ProSoft
TECHNOLOGY

Where Automation Connects.

PLX51-DNPS
Micro800 DNP3 Outstation

TECHNICAL NOTE

CONTENTS

1. PREFACE

1.1. PURPOSE OF THIS DOCUMENT

This document will assist the user to setup a Micro850 Controller as a DNP3 Outstation RTU using ProSoft Technology's PLX51-DNPS module.

1.2. ADDITIONAL INFORMATION

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

Resource	Link
PLX50 Configuration Utility Installation	www.prosoft-technology.com
PLX51-DNPx User Manual PLX51-DNPx Datasheet Example Code & UDTs	www.prosoft-technology.com
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
DNP3	http://www.dnp.org

Table 1.1. - Additional Information

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:

Resource	Link
Contact Us web link	www.prosoft-technology.com
Support email	www.prosoft-technology.com

Table 1.2. – Support Details

2. APPLICATION DESCRIPTION

The ProSoft PLX51-DNPS can be used with a Micro800 Controller (M850) to operate as a DNP3 Outstation RTU. The diagrams below show examples of typical network configurations.
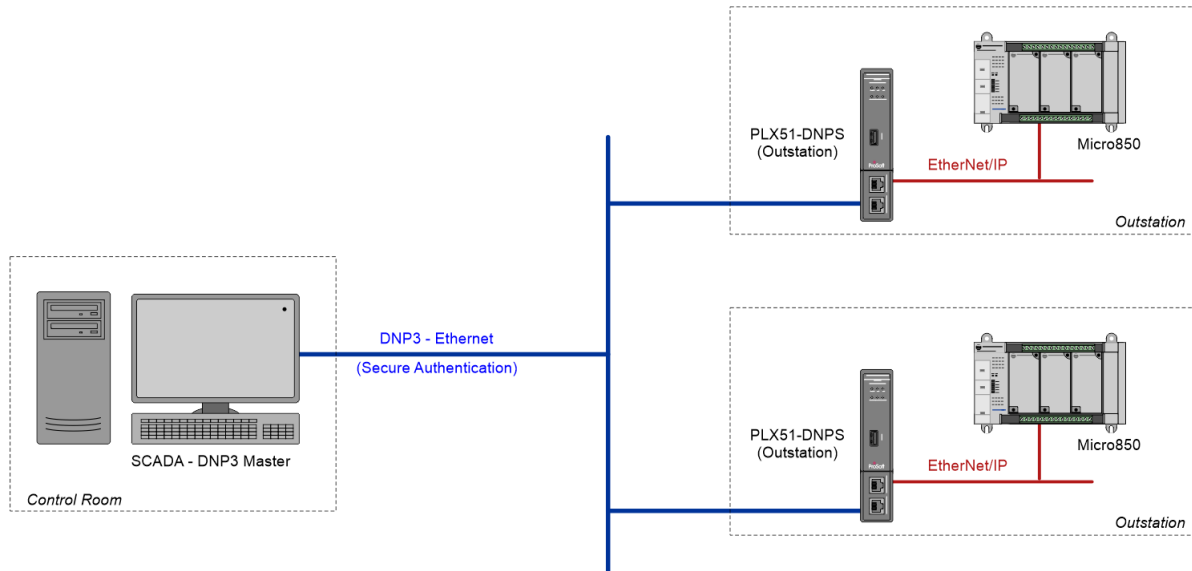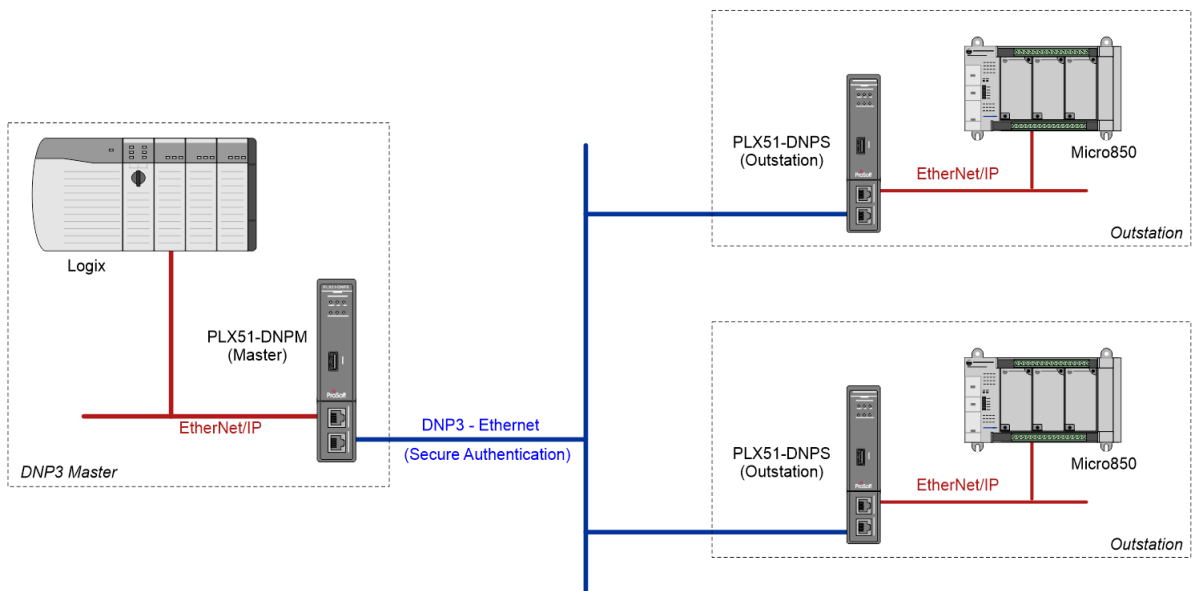

