

ControlNet Router

(Enhanced)

Example Pack 1

Technical Application Note

A-CNR

Document No. D138-018

Revision 1.0

02/2023

CONTENTS

1. Preface	2
1.1. Purpose of this Document.....	2
1.2. Additional Information.....	2
1.3. Support.....	2
2. Application Description.....	3
3. Setup	4
3.1. Logix.....	4
3.2. Slate.....	6
3.2.1. General.....	6
3.2.2. ControlNet Target	6
3.2.3. EtherNet/IP Originator – Class 1	7
3.2.4. EtherNet/IP Originator – Explicit Messaging	8
3.2.5. Internal Mapping	9



1. PREFACE

1.1. PURPOSE OF THIS DOCUMENT

This document will assist and provide information to the user with the ControlNet Router Example Pack 1.

1.2. ADDITIONAL INFORMATION

The following resources contain additional information that can assist the user with the module installation and operation.

Resource	Link
Slate Installation	http://www.aparian.com/software/slate
ControlNet Router User Manual ControlNet Router Datasheet Example Code & UDTs	http://www.aparian.com/products/controlnetrouter
Ethernet wiring standard	www.cisco.com/c/en/us/td/docs/video/cds/cde/cde205_220_420/installation/guide/cde205_220_420_hig/Connectors.html
CIP Routing	The CIP Networks Library, Volume 1, Appendix C:Data Management
ControlNet	http://www.odva.org
ControlNet Cabling	ControlNet Coax Media Planning and Installation Guide https://literature.rockwellautomation.com/idc/groups/literature/documents/in/cnet-in002_-en-p.pdf

1.3. SUPPORT

Technical support will be provided via the Web (in the form of user manuals, FAQ, datasheets etc.) to assist with installation, operation, and diagnostics.

For additional support the user can use either of the following:

Contact Us web link	https://www.prosoft-technology.com/Services-Support/Customer-Support
Support email	support@prosoft-technology.com

2. APPLICATION DESCRIPTION

The example 1 application has the ControlNet Router operating as an ControlNet Target and EtherNet/IP Originator. This will allow legacy Logix Controllers with existing ControlNet infrastructure to communicate with newer EtherNet/IP devices.

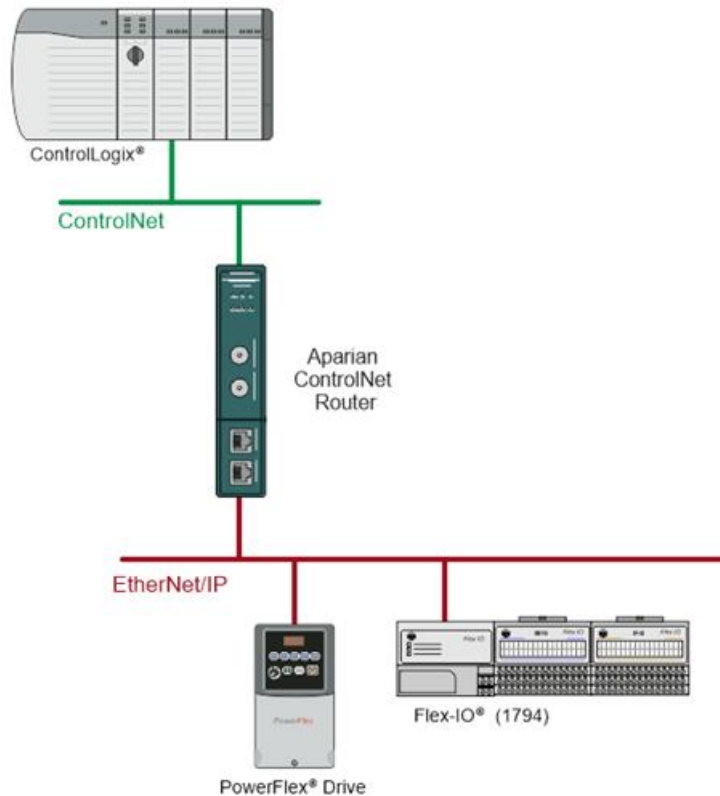


FIGURE 2.1. - EXAMPLE 1 TYPICAL NETWORK SETUP

When the ControlNet Router is configured as a ControlNet Target and EtherNet/IP Originator, a controller (e.g. Logix Controller) can own the ControlNet Router via Scheduled ControlNet (e.g. via 1756-CNB/R) while the ControlNet Router owns EtherNet/IP IO using a class 1 connection.

The data from the controller (e.g., Logix Controller) can be exchanged with the ControlNet Router using the Scheduled ControlNet input and output assembly. This data, in turn, can be mapped to any of the configured Class 1 EtherNet/IP IO input and output assemblies being owned by the ControlNet Router.

The ControlNet Router can also exchange data with EtherNet/IP devices using the Explicit Messaging Map. A CIP message with specific service, class, instance, and attribute can be configured to execute at a configured rate with the data being stored in the internal data space (IDS).

3. SETUP

The following sections will provide information regarding the Example 1 setup and operation.

3.1. LOGIX

In the example Logix project, a CNR was added to the ControlNet bridge and scheduled using RSNetworx for ControlNet.

