PULSEROLLER

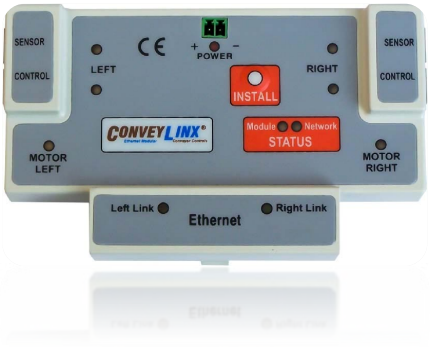
CONVEYLINX

Supplement to PLC Developer's Guide

Connecting to Siemens S7-300 PLCs

Version 1.0

September 2014



Publication *ERSC*-1525

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

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	September 2014	Initial Release

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 to utilize a Siemens PLC equipped with Profinet I/O capability to connect to a *ConveyLinx* Ethernet network to access module status and control conveyor operation.

PREREQUISITES

You should have reviewed and understood either the *ConveyLinx PLC Developer's Guide* (Insight Automation publication ERSC-1500) or *ConveyLinx-Ai PLC Developer's Guide* (Insight Automation publication ERSC-1510) before utilizing this manual's instructions to physically connect your Siemens PLC to a ConveyLinx network.

This manual also assumes you have a solid working knowledge of both Siemens PLC's and the TIA Portal Step7 development environments.

NOT INCLUDED IN THIS MANUAL

Because system applications vary; this manual assumes users and application engineers have properly sized their PLC's Ethernet port capacity to accommodate the quantity of ConveyLinx module connections desired. Please refer to you particular PLC's specifications.

INTRODUCTION

S7-300 PLCs from Siemens can use ConveyLinx modules because they support PROFINET IO communication protocol and can act as PROFINET IO-devices with the S7-300 PLC acting as a PROFINET IO controller.

MODES OF OPERATION

When installed, ConveyLinx modules can be configured in 4 basic modes:

FULL ZPA MODE

In this mode, ConveyLinx works as a one or two zone ZPA controller. When communicating with module in this mode, the S7-300 PLC can control mainly material handling properties such as accumulation, release, wake-up/full stop, read/write tracking, MDR parameters (speed, acceleration and deceleration), and ConveyStop. In this mode the PLC does not have access to directly control start/stop of MDRs because internal ZPA logic is operating the module. When communicating with ConveyLinx in ZPA mode, the PLC does not require fast reaction time as would be expected for a remote I/O device. Typically, 32 ms to 256 ms is the range of response time utilized between the PLC and a ConveyLinx module in ZPA mode. The data array instance lengths for input and output data mode are 64 bytes.

REDUCED ZPA MODE

In this mode, the ConveyLinx modules operates as a one or two zone ZPA controller just like the Full ZPA Mode. However, the instance data presented to the S7-300 PLC is limited to basic material handling properties such as accumulation, release, wake-up/full stop, and MDR speed. In this mode, because the instances length is reduced to 30 bytes; the PLC may be able to accommodate more modules especially in applications where all ZPA features (such as tracking) are not required. Similarly to the Full ZPA Mode, the 32 ms to 256 ms response time is typical for this mode.

Please note, that the Reduced ZPA Mode instance is available with Firmware version 4.24 or higher.

FULL PLC CONTROLLED MODE

In this mode, the ConveyLinx module is placed into PLC I/O mode with the EasyRoll software tool. When the module is in PLC I/O mode, all internal ZPA logic is suspended and the module requires an external PLC to read port inputs and run/stop MDRs connected to the module. The PLC contains any and all logic to process inputs from sensor/control ports, run stop MDRs, set direction/speed/accel/decel of MDRs, and more. Also, in this mode the PLC has access to registers in the PLC I/O module that can interface to neighbouring ZPA modules for material handling control and tracking data support. A ConveyLinx module in PLC I/O mode is also capable of Servo functions that are made available to the PLC. In this mode, the ConveyLinx module responds as would be expected for a remote I/O device. ConveyLinx will connect to the PLC with a 4 msec communication response time. The data array instance lengths for input and output data in this mode are 64 bytes.

Reduced PLC Controlled mode

In this mode, the ConveyLinx module is configured and operates without ZPA logic the same as it does in Full PLC Controlled Mode. However in this mode, only port I/O and MDR run/stop/speed functions are available to the PLC. In this mode, because the instances length is reduced; the PLC may be able to accommodate more modules especially in applications where all the features in Full PLC Controlled Mode are not required. ConveyLinx modules in this mode communicate at the same 4 msec response time as the Full PLC Controlled Mode. The data array instance lengths for input and output data in this mode are 16 bytes.

PREPARING YOUR PROGRAMMING ENVIRONMENT

In order to use ConveyLinx with your PLC, you need to download the proper configuration files from www.pulseroller.com. This section provides instructions on how to install these configuration files into either the TIA Portal V11 environment or STEP 7 environment.

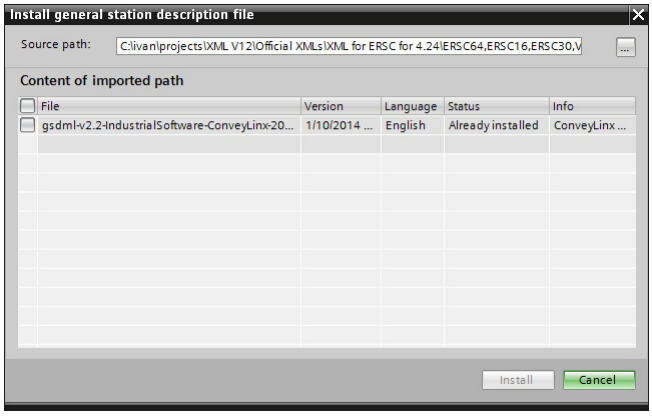
INSTALLATION FILES

The downloaded files include the following:

Filename	Description
gsdml-v2.2-IndustrialSoftware-ConveyLinx-20140510.xml	File which describes communication properties of ConveyLinx card. The filename is encoded with the file revision (v2.2) and date (10 May 2012). *Please note that downloaded file may be more recent that the filename shown*.
ConveyLinx.bmp	Icon image file which represents how a Full ZPA or Full PLC Controlled Mode ConveyLinx module will look in your programming tools
ConveyLinxmini.bmp	Icon image file which represents how a Reduced ZPA or Reduced PLC Controlled Mode ConveyLinx module will look in your programming tools

It is important to place these files in the same folder. Once you install the gsdml file into your environment, it will look for the other bmp files to be in the same folder

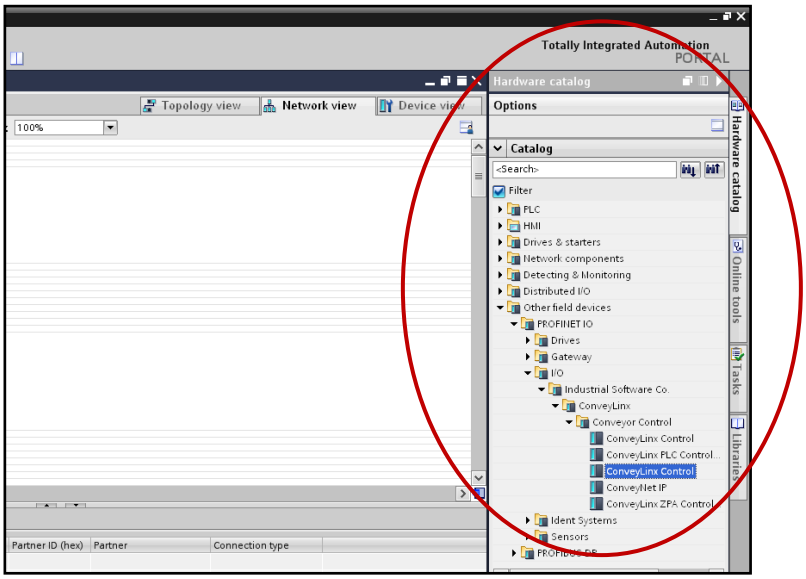
INSTALLING GSD FILE WITH TIA PORTAL

In PORTAL V11 select "Options"/"install general station description file (GSD)". Select the xml file you downloaded and click "Install".	

