

ControlNet Router

(Enhanced)

Example Pack 2

Technical Application Note

A-CNR

Document No. D138-019

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 - ControlNet	4
3.2. Logix – EtherNet/IP	5
3.3. Slate.....	7
3.3.1. General.....	7
3.3.2. ControlNet Target	7
3.3.3. EtherNet/IP Target.....	8
3.3.4. 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 2.

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

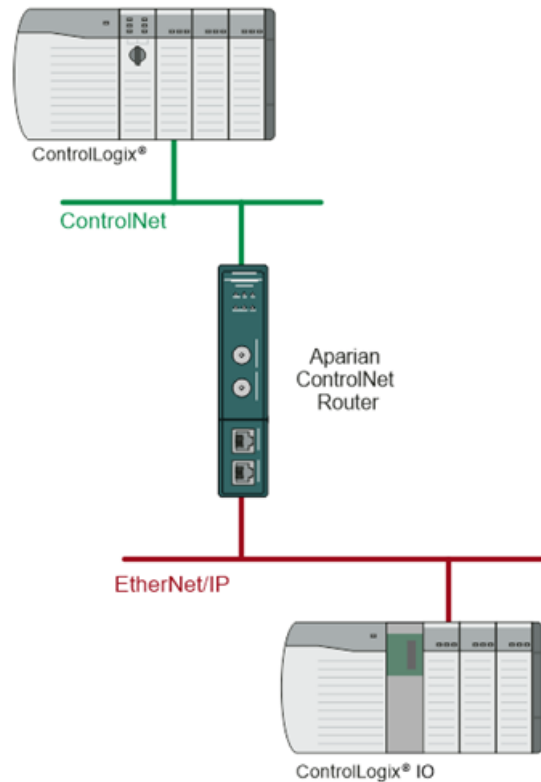


FIGURE 2.1. - EXAMPLE 2 TYPICAL NETWORK SETUP

When the ControlNet Router is configured as a ControlNet Target and EtherNet/IP Target, a controller (e.g. Logix Controller) can own the ControlNet Router using Scheduled ControlNet (e.g. via 1756-CNB/R) while another Logix Controller can own the ControlNet Router via a Class 1 EtherNet/IP 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 the input and output assemblies of the EtherNet/IP Class 1 connection from another Logix controller owning the CNR.

3. SETUP

The following sections will provide information regarding the Example 2 setup and operation.

3.1. LOGIX - CONTROLNET

For this example, there are two Logix projects (one for each interface). One Logix project is connecting to the CNR using Scheduled ControlNet (scheduled using RSNetworkx for ControlNet), while the other is connecting to the CNR using EtherNet/IP Class 1.

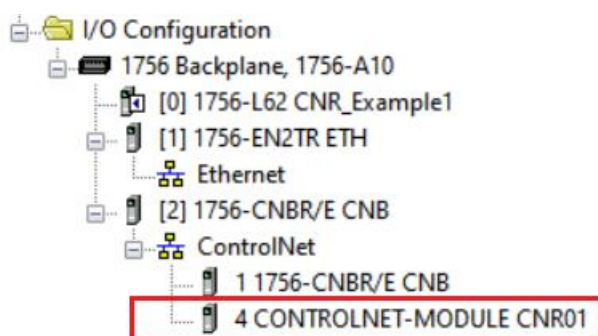


FIGURE 3.1. – CNR LOGIX CONTROLNET IO TREE

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

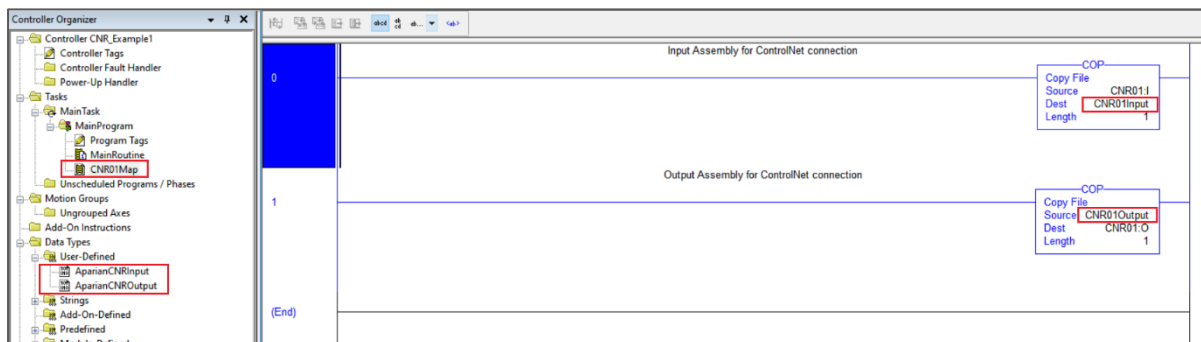


FIGURE 3.2. – CNR LOGIX MAPPING ROUTINE (CONTROLNET)

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 (CONTROLNET)

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 (CONTROLNET)

