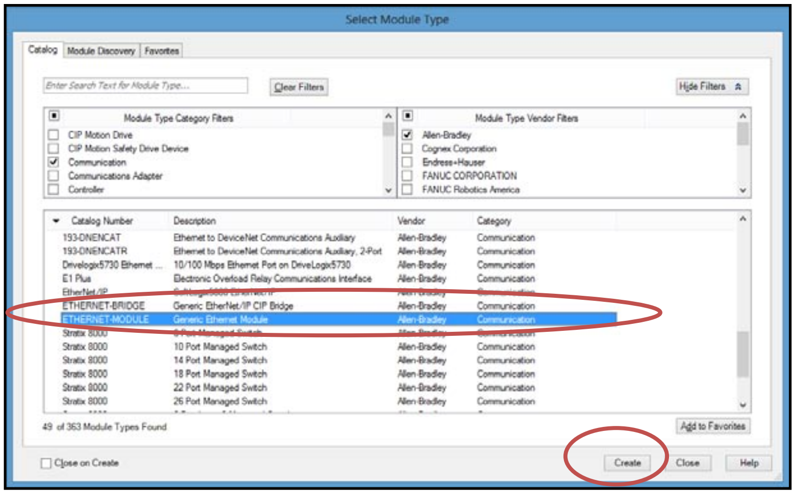
This section will provide the set-by-step procedure for creating an instance of a *CNIP* into the I/O configuration for an Allen-Bradley CompactLogix processor in RSLogix 5000 software.



Step #1

Add a New Module to the processor's I/O configuration by highlighting the processor's local Ethernet port in the I/O configuration tree.

Right-clicking will show the context menu. Select "New Module…"



Step #2

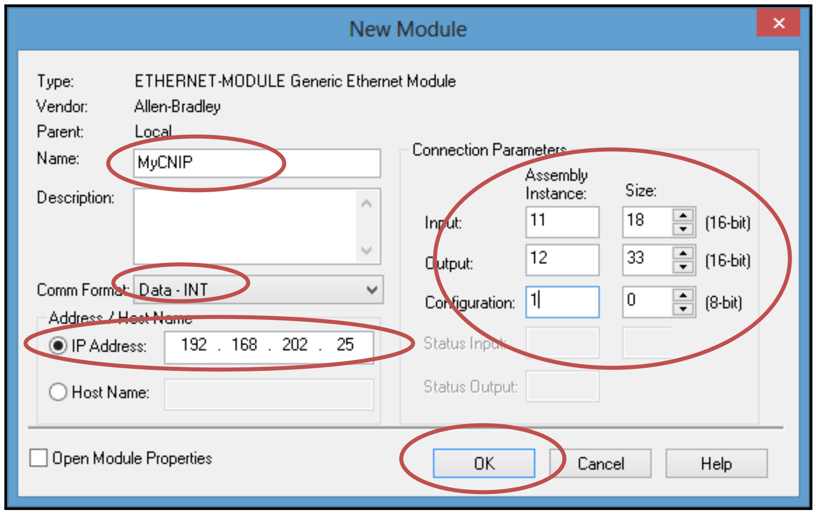
From the Select Module pop-up window, expand the Communications tree and select "Generic Ethernet Module" and click OK

Step #3

Fill in the Name field. This will be the *ModuleName* that will appear in your program Tag Database for any addressing.

Select Comm Format to be "Data – INT" and fill in the I.P. address of the *CNIP*.

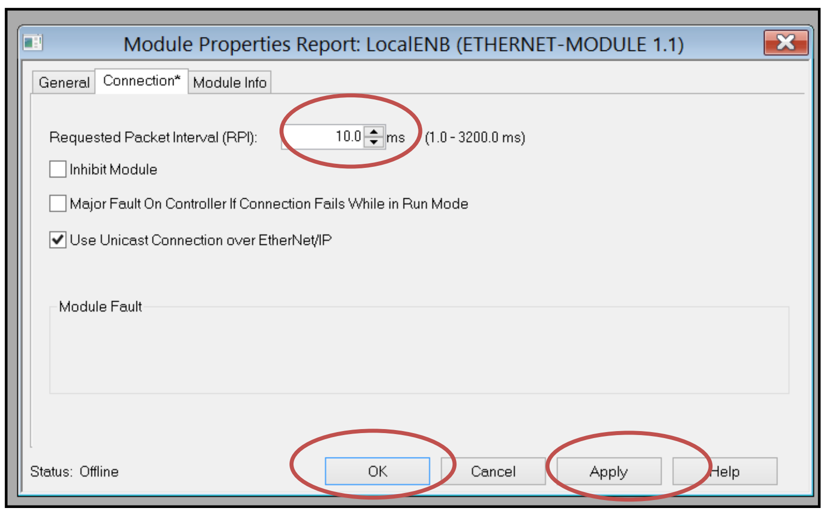
Fill in the Connection Parameters as shown.



Step #4

Set RPI to a value no lower than 10ms. You may also optionally select Unicast Connection.

Click "Apply" to update the value and then "OK" to exit the window.



It is <u>very important</u> to select *Comm Format* data type to be <u>*Data-INT*</u> or interface to *CNIP* will not operate correctly!

Access Data with Controller Tags

Once you have created your CNIP module, you access the input and output data from the controller tags automatically generated. Figure 12 and Figure 13 show the controller tags for the input instance assembly and the output instance assembly respectively. Note that the number of elements in each data arrays matches the size values entered when you created the *CNIP*.

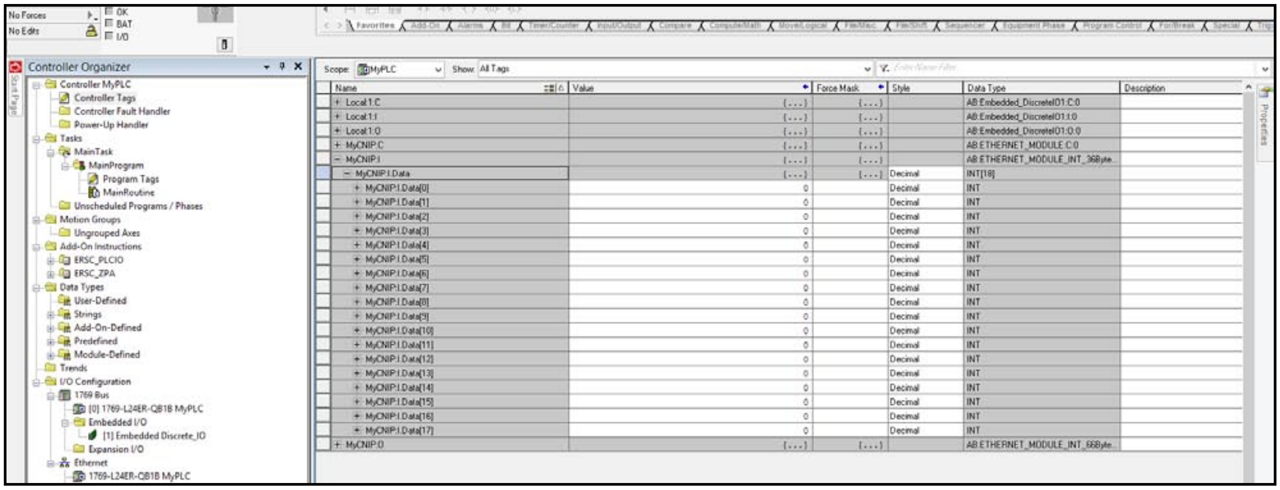


FIGURE 12 - CONTROLLER TAGS FOR THE INPUT INSTANCE

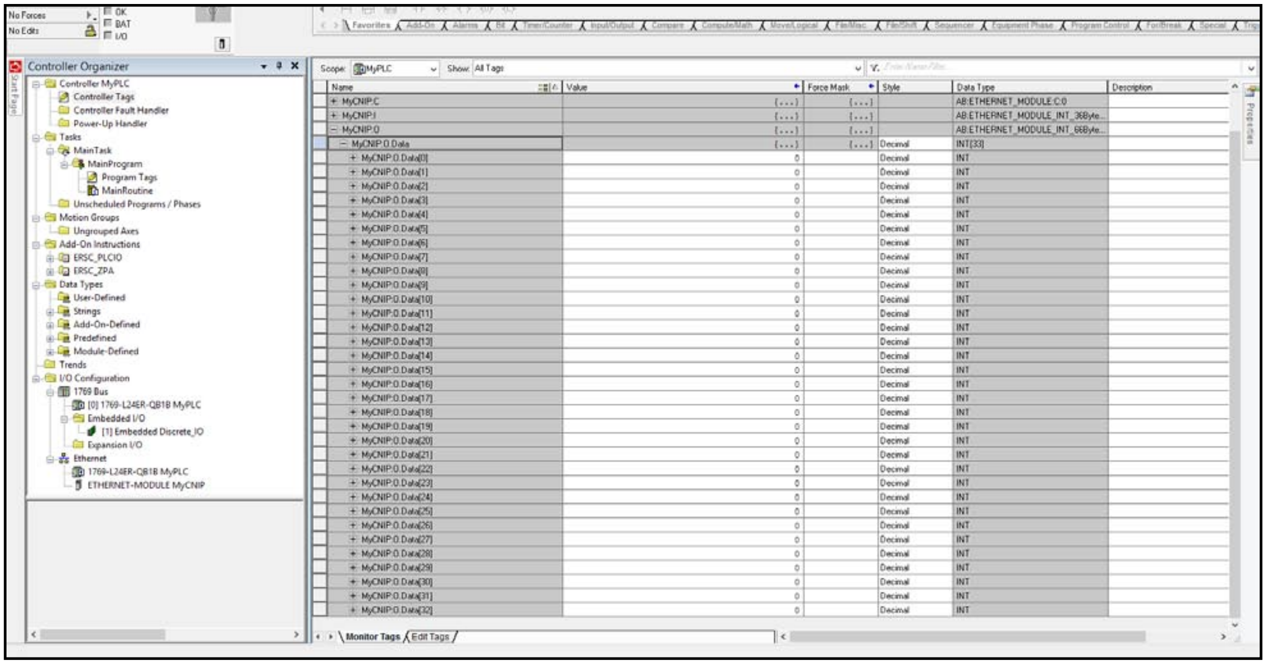


FIGURE 13 - CONTROLLER TAGS FOR THE OUTPUT INSTANCE

Modbus Addresses for Firmware 5.2

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

For Firmware 5.2, the original addresses are no longer valid and this appendix will show each table with the new ODVA compliant Modbus register address ranges for each assembly.