ADDING MODULES TO YOUR PROJECT

Once you have installed the GSD file into your environment, ConveyLinx devices are available to be used in your project. You can find ConveyLinx devices in the *Hardware Catalog* window.

Drill down from Other field devices as shown to get to the folder Conveyor Control to see the available modules



The Mode of operation of a given ConveyLinx module will determine which of the devices do drag and drop from the Hardware Catalog window into your project. Also, once the module has been dragged and dropped into your project, there are other parameters to enter to tell the PLC what Mode of operation you wish to use for the given module.

PROFINET NAME FOR THE DEVICE

In order for the *ConveyLinx* device to transfer the expected data to/from the PLC, the PLC needs to connect using a specific *Profinet Name* that the *ConveyLinx* device recognizes. The ConveyLinx module packages the data for a specific Mode of operation based upon the syntax of the *Profinet Name*. The following chart lists the four possible modes and the syntax to use for each.

Mode	Syntax	Remarks
Full ZPA	*conveylinxzpa-xxx-yyy*	xxx = 3rd Octet of I.P. Address yyy = 4th Octet of I.P. Address
Full PLC Controlled	*conveylinxplc-xxx-yyy*	
Reduced ZPA	*conveylinxzpa-xxx-yyy*	
Reduced PLC Controlled	*conveylinxplc-xxx-yyy*	

Please do not confuse the Module's "General Name" or "Project Name" that you assign for a given device with the *Profinet Name* you give to a device. The *Profinet Name* is required to be in the proper syntax or the module will not connect to the PLC. The module's "Project Name" is completely user's choice to identify the module in the TIA Portal environment and is what is displayed in your project's Network View(s) and Project Tree.

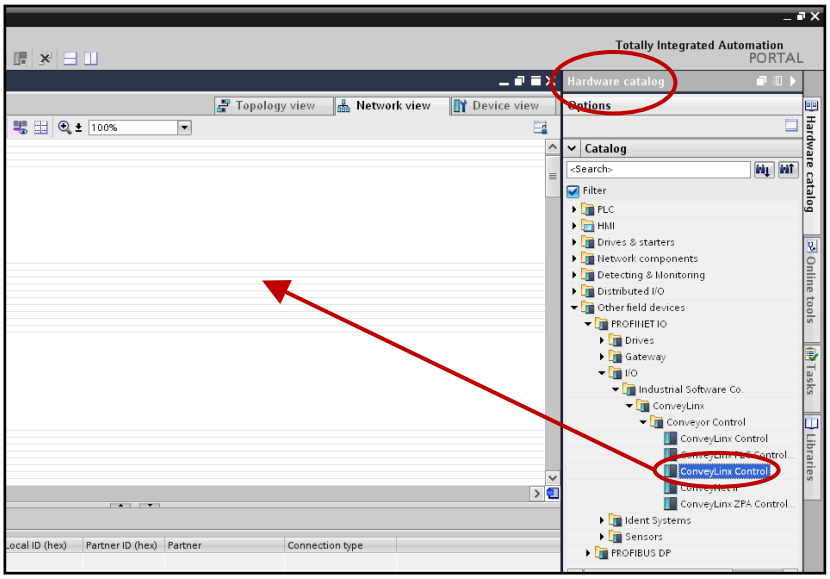
EXAMPLE

We will provide an example that will add a single module of each Mode to a project. This will require a combination of selecting the proper *Device* from the *Hardware Catalog* coupled with using the correct *Profinet Name* to achieve our desired result. The following chart lists the 4 modules we want to add to our project and the proper *Device* and *Profinet Name* syntax needed:

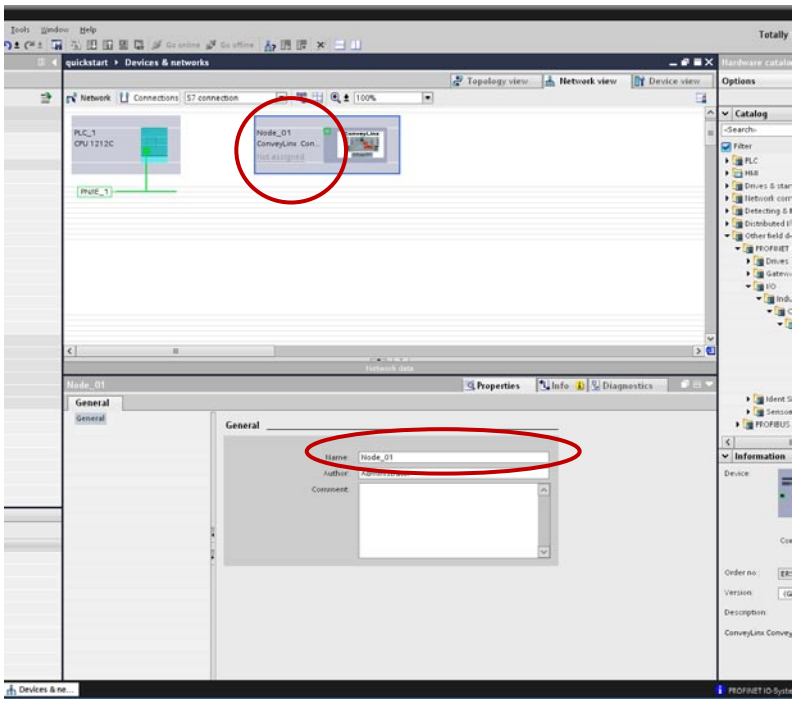
Mode	I.P. Address	Hardware Catalog Item	Project Name	Profinet Name
Full ZPA	192.168.200.20	*ConveyLinx Control*	Node_01	conveylinxzpa-200-20
Full PLC Controlled	192.168.200.21	*ConveyLinx Control*	Node_02	conveylinxplc-200-21
Reduced ZPA	192.168.200.22	*ConveyLinx ZPA Control with Reduced I/O*	Node_03	conveylinxzpa-200-22
Reduced PLC	192.168.200.23	*ConveyLinx PLC Control with Reduced I/O*	Node_04	conveylinxplc-200-23