3.2. LOGIX – ETHERNET/IP

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

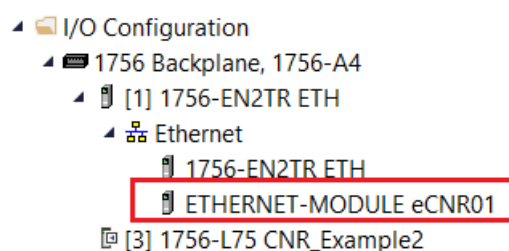


FIGURE 3.5. – CNR LOGIX ETHERNET IO TREE

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

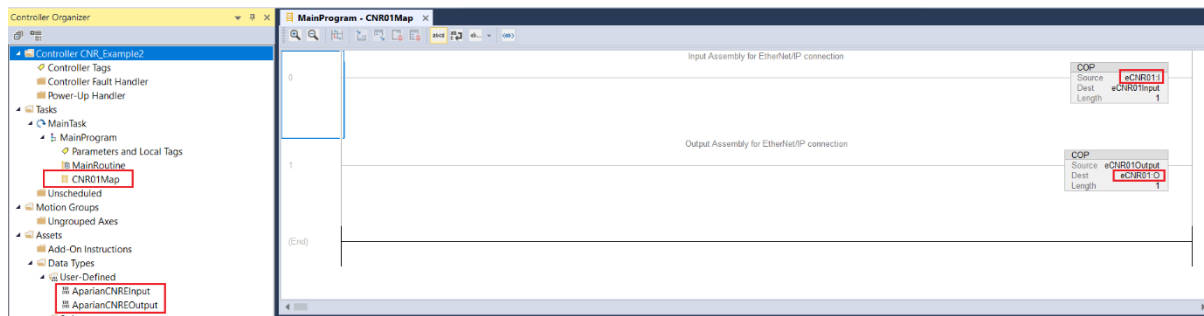


FIGURE 3.6. – CNR LOGIX MAPPING ROUTINE (ETHERNET)

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 3.7. – 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 3.8. – OUTPUT ASSEMBLY MAPPEDDATA (ETHERNET)

3.3. 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) exchange data with a Class 1 connection to a EtherNet/IP bridge (e.g., 1756-EN2TR).

3.3.1. GENERAL

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

The screenshot shows the 'CNR01 - Configuration' dialog box with the 'General' tab selected. The fields are as follows:

- Instance Name: CNR01
- Description: (empty)
- IP Address: 192 . 168 . 1 . 168
- Major Revision: 1
- Main Mode: Operational
- ControlNet Mode: Target (highlighted with a red box)
- Ethernet Mode: EtherNet/IP Target (highlighted with a red box)
- PCCC / PLC5 Emulation: (empty)

At the bottom, there are buttons for 'Ok', 'Apply', 'Cancel', and 'Help'.

FIGURE 3.9. — GENERAL CONFIGURATION

3.3.2. 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 RSNetworx for ControlNet setup with the CNR module.

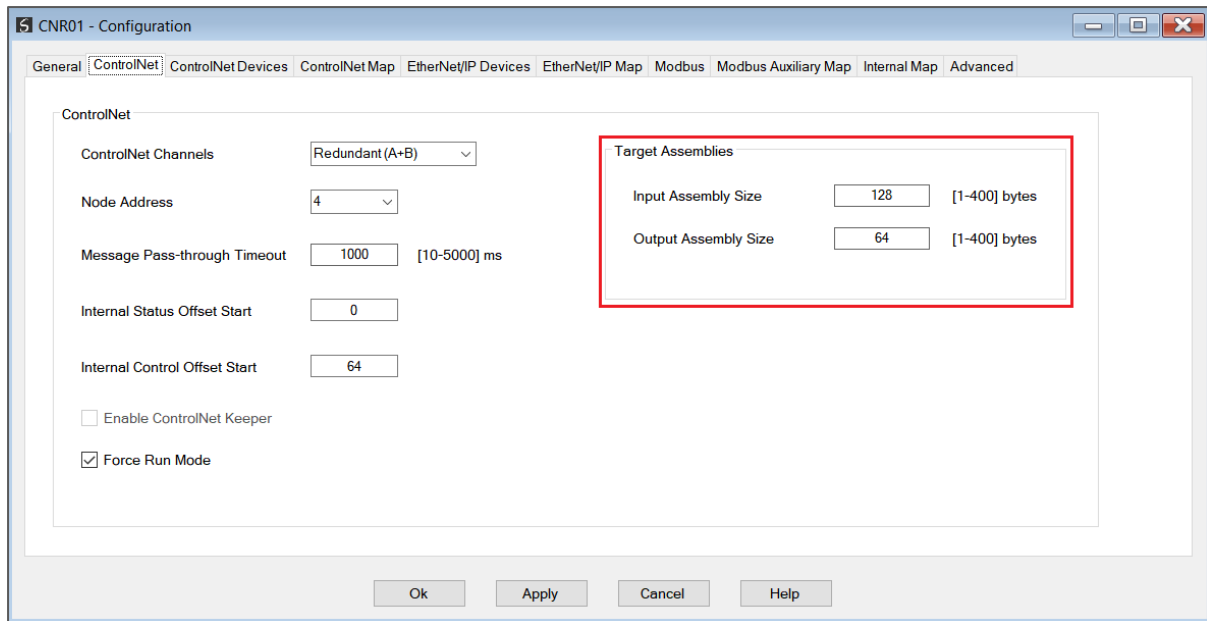


FIGURE 3.10. – CONTROLNET CONFIGURATION

3.3.3. ETHERNET/IP TARGET

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

3.3.4. 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 one Logix controller over Ethernet/IP using the Ethernet CNR output assembly and 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 ControlNet Logix controller to access the data from the Ethernet/IP Logix controller.