All register Assembly tables in this guide show the Modbus addresses with their original assignments that are still valid for Firmware versions 4.25 and older.

Input Assembly

ERSC Internal Address	*Register Name*	*Assembled Address for FW 5.2*	*Original Assembled Address for FW 4.25 and earlier*
4:0040	RS-485 Input 0	M: 4:12100	M: 4:2100
4:0041	RS-485 Input 1	M: 4:12101	M: 4:2101
4:0042	RS-485 Input 2	M: 4:12102	M: 4:2102
4:0043	RS-485 Input 3	M: 4:12103	M: 4:2103
4:0044	RS-485 Input 4	M: 4:12104	M: 4:2104
4:0045	RS-485 Input 5	M: 4:12105	M: 4:2105
4:0046	RS-485 Input 6	M: 4:12106	M: 4:2106
4:0047	RS-485 Input 7	M: 4:12107	M: 4:2107
4:0048	RS-485 Input 8	M: 4:12108	M: 4:2108
4:0049	RS-485 Input 9	M: 4:12109	M: 4:2109
4:0079	RS-485 Errors	M: 4:12110	M: 4:2110
4:0080	Digital Inputs	M: 4:12111	M: 4:2111
4:0019	ConveyStop Status	M: 4:12112	M: 4:2112
N/A	Reserved	M: 4:12113	M: 4:2113
4:0134	Upstream Module Status	M: 4:12114	M: 4:2114
4:0232	Downstream Module Status	M: 4:12115	M: 4:2115
4:0139	From Upstream Tracking Word 1	M: 4:12116	M: 4:2116
4:0140	From Upstream Tracking Word 2	M: 4:12117	M: 4:2117

Output Assembly

ERSC Internal Address	Register Name	Assembled Address for FW 5.2	Original Assembled Address for FW 4.25 and earlier
4:0050	RS-485 Output 0	M: 4:12200	M: 4:2200
4:0051	RS-485 Output 1	M: 4:12201	M: 4:2201
4:0052	RS-485 Output 2	M: 4:12202	M: 4:2202
4:0053	RS-485 Output 3	M: 4:12203	M: 4:2203
4:0054	RS-485 Output 4	M: 4:12204	M: 4:2204
4:0055	RS-485 Output 5	M: 4:12205	M: 4:2205
4:0056	RS-485 Output 6	M: 4:12206	M: 4:2206
4:0057	RS-485 Output 7	M: 4:12207	M: 4:2207
4:0058	RS-485 Output 8	M: 4:12208	M: 4:2208
4:0059	RS-485 Output 9	M: 4:12209	M: 4:2209
4:0116	Set Status to Upstream Module	M: 4:12210	M: 4:2210
4:0196	Set Status to Downstream Module	M: 4:12211	M: 4:2211
4:0201	Set Downstream Tracking Word 1	M: 4:12212	M: 4:2212
4:0202	Set Downstream Tracking Word 2	M: 4:12213	M: 4:2213
N/A	Reserved	M: 4:12214	M: 4:2214
N/A	Reserved	M: 4:12215	M: 4:2215
N/A	Reserved	M: 4:12216	M: 4:2216
N/A	Reserved	M: 4:12217	M: 4:2217
N/A	Reserved	M: 4:12218	M: 4:2218
N/A	Reserved	M: 4:12219	M: 4:2219
4:0070	RS-485 Slave ID	M: 4:12220	M: 4:2220
4:0071	RS-485 Read Block Start Address	M: 4:12221	M: 4:2221
4:0072	RS-485 Number or Registers to Read	M: 4:12222	M: 4:2222
4:0073	RS-485 Write Block Start Address	M: 4:12223	M: 4:2223
4:0074	RS-485 Number or Registers to Write	M: 4:12224	M: 4:2224
4:0075	RS-485 Baud Rate	M: 4:12225	M: 4:2225
4:0076	RS-485 Port Settings	M: 4:12226	M: 4:2226
4:0077	RS-485 Scan Rate	M: 4:12227	M: 4:2227
4:0078	RS-485 Response Time Out	M: 4:12228	M: 4:2228
4:0081	Digital Outputs	M: 4:12229	M: 4:2229
4:0082	Digital Outputs Defaults	M: 4:12230	M: 4:2230
N/A	Reserved	M: 4:12231	M: 4:2231
4:0020	ConveyStop Command	M: 4:12232	M: 4:2232

Input Assembly with Reset Protection

ERSC Internal Address	Register Name	Assembled Address for FW 5.2	Original Assembled Address for FW 4.25 and earlier
4:0040	RS-485 Input 0	M: 4:32100	M: 4:4100
4:0041	RS-485 Input 1	M: 4:32101	M: 4:4101
4:0042	RS-485 Input 2	M: 4:32102	M: 4:4102
4:0043	RS-485 Input 3	M: 4:32103	M: 4:4103
4:0044	RS-485 Input 4	M: 4:32104	M: 4:4104
4:0045	RS-485 Input 5	M: 4:32105	M: 4:4105
4:0046	RS-485 Input 6	M: 4:32106	M: 4:4106
4:0047	RS-485 Input 7	M: 4:32107	M: 4:4107
4:0048	RS-485 Input 8	M: 4:32108	M: 4:4108
4:0049	RS-485 Input 9	M: 4:32109	M: 4:4109
4:0079	RS-485 Errors	M: 4:32110	M: 4:4110
4:0080	Digital Inputs	M: 4:32111	M: 4:4111
4:0019	ConveyStop Status	M: 4:32112	M: 4:4112
N/A	Current Module Reset Count	M: 4:32113	M: 4:4113
4:0134	Upstream Module Status	M: 4:32114	M: 4:4114
4:0232	Downstream Module Status	M: 4:32115	M: 4:4115
4:0139	From Upstream Tracking Word 1	M: 4:32116	M: 4:4116
4:0140	From Upstream Tracking Word 2	M: 4:32117	M: 4:4117

Output Assembly with Reset Protection

ERSC Internal Address	Register Name	Assembled Address for FW 5.2	Original Assembled Address for FW 4.25 and earlier
4:0050	RS-485 Output 0	M: 4:32200	M: 4:4200
4:0051	RS-485 Output 1	M: 4:32201	M: 4:4201
4:0052	RS-485 Output 2	M: 4:32202	M: 4:4202
4:0053	RS-485 Output 3	M: 4:32203	M: 4:4203
4:0054	RS-485 Output 4	M: 4:32204	M: 4:4204
4:0055	RS-485 Output 5	M: 4:32205	M: 4:4205
4:0056	RS-485 Output 6	M: 4:32206	M: 4:4206
4:0057	RS-485 Output 7	M: 4:32207	M: 4:4207
4:0058	RS-485 Output 8	M: 4:32208	M: 4:4208
4:0059	RS-485 Output 9	M: 4:32209	M: 4:4209
4:0116	Set Status to Upstream Module	M: 4:32210	M: 4:4210
4:0196	Set Status to Downstream Module	M: 4:32211	M: 4:4211
4:0201	Set Downstream Tracking Word 1	M: 4:32212	M: 4:4212
4:0202	Set Downstream Tracking Word 2	M: 4:32213	M: 4:4213
N/A	Reserved	M: 4:32214	M: 4:4214
N/A	Reserved	M: 4:32215	M: 4:4215
N/A	Reserved	M: 4:32216	M: 4:4216
N/A	Reserved	M: 4:32217	M: 4:4217
N/A	Reserved	M: 4:32218	M: 4:4218
N/A	Reserved	M: 4:32219	M: 4:4219
4:0070	RS-485 Slave ID	M: 4:32220	M: 4:4220
4:0071	RS-485 Read Block Start Address	M: 4:32221	M: 4:4221
4:0072	RS-485 Number or Registers to Read	M: 4:32222	M: 4:4222
4:0073	RS-485 Write Block Start Address	M: 4:32223	M: 4:4223
4:0074	RS-485 Number or Registers to Write	M: 4:32224	M: 4:4224
4:0075	RS-485 Baud Rate	M: 4:32225	M: 4:4225
4:0076	RS-485 Port Settings	M: 4:32226	M: 4:4226
4:0077	RS-485 Scan Rate	M: 4:32227	M: 4:4227
4:0078	RS-485 Response Time Out	M: 4:32228	M: 4:4228
4:0081	Digital Outputs	M: 4:32229	M: 4:4229
4:0082	Digital Outputs Defaults	M: 4:32230	M: 4:4230
N/A	Set Module Reset Count	M: 4:32231	M: 4:4231
4:0020	ConveyStop Command	M: 4:32232	M: 4:4232

PowerFlex 4 VFD Example

Allen-Bradley PowerFlex 4 VFD's are equipped with RS-485 ports that communicate Modbus RTU. A straight-forward application for a *CNIP* module is to connect the *CNIP* RS-485 port to a Power-Flex 4 VFD and use the PLC to command the VFD to run and change its speed reference over Ethernet I/P using the *CNIP* as a gateway. Our example will show how to populate the *CNIP* Input Assembly registers with the PLC in order to configure the *CNIP* to connect to the VFD. Because we will have read/write capability with the Power-Flex 4, the example will also read the speed and error status from the VFD.

The RS-485 / Modbus RTU information for this example was taken from Allen-Bradley Publication 22A-UM001G-EN-E – April 2007 – Appendix C. Please refer to the latest pertinent revision of Allen-Bradley publications for Power-Flex 4 drives for your application to verify that the information in this example is still valid.