ADDING FULL ZPA MODULE

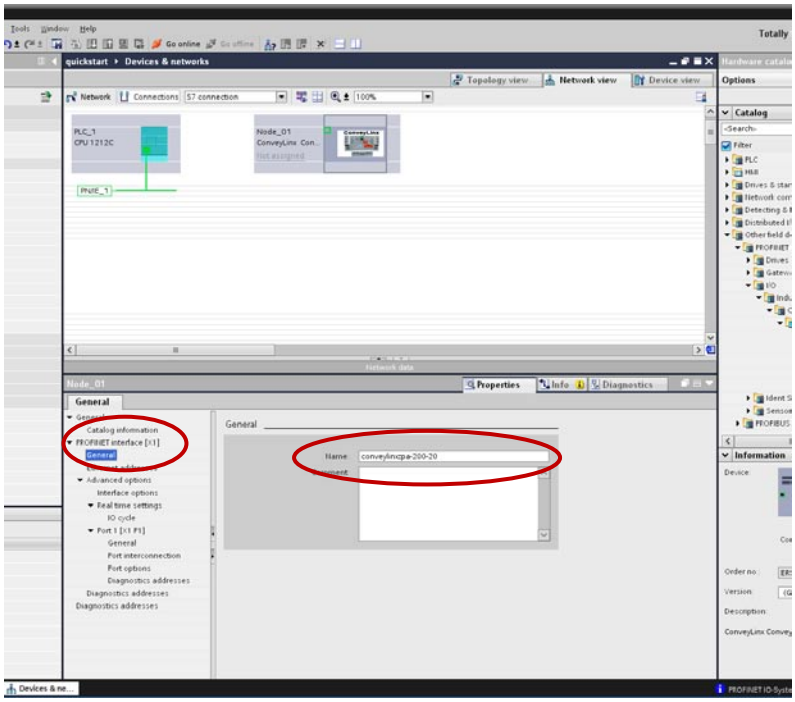
From the *Hardware Catalog* window, drill down the tree as shown and select *ConveyLinx Control* as shown. Drag and drop the device into the *Network View* window



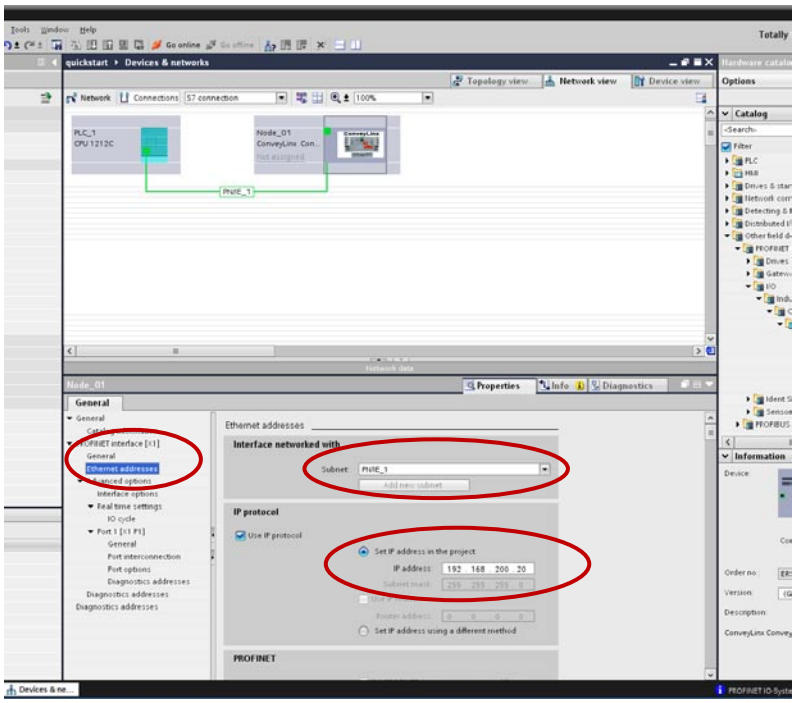
Double Click the module in Network View to enter the General or "Project Name". For our example this is "Node_01"



Double click the module icon to open the Profinet Interface settings and select "General" and enter in the exact *Profinet Name* syntax for the specific module. In this case we enter "conveylinxzpa-200-20"



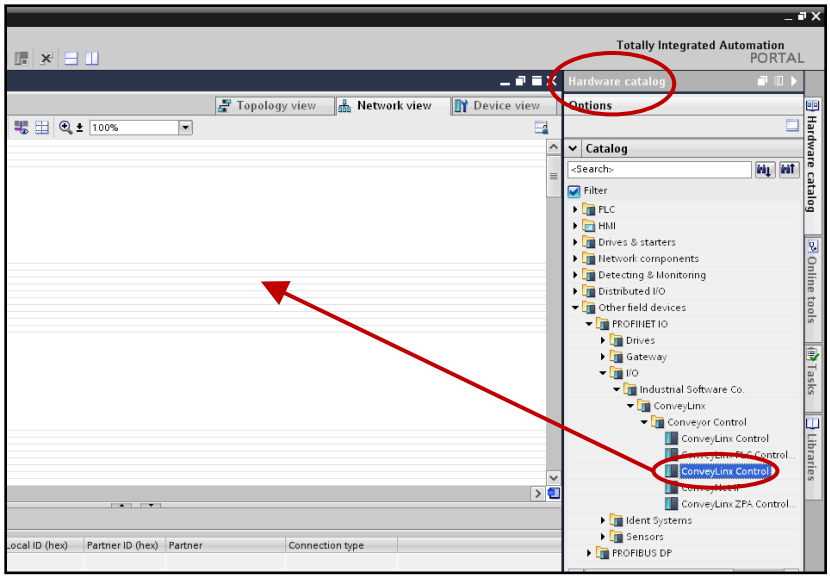
Select *Ethernet Addresses* and select you PLC's subnet and enter the I.P. address of the module you are adding. The Full ZPA Mode module has been added to your project.



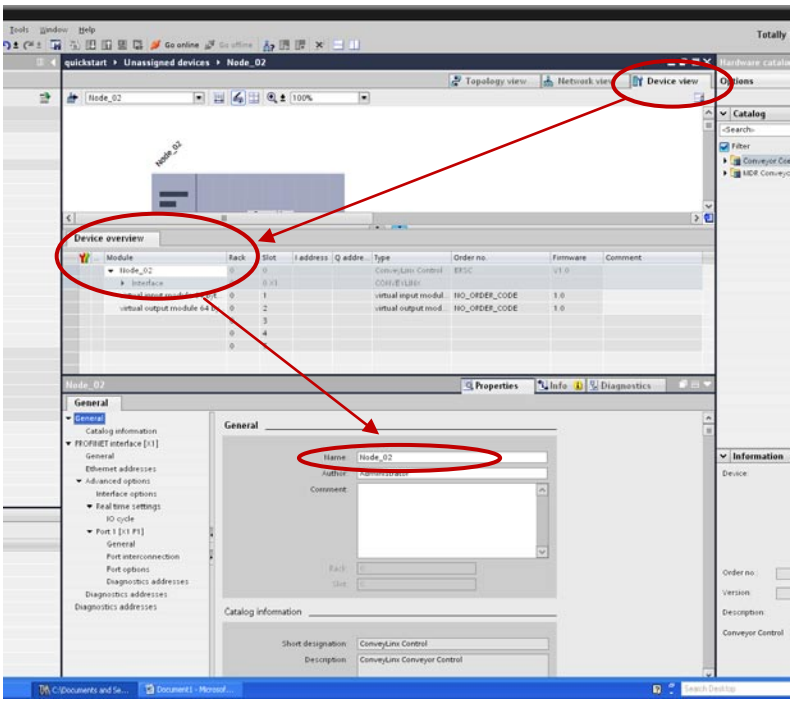
Adding Full PLC Controlled Mode Module

From the *Hardware Catalog* window, drill down the tree as shown and select *ConveyLinx Control* as shown. Drag and drop the device into the *Network View* window