Figure 1 - Example 1 - typical network setup

Figure 2 - Example 2 - typical network setup

3.   SETUP – MICRO800

The following sections describe the configuration required for a Micro800 controller to operate as a DNP3 Outstation RTU.

An example project is available at www.prosoft-technology.com, which contains all the necessary DNP3 data type and mapping function blocks. Use this project as a starting point for any DNP3 Outstation application.

Using Rockwell Automation's Connected Components Workbench (CCW), import the aforementioned example project, by selecting the *Import Project* item under the *File* menu.

NOTE: Creating a DNP3 Outstation project requires **only** the CCW Standard Edition. Since the data structures and function blocks already exist in the example project, the CCW Developer Edition can be used, but is not required.

Figure 3 - CCW - Import Example Project
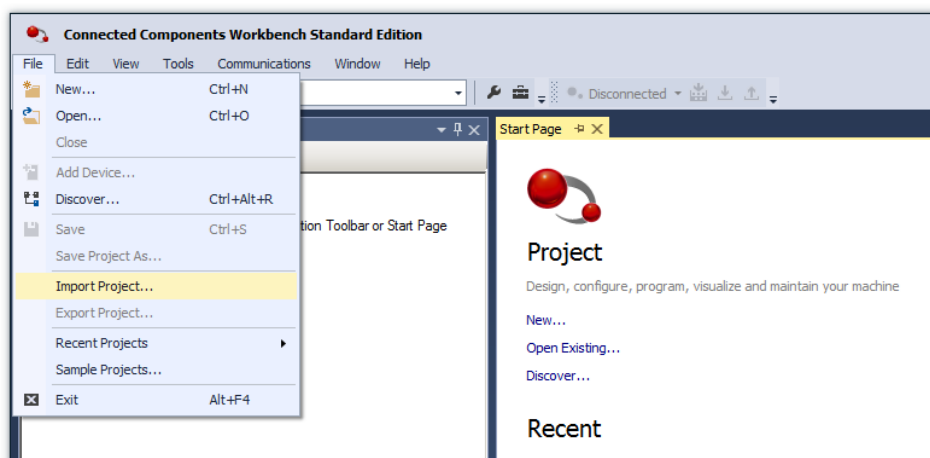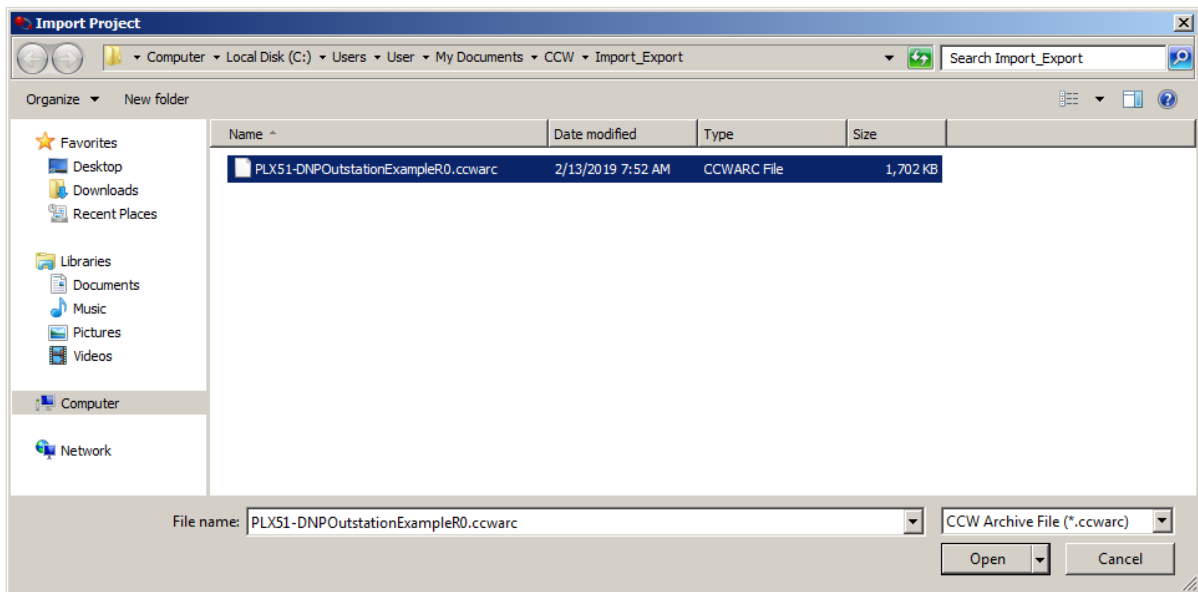
Figure 4 - Selecting Example Project

