

ControlNet Router

(Enhanced)

Example Pack 3

Technical Application Note

A-CNR

Document No. D138-020

Revision 1.0

03/2023

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 – EtherNet/IP	4
3.2. Slate.....	6
3.2.1. General.....	6
3.2.2. ControlNet Originator – Scheduled ControlNet	6
3.2.3. ControlNet Originator – Unscheduled ControlNet	8
3.2.4. EtherNet/IP Target.....	8
3.2.5. Internal Mapping	8



1. PREFACE

1.1. PURPOSE OF THIS DOCUMENT

This document will assist and provide information to the user with the ControlNet Router Example Pack 3.

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 3 application has the ControlNet Router operating as an ControlNet Originator and EtherNet/IP Target. This will allow newer EtherNet/IP only Logix Controllers to communicate with existing ControlNet IO.

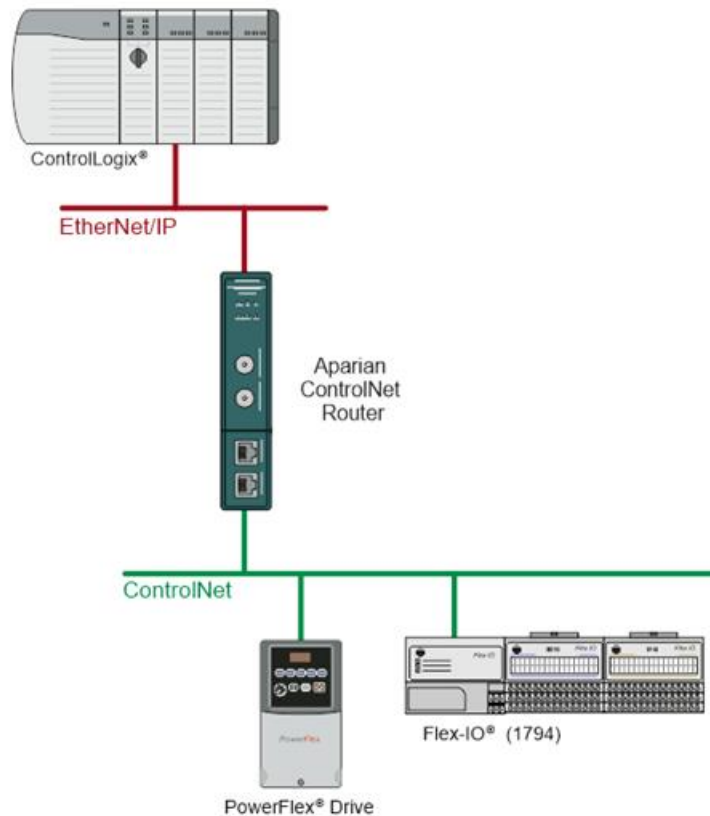


FIGURE 2.1. - EXAMPLE 3 TYPICAL NETWORK SETUP

When the ControlNet Router is configured as a ControlNet Originator and EtherNet/IP Target, a Logix Controller can own the ControlNet Router (CNR) via a Class 1 EtherNet/IP connection while the CNR module exchanges data with ControlNet IO using a scheduled ControlNet connection.

The data from the Logix controller can be exchanged with the ControlNet Router using the Class 1 EtherNet/IP connection input and output assemblies. This data, in turn, can be exchanged with the ControlNet IO devices using the input and output assemblies from the scheduled ControlNet connection originating from the CNR module.

3. SETUP

The following sections will provide information regarding the Example 3 setup and operation.

3.1. LOGIX – ETHERNET/IP

For the EtherNet/IP side, a Logix example project is used to connect the Logix controller to the CNR module using Class 1 EtherNet/IP.

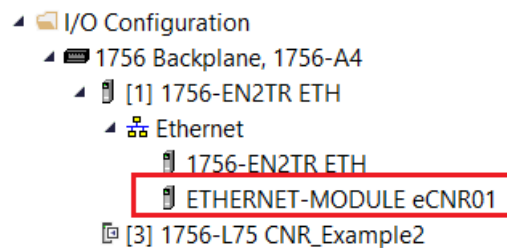


FIGURE 3.1. – CNR LOGIX ETHERNET IO TREE

The CNR01Map Routine is added which will have the COP instructions for the Ethernet CNR input and output UDTs.