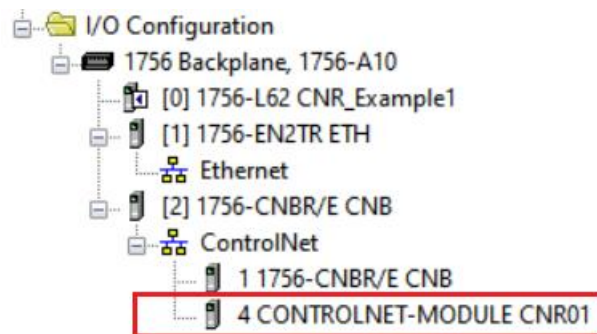


FIGURE 3.1. – CNR LOGIX CONTROLNET IO TREE

In the example project, the CNR01Map Routine is added which will have the COP instructions for the CNR input and output UDTs.

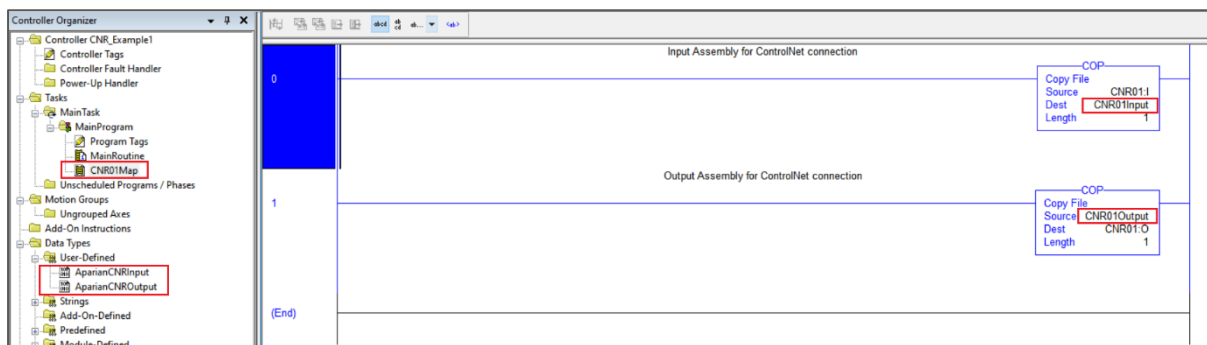


FIGURE 3.2. – CNR LOGIX MAPPING ROUTINE

The data being exchanged in the internal mapping (which is explained later in this document) will be in the *MappedData* areas of the UDTs.

Name	Value	For	Style	Data Type	Description
- CNR01Input	{...}	{..}		AparianCNRInput	
+ CNR01Input.GeneralStatus	0		Decimal	SINT	
+ CNR01Input.CNetNodeNumber	0		Decimal	SINT	
+ CNR01Input.CNetLedState	0		Decimal	SINT	
+ CNR01Input.CNetMode	0		Decimal	SINT	
- CNR01Input.Temperature	0.0		Float	REAL	
- CNR01Input.MappedData	{...}	{..}	Decimal	SINT[400]	
+ CNR01Input.MappedData[0]	0		Decimal	SINT	
+ CNR01Input.MappedData[1]	0		Decimal	SINT	
+ CNR01Input.MappedData[2]	0		Decimal	SINT	
+ CNR01Input.MappedData[3]	0		Decimal	SINT	
+ CNR01Input.MappedData[4]	0		Decimal	SINT	
+ CNR01Input.MappedData[5]	0		Decimal	SINT	

FIGURE 3.3. – INPUT ASSEMBLY MAPPEDDATA

Name	Value	For	Style	Data Type	Description
- CNR01Output	{...}	{..}		AparianCNROutput	
+ CNR01Output.GeneralCtrl	0		Decimal	DINT	
- CNR01Output.MappedData	{...}	{..}	Decimal	SINT[400]	
+ CNR01Output.MappedData[0]	0		Decimal	SINT	
+ CNR01Output.MappedData[1]	0		Decimal	SINT	
+ CNR01Output.MappedData[2]	0		Decimal	SINT	
+ CNR01Output.MappedData[3]	0		Decimal	SINT	
+ CNR01Output.MappedData[4]	0		Decimal	SINT	

FIGURE 3.4. – OUTPUT ASSEMBLY MAPPEDDATA

Two tags were also created to demonstrate the CNR Logix Tag Read/Write functionality.

Name	Value	For	Style	Data Type	Description
+ CNR01:C	{...}	{..}		AB:CONTROLNET_MOD...	
+ CNR01:I	{...}	{..}		AB:CONTROLNET_MOD...	
+ CNR01:O	{...}	{..}		AB:CONTROLNET_MOD...	
+ CNR01Input	{...}	{..}		AparianCNRInput	
+ CNR01Output	{...}	{..}		AparianCNROutput	
+ TestReadTag	0		Decimal	DINT	
+ TestWriteTag	0		Decimal	DINT	

FIGURE 3.5. – CNR LOGIX TAGS

CNR01 - Configuration													
General ControlNet ControlNet Devices ControlNet Map EtherNet/IP Devices EtherNet/IP Map Modbus Modbus Auxiliary Map Internal Map Advanced													
Explicit EtherNet/IP Map (max. of 50 items.)													
Device	Function	Scan	Service	Class	Instance	Attribute	Input Offset	Get Length	Output Offset	Set Length	Data Type	Tag / Static Value	...
PLX51-DLP	Get	A	0x00	0x0001	1	7	4000	10	0	0		0	
1756-L62	Read Tag	A	0x00	0x0000	0	0	4020	0	0	0		TestReadTag	...
PLX51-PBS	Get	A	0x00	0x0001	1	7	4040	10	0	0		0	
1756-L62	Write Tag	A	0x00	0x0000	0	0	0	0	4060	1		TestWriteTag	...
PLX51-DLP	Set	A	0x00	0x0409	1	4	0	0	4080	4		0	

FIGURE 3.6. – SLATE EXPLICIT MAPPING FOR LOGIX TAGS.