The imported project includes the following:

- DNP3 User-defined Data Structures
- DNP3 User-defined Function Blocks
- DNP3 Mapping Routine (Structured Text)
- DNP3 Statistics Routine (Ladder)
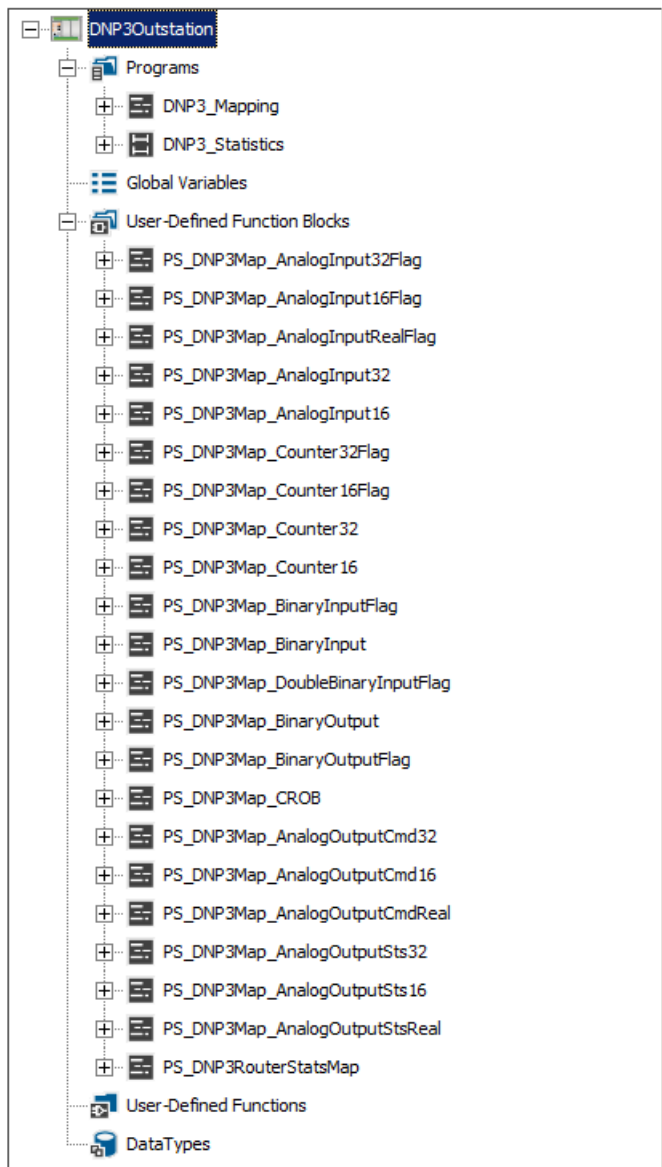- Global Variables (Example DNP3 Object instances and mapping block arrays)