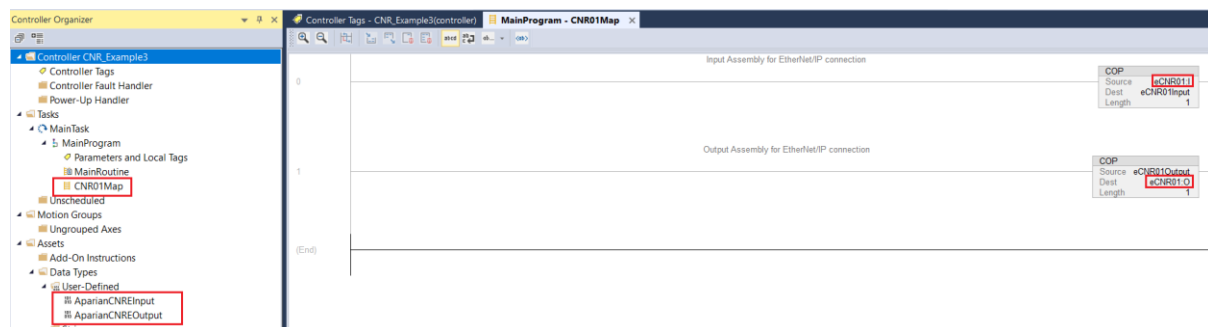


FIGURE 3.2. – CNR LOGIX MAPPING ROUTINE (ETHERNET)

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	Force	Style	Data Type
▲ eCNR01Input	{...}	{...}		AparianCNREInput
▸ eCNR01Input.GeneralStatus	97		Decimal	SINT
▸ eCNR01Input.CNetNodeNumber	4		Decimal	SINT
▸ eCNR01Input.CNetLedState	1		Decimal	SINT
▸ eCNR01Input.CNetMode	6		Decimal	SINT
eCNR01Input.Temperature	27.0		Float	REAL
▲ eCNR01Input.MappedData	{...}	{...}	Decimal	SINT[492]
▸ eCNR01Input.MappedData[0]	22		Decimal	SINT
▸ eCNR01Input.MappedData[1]	0		Decimal	SINT
▸ eCNR01Input.MappedData[2]	0		Decimal	SINT

FIGURE 3.3. – INPUT ASSEMBLY MAPPEDDATA (ETHERNET)

▲ eCNR01Output	{...}	{...}		AparianCNREOutput
▸ eCNR01Output.GeneralCtrl	0		Decimal	DINT
▲ eCNR01Output.MappedData	{...}	{...}	Decimal	SINT[492]
▸ eCNR01Output.MappedData[0]	33		Decimal	SINT
▸ eCNR01Output.MappedData[1]	0		Decimal	SINT
▸ eCNR01Output.MappedData[2]	0		Decimal	SINT
▸ eCNR01Output.MappedData[3]	0		Decimal	SINT

FIGURE 3.4. – OUTPUT ASSEMBLY MAPPEDDATA (ETHERNET)

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 the added ControlNet IO exchanging data with a Class 1 connection to a EtherNet/IP bridge (e.g., 1756-EN2TR).

3.2.1. GENERAL

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

The screenshot shows the 'CNR01 - Configuration' dialog box with the 'General' tab selected. The dialog has several tabs: General, ControlNet, ControlNet Devices, ControlNet Map, EtherNet/IP Devices, EtherNet/IP Map, Modbus, Modbus Auxiliary Map, Internal Map, and Advanced. The 'General' tab contains the following fields and controls:

- Instance Name:** A text box containing 'CNR01'.
- Description:** An empty text box.
- IP Address:** A field with four segments showing '192', '168', '1', and '168', followed by an ellipsis button and a 'Major Revision' dropdown set to '1'.
- Main Mode:** A dropdown menu set to 'Operational'.
- ControlNet Mode:** A dropdown menu set to 'Originator', highlighted with a red rectangle.
- Ethernet Mode:** A dropdown menu set to 'EtherNet/IP Target', highlighted with a red rectangle.
- PCCC / PLC5 Emulation:** A section containing an 'IP Address' field with segments '192', '168', '1', and '167'.

At the bottom of the dialog are four buttons: 'Ok', 'Apply', 'Cancel', and 'Help'.