3.2. SLATE

The ControlNet Router will need to be configured, using the Slate software utility, so that it can have a scheduled connection to a ControlNet bridge (e.g., 1756-CNB) as well as Class 1 and Explicit Message data exchange with a EtherNet/IP IO.

3.2.1. GENERAL

The user need to configure the CNR as a ControlNet Target and a EtherNet/IP Originator (as shown below).

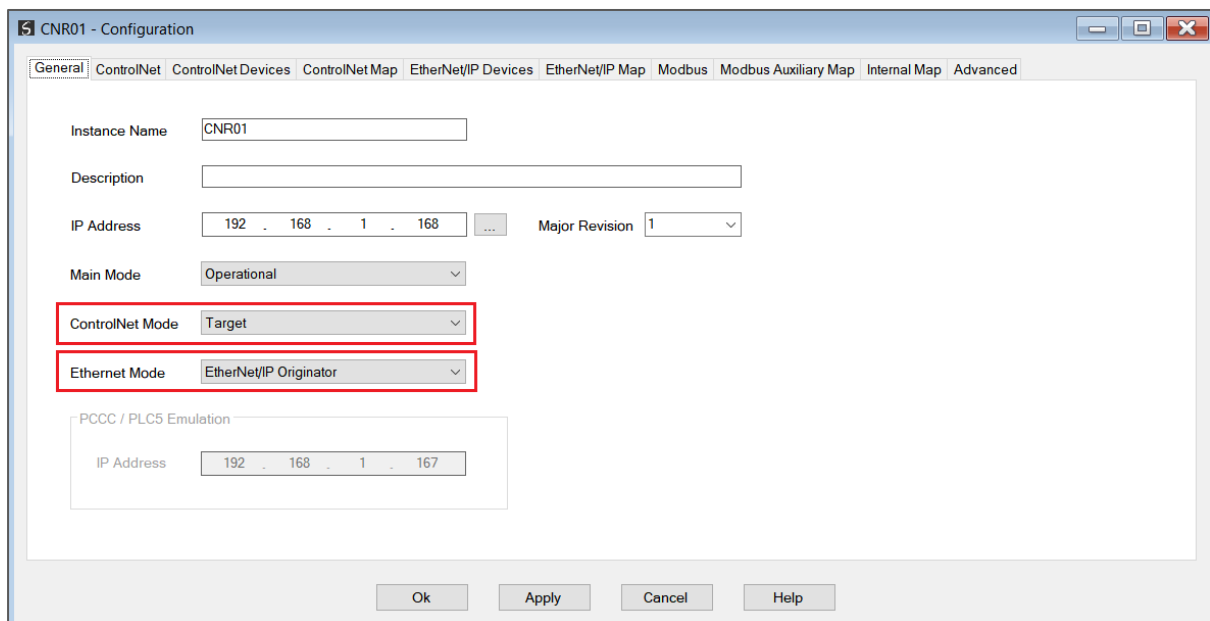


FIGURE 3.7. – GENERAL CONFIGURATION

3.2.2. CONTROLNET TARGET

Next the ControlNet Input and Output assembly sizes for the CNR ControlNet scheduled connection must be configured. In the example below the CNR will produce 128 bytes of data (the input assembly in Logix) and consume 64 bytes of data (the output assembly in Logix).

See section 3.5.1.2 in the ControlNet Router user manual for information regarding the RSNetworkx for ControlNet setup with the CNR module.

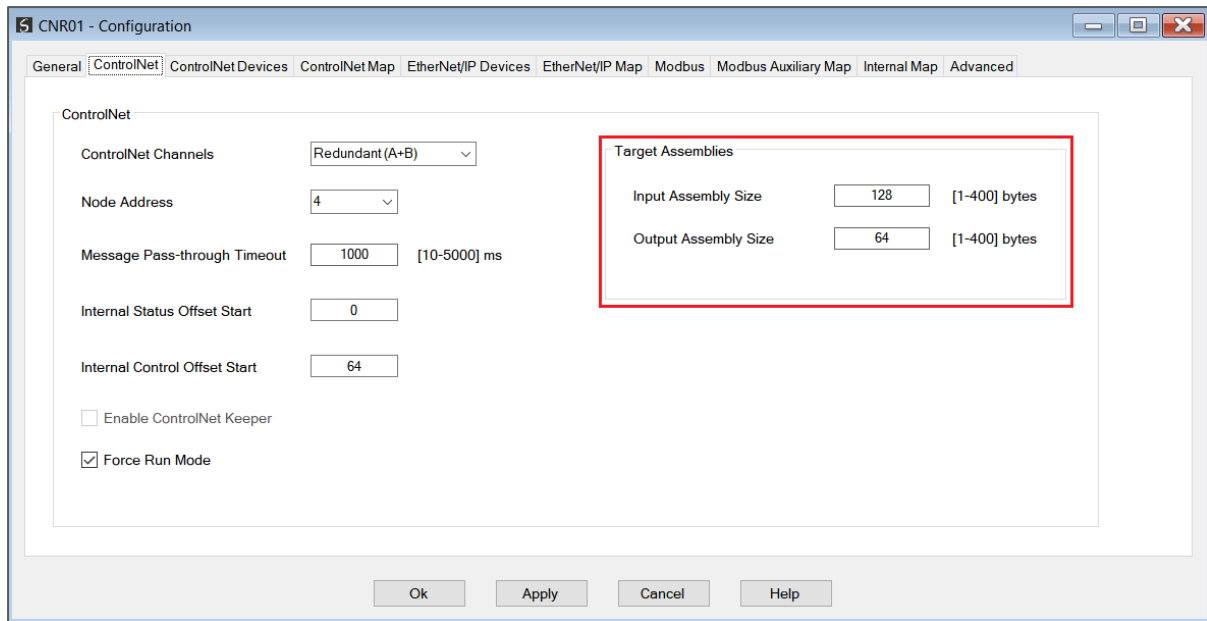


FIGURE 3.8. – CONTROLNET CONFIGURATION

3.2.3. ETHERNET/IP ORIGINATOR – CLASS 1

To add EtherNet/IP devices to the CNR EtherNet/IP connections tree, see section 3.6.5.1 of the CNR user manual which will provide information regarding the EtherNet/IP IO device setup.