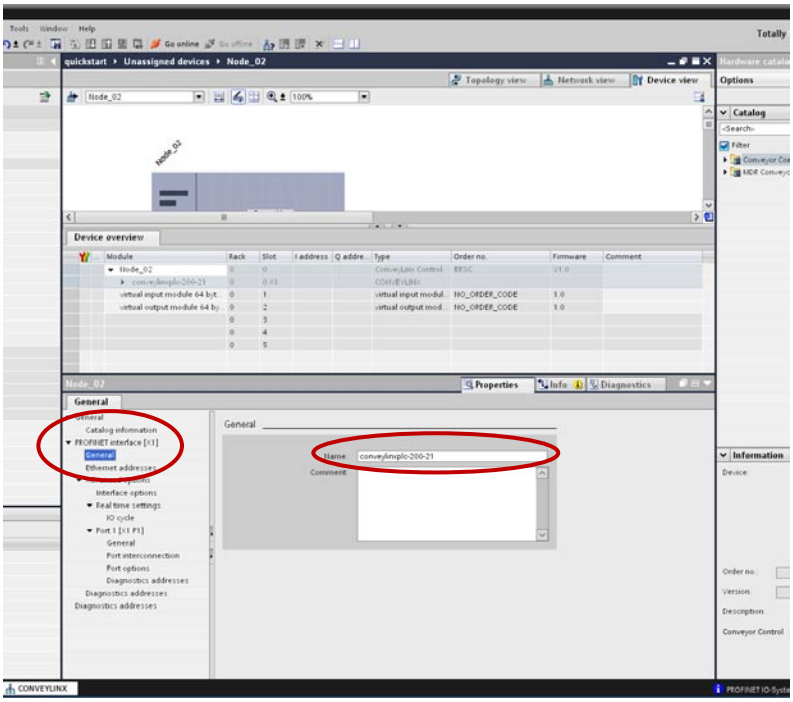
Note that this is the same device from the Hardware Catalog that was used for Full ZPA Mode Module

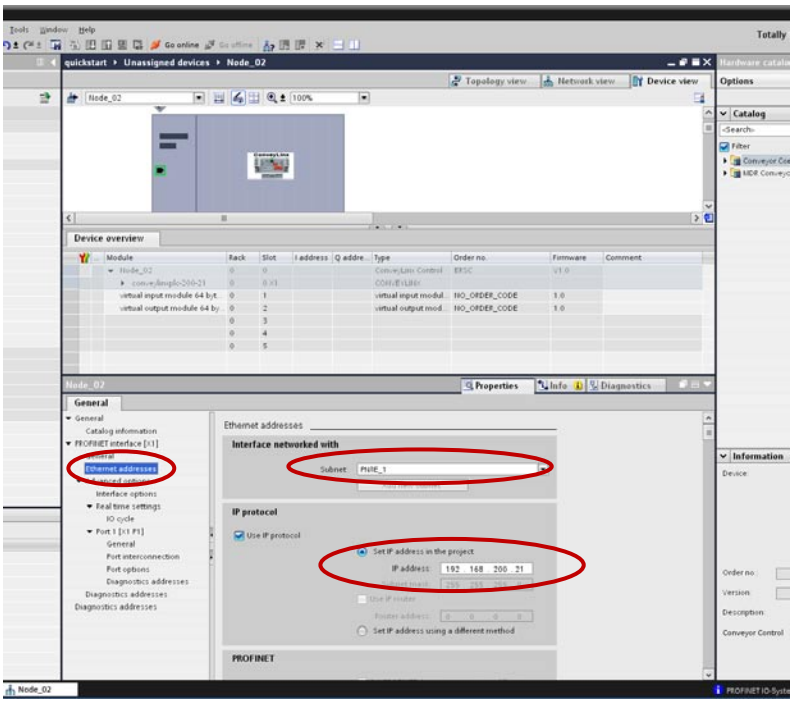


Just to show another way to enter the Project Name for the module, you can select the module in Network View and then change to Device View. Double click in the module's name in the Device Overview window and the properties are displayed. In our example we entered "Node_02" as the Project name for this module.



Select "General" under the "Profinet Interface" and enter in the exact *Profinet Name* syntax for the specific module. In this case we enter "conveylinxplc-200-21"

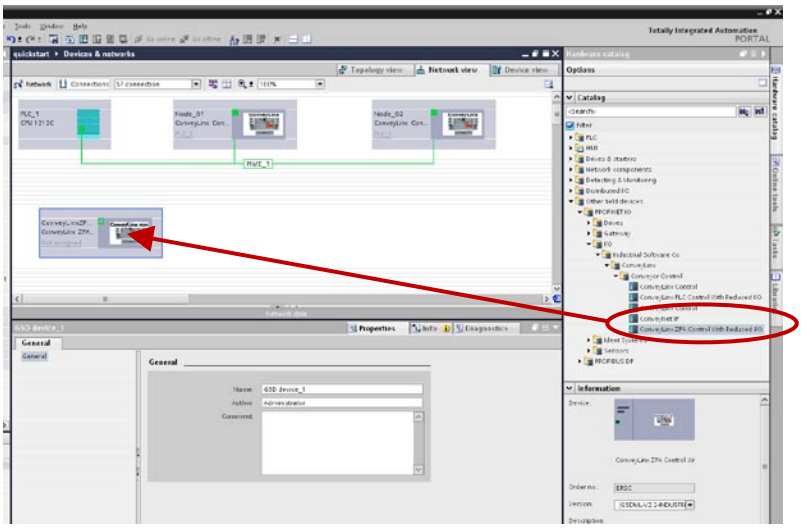




Select *Ethernet Addresses* and select you PLC's subnet and enter the I.P. address of the module you are adding. In this case it is "192.168.200.21".

The Full PLC Controlled Mode module has been added to your project.

ADD REDUCED ZPA MODE MODULE



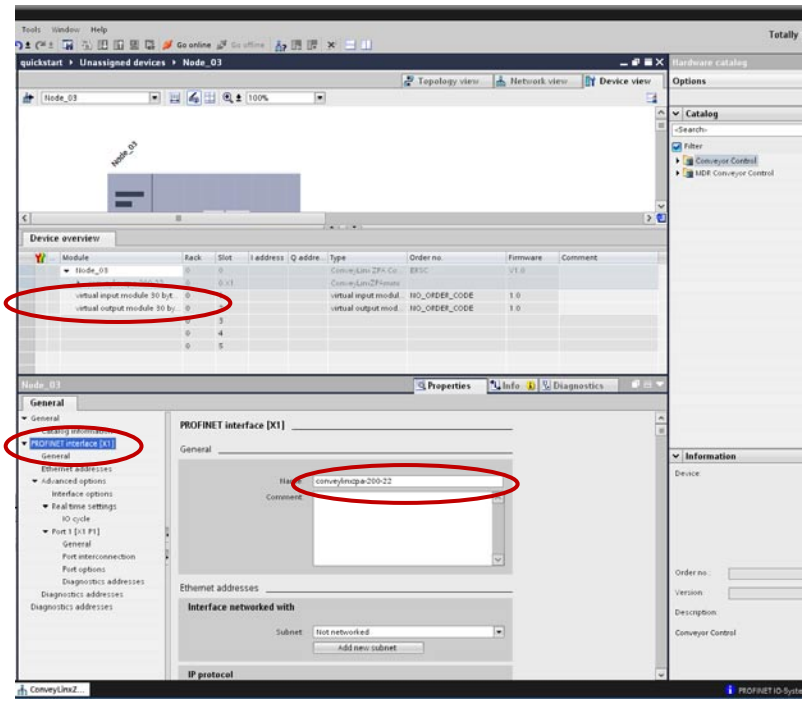
For this module we need to drag and drop a different item from the Hardware Catalog. Drag and drop the *ConveyLinx ZPA Control with Reduced IO* device to the Network View window.