FIGURE 3.5. – GENERAL CONFIGURATION

3.2.2. CONTROLNET ORIGINATOR – SCHEDULED CONTROLNET

The ControlNet node number must be set for the CNR as well as enabling the ControlNet Keeper functionality (as shown below):

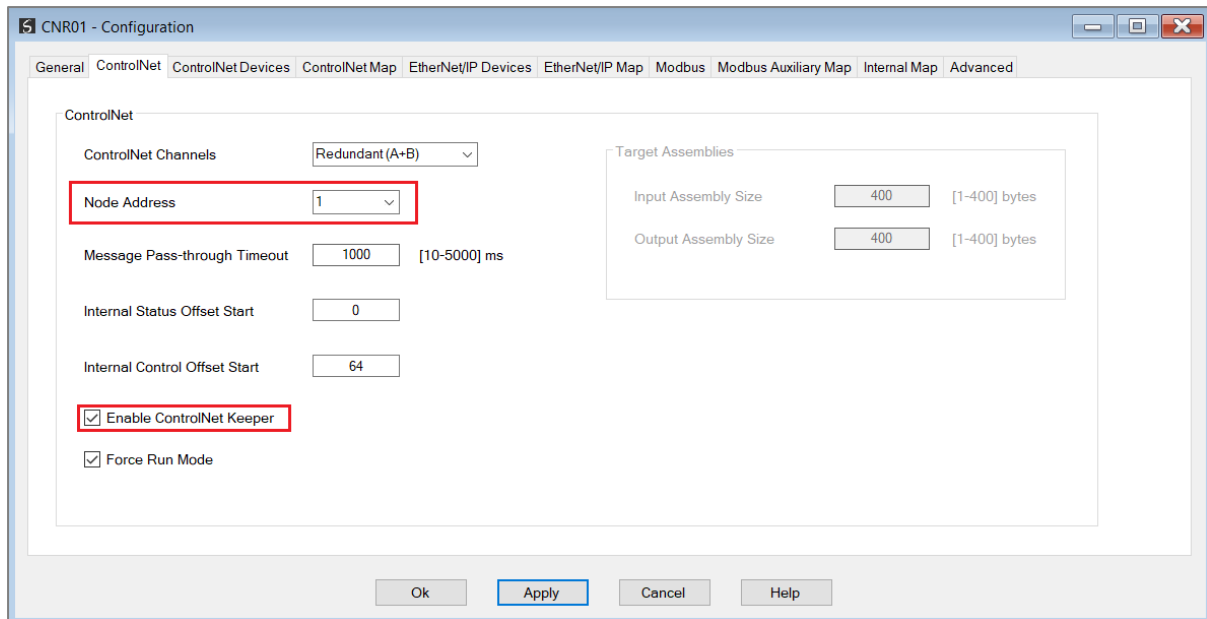


FIGURE 3.6. – CONTROLNET CONFIGURATION

To add ControlNet IO devices to the CNR ControlNet connections tree, see section 3.5.2.1 of the CNR user manual which will provide information regarding the ControlNet IO device setup.

Each ControlNet device will require a path, which could be directed to the target or routed via a ControlNet bridge (e.g., 1756-CNB or 1794-ACN15). For direct connections only the target ControlNet node address must be entered (as shown below).