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]*.

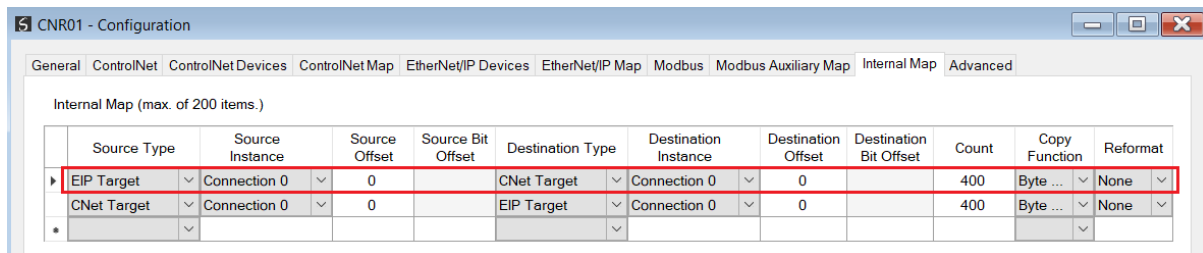


FIGURE 3.11. – INTERNAL MAPPING – EIP TARGET TO CNET TARGET

In the above example, the data received from the Ethernet Logix controller using the CNR Output Assembly UDT Tag Mapped Data (as shown below) will be sent to the ControlNet Logix controller using the CNR Input Assembly UDT Tag Mapped Data (as shown below).

Name	Value	Force	Style	Data Type
eCNR01Output		{...}	{...}	AparianCNREOutput
eCNR01Output.GeneralCtrl	0		Decimal	DINT
eCNR01Output.MappedData		{...}	{...}	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

FIGURE 3.12. – ETHERNET LOGIX CNR OUTPUT ASSEMBLY (COPY FROM)

Name	Value	Force	Style	Data Type
CNR01Input	{...}	{..}		AparianCNRInput
CNR01Input.GeneralStatus	97		Decimal	SINT
CNR01Input.CNetNodeNumber	4		Decimal	SINT
CNR01Input.CNetLedState	1		Decimal	SINT
CNR01Input.CNetMode	6		Decimal	SINT
CNR01Input.Temperature	28.287554		Float	REAL
CNR01Input.MappedData	{...}	{..}	Decimal	SINT[400]
CNR01Input.MappedData[0]	33		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
CNR01Input.MappedData[6]	0		Decimal	SINT
CNR01Input.MappedData[7]	0		Decimal	SINT
CNR01Input.MappedData[8]	0		Decimal	SINT
CNR01Input.MappedData[9]	0		Decimal	SINT
CNR01Input.MappedData[10]	0		Decimal	SINT

FIGURE 3.13. – CONTROLNET LOGIX CNR INPUT ASSEMBLY (COPY TO)

In the example, the data received from the ControlNet Logix controller using the CNR Output Assembly UDT Tag Mapped Data (as shown below) will be sent to the Ethernet Logix controller using the CNR Input Assembly UDT Tag Mapped Data (as shown below).

Name	Value	Force	Style	Data Type
- CNR01Output	{...}	{..}		AparianCNROutput
+ CNR01Output.GeneralCtrl	0		Decimal	DINT
- CNR01Output.MappedData	{...}	{..}	Decimal	SINT[400]
+ CNR01Output.MappedData[0]	22		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

FIGURE 3.14. – CONTROLNET LOGIX CNR OUTPUT ASSEMBLY (COPY FROM)

Name	Value	Force	Style	Data Type
▲ eCNR01Input	{...}	{...}		AparianCNRInput
▸ 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
▸ eCNR01Input.MappedData[3]	0		Decimal	SINT
▸ eCNR01Input.MappedData[4]	0		Decimal	SINT
▸ eCNR01Input.MappedData[5]	0		Decimal	SINT
▸ eCNR01Input.MappedData[6]	0		Decimal	SINT
▸ eCNR01Input.MappedData[7]	0		Decimal	SINT
▸ eCNR01Input.MappedData[8]	0		Decimal	SINT
▸ eCNR01Input.MappedData[9]	0		Decimal	SINT
▸ eCNR01Input.MappedData[10]	0		Decimal	SINT

FIGURE 3.15. – ETHERNET LOGIX CNR INPUT ASSEMBLY (COPY TO)