Each EtherNet/IP device will require a path, which could be directed to the target or routed via an Ethernet bridge (e.g., 1756-EN2TR or 1794-AENT). For direct connections only the target IP address must be entered (as shown below).

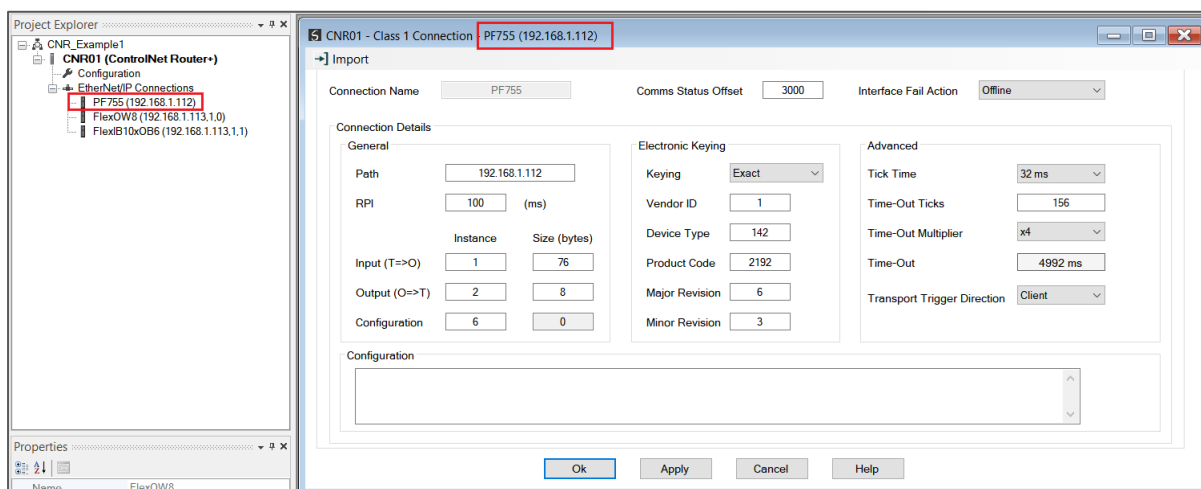


FIGURE 3.9. – ETHERNET/IP ORIGINATOR CLASS 1 DIRECT CONFIGURATION

For routed connections the target IP address must be entered followed by the port and then the target slot or node (as shown below). For example if you need a connection to a 1794 Flex IO module, then the IP address of the Flex adapter will be entered, followed by a comma and

then the internal port number, followed by a comma and then the Flex IO slot number (starting at 0).

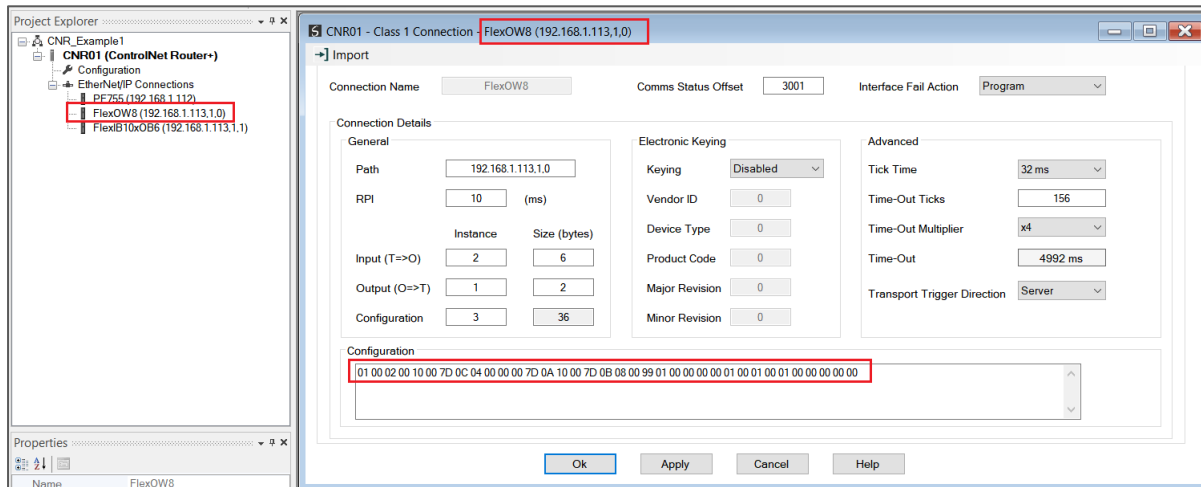


FIGURE 3.10. – ETHERNET/IP ORIGINATOR CLASS 1 ROUTED CONFIGURATION

3.2.4. ETHERNET/IP ORIGINATOR – EXPLICIT MESSAGING

Explicit EtherNet/IP messaging can also be setup allowing the CNR module to communicate at a configured rate with EtherNet/IP devices using either Class3 or UCMM connections. See section 3.6.5.2 of the CNR user manual for more information regarding the explicit messaging setup. Note that the module can also communicate directly with Logix tags.