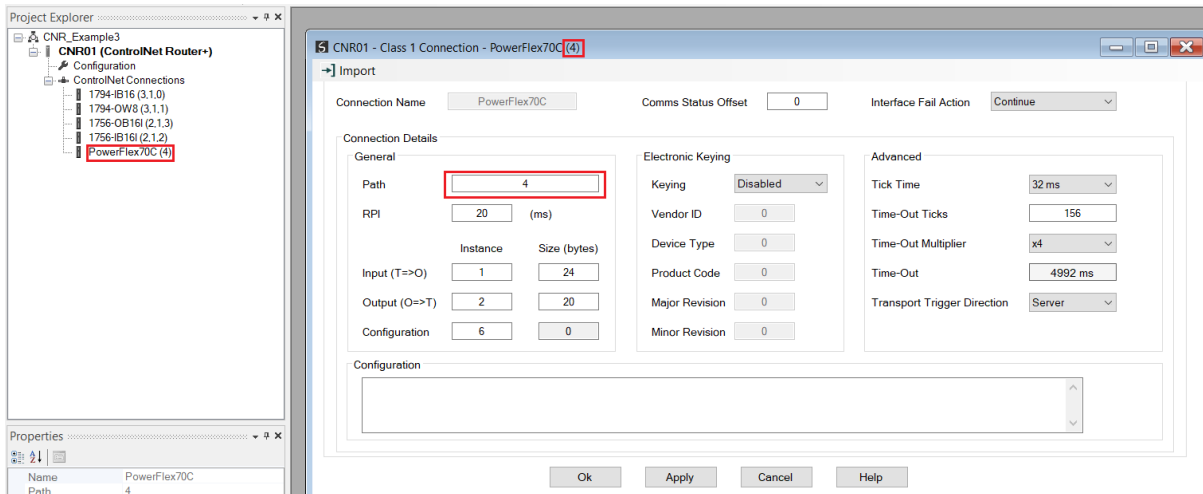


FIGURE 3.7. – SCHEDULED CONTROLNET DIRECT CONFIGURATION

For routed connections the target ControlNet node 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 ControlNet node 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).

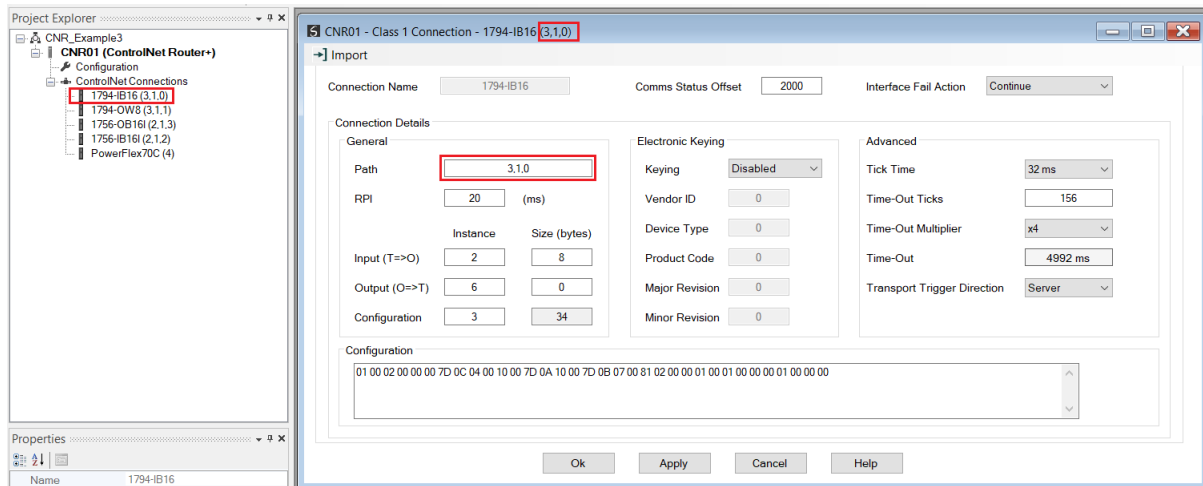


FIGURE 3.8. — SCHEDULED CONTROLNET ROUTED CONFIGURATION

Once the ControlNet IO devices have been added to the CNR ControlNet IO tree, then the ControlNet network must be scheduled and the ControlNet keeper must be updated. See section 3.5.2.3 of the CNR User Manual for information regarding the ControlNet Scheduling.

3.2.3. CONTROLNET ORIGINATOR — UNSCHEDULED CONTROLNET

Explicit Unscheduled ControlNet messaging can also be setup allowing the CNR module to communicate at a configured rate with ControlNet devices using unscheduled ControlNet connections. See section 3.5.2.2 of the CNR user manual for more information regarding the explicit messaging setup.

3.2.4. ETHERNET/IP TARGET

No Slate configuration is required for EtherNet/IP Target mode parameters.

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 ControlNet devices connected to the CNR (using scheduled ControlNet connections) is copied to the input assembly of the CNR EtherNet/IP Class 1 connection (i.e., the data being sent by the CNR on Ethernet). This will allow the Logix controller to access the data from the ControlNet 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]*.