Please note that several VFD parameters have to be set manually on the operator keypad to enable the drive to operate using its internal RS-485 port.

According to the Power-Flex 4 documentation there are two registers that can be written to by an external device to command the drive to run and to give it a speed reference. These are

- Address 8192 – Bit-wise values to start/stop the drive as well as set the direction, etc.
- Address 8193 – Holds integer value for speed reference (i.e. 600 = 60 Hz, etc.)

The statuses that can be read from the Power-Flex 4 are in the following registers:

- Address 8448 – Bit-wise logic status
- Address 8449 – Drive error code values
- Address 8451 – Feedback of output frequency

Let's assume we have set up the VFD's communications to be:

- 9600 Baud Rate
- 8 Data Bits, 1 Stop Bit, No Parity
- Scan Rate of 100 msec
- Response Timeout of 200 msec
- Writing to the VFD will be single register mode and not multiple register mode

The following chart shows what the data and the Output Assembly registers to be used to set up the *CNIP* to read/write to the example VFD:

Item	Output Assembly Register	Value	Comments
Slave ID	O.Data[20]	1	*CNIP* only addresses one and only Slave ID. This value is always 1
Start Address of registers to READ	O.Data[21]	8448	Starting Address of drive status registers per Power-Flex 4 documentation
Quantity of registers to READ	O.Data[22]	4	Need to read from 8448 up to 8451 which is a span of 4 registers
Start Address of registers to WRITE	O.Data[23]	8192	Starting Address of drive command registers per Power-Flex 4 documentation
Quantity of registers to WRITE	O.Data[24]	2	Need to write to 8192 and 8193 only which is a span of 2 registers
Baud Rate	O.Data[25]	3	Per the Instance #12 charts a value of 3 corresponds to a 9600 baud rate setting
RS-485 Settings	O.Data[26]	256	8, None, 1 setting is all 0's for the low byte. Bit 8 needs to be set for single register write mode, which corresponds to a decimal value of 256
Scan Rate	O.Data[27]	100	Must be non-zero and non-negative
Response Timeout	O.Data[28]	200	Must be non-zero and non-negative

Once implemented and the PLC is reading and writing to the *CNIP* and the *CNIP* is exchanging data with the VFD; the PLC program will receive the following data in the Input Assembly:

- I.Data[0] = VFD Register 8448
- I.Data[1] = VFD Register 8449
- I.Data[2] = VFD Register 8450
- I.Data[3] = VFD Register 8451

The PLC can write the speed reference command to O.Data[1] and then set the appropriate bits in O.Data[0] to command the VFD to run, stop, change direction, etc. per the Power-Flex 4 documentation.

If you do not need to use the RS-485 port, be sure to enter "0" values in the all of the above RS-485 registers. This will indicate to the *CNIP* that the RS-485 port is not being used and will not report error status back to the PLC.

Please note that if there are non-zero values in the RS-485 registers in the above chart and no device is connected to the RS-485 port; the *CNIP* will report errors in the *RS-485 Errors* register and the Module Status LED will blink red indicating a connection error.

Please note that if there is a loss of communications between the *CNIP* and the PLC; all read/write activity on the RS-485 port will stop. Take care to understand and configure your RS-485 device accordingly in the event of this condition.

Appendix A – Mounting Dimensions

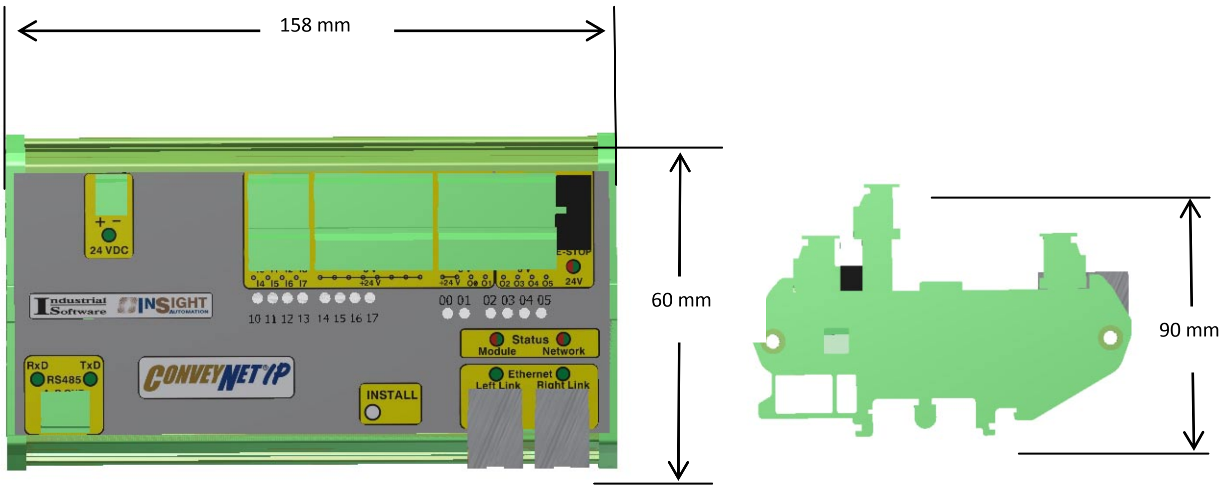


FIGURE 14 : BASIC DIMENSIONING OF THE *CNIP* MODULE

CNIP Module Mounting

The *CNIP* module is designed to be mounted on a DIN rail (EN50022-35x7.5 or equivalent).

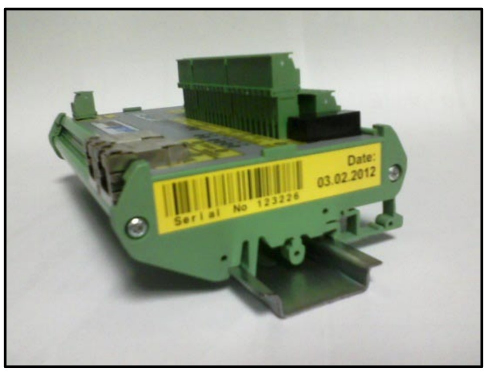


FIGURE 30: *CNIP* MODULE MOUNTED ON A TOP HAT EN50022 DIN RAIL

Appendix B – Specifications

Overall Module Electrical

Specification	Values
Power supply voltage – Min. and Max.	18 to 30 VDC
Standby current consumption	<120mA
Terminal Pitch	3.81 mm
Terminal Wire Size Range	16 to 28 AWG (0.14 to 1.5 mm^2)
Maximum Terminal Screw Tightening Torque	0.25 Nm
Terminal Screw Type	M2

Digital Inputs

Specification	Values
Quantity	8
Input Delay ON	8 ms
Input Delay OFF	8 ms
Minimum ON State Voltage	10 VDC
Minimum ON State Current	1.5 mA
Maximum OFF State Voltage	5 VDC
Maximum OFF State Current	0.2 mA
Inrush Current	250 mA
Input Impedance, nominal	3 kΩ
IEC Input Compatibility	Type 1+

Digital Outputs

Specification	Values
Quantity	6
Output Delay ON	0.1 ms
Output Delay OFF	1.0 ms
OFF State Leakage Current	1.0 mA
Minimum ON State Current	1.0 mA
Maximum ON State Voltage Drop	1.0 VDC @ 1A
Maximum Current per Point	0.5 A
Maximum Current for Total Output Bus	2.0 A
Surge Current	2.0A for 10 ms, repeatable every 2s
Isolation Voltage (Output Bus to Module Bus)	1.5 KV

ENVIRONMENTAL

Operating outside these parameters may result in permanent module failure or unexpected device behavior.

Specification	Values
Storage temperature	-40ºC to 150º C (-40ºF to 300ºF)
Ambient Operating temperature	0ºC to 60ºC (32ºF to 140ºF)
Humidity	5% to 95% non-condensing
Vibration	0.152 mm (0.006 in.) displacement, 1G peak
Mechanical Shock	20G peak for 10ms duration (1.0 ms)
Enclosure IP rating	IP20

COMPLIANT STANDARDS

This part is *CE Certified* and tested to comply with the following standards:

Standard	Description
IEC EN 61131-2:2008	Programmable controllers -- Part 2: Equipment requirements and tests
IEC EN 61000-6-2:2006	Electromagnetic compatibility (EMC) -- Part 6-2: Generic standards - Immunity for industrial environments
IEC EN 61000-6-4:2007	Electromagnetic compatibility (EMC) -- Part 6-4: Generic standards - Emission standard for industrial environments
IEC EN 55016-2-1+A1:2006	Specification for radio disturbance and immunity measuring apparatus and methods Part 2-1 Methods of measurement of disturbances and immunity. Conducted disturbance measurements
IEC EN 55014-1:2007	Electromagnetic compatibility - Requirements for household appliances, electric tools and similar apparatus -- Part 1: Emission
IEC EN 61000-4-2+A1+A2:2004	Electromagnetic compatibility (EMC) Part 4-2: Electromagnetic discharge Immunity test
IEC EN 61000-4-3/A1:2008	Electromagnetic compatibility (EMC) Part 4-3 Radiated radio-frequency, electromagnetic field immunity test.
IEC EN 61000-4-4:2006	Electromagnetic compatibility (EMC) Part 4-4 Electrical fast transient/burst immunity test.
IEC EN 61000-4-5:2007	Electromagnetic compatibility (EMC) Part 4-5 Surge immunity test.
IEC EN 61000-4-6:2007	Electromagnetic compatibility (EMC) Part 4-6 Immunity to conducted disturbances, induced by radio-frequency field.
IEC EN 61000-4-11:2006	Electromagnetic compatibility (EMC) Part 4-11 Voltage dips, short interruptions and voltage variations immunity tests.

ETHERNET SPECIFICATION

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

Appendix C – Connecting to Rockwell PLC with EDS and AOI

This section is applicable for CNIP's with firmware 4.25 and later as well as firmware 5.2 and later. If you are using firmware 4.24 and earlier, you can only connect using the Generic Ethernet Module method described earlier in this document.