This module will be "Node_03". Follow the same procedure as in the previous section to assign the "Project Name" to the module

Similarly as previous modules, enter the *Profinet Name* for the module in the "General" area of the "Profinet Addresses" selection. In this case the Profinet Name is "conveylinxzpa-200-22"

Also note that the I/O byte size is 30 bytes which is reduced from the 64 bytes used for the Full ZPA Mode.

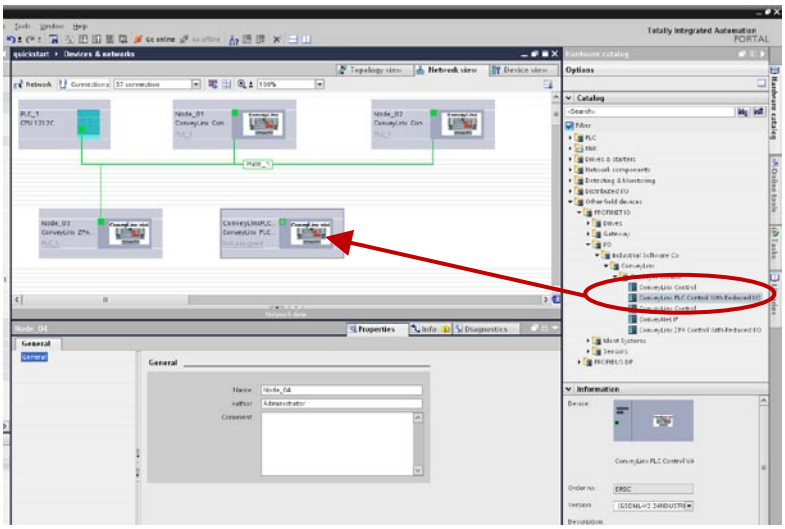
Enter the I.P. address and select the PLC subnet in the same way as in the previous examples, in this case "192.168.200.22". The Reduced ZPA Mode module will then be added to your project



ADD REDUCED PLC CONTROLLED MODE MODULE

For this module we need to drag and drop a different item from the Hardware Catalog. Drag and drop the *ConveyLinx PLC Control with Reduced IO* device to the Network View window.

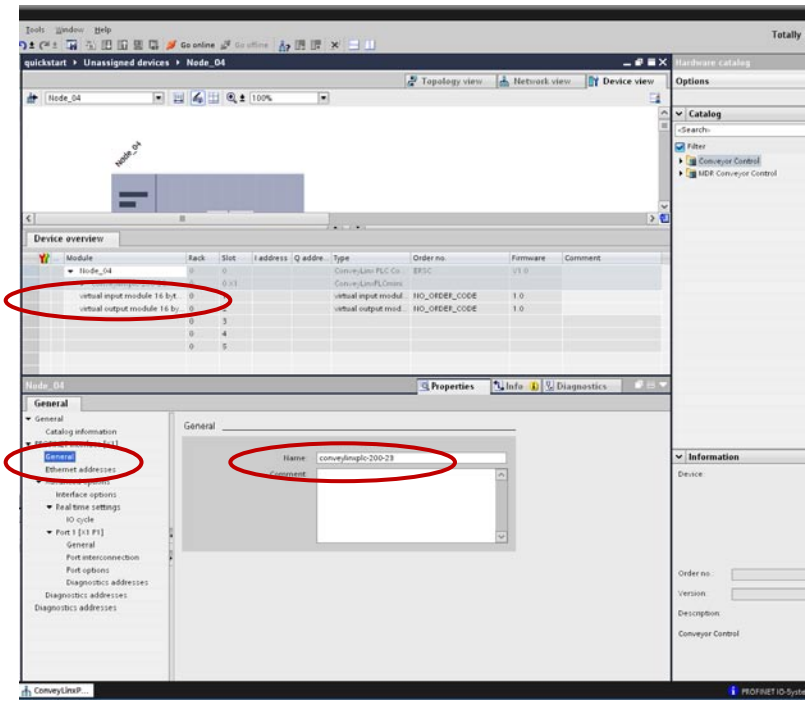
This module will be "Node_04". Follow the same procedure as in the previous section to assign the "Project Name" to the module



Similarly as previous modules, enter the *Profinet Name* for the module in the "General" area of the "Profinet Addresses" selection. In this case the Profinet Name is "conveylinxzplc-200-23"

Also note that the I/O byte size is 16 bytes which is reduced from the 64 bytes used for the Full PLC Controlled Mode.

Enter the I.P. address and select the PLC subnet in the same way as in the previous examples, in this case "192.168.200.23". The Reduced ZPA Mode module will then be added to your project



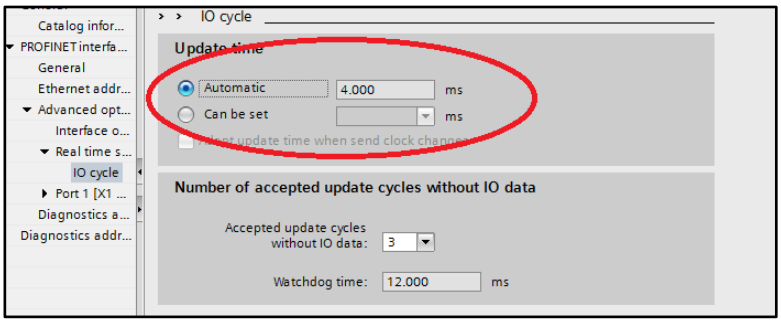
TIA Portal does not have a feature to "uninstall" a previous GSD file so, if you have installed previous versions of the ConveyLinx GSD , there may be items listed in the device tree that are older versions. The "Order No." field in the "Information" area in the bottom left should indicate "ERSC" for the correct selection.

Assigning Module Update Time

The module update time can be adjusted from default and should be adjusted depending upon the Mode being used.

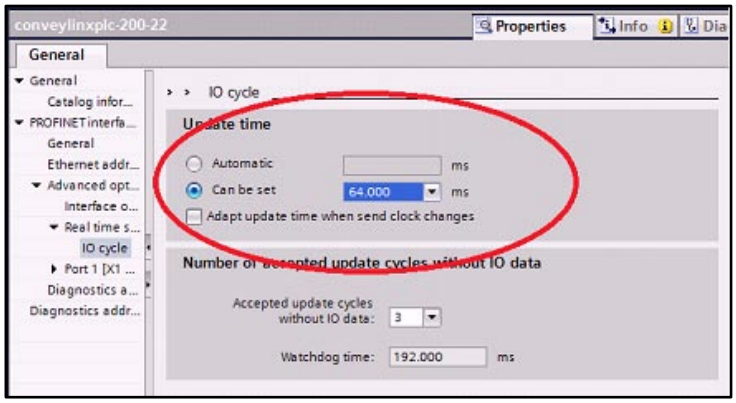
Full and Reduced PLC Controlled Modes

For Full and Reduced PLC Controlled Modes, you can leave the default 4 ms Update Time Interval.