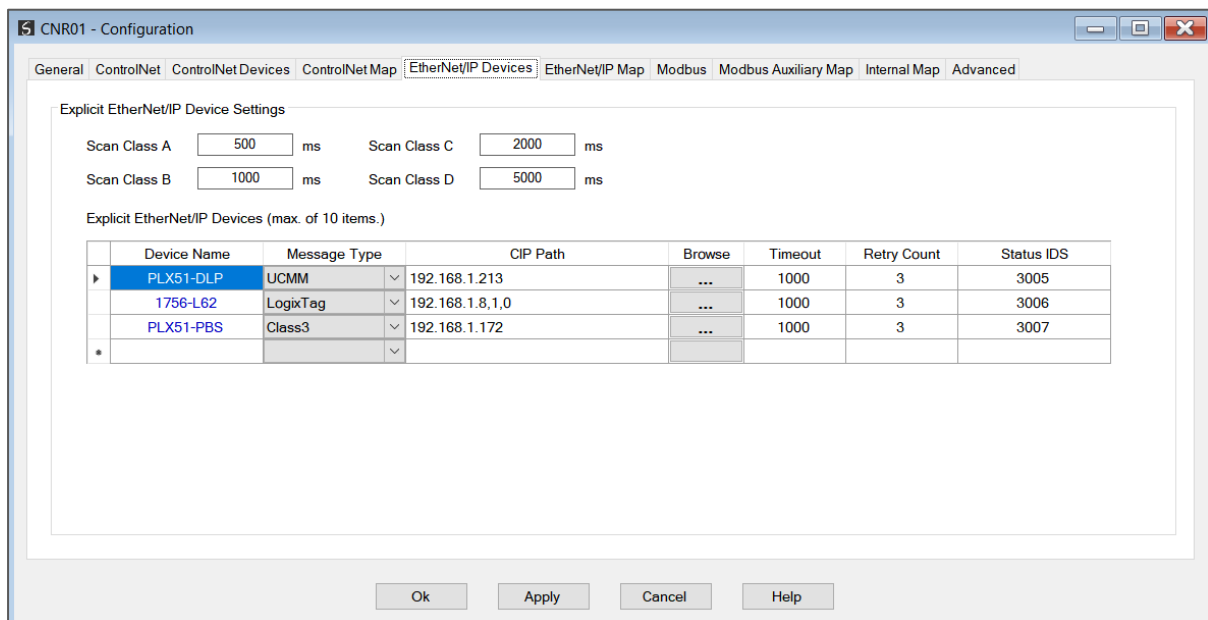


FIGURE 3.11. – ETHERNET/IP ORIGINATOR EXPLICIT DEVICE CONFIGURATION

For each EtherNet/IP explicit messaging device, a range of message instructions can be configured. The data for these functions will be read from or written to the internal data space

(IDS) in the CNR. In the example below the data for the PLX51-PBS Get function will be stored at IDS address 4040.

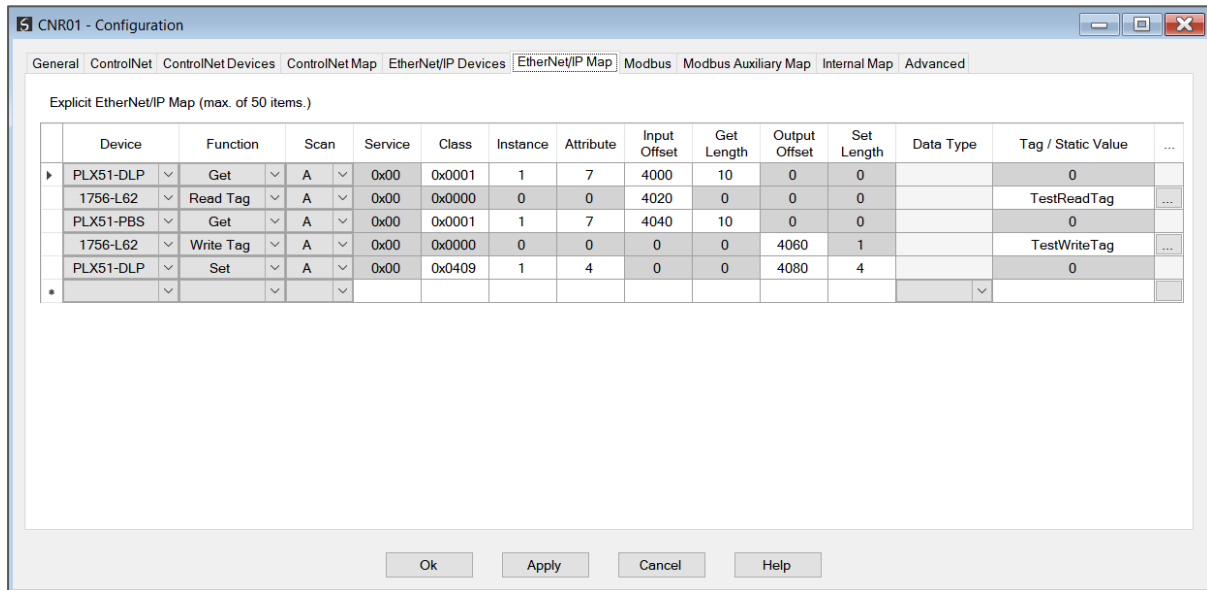


FIGURE 3.12. – ETHERNET/IP ORIGINATOR EXPLICIT MESSAGE CONFIGURATION

3.2.5. INTERNAL MAPPING

The internal mapping will need to be setup once the ControlNet and EtherNet/IP setup has been complete. This will copy data from one interface or data space to another, allowing interfaces to exchange data.