Please refer to publication ERSC-1520 Connecting to Rockwell PLCs Version 2.0 for general details on installing an EDS file into your RSLogix 5000 environment. The remainder of this section will assume you have properly installed the CNIP EDS file into you RSLogix 5000 environment

Installing the EDS file provided on pulseroller.com into your RSLogix 5000 environment will allow you to select the CNIP module from your list of known devices without having to use the Generic Ethernet Module method described previously. The EDS file contains the Instance and size parameters so you do not have to fill in this information. When you connect to a CNIP, the data is arranged in assembled registers as described in the previous sections of this document with the data appearing in your Controller Tags similarly

From the pulseroller.com website, you need to download the following file:

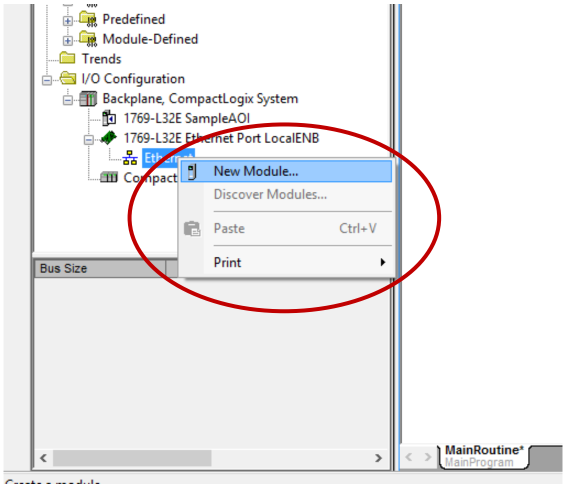
ConveyNetIP_V5_2.eds

This is the filename as of the publication of this document. Please go to pulseroller.com to download the latest versions of this EDS file.

Creating a CNIP Module in the Ethernet Tree

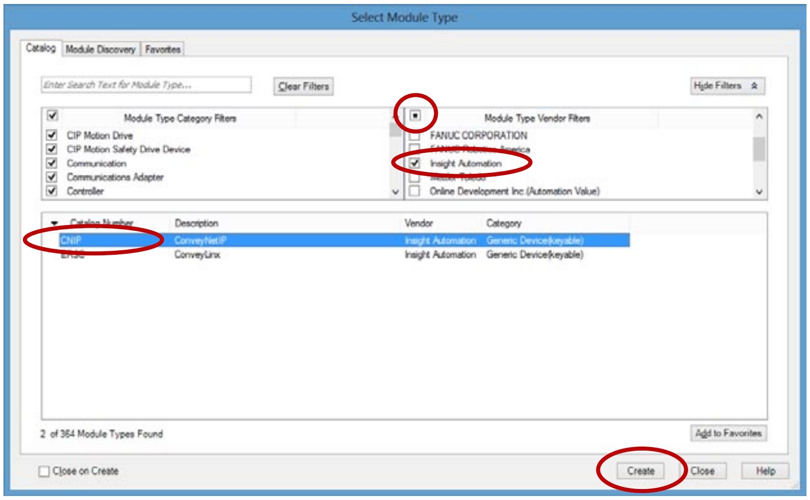
Once you have installed the EDS file into your RSLogix 5000 environment, you can now add specific instances of CNIP modules into your project. You follow a similar procedure as described for the Generic Ethernet Module method.

We are going to show adding a CNIP with Reset Protection to your program as an example.



Step 1

Right click on your Ethernet Tree and select *New Module* to open the *Select Module Type* window.



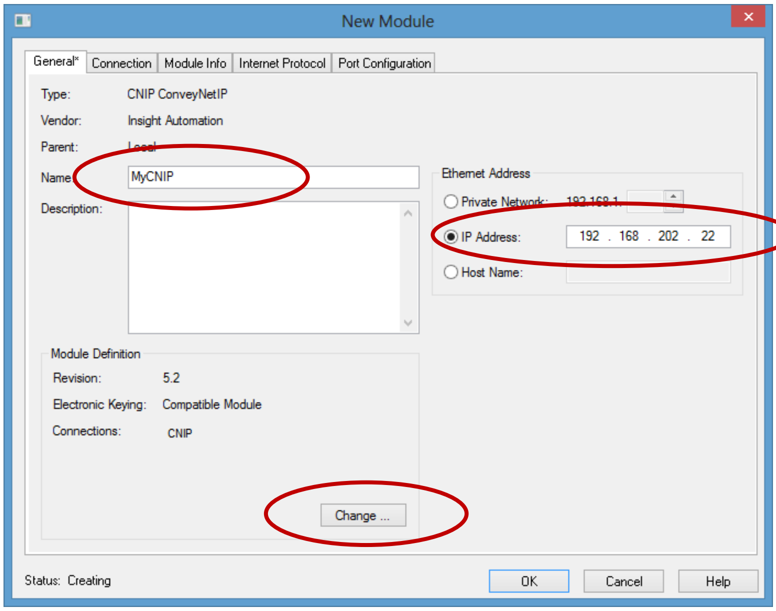
Step 2

In the *Select Module Type* window, locate the CNIP catalog number. In this example we cleared the vendor filter and checked Insight Automation. Once you select CNIP, click the *Create* button to open the *New Module* window.

Note: Your list may look different depending on what you have already installed in your RSLogix5000 environment.

Step 3

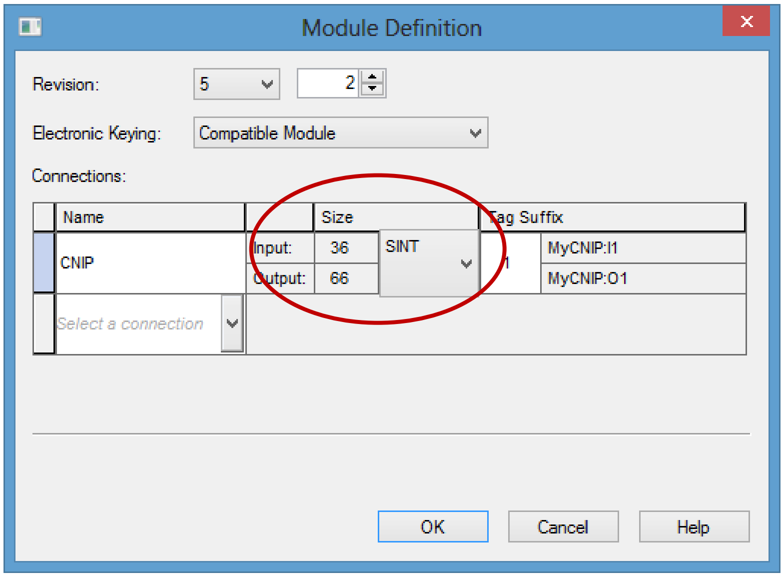
For our example, we entered the Name and IP address information as shown. You can choose whatever name you desire and enter the proper IP address for your application. **Then you must click the *Change* button to open the *Module Definition* window.**



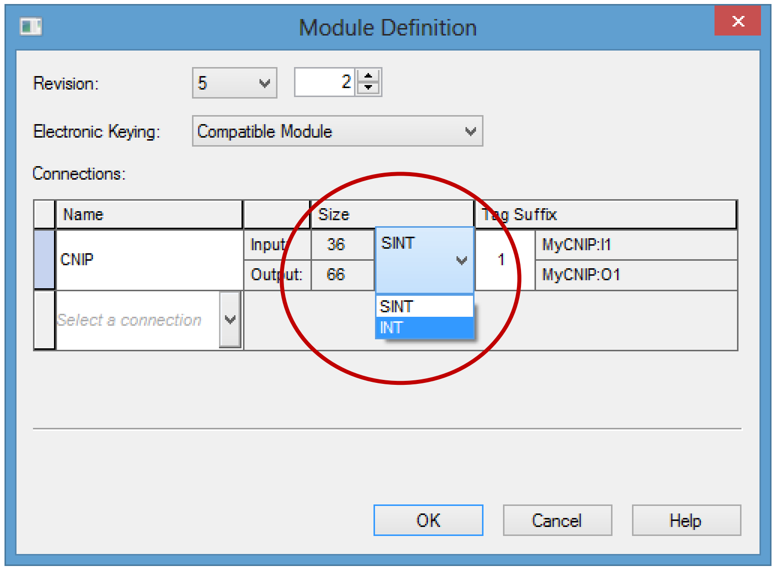
Step 4

In the Module Definition window you will see the default settings from the EDS file. The EDS file only allows SINT data type to be default. This needs to be changed to INT.

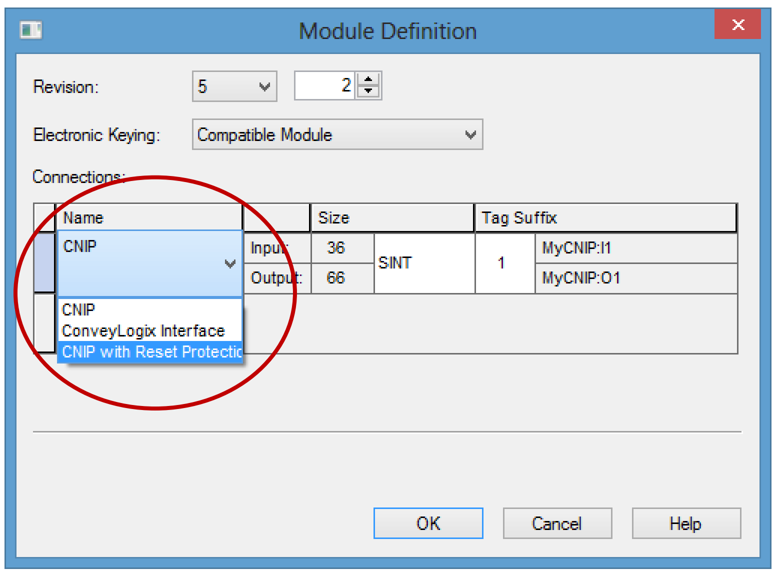
See Note ① below.