Full and Reduced ZPA Modes

For Full and Reduced ZPA Modes, you should use a value no smaller than 32 ms. In this example a value of 64 is used.



Using an I/O cycle update time of less than 32 ms on a module in ZPA mode (Full or Reduced) could result in communication and/or performance degradation.

USING DATA FROM CONVEYLINX MODULES

Depending on which module type and mode you added to your project, the memory bytes used in the PLC can vary. The following chart indicates each module type and mode and the number of PLC bytes used:

Mode	Hardware Catalog Item	Input Bytes	Output Bytes
Full ZPA	*ConveyLinx Control*	64	64
Full PLC Controlled	*ConveyLinx Control*	64	64
Reduced ZPA	*ConveyLinx ZPA Control with Reduced I/O*	30	30
Reduced PLC	*ConveyLinx PLC Control with Reduced I/O*	16	16

You may work directly with the PLC I/O data as it appears in your PLC's data memory table. For example when you attach a Full ZPA module at address %IB256, you can access its 64 bytes in order in the block from %IB256 through %IB320.

To work directly with ConveyLinx module data, please refer to either ConveyLinx PLC Developer's Guide (Insight Automation publication ERSC-1500) or ConveyLinx-Ai PLC Developer's Guide (Insight Automation publication ERSC-1510) for details on the meaning and use of this data.

USING USER DEFINED DATA TYPES (UDT'S)

Dealing directly with the "raw" module data in the PLC will work, but it may be difficult for some programmers to follow another's program. Some programmers prefer using UDT's to help make their programs easier to follow and debug. UDT structure files (.sci files) are available for download for you to install in your programming environment. You simply use GETIO and SETIO function blocks to map the "raw" data to these useful UDT's. The following sections list the contents of these UDT files.

Please see

Appendix A – UDT File Examples **for examples of the full .sci file syntax for each UDT**

UDT FOR FULL ZPA MODE

The following 2 sections show the UDT structure elements and data types for the Input and Output data for Full ZPA Mode.

CLXZPA_IN

This chart is for the data coming from the ConveyLinx module that is input to the PLC

Structure Element Name	Data Type
StateUpstreamZoneRev	Byte
StateUpstreamZoneFwd	Byte
StateDownstreamZoneRev	Byte
StateDownstreamZoneFwd	Byte
ArrivalCounterUpstreamZone	Int
DepartureCounterUpstreamZone	Int
ArrivalCounterDownstreamZone	Int
DepartureCounterDownstreamZone	Int
Diagnostic	Array[0..31] of Bool
TrackingUpstreamZone	DWord
TrackingDownstreamZone	DWord
ReleaseCounterUpstreamZone	Int
ReleaseCounterDownstreamZone	Int
ModuleDischargeTrackingFwd	DWord
ModuleDischargeTrackingRev	DWord
SensorControlPortInputs	Array [0 .. 15] of Bool
Reserved	Word
ConveyStop_Status	Word
Future	Array [21 .. 31] of Word

CLXZPA_OUT

This chart is for the data coming from the PLC as output to the module

Structure Element Name	Data Type
InductTrackingOnUpstreamZone	DWord
InductTrackingOnDownstreamZone	DWord
AccumulateControlUpstream	Array[0..15] of Bool
AccumulateControlDownstream	Array[0..15] of Bool
SpeedLeftMDR	Int
SpeedRightMDR	Int
ReleaseControlUpstream	Int
ReleaseControlDownstream	Int
InductControlState	Word
DishargeControlState	Word
ModuleInductTrackingOnInductSide	DWord
ModuleInductTrackingOnDishargeSide	DWord
ClearMotorError	Word
ControlPortOututs	Array [0 .. 15] of Bool
Reserved	Word

Structure Element Name	Data Type
ConveyStop_Command	Word
JamClearUpstream	Word
JamClearDownstream	Word
GlobalDirectionControlUpstream	Word
GlobalDirectionControlDownstream	Word
Future	Array[24..31]of Word

UDT FOR FULL PLC CONTROLLED MODE

The following 2 sections show the UDT structure elements and data types for the Input and Output data for Full PLC Controlled Mode.

CLXPLC_IN

This chart is for the data coming from the ConveyLinx module that is input to the PLC

Structure Element Name	Data Type
ConveyStop_Status	Word
AllSensors	Array[0..15] of Bool
SensorDetect	Array[0..15] of Bool
Voltage	Int
LeftCurrent	Int
LeftFreq	Int
LeftCalcTemp	Byte
LeftMeasuredTemp	Byte
LeftMDRDiagnostic	Word
RightCurrent	Int
RightFreq	Int
RightCalcTemp	Byte
RightMeasuredTemp	Byte
RightMDRDiagnostic	Word
LeftDigitalIOStatus	Word
RightDigitalIOStatus	Word
UpstreamModuleStatus	Word
DownstreamModuleStatus	Word
TrackingFromUpstream	DWord
Reserved	Word
DistanceLeft	Int
DistanceRight	Int
ServoStatusLeft	Word
ServoStatusRight	Word
Future	Array[23..31] of Word

CLXPLC_OUT

This chart is for the data coming from the PLC as output to the module

Structure Element Name	Data Type
ConveyStop_Command	Word
LeftMDRAsDigitalIO	Array [0 .. 15] of Bool
RightMDRAsDigitalIO	Array [0 .. 15] of Bool
ControlPortDigitalOutputs	Array [0 .. 15] of Bool
LeftMDRControl	Array [0 .. 15] of Bool
LeftMDRBrakeMode	Word
LeftMDRPIMode	Word
RightMDRControl	Array [0 .. 15] of Bool
RightMDRBrakeMode	Word
RightMDRPIMode	Word
LeftMDRSpeed	Int
RightMDRSpeed	Int
LeftMDRAccel	Int
LeftMDRDeccel	Int
RightMDRAccel	Int
RightMDRDeccel	Int
ClearMDRError	Word
StatusToDownstream	Word
StatusToUpstream	Word
SensorPolarity	Array [0 .. 15] of Bool
TrackingToDownstream	DWord
Reserved	Word
ServoControlDistanceLeft	Int
ServoControlCommandLeft	Word
ServoControlDistanceRight	Int
ServoControlCommandRight	Word
Future	Array [27 ..31] of Word

UDT for Reduced PLC Controlled Mode

The following 2 sections show the UDT structure elements and data types for the Input and Output data for Reduced PLC Controlled Mode.

CLXPLCMINI_IN

This chart is for the data coming from the ConveyLinx module that is input to the PLC

Structure Element Name	Data Type
ControlPortInputs	Array[0..15] of Bool
SensorDetect	Array[0..15] of Bool
LeftCalcTemp	Byte
LeftMeasuredTemp	Byte
LeftMDRDiagnostic	Word
RightCalcTemp	Byte
RightMeasuredTemp	Byte
RightMDRDiagnostic	Word
LeftDigitalIOStatus	Word
RightDigitalIOStatus	Word

CLXPLCMINI_OUT

This chart is for the data coming from the PLC as output to the module

Structure Element Name	Data Type
LeftMDRAsDigitalIO	Array[0..15] of Bool
RigtMDRAsDigitalIO	Array[0..15] of Bool
ControlPortDigitalOutputs	Array[0..15] of Bool
LeftMDRControl	Array[0..15] of Bool
RightMDRControl	Array[0..15] of Bool
LeftMDRSpeed	Int
RightMDRSpeed	Int
ClearMDRError	Word

UDT FOR REDUCED ZPA MODE

The following 2 sections show the UDT structure elements and data types for the Input and Output data for Reduced ZPA Mode.