In the example below the data received from the EtherNet/IP devices connected to the CNR (using class 1 EtherNet/IP connections) is copied to the input assembly of the CNR ControlNet scheduled connection (i.e., the data being produced by the CNR on ControlNet). This will allow the Logix controller to access the data from the EtherNet/IP devices via the CNR module.



NOTE: The data being exchange with the CNR Target will be in the *MappedData* tag of the input and output assembly UDTs. So, the source or destination Offset is the offset in the *MappedData* tag, where offset 0 will be *MappedData[0]*.

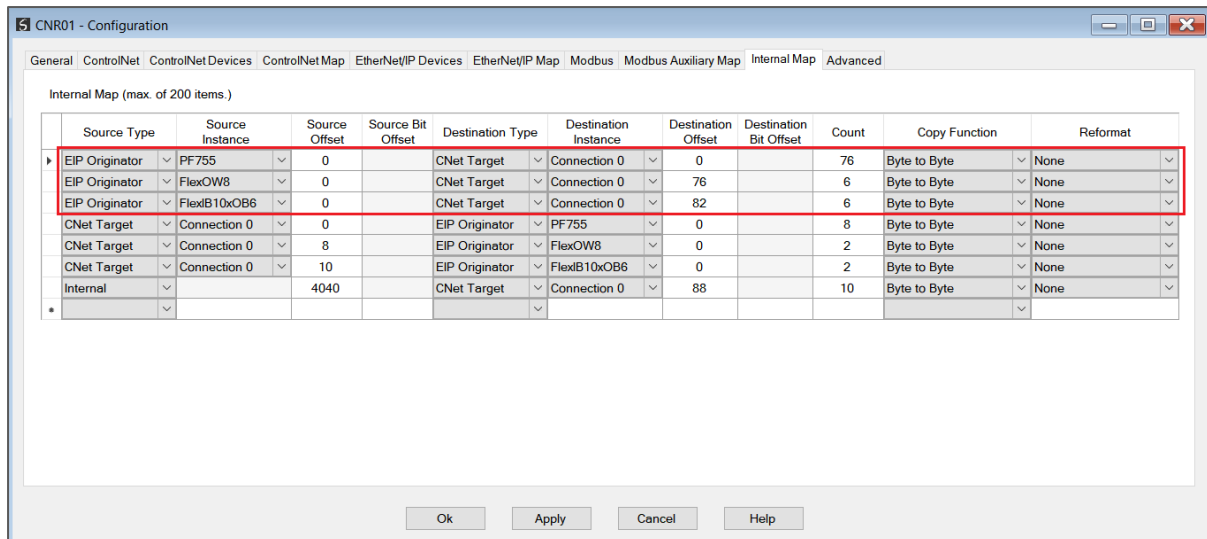


FIGURE 3.13. – INTERNAL MAPPING – EIP ORIGINATOR TO CNET TARGET

In the above example, the data received from EtherNet/IP device PF755 will be copied to the input assembly *MappedData[0]* to *MappedData[75]*, the data received from the FlexOW8 will be copied to the input assembly *MappedData[76]* to *MappedData[81]*, and the data received from the FlexIB10xOB6 will be copied to the input assembly *MappedData[82]* to *MappedData[87]* as shown below.

Name	Value	For	Style	Data Type
+ CNR01Input.MappedData[80]	0		Decimal	SINT
+ CNR01Input.MappedData[81]	0		Decimal	SINT
+ CNR01Input.MappedData[82]	0		Decimal	SINT
+ CNR01Input.MappedData[83]	0		Decimal	SINT
+ CNR01Input.MappedData[84]	0		Decimal	SINT
+ CNR01Input.MappedData[85]	0		Decimal	SINT
+ CNR01Input.MappedData[86]	0		Decimal	SINT
+ CNR01Input.MappedData[87]	0		Decimal	SINT
+ CNR01Input.MappedData[88]	0		Decimal	SINT

FIGURE 3.14. – INTERNAL MAPPING – FLEXIB10xOB6 MAPPED DATA

Next the data from the CNR scheduled ControlNet output assembly (i.e., the data being consumed by the CNR on ControlNet) will need to be copied to the data sent to the EtherNet/IP devices via the Class 1 EtherNet/IP connection.