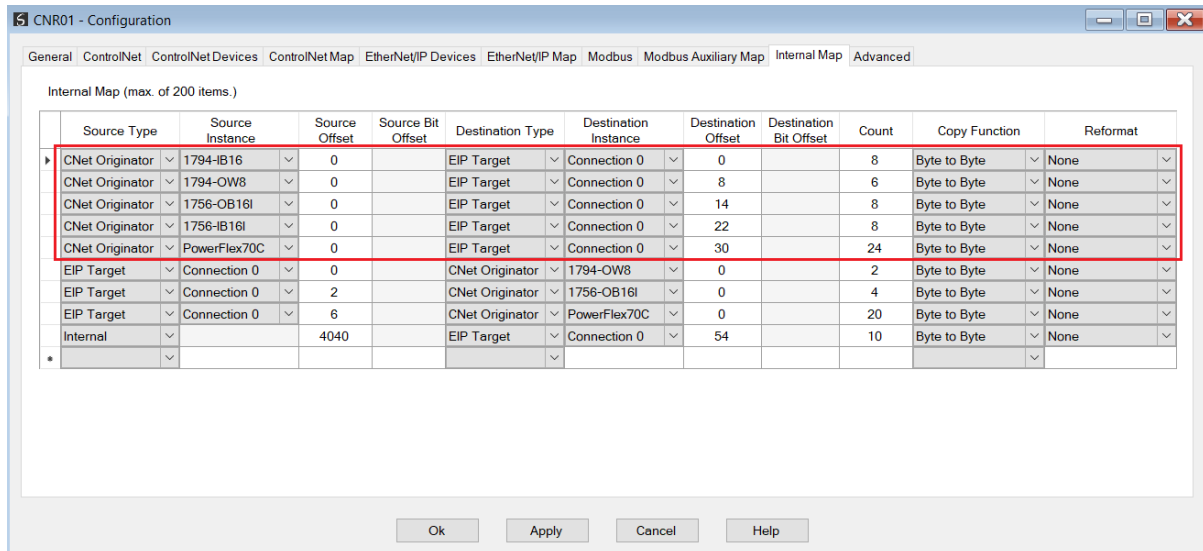


FIGURE 3.9. – INTERNAL MAPPING – CNET ORIGINATOR TO EIP TARGET

In the above example, the data received from EtherNet/IP device PowerFlex70C will be copied to the input assembly *MappedData[30]* to *MappedData[53]*, the data received from the 1794-OW8 will be copied to the input assembly *MappedData[8]* to *MappedData[13]*, and the data received from the 1756-IB16I will be copied to the input assembly *MappedData[22]* to *MappedData[29]* as shown below.