CLXZPAMINI_IN

This chart is for the data coming from the ConveyLinx module that is input to the PLC

Structure Element Name	Data Type
StateUpstreamZoneRev	Byte
StateUpstreamZoneFwd	Byte
StateDownstreamZoneRev	Byte
StateDownstreamZoneFwd	Byte
ArrivalCounterUpstreamZone	Int
DepartureCounterUpstreamZone	Int
ArrivalCounterDownstreamZone	Int
DepartureCounterDownstreamZone	Int
Diagnostic	Array[0..31] of Bool
ReleaseCounterUpstreamZone	Int
ReleaseCounterDownstreamZone	Int
SensorControlPortInputs	Array [0 .. 15] of Bool
Future	Array [12 .. 15] of Word

CLXZPAMINI_OUT

This chart is for the data coming from the PLC as output to the module

Structure Element Name	Data Type
AccumulateControlUpstream	Array[0..15] of Bool
AccumulateControlDownstream	Array[0..15] of Bool
SpeedLeftMDR	Int
SpeedRightMDR	Int
ReleaseControlUpstream	Int
ReleaseControlDownstream	Int
InductControlState	Word
DishargeControlState	Word
ClearMotorError	Word
ControlPortOututs	Array [0 .. 15] of Bool
Reserved	Word
JamClearUpstream	Word
JamClearDownstream	Word
GlobalDirectionControlUpstream	Word
GlobalDirectionControlDownstream	Word

TROUBLESHOOTING TIPS

Issue	Possible Cause	Action
PLC Cannot Communicate with a module	Incorrect Name Syntax	Verify that the Profinet Name follows the syntax as described in section If online, you can check module connections to your PLC with the Accessible Nodes function in the TIA Portal.
	Incorrect I.P. Address	Make sure module's IP address is correct per section *Adding Modules to Your* Project
	IP Subnet Mask Mismatch	Make sure PLC's subnet mask matches subnet mask in all ConveyLinx modules.
	Firmware Revision	For Full ZPA, Full PLC Controlled, and Reduced PLC Controlled modes - ConveyLinx firmware 4.3 and higher For Reduced ZPA Mode – ConveyLinx firmware 4.24 and higher
Intermittent Communication drop out between PLC and module	Module Update Time	For ZPA modules, update time must be 32 msec or higher. Refer to section **TIA Portal does not have a feature to "uninstall" a previous GSD file so, if you have installed previous versions of the ConveyLinx GSD , there may be items listed in the device tree that are older versions. The "Order No." field in the "Information" area in the bottom left should indicate "ERSC" for the correct selection.** Assigning Module Update Time
	PLC Connection Limit Reached	Verify quantity of connections and memory limitations for the PLC processor being used. Reduce number of connected modules or their respective Modes (i.e. change from "Full" to "Reduced") or upgrade PLC processor
	Faulty Cabling or Ethernet Switch	Verify Ethernet cable connections and cables as well as any Ethernet switches being used