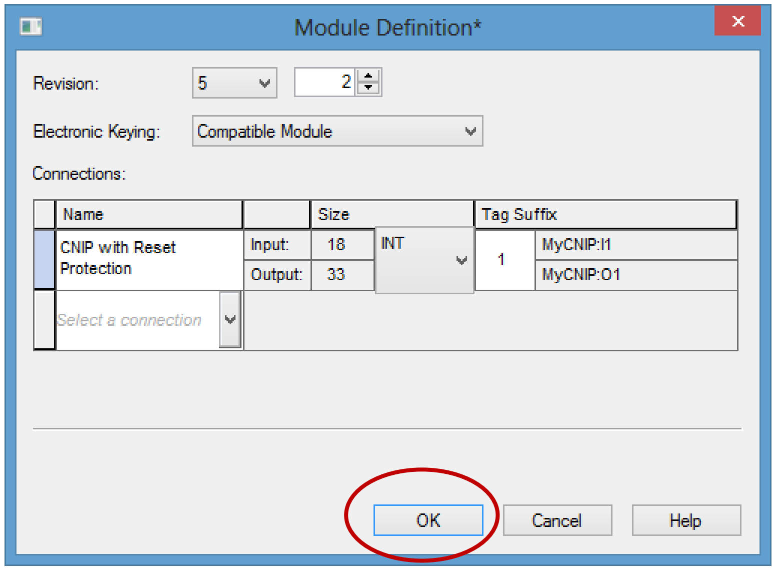
Click the right of the *Size* box to show a drop down box and then select INT



Once you have selected INT as the data size, if you want to use the AOI, you will need to change the connection type to *CNIP with Reset Protection*. If you are not going to use the AOI, then you can leave the connection type as *CNIP*.

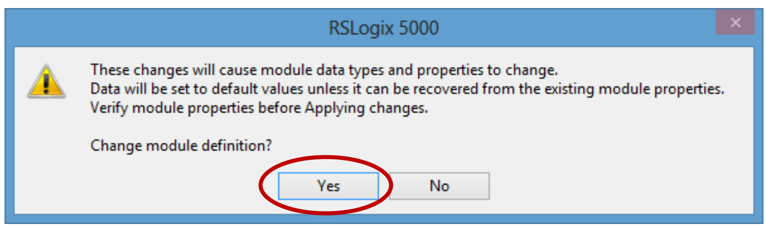


Once you have made your selections, click *OK*.



Step 5

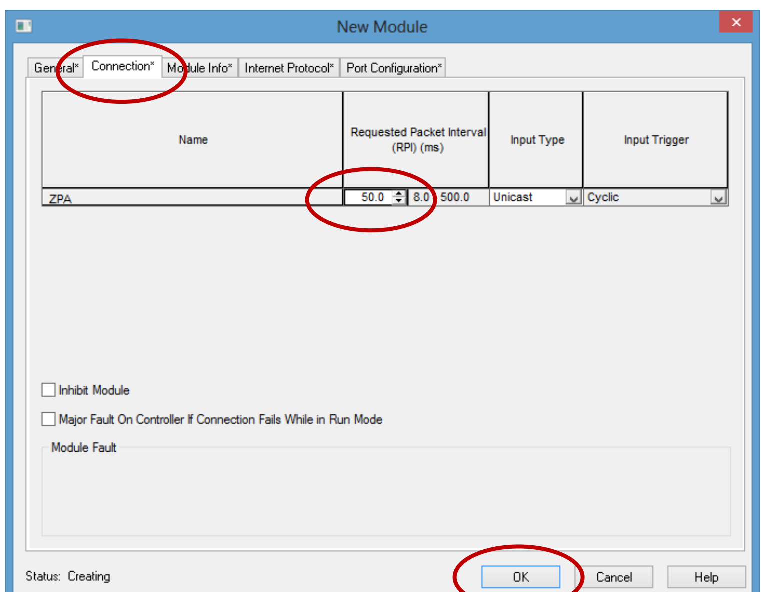
When you click OK, a warning will appear to tell you that you are changing the default parameters. Click *Yes*.



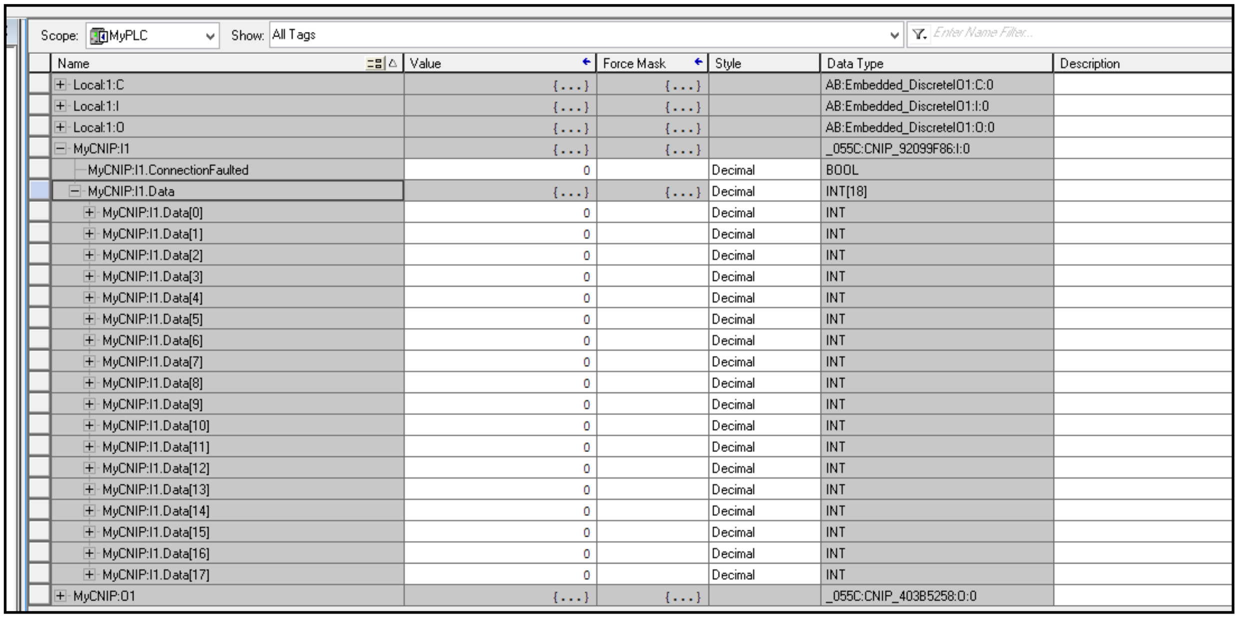
Step 6

You should be back to the New Module window. You can change the RPI of the connection to your CNIP by clicking the Connection tab and modify the value as desired. Click OK to finish creating the CNIP module

See Note ② - RPI Settings for more details.



For our example, Figure 15 and Figure 16 show the PLC’s Controller tags generated when the CNIP was created. You can see that the quantities of INT registers corresponds with the registers defined earlier in section *Ethernet I/P Instances for CNIP*.



Scope: MyPLC Show: All Tags Enter Name Filter...

Name	Value	Force Mask	Style	Data Type	Description
+ Local:1:C	{...}	{...}		AB:Embedded_DiscreteIO1:C:0	
+ Local:1:I	{...}	{...}		AB:Embedded_DiscreteIO1:I:0	
+ Local:1:O	{...}	{...}		AB:Embedded_DiscreteIO1:O:0	
− MyCNIP:I1	{...}	{...}		_055C:CNIP_92099F86:I:0	
MyCNIP:I1.ConnectionFaulted	0		Decimal	BOOL	
− MyCNIP:I1.Data	{...}	{...}	Decimal	INT[18]	
+ MyCNIP:I1.Data[0]	0		Decimal	INT	
+ MyCNIP:I1.Data[1]	0		Decimal	INT	
+ MyCNIP:I1.Data[2]	0		Decimal	INT	
+ MyCNIP:I1.Data[3]	0		Decimal	INT	
+ MyCNIP:I1.Data[4]	0		Decimal	INT	
+ MyCNIP:I1.Data[5]	0		Decimal	INT	
+ MyCNIP:I1.Data[6]	0		Decimal	INT	
+ MyCNIP:I1.Data[7]	0		Decimal	INT	
+ MyCNIP:I1.Data[8]	0		Decimal	INT	
+ MyCNIP:I1.Data[9]	0		Decimal	INT	
+ MyCNIP:I1.Data[10]	0		Decimal	INT	
+ MyCNIP:I1.Data[11]	0		Decimal	INT	
+ MyCNIP:I1.Data[12]	0		Decimal	INT	
+ MyCNIP:I1.Data[13]	0		Decimal	INT	
+ MyCNIP:I1.Data[14]	0		Decimal	INT	
+ MyCNIP:I1.Data[15]	0		Decimal	INT	
+ MyCNIP:I1.Data[16]	0		Decimal	INT	
+ MyCNIP:I1.Data[17]	0		Decimal	INT	
+ MyCNIP:O1	{...}	{...}		_055C:CNIP_403B5258:O:0	