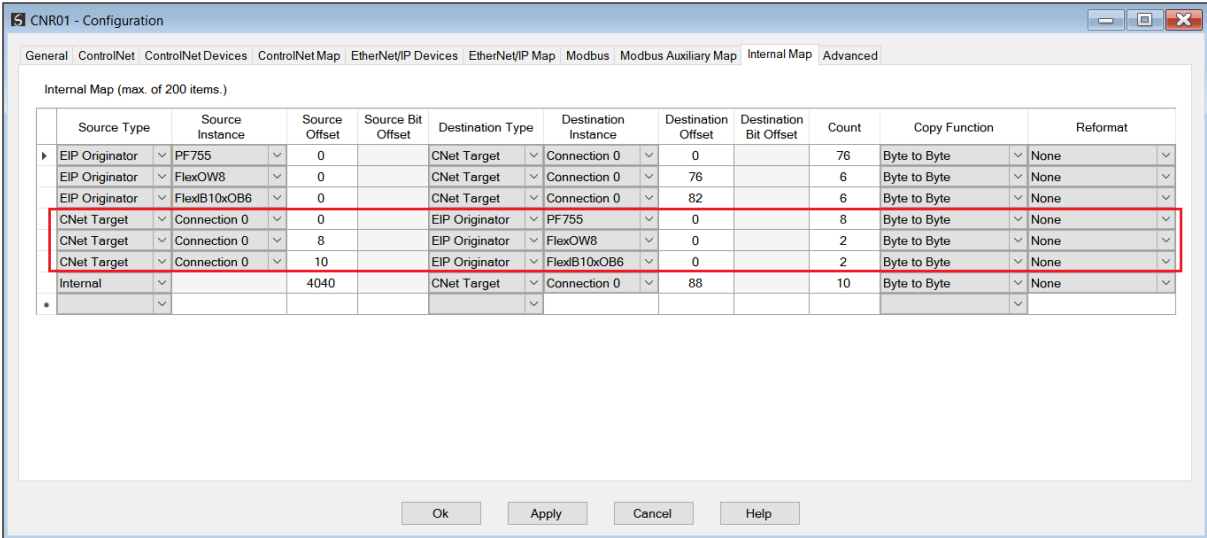


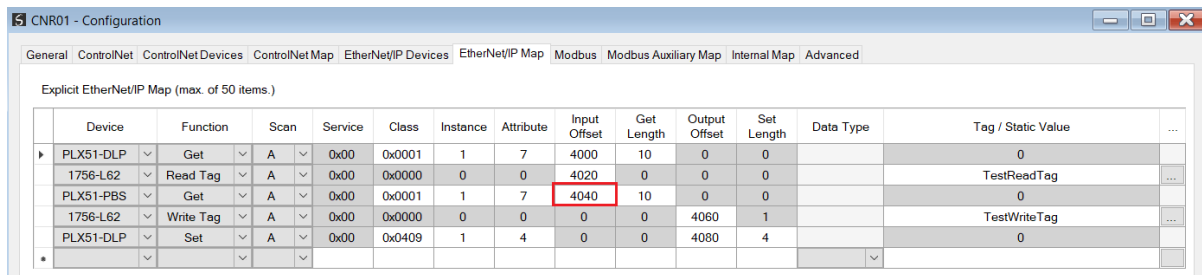
FIGURE 3.15. – INTERNAL MAPPING – CNET TARGET TO EIP ORIGINATOR

In the example, the data from the ControlNet output assembly *MappedData[0]* to *MappedData[7]* is sent to the PF755 EtherNet/IP device (in red below), *MappedData[8]* to *MappedData[9]* is sent to the FlexOW8 EtherNet/IP device (in orange below), and *MappedData[10]* to *MappedData[11]* is sent to the FlexIB10xOB6 EtherNet/IP device (in green below), using the Class 1 EtherNet/IP connection from the CNR.

Name	Value	For	Style	Data Type
- CNR01Output.MappedData	{...}	{..	Decimal	SINT[400]
+ CNR01Output.MappedData[0]	0		Decimal	SINT
+ CNR01Output.MappedData[1]	0		Decimal	SINT
+ CNR01Output.MappedData[2]	0		Decimal	SINT
+ CNR01Output.MappedData[3]	0		Decimal	SINT
+ CNR01Output.MappedData[4]	0		Decimal	SINT
+ CNR01Output.MappedData[5]	0		Decimal	SINT
+ CNR01Output.MappedData[6]	0		Decimal	SINT
+ CNR01Output.MappedData[7]	0		Decimal	SINT
+ CNR01Output.MappedData[8]	0		Decimal	SINT
+ CNR01Output.MappedData[9]	0		Decimal	SINT
+ CNR01Output.MappedData[10]	0		Decimal	SINT
+ CNR01Output.MappedData[11]	0		Decimal	SINT
+ CNR01Output.MappedData[12]	0		Decimal	SINT

FIGURE 3.16. — INTERNAL MAPPING — MAPPED DATA OUTPUT ASSEMBLY

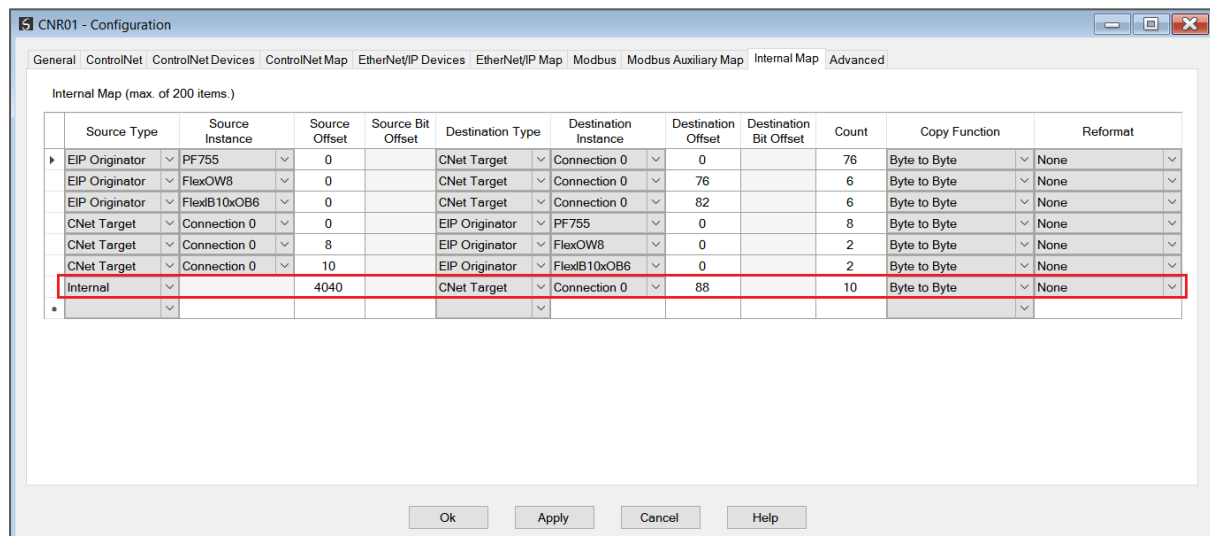
Next the data received from the explicit EtherNet/IP messaging is copied to the CNR Scheduled ControlNet input assembly. The data for the Get function for the PLX51-PBS is stored at address 4040 in the Internal Data Space (IDS).