APPENDIX A – UDT FILE EXAMPLES

```
TYPE "CLXZPA_IN"
VERSION : 1.2
   STRUCT
      "StateUpstreamZoneRev" { S7_HMI_Visible := 'False' }  : Byte;
      "StateUpstreamZoneFwd" { S7_HMI_Visible := 'False' }  : Byte;
      "StateDownstreamZoneRev" { S7_HMI_Visible := 'False' }  : Byte;
      "StateDownstreamZoneFwd" { S7_HMI_Visible := 'False' }  : Byte;
      "ArrivalCounterUpstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "DepartureCounterUpstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "ArrivalCounterDownstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "DepartureCounterDownstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "Diagnostic" { S7_HMI_Visible := 'False' }  : Array[0..31] of Bool;
      "TrackingUpstreamZone" { S7_HMI_Visible := 'False' }  : DWord;
      "TrackingDownstreamZone" { S7_HMI_Visible := 'False' }  : DWord;
      "ReleaseCounterUpstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "ReleaseCounterDownstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "ModuleDischargeTrackingFwd" { S7_HMI_Visible := 'False' }  : DWord;
      "ModuleDichargeTrackingRev" { S7_HMI_Visible := 'False' }  : DWord;
      "ControlPortInputs" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "Reserved" { S7_HMI_Visible := 'False' }  : Word;
      "ConveyStop_Status" { S7_HMI_Visible := 'False' }  : Word;
      "Future" { S7_HMI_Visible := 'False' }  : Array [21 .. 31] of Word;
   END_STRUCT;

END_TYPE
```

```
TYPE "CLXZPA_OUT"
VERSION : 1.2
   STRUCT
      "InductTrackingOnUpstreamZone" { S7_HMI_Visible := 'False' }  : DWord;
      "InductTrackingOnDownstreamZone" { S7_HMI_Visible := 'False' }  : DWord;
      "AccumulateControlUpstream" { S7_HMI_Visible := 'False' }  : Array[0..15] of  Bool;
      "AccumulateControlDownstream" { S7_HMI_Visible := 'False' }  : Array[0..15] of  Bool;
      "SpeedLeftMDR" { S7_HMI_Visible := 'False' }  : Int;
      "SpeedRightMDR" { S7_HMI_Visible := 'False' }  : Int;
      "ReleaseControlUpstream" { S7_HMI_Visible := 'False' }  : Int;
      "ReleaseControlDownstream" { S7_HMI_Visible := 'False' }  : Int;
      "InductControlState" { S7_HMI_Visible := 'False' }  : Word;
      "DishargeControlState" { S7_HMI_Visible := 'False' }  : Word;
      "ModuleInductTrackingOnInductSide" { S7_HMI_Visible := 'False' }  : DWord;
      "ModuleInductTrackingOnDishargeSide" { S7_HMI_Visible := 'False' }  : DWord;
      "ClearMotorError" { S7_HMI_Visible := 'False' }  : Word;
      "ControlPortOututs" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "Reserved" { S7_HMI_Visible := 'False' }  : Word;
      "ConveyStop_Control" { S7_HMI_Visible := 'False' }  : Word;
      "JamClearUpstream" { S7_HMI_Visible := 'False' }  : Word;
      "JamClearDownstream" { S7_HMI_Visible := 'False' }  : Word;
      "GlobalDirectionControlUpstream" { S7_HMI_Visible := 'False' }  : Word;
      "GlobalDirectionControlDownstream" { S7_HMI_Visible := 'False' }  : Word;
      "Future" : Array[24..31]of Word;
   END_STRUCT;

END_TYPE
```

```
TYPE "CLXPLC_IN"
VERSION : 1.2
   STRUCT
      "ConveyStop_status" { S7_HMI_Visible := 'False' }  : Word;
      "AllSensors" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "SensorDetect" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "Voltage" { S7_HMI_Visible := 'False' }  : Int;
      "LeftCurrent" { S7_HMI_Visible := 'False' }  : Int;
      "LeftFreq" { S7_HMI_Visible := 'False' }  : Int;
      "LeftCalcTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "LeftMeasuredTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "LeftMDRDiagnostic" { S7_HMI_Visible := 'False' }  : Word;
      "RightCurrent" { S7_HMI_Visible := 'False' }  : Int;
      "RightFreq" { S7_HMI_Visible := 'False' }  : Int;
      "RightCalcTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "RightMeasuredTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "RightMDRDiagnostic" { S7_HMI_Visible := 'False' }  : Word;
      "LeftDigitalIOStatus" { S7_HMI_Visible := 'False' }  : Word;
      "RightDigitalIOStatus" { S7_HMI_Visible := 'False' }  : Word;
      "UpstreamModuleStatus" { S7_HMI_Visible := 'False' }  : Word;
      "DownstreamModuleStatus" { S7_HMI_Visible := 'False' }  : Word;
      "TrackingFromUpstream" { S7_HMI_Visible := 'False' }  : DWord;
      "Reserved" { S7_HMI_Visible := 'False' }  : Word;
      "DistanceLeft" { S7_HMI_Visible := 'False' }  : Int;
      "DistanceRight" { S7_HMI_Visible := 'False' }  : Int;
      "ServoStatusLeft" : Word;
      "ServoStatusRight" : Word;
      "Future" { S7_HMI_Visible := 'False' }  : Array[23..31] of Word;
   END_STRUCT;

END_TYPE

TYPE "CLXPLC_OUT"
VERSION : 1.2
   STRUCT
      "ConveyStop_Command" { S7_HMI_Visible := 'False' }  : Word;
      "LeftMDRAsDigitalIO" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "RigtMDRAsDigitalIO" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "ControlPortDigitalOutputs" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "LeftMDRControl" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "LeftMDRBrakeMode" { S7_HMI_Visible := 'False' }  : Word;
      "LeftMDRPIMode" { S7_HMI_Visible := 'False' }  : Word;
      "RightMDRControl" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "RightMDRBrakeMode" { S7_HMI_Visible := 'False' }  : Word;
      "RightMDRPIMode" { S7_HMI_Visible := 'False' }  : Word;
      "LeftMDRSpeed" { S7_HMI_Visible := 'False' }  : Int;
      "RightMDRSpeed" { S7_HMI_Visible := 'False' }  : Int;
      "LeftMDRAccel" { S7_HMI_Visible := 'False' }  : Int;
      "LeftMDRDeccel" { S7_HMI_Visible := 'False' }  : Int;
      "RightMDRAccel" { S7_HMI_Visible := 'False' }  : Int;
      "RightMDRDeccel" { S7_HMI_Visible := 'False' }  : Int;
      "ClearMDRError" { S7_HMI_Visible := 'False' }  : Word;
      "StatusToDownstream" { S7_HMI_Visible := 'False' }  : Word;
      "StatusToUpstream" { S7_HMI_Visible := 'False' }  : Word;
      "SensorPolarity" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "TrackingToDownstream" { S7_HMI_Visible := 'False' }  : DWord;
      "Reserved" { S7_HMI_Visible := 'False' }  : Word;
      "ServoControlDistanceLeft" { S7_HMI_Visible := 'False' }  : Int;
      "ServoControlCommandLeft" { S7_HMI_Visible := 'False' }  : Word;
      "ServoControlDistanceRight" { S7_HMI_Visible := 'False' }  : Int;
      "ServoControlCommandRight" { S7_HMI_Visible := 'False' }  : Word;
      "Future" { S7_HMI_Visible := 'False' }  : Array [27 ..31] of Word;
   END_STRUCT;

END_TYPE
```

```
TYPE "CLXPLCMINI_IN"
VERSION : 1.2
   STRUCT
      "ControlPortInputs" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "SensorDetect" { S7_HMI_Visible := 'False' }  : array[0..15] of Bool;
      "LeftCalcTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "LeftMeasuredTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "LeftMDRDiagnostic" { S7_HMI_Visible := 'False' }  : Word;
      "RightCalcTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "RightMeasuredTemp" { S7_HMI_Visible := 'False' }  : Byte;
      "RightMDRDiagnostic" { S7_HMI_Visible := 'False' }  : Word;
      "LeftDigitalIOStatus" { S7_HMI_Visible := 'False' }  : Word;
      "RightDigitalIOStatus" { S7_HMI_Visible := 'False' }  : Word;
   END_STRUCT;

END_TYPE

TYPE "CLXPLCMINI_OUT"
VERSION : 1.2
   STRUCT
      "LeftMDRAsDigitalIO" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "RigtMDRAsDigitalIO" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "ControlPortDigitalOutputs" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "LeftMDRControl" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "RightMDRControl" { S7_HMI_Visible := 'False' }  : Array[0..15] of Bool;
      "LeftMDRSpeed" { S7_HMI_Visible := 'False' }  : Int;
      "RightMDRSpeed" { S7_HMI_Visible := 'False' }  : Int;
      "ClearMDRError" { S7_HMI_Visible := 'False' }  : Word;
   END_STRUCT;

END_TYPE
```

```
TYPE "CLXZPAMINI_IN"
VERSION : 1.2
   STRUCT
      "StateUpstreamZoneRev" { S7_HMI_Visible := 'False' }  : Byte;
      "StateUpstreamZoneFwd" { S7_HMI_Visible := 'False' }  : Byte;
      "StateDownstreamZoneRev" { S7_HMI_Visible := 'False' }  : Byte;
      "StateDownstreamZoneFwd" { S7_HMI_Visible := 'False' }  : Byte;
      "ArrivalCounterUpstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "DepartureCounterUpstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "ArrivalCounterDownstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "DepartureCounterDownstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "Diagnostic" { S7_HMI_Visible := 'False' }  : Array[0..31] of Bool;
      "ReleaseCounterUpstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "ReleaseCounterDownstreamZone" { S7_HMI_Visible := 'False' }  : Int;
      "SensorContrlPortInputs" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "Future" { S7_HMI_Visible := 'False' }  : Array [12 .. 15] of Word;
   END_STRUCT;

END_TYPE


TYPE "CLXZPAMINI_OUT"
VERSION : 1.2
   STRUCT
      "AccumulateControlUpstream" { S7_HMI_Visible := 'False' }  : Array[0..15] of  Bool;
      "AccumulateControlDownstream" { S7_HMI_Visible := 'False' }  : Array[0..15] of  Bool;
      "SpeedLeftMDR" { S7_HMI_Visible := 'False' }  : Int;
      "SpeedRightMDR" { S7_HMI_Visible := 'False' }  : Int;
      "ReleaseControlUpstream" { S7_HMI_Visible := 'False' }  : Int;
      "ReleaseControlDownstream" { S7_HMI_Visible := 'False' }  : Int;
      "InductControlState" { S7_HMI_Visible := 'False' }  : Word;
      "DishargeControlState" { S7_HMI_Visible := 'False' }  : Word;
      "ClearMotorError" { S7_HMI_Visible := 'False' }  : Word;
      "ControlPortOututs" { S7_HMI_Visible := 'False' }  : Array [0 .. 15] of Bool;
      "Reserved" { S7_HMI_Visible := 'False' }  : Word;
      "JamClearUpstream" { S7_HMI_Visible := 'False' }  : Word;
      "JamClearDownstream" { S7_HMI_Visible := 'False' }  : Word;
      "GlobalDirectionControlUpstream" { S7_HMI_Visible := 'False' }  : Word;
      "GlobalDirectionControlDownstream" { S7_HMI_Visible := 'False' }  : Word;
       END_STRUCT;

END_TYPE
```

NOTES:

PULSE ROLLER

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

Publication ERSC-1525 Rev 1.0 – September 2014