FIGURE 15 - CONTROLLER TAGS FOR CNIP INPUTS TO PLC

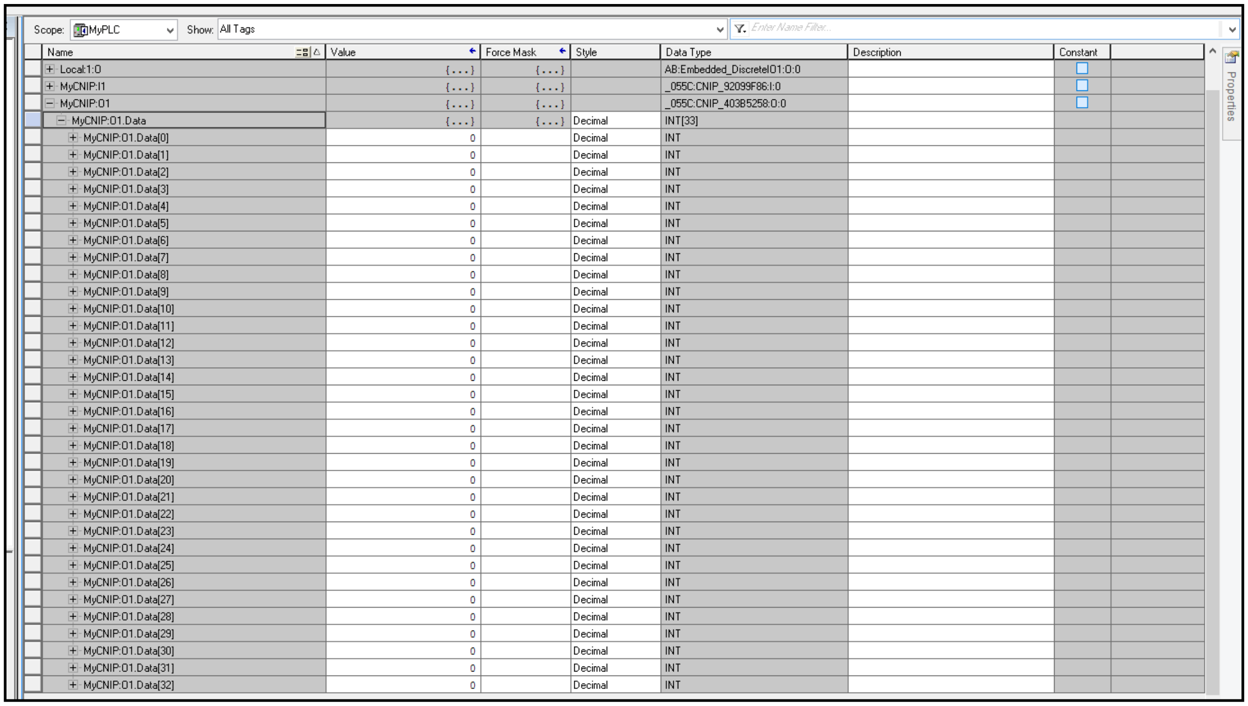


FIGURE 16 - CONTROLLER TAGS FOR OUTPUTS FROM PLC TO CNIP

NOTE ① - DATA TYPE SIZE

As noted, the EDS specification only allows for SINT data type as default. You can leave SINT as the default data type if this fits your particular programming preferences. However, keep in mind the documented register types in described in section *Ethernet I/P Instances for CNIP* earlier in this document are described as 16 bit INT and this could lead to cross-referencing confusion. Furthermore, if you also wish to use Insight Automation's **Add On Instructions** (AOIs – described in the next section), you **must change the data type to INT** because these items are written expecting INT data type.

NOTE ② - RPI SETTINGS

Please note that RPI settings do not affect the CNIP nearly as much as the PLC's Ethernet port's throughput. A combination of the quantity of device connections along with small RPI values can create a bottleneck at the PLC's Ethernet port. A higher quantity of device connections coupled with a small RPI for each can result in dropped connections to devices. <u>This is not an issue with the CNIP (or other device); it is an issue with the PLC</u>. It is always recommended to use the largest RPI value you can for a given connection while maintaining reasonable device response.

USING CNIP ADD ON INSTRUCTIONS (AOI) WITH RSLOGIX 5000

Insight Automation has authored and made available an Add On Instruction (AOI) in order to make your programming easier to follow. In this document up until this section, when connecting to a CNIP module; your PLC program needs to directly access the register data array tags created when you created the CNIP instance. The AOI attaches to the CNIP's register data arrays and maps the data into user tags and functions with meaningful names..

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

INSTALLING THE AOI INTO RSLOGIX 5000

From the pulseroller.com website, you need to download the following file depending on your needs:

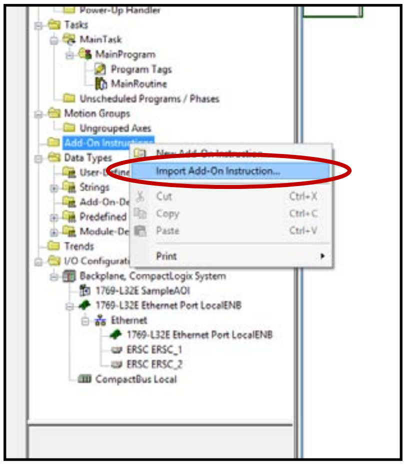
CNIP_5_2.L5X

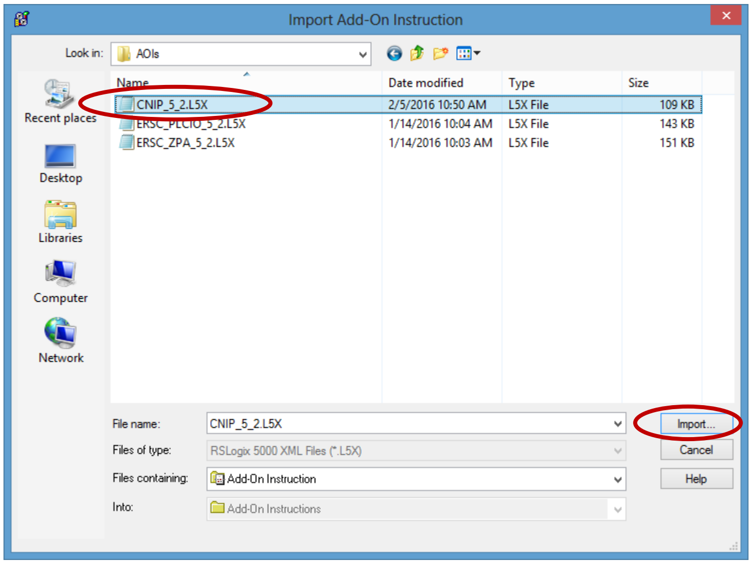
These are the filenames as of the publication of this document. Please go to pulseroller.com to download the latest versions of any AOI files.

After your EDS file has been installed; the next procedure is to install the Add On Instruction (AOI) file that you downloaded.

Step 1

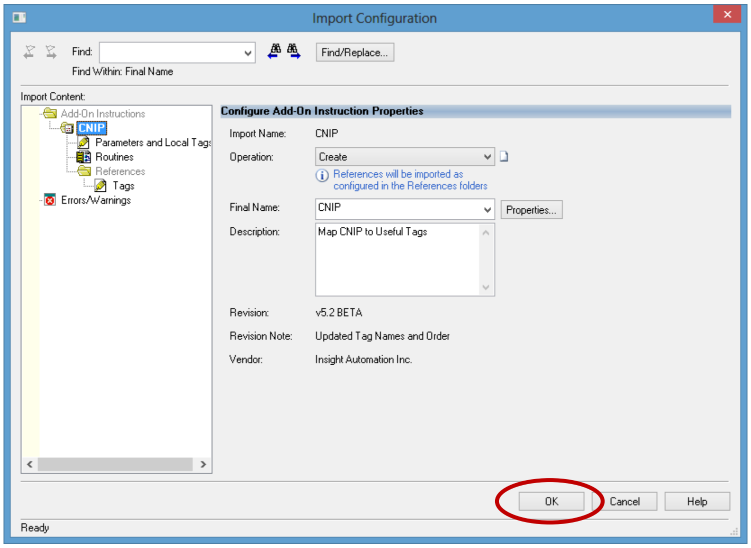
Right click on the *Add On Instruction* folder in the explorer tree. From the pop-up menu select Import *Add On Instruction…*





Step 2

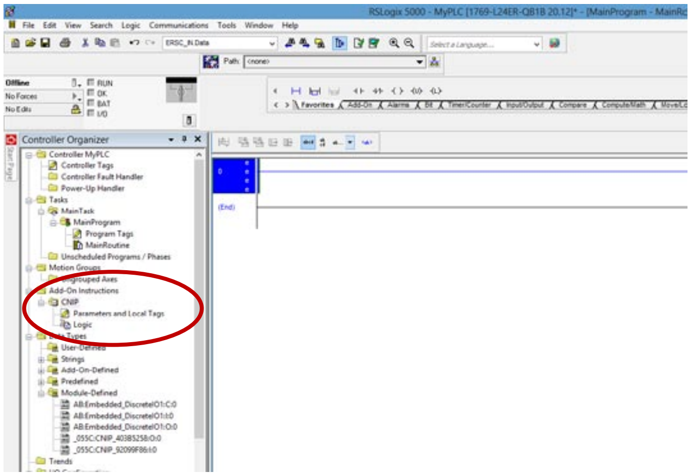
Navigate to the folder location where you downloaded your AOIs, select the file then click import.



Step 3

A window will appear indicating the details about the AOI you are about to import. There should be no errors or warnings. Click OK to proceed with the import.

When you are done, the AOI will appear in the explorer tree as shown.



EXAMPLE FOR ASSIGNING AOI TO CNIP MODULE IN YOUR PROJECT

For our example, we are going to assign the CNIP depicted in our previous examples to our new AOI. Also, the CNIP is attached using the *CNIP with Reset Protection* connection.

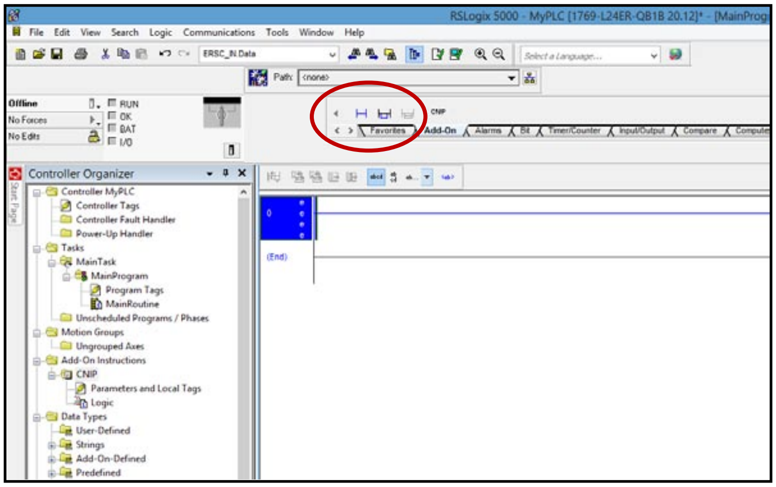
Insight Automation provided AOI's require connections to be "with Reset Protection". Also refer to section *Input and Output Assemblies with Reset Protection* for details on reset protection assemblies.