Figure 5 - Project Explorer

Figure 6 - User-defined Data Structures

The following table contains the supported DNP3 Group Variations types.
Typically, only one Variation per Group is used in an application, and thus the unused variations' Function Blocks and associated Tags and Mapping code can be deleted.

The matching PLX50 Configuration Utility project (PLX51-DNPS configuration) includes only the recommended Variations of each group, but can be easily modified to suit.

Group	Group Description	Variation Code	Variation Description	Recommended
1	Binary Inputs	1	Packed Format	
		2	With Flags	Yes
3	Double Binary Inputs	2	With Flags	
10	Binary Outputs	1	Packed Format	
		2	Output Status With Flags	Yes
12	Binary Output Commands	1	Control Relay Output Block	Yes
20	Counters	1	32-bit With Flag	Yes
		2	16-bit With Flag	
		5	32-bit Without Flag	
		6	16-bit Without Flag	
30	Analog Inputs	1	32-bit With Flag	
		2	16-bit With Flag	
		3	32-bit Without Flag	
		4	16-bit Without Flag	
		5	Single Floating Point With Flag	Yes
40	Analog Output Status	1	32-bit With Flag	
		2	16-bit With Flag	
		3	Single Floating Point With Flag	Yes
41	Analog Outputs	1	32-bit	
		2	16-bit	
		3	Single Floating Point	Yes

Table 3 - Supported DNP3 Group-Variation Combinations

3.1. MAPPING RATIONALE

A tag, of the correct data type, must be created for each required DNP3 object instance (e.g. Analog Input point). Depending on the DNP3 object type (Group/Variation combination) this tag may be either an atomic (standard) type or a DNP3 specific structured data type. The correct choice of data type for each object is tabulated below.

Group	Group Description	Var. Code	Variation Description	DNP3 Object Instance Data Type	Objects per Block	Mapping Block Data Type
1	Binary Inputs	1	Packed Format	BOOL	8	SINT [0..1]
		2	With Flags	PSDNP3BinaryInputWithFlag	8	SINT [0..31]
3	Double Binary Inputs	2	With Flags	PSDNP3DoubleBinaryInputWithFlag	8	SINT [0..31]
10	Binary Outputs	1	Packed Format	BOOL	8	SINT [0..1]
		2	Output Status With Flags	PSDNP3BinaryOutputWithFlag	8	SINT [0..31]
12	Binary Output Commands	1	Control Relay Output Block	BOOL	1	SINT [0..19]
20	Counters	1	32-bit With Flag	PSDNP3Counter32WithFlag	8	SINT [0..63]
		2	16-bit With Flag	PSDNP3Counter16WithFlag	8	SINT [0..31]
		5	32-bit Without Flag	DINT	8	SINT [0..31]
		6	16-bit Without Flag	INT	8	SINT [0..15]
30	Analog Inputs	1	32-bit With Flag	PSDNP3AnalogInput32Flag	8	SINT [0..63]
		2	16-bit With Flag	PSDNP3AnalogInput16Flag	8	SINT [0..31]
		3	32-bit Without Flag	DINT	8	SINT [0..31]
		4	16-bit Without Flag	INT	8	SINT [0..15]
		5	Single Floating Point With Flag	PSDNP3AnalogInputRealFlag	8	SINT [0..63]
40	Analog Output Status	1	32-bit With Flag	PSDNP3AnalogOutputStatus32Flag	8	SINT [0..63]
		2	16-bit With Flag	PSDNP3AnalogOutputStatus16Flag	8	SINT [0..31]
		3	Single Floating Point With Flag	PSDNP3AnalogOutputStatusRealFlag	8	SINT [0..63]
41	Analog Outputs	1	32-bit	DINT	1	SINT [0..11]
		2	16-bit	INT	1	SINT [0..7]
		3	Single Floating Point	REAL	1	SINT [0..11]

TABLE 2 – REQUIRED DATA TYPES

NOTE: Individual DNP3 object instances are then mapped into mapping blocks, which are in turn read, or written to, from the PLX51-DNPS. The number of objects packed into a mapping block is listed in the table, but typically objects being produced by the outstation will be packed in blocks of 8, and objects consumed by the outstation will be mapped to a dedicated block.