Explicit EtherNet/IP Map (max. of 50 items.)

Device	Function	Scan	Service	Class	Instance	Attribute	Input Offset	Get Length	Output Offset	Set Length	Data Type	Tag / Static Value
PLX51-DLP	Get	A	0x00	0x0001	1	7	4000	10	0	0		0
1756-L62	Read Tag	A	0x00	0x0000	0	0	4020	0	0	0		TestReadTag
PLX51-PBS	Get	A	0x00	0x0001	1	7	4040	10	0	0		0
1756-L62	Write Tag	A	0x00	0x0000	0	0	0	0	4060	1		TestWriteTag
PLX51-DLP	Set	A	0x00	0x0409	1	4	0	0	4080	4		0

FIGURE 3.17. – EXPLICIT ETHERNET/IP MAP

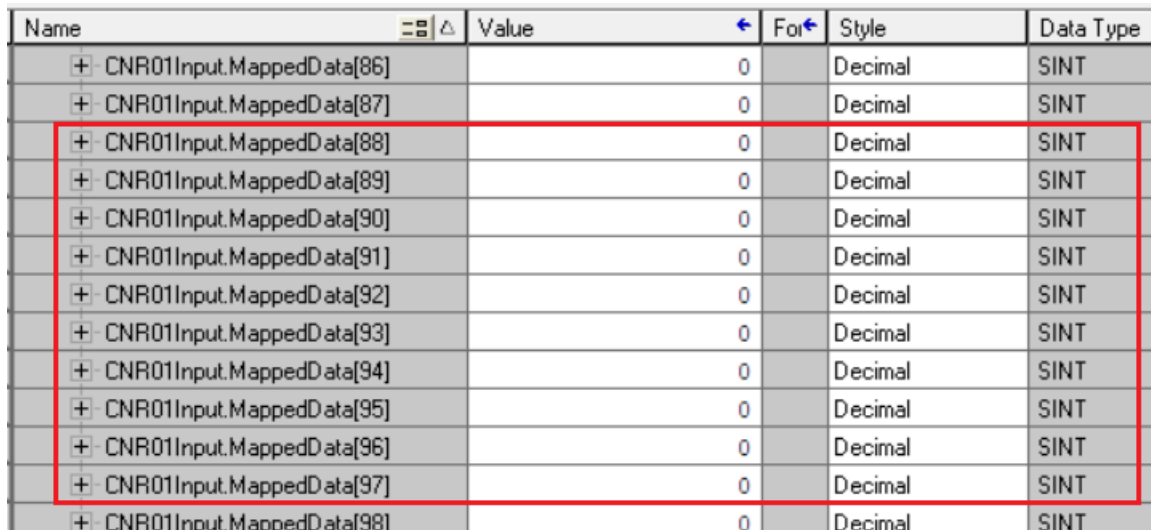


Internal Map (max. of 200 items.)

Source Type	Source Instance	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
EIP Originator	PF755	0		CNet Target	Connection 0	0		76	Byte to Byte	None
EIP Originator	FlexQW8	0		CNet Target	Connection 0	76		6	Byte to Byte	None
EIP Originator	FlexB10xOB6	0		CNet Target	Connection 0	82		6	Byte to Byte	None
CNet Target	Connection 0	0		EIP Originator	PF755	0		8	Byte to Byte	None
CNet Target	Connection 0	8		EIP Originator	FlexQW8	0		2	Byte to Byte	None
CNet Target	Connection 0	10		EIP Originator	FlexB10xOB6	0		2	Byte to Byte	None
Internal		4040		CNet Target	Connection 0	88		10	Byte to Byte	None

FIGURE 3.18. – INTERNAL MAPPING – IDS DATA TO CNET TARGET

The data from the IDS is copied to the input assembly *MappedData[88]* to *MappedData[97]*.



Name	Value	For	Style	Data Type
+ CNR01Input.MappedData[86]	0		Decimal	SINT
+ CNR01Input.MappedData[87]	0		Decimal	SINT
+ CNR01Input.MappedData[88]	0		Decimal	SINT
+ CNR01Input.MappedData[89]	0		Decimal	SINT
+ CNR01Input.MappedData[90]	0		Decimal	SINT
+ CNR01Input.MappedData[91]	0		Decimal	SINT
+ CNR01Input.MappedData[92]	0		Decimal	SINT
+ CNR01Input.MappedData[93]	0		Decimal	SINT
+ CNR01Input.MappedData[94]	0		Decimal	SINT
+ CNR01Input.MappedData[95]	0		Decimal	SINT
+ CNR01Input.MappedData[96]	0		Decimal	SINT
+ CNR01Input.MappedData[97]	0		Decimal	SINT
+ CNR01Input.MappedData[98]	0		Decimal	SINT

FIGURE 3.19. – INTERNAL MAPPING – IDS MAPPED DATA