ASSIGNING NEW MODULES TO AOI

Now that we have our CNIP defined with the correct connection types and the AOI instructions installed into our RSLogix5000 project; the next step is to create an instance of the AOI for the physical CNIP.

Step 1

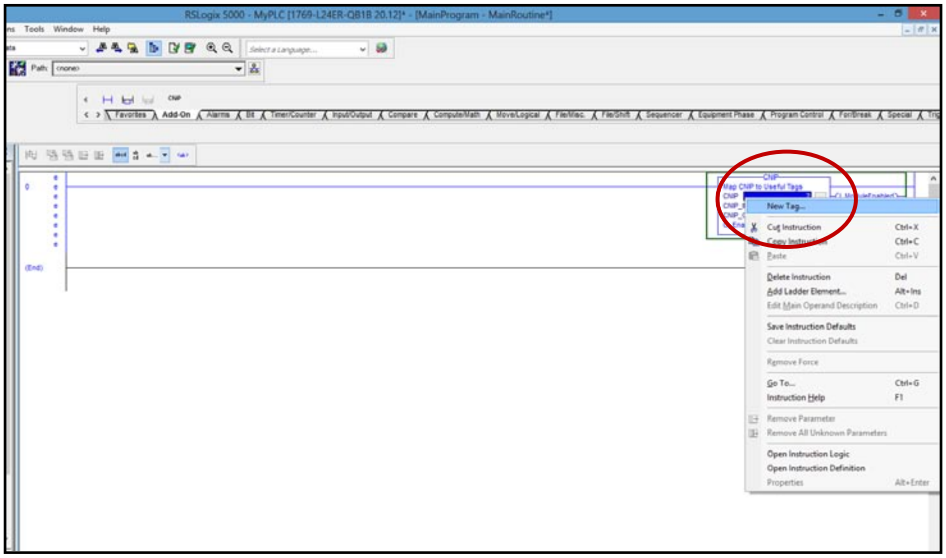
Locate the Add-On tab and find the CNIP instruction button. Click to place the instruction on the selected rung in your ladder diagram.



Step 2

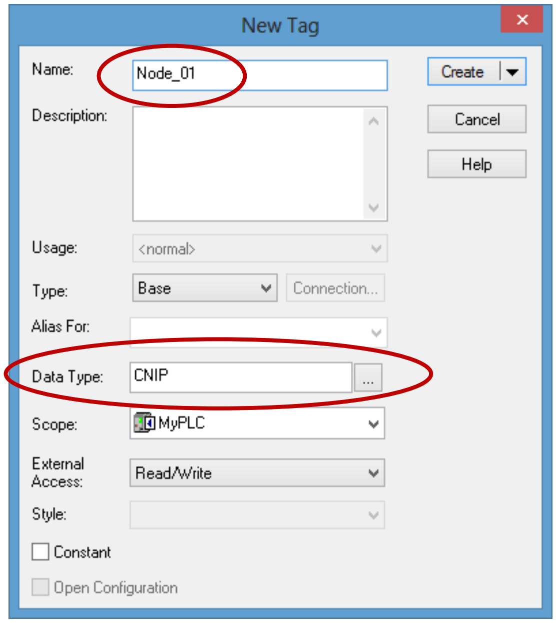
Once the instruction has been added to the ladder, we need to create a tag that will be how you access the modules data.

For our example we entered "Node_01" and then created the new tag



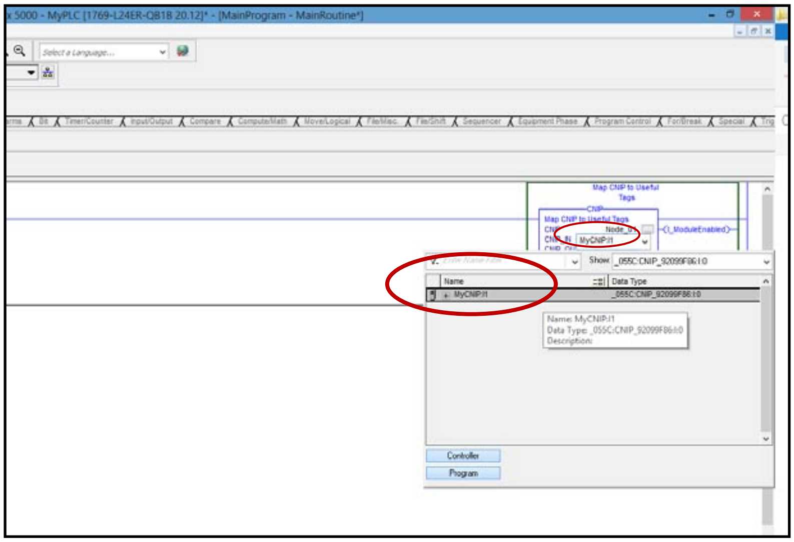
Step 3

This is the typical New Tag window you invoke from the ladder diagram screen. Note that the Data Type defaults to the AOI's data type. Click Create to create the new tag



Step 4

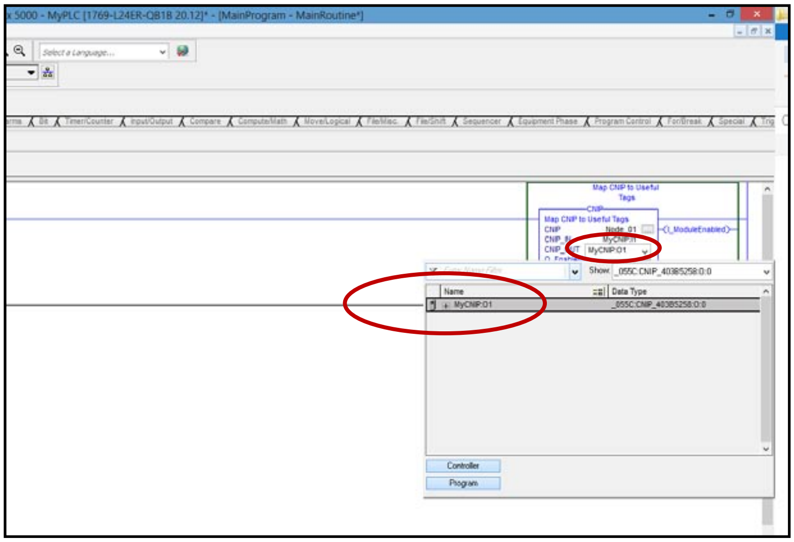
The AOI requires two other parameters; "CNIP_IN" for the data coming from the module and "CNIP_OUT for data coming from the PLC to the module. These will be from the physical modules we previously added. Here we will add the CNIP_IN parameter by clicking the drop down box arrow will automatically show all tags that match the data type for the CNIP_IN parameter. In this case, "MyCNIP" is the only module we created, so it is the only selection. Double click this and it will be assigned to the Node_01 instance of our AOI's "CNIP_IN" parameter.



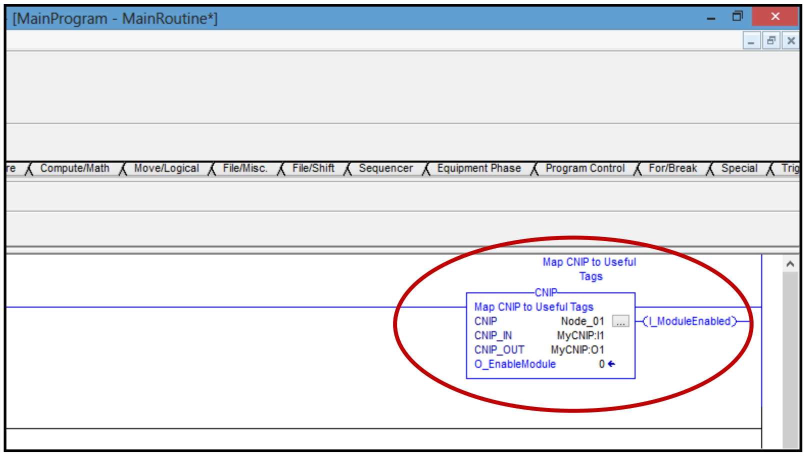
Step 5

Similarly to ***Step 4***, we need to do the select the physical module for the CNIP_OUT parameter. Clicking the drop down box arrow will show all physical modules that have the matching data type for the CNIP_OUT parameter.

Double click this and it will be assigned to the Node_01 instance of our AOI's "CNIP_OUT" parameter.



At this point, the AOI has been set up to use in your logic program. All of the tags associated with using the ERSC_1 in ZPA mode are in the structured tag “Node_01”.



You simply follow this same 5 step procedure for each physical CNIP module you add to your Ethernet Tree.

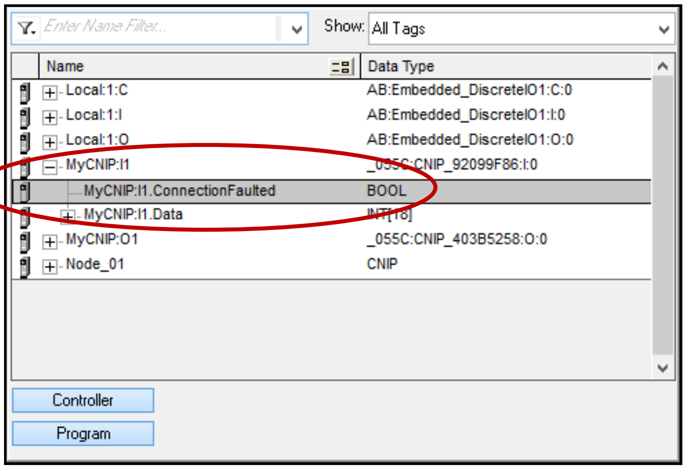
ENABLING THE MODULE FOR OPERATION

Before using the AOI in your program, you need to add some logic to enable the outputs on the physical module. The connection defined in the EDS file use the "with reset Protection" assemblies that require the PLC to instruct the CNIP module to process output data coming from the PLC.