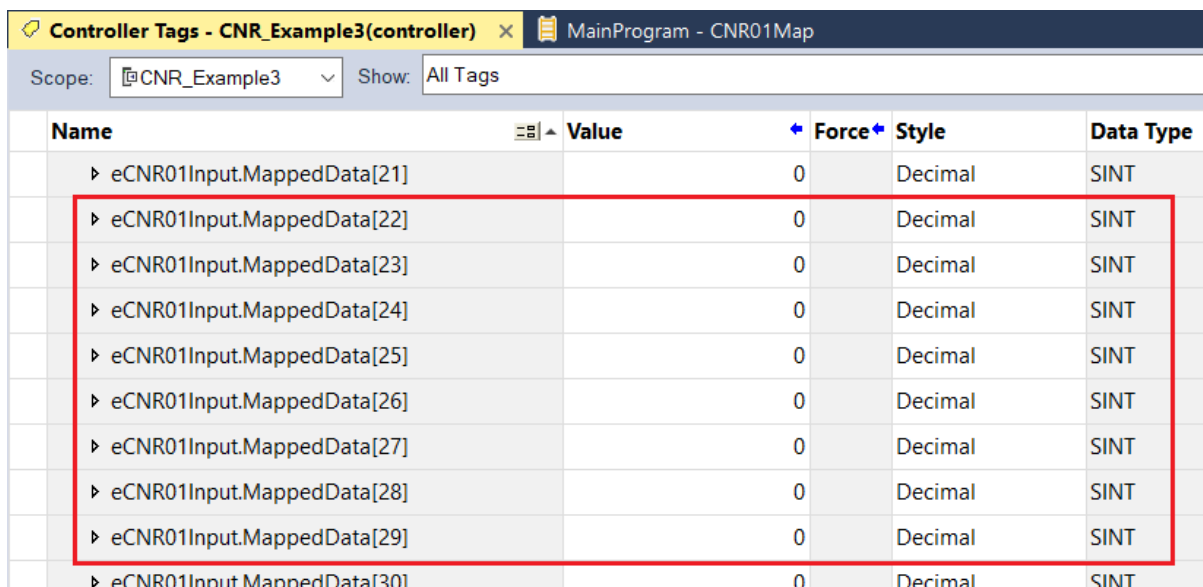


FIGURE 3.10. – INTERNAL MAPPING – 1756-IB16I MAPPED DATA

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

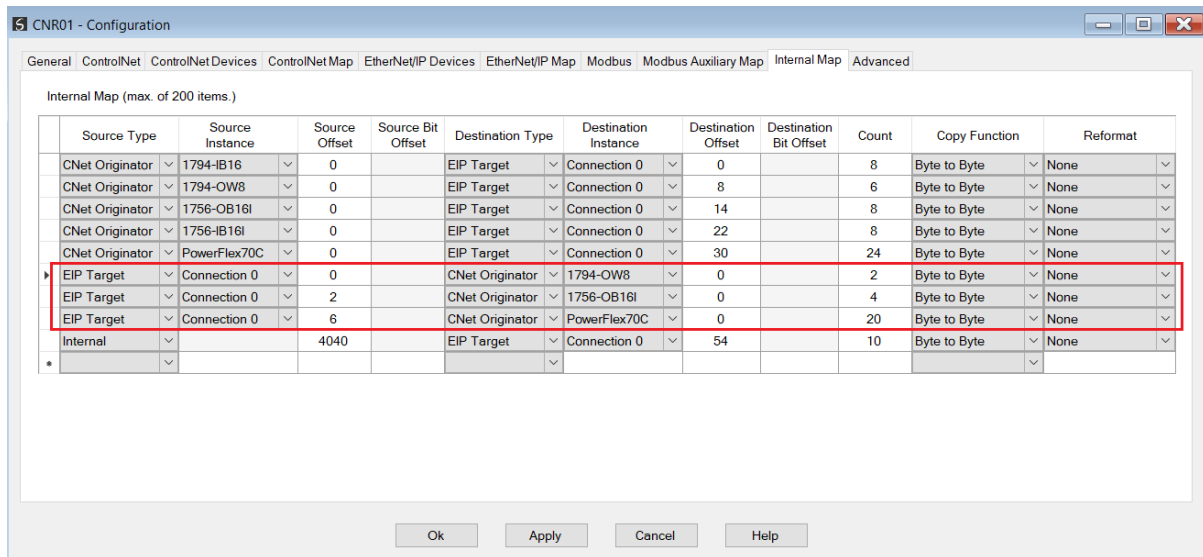


FIGURE 3.11. – INTERNAL MAPPING – EIP TARGET TO CNET ORIGINATOR

In the example, the data from the ControlNet output assembly *MappedData[0]* to *MappedData[1]* is sent to the 1794-OW8 ControlNet device (in red below), *MappedData[2]* to *MappedData[5]* is sent to the 1756-OB16I ControlNet device (in orange below), and *MappedData[6]* to *MappedData[25]* is sent to the PowerFlex70C ControlNet device (in green below), using the scheduled ControlNet connection from the CNR.

Name	Value	Force	Style	Data Type
▲ eCNR01Output.MappedData	{...}	{...}	Decimal	SINT[492]
▶ eCNR01Output.MappedData[0]	33		Decimal	SINT
▶ eCNR01Output.MappedData[1]	0		Decimal	SINT
▶ eCNR01Output.MappedData[2]	0		Decimal	SINT
▶ eCNR01Output.MappedData[3]	0		Decimal	SINT
▶ eCNR01Output.MappedData[4]	0		Decimal	SINT
▶ eCNR01Output.MappedData[5]	0		Decimal	SINT
▶ eCNR01Output.MappedData[6]	0		Decimal	SINT
▶ eCNR01Output.MappedData[7]	0		Decimal	SINT
▶ eCNR01Output.MappedData[8]	0		Decimal	SINT
▶ eCNR01Output.MappedData[9]	0		Decimal	SINT
▶ eCNR01Output.MappedData[10]	0		Decimal	SINT
▶ eCNR01Output.MappedData[11]	0		Decimal	SINT
▶ eCNR01Output.MappedData[12]	0		Decimal	SINT
▶ eCNR01Output.MappedData[13]	0		Decimal	SINT
▶ eCNR01Output.MappedData[14]	0		Decimal	SINT
▶ eCNR01Output.MappedData[15]	0		Decimal	SINT
▶ eCNR01Output.MappedData[16]	0		Decimal	SINT
▶ eCNR01Output.MappedData[17]	0		Decimal	SINT
▶ eCNR01Output.MappedData[18]	0		Decimal	SINT
▶ eCNR01Output.MappedData[19]	0		Decimal	SINT
▶ eCNR01Output.MappedData[20]	0		Decimal	SINT
▶ eCNR01Output.MappedData[21]	0		Decimal	SINT

FIGURE 3.12. – INTERNAL MAPPING – MAPPED DATA OUTPUT ASSEMBLY

Next the data received from the explicit unscheduled ControlNet messaging is copied to the CNR Class 1 EtherNet/IP input assembly. The data for the Get function for the PowerFlex70C is stored at address 4040 in the Internal Data Space (IDS).