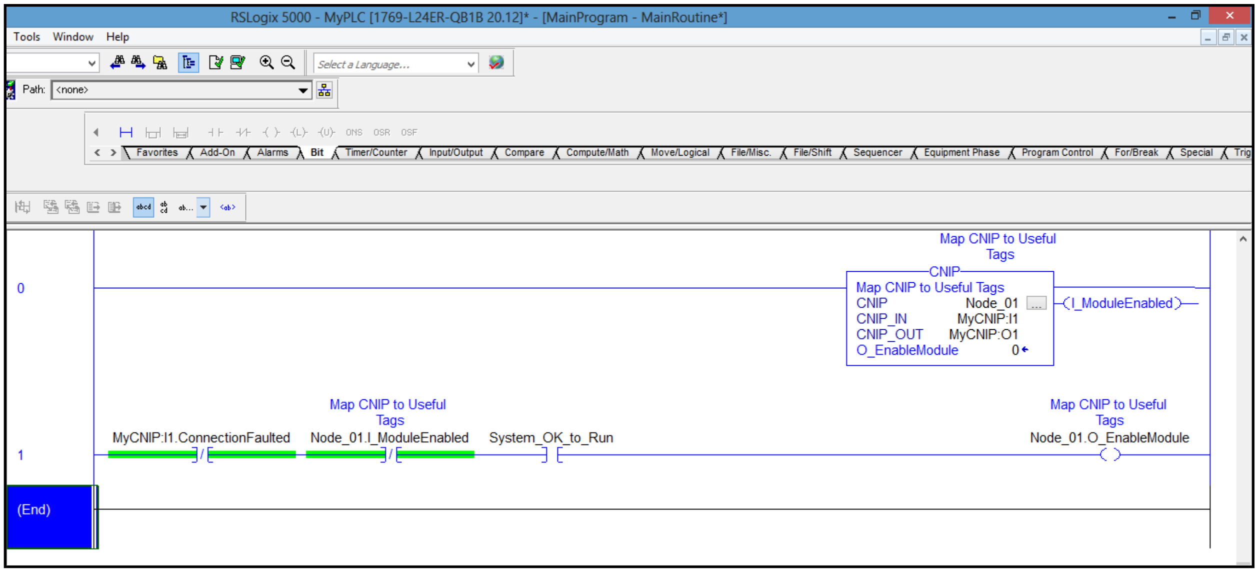
The Reset Protection topic is covered in PLC Developer's Guide

Another function that is built-in when you created the module is indication of whether the PLC is communicating with the module. For example, if you look in the Controller tags for the input data coming from the module, there is a Boolean value that indicates "Connection Faulted".

From our example, when you expand the "MyCNIP:I" structure, there is a BOOL that indicates "Connection Faulted". This tag can be used in your logic to assure connection is OK prior to enabling the module.



We recommend a simple rung of logic for each module that will allow the module to be enabled when its connection is OK and when "the system" is OK to start. This "system OK" state is wholly up to you as the programmer to determine or omit as desired.



For our example, we added a N.C. contact for the module's "ConnectionFaulted" tag, a N.C. contact for the AOI's tag that indicates that the module is not enabled, and a N.O. contact for the programmer's "System_OK_to_Run" condition as previously described. When this logic becomes true, then AOI's input to "EnableModule" is energized. When the module becomes enabled, the "ModuleEnabled" contact becomes true so that the "EnableModule" input to the AOI does not need to be held ON to keep the module enabled.

Simply repeat this rung of logic for each instance of the AOI.

If you do not include some logic to set the "O_EnableModule" bit in your program, the CNIP will NOT respond to any data written to it by the PLC.

AOI Tag Descriptions

Tag Name	Data Type	Internal Register	Bit	Description
I_ConveyStopByLeftControlPort	BOOL	4:0019	8	ConveyStop
I_ConveyStopByLostConnection	BOOL	4:0019	6	ConveyStop
I_ConveyStopByPLCCmd	BOOL	4:0019	10	ConveyStop
I_ConveyStopByPLCDisconnect	BOOL	4:0019	7	ConveyStop
I_ConveyStopByRemoteModule	BOOL	4:0019	5	ConveyStop
I_ConveyStopByRightControlPort	BOOL	4:0019	9	ConveyStop
I_Heartbeat	BOOL	4:0080	15	Module Digital Inputs
I_ModuleEnabled	BOOL	AOI Logic	-	Module Enabled to accept Outputs from PLC
I_Register_0_From_Slave	INT	4:0040	-	RS-485 Input Data
I_Register_1_From_Slave	INT	4:0041	-	RS-485 Input Data
I_Register_2_From_Slave	INT	4:0042	-	RS-485 Input Data
I_Register_3_From_Slave	INT	4:0043	-	RS-485 Input Data
I_Register_4_From_Slave	INT	4:0044	-	RS-485 Input Data
I_Register_5_From_Slave	INT	4:0045	-	RS-485 Input Data
I_Register_6_From_Slave	INT	4:0046	-	RS-485 Input Data
I_Register_7_From_Slave	INT	4:0047	-	RS-485 Input Data
I_Register_8_From_Slave	INT	4:0048	-	RS-485 Input Data
I_Register_9_From_Slave	INT	4:0049	-	RS-485 Input Data
I_RS485PortErrorWord	INT	4:0079	-	RS-485 Error Data
I_Status_From_Downstream	INT	4:0232	-	Adjacent Module's Status
I_Status_From_Upstream	INT	4:0134	-	Adjacent Module's Status
I_Tracking_From_Upstream	DINT	4:0139 (MSW) 4:0140 (LSW)	-	Upstream Module Tracking
Input_0	BOOL	4:0080	0	Module Digital Inputs
Input_1	BOOL	4:0081	1	Module Digital Inputs
Input_2	BOOL	4:0082	2	Module Digital Inputs
Input_3	BOOL	4:0083	3	Module Digital Inputs
Input_4	BOOL	4:0084	4	Module Digital Inputs
Input_5	BOOL	4:0085	5	Module Digital Inputs
Input_6	BOOL	4:0086	6	Module Digital Inputs
Input_7	BOOL	4:0087	7	Module Digital Inputs
O_BaudRate	INT	4:0075	-	RS-485 Baud Rate
O_ConveyStopCommand	INT	4:0020	-	ConveyStop
O_Default_Output_0_Enable	BOOL	4:0082	0	Digital Outputs Defaults
O_Default_Output_0_State	BOOL	4:0083	1	Digital Outputs Defaults
O_Default_Output_1_Enable	BOOL	4:0084	2	Digital Outputs Defaults
O_Default_Output_1_State	BOOL	4:0085	3	Digital Outputs Defaults
O_Default_Output_2_Enable	BOOL	4:0086	4	Digital Outputs Defaults
O_Default_Output_2_State	BOOL	4:0087	5	Digital Outputs Defaults
O_Default_Output_3_Enable	BOOL	4:0088	6	Digital Outputs Defaults
O_Default_Output_3_State	BOOL	4:0089	7	Digital Outputs Defaults
O_Default_Output_4_Enable	BOOL	4:0090	8	Digital Outputs Defaults
O_Default_Output_4_State	BOOL	4:0091	9	Digital Outputs Defaults
O_Default_Output_5_Enable	BOOL	4:0092	10	Digital Outputs Defaults
O_Default_Output_5_State	BOOL	4:0093	11	Digital Outputs Defaults
O_EnableModule	BOOL	AOI Logic	-	Command to Enable accepting outputs from PLC
O_QtyOfReadRegisters	INT	4:0072	-	RS-485 Settings
O_QtyOfWriteRegisters	INT	4:0074	-	RS-485 Settings
O_ReadRegisterStartAddress	INT	4:0071	-	RS-485 Settings
O_Register_0_To_Slave	INT	4:0050	-	RS-485 Settings
O_Register_1_To_Slave	INT	4:0051	-	RS-485 Settings
O_Register_2_To_Slave	INT	4:0052	-	RS-485 Settings
O_Register_3_To_Slave	INT	4:0053	-	RS-485 Settings

Tag Name	Data Type	Internal Register	Bit	Description
O_Register_4_To_Slave	INT	4:0054	-	RS-485 Settings
O_Register_5_To_Slave	INT	4:0055	-	RS-485 Settings
O_Register_6_To_Slave	INT	4:0056	-	RS-485 Settings
O_Register_7_To_Slave	INT	4:0057	-	RS-485 Settings
O_Register_8_To_Slave	INT	4:0058	-	RS-485 Settings
O_Register_9_To_Slave	INT	4:0059	-	RS-485 Settings
O_ResponseTimeout	INT	4:0078	-	RS-485 Settings
O_RS485Settings	INT	4:0076	-	RS-485 Settings
O_ScanRate	INT	4:0077	-	RS-485 Settings
O_SlaveID	INT	4:0070	-	RS-485 Settings
O_StatusToDownstreamModule	INT	4:0196	-	Status to Adjacent Module
O_StatusToUpstreamModule	INT	4:0116	-	Status to Adjacent Module
O_TrackingToDownstream	DINT	4:0201 (MSW) 4:0202 (LSW)	-	Tracking to Adjacent Module
O_WriteRegisterStartAddress	INT	4:0073	-	RS-485 Settings
Output_0	BOOL	4:0081	0	Module Digital Outputs
Output_1	BOOL	4:0082	1	Module Digital Outputs
Output_2	BOOL	4:0083	2	Module Digital Outputs
Output_3	BOOL	4:0084	3	Module Digital Outputs
Output_4	BOOL	4:0085	4	Module Digital Outputs
Output_5	BOOL	4:0086	5	Module Digital Outputs

NOTES:

PULSE ROLLER
WWW.PULSEROLLER.COM
SALES@PULSEROLLER.COM
SUPPORT@PULSEROLLER.COM

Publication CNIP-1000 Rev 1.4 – January 2016