CNR01 - Configuration													
General ControlNet ControlNet Devices ControlNet Map EtherNet/IP Devices EtherNet/IP Map Modbus Modbus Auxiliary Map Internal Map Advanced													
Explicit ControlNet Map (max. of 50 items.)													
	Device	Function	Scan	Service	Class	Instance	Attribute	Input Offset	Get Length	Output Offset	Set Length	Data Type	Static Value
▶	PowerFlex70C	Get	B	0x00	0x0001	1	7	4040	10	0	0		0
•													

FIGURE 3.13. – EXPLICIT UNSCHEDULED CONTROLNET MAP

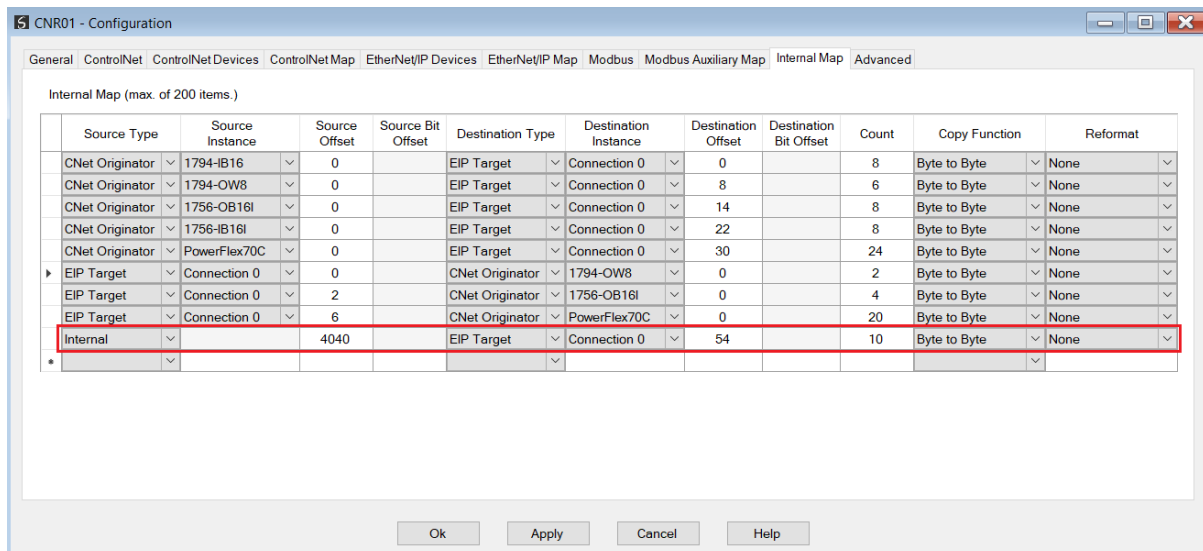


FIGURE 3.14. – INTERNAL MAPPING – IDS DATA TO EIP TARGET

The data from the IDS is copied to the input assembly *MappedData[54]* to *MappedData[63]*.

Name	Value	Force	Style	Data Type
▶ eCNR01Input.MappedData[53]		0	Decimal	SINT
▶ eCNR01Input.MappedData[54]		0	Decimal	SINT
▶ eCNR01Input.MappedData[55]		0	Decimal	SINT
▶ eCNR01Input.MappedData[56]		0	Decimal	SINT
▶ eCNR01Input.MappedData[57]		0	Decimal	SINT
▶ eCNR01Input.MappedData[58]		0	Decimal	SINT
▶ eCNR01Input.MappedData[59]		0	Decimal	SINT
▶ eCNR01Input.MappedData[60]		0	Decimal	SINT
▶ eCNR01Input.MappedData[61]		0	Decimal	SINT
▶ eCNR01Input.MappedData[62]		0	Decimal	SINT
▶ eCNR01Input.MappedData[63]		0	Decimal	SINT
▶ eCNR01Input.MappedData[64]		0	Decimal	SINT

FIGURE 3.15. – INTERNAL MAPPING – IDS MAPPED DATA