Each mapped block will be represented by an item in the PLX51-DNPS mapping configuration.

The mapping program (***DNP3_Mapping***) makes use of structured text to map each DNP3 object to its corresponding mapping block using the matching Function Block.

The example code provides the mapping code of two instances for each of the supported Group-Variation combinations.

```
 1   (* BINARY INPUTS *)
 2
 3   (* Map Binary Inputs (Group 1 Variant 1 - Packed) *)
 4   PS_DNP3Map_BinaryInput_00(BlockIndex := 0, DNP3BinaryInput := MyBinaryInputV100, MapBlockIn := DNP3_BinaryInputBlockV1A, MapBloc
 5   PS_DNP3Map_BinaryInput_01(BlockIndex := 1, DNP3BinaryInput := MyBinaryInputV101, MapBlockIn := DNP3_BinaryInputBlockV1A, MapBloc
 6
 7   (* Map Binary Inputs (Group 1 Variant 2 - With Flag) *)
 8   PS_DNP3Map_BinaryInputFlag_00(BlockIndex := 0, DNP3BinaryInput := MyBinaryInputV200, MapBlockIn := DNP3_BinaryInputBlockV2A, MapI
 9   PS_DNP3Map_BinaryInputFlag_01(BlockIndex := 1, DNP3BinaryInput := MyBinaryInputV201, MapBlockIn := DNP3_BinaryInputBlockV2A, MapI
10
11   (* DOUBLE BINARY INPUTS *)
12
13   (* Map Double Binary Inputs (Group 3 Variant 2 - With Flag) *)
14   PS_DNP3Map_DoubleBinaryInputFlag_00(BlockIndex := 0, DNP3DoubleBinaryInput := MyDoubleBinaryInputV200, MapBlockIn := DNP3_DoubleI
15   PS_DNP3Map_DoubleBinaryInputFlag_01(BlockIndex := 1, DNP3DoubleBinaryInput := MyDoubleBinaryInputV201, MapBlockIn := DNP3_DoubleI
16
17   (* BINARY OUTPUTS STATUS *)
18
19   (* Map Binary Output Status (Group 10 Variant 1 - Packed) *)
20   PS_DNP3Map_BinaryOutput_00(BlockIndex := 0, DNP3BinaryOutput := MyBinaryOutputV100, MapBlockIn := DNP3_BinaryOutputBlockV1A, MapI
21   PS_DNP3Map_BinaryOutput_01(BlockIndex := 1, DNP3BinaryOutput := MyBinaryOutputV101, MapBlockIn := DNP3_BinaryOutputBlockV1A, MapI
22
23   (* Map Binary Output Status (Group 10 Variant 2 - With Flag) *)
24   PS_DNP3Map_BinaryOutputFlag_00(BlockIndex := 0, DNP3BinaryOutput := MyBinaryOutputV200, MapBlockIn := DNP3_BinaryOutputBlockV2A,
25   PS_DNP3Map_BinaryOutputFlag_01(BlockIndex := 1, DNP3BinaryOutput := MyBinaryOutputV201, MapBlockIn := DNP3_BinaryOutputBlockV2A,
```

Figure 7 - DNP3 Mapping Routine

3.2. ANALOG INPUT EXAMPLE

The diagram below illustrates the example mapping of two Analog Input object instances, **MyAnalogInputV500** and **MyAnalogInputV501**. These tags are both of type *PSDNP3AnalogInputRealFlag*.

The application code (beyond the scope of the example) is responsible for setting the correct values in the object, using the Micro800's physical inputs are other calculations.

The mapping routine then packs each instance into a mapping block tag (**DNP3_AnalogInputBlockV5A** of type *SINT[0..63]*). In this example, only two of the possible eight objects, are mapped into the block.

The mapping code (Structured Text) to map these two Analog Input examples is as follows:

```
(* Map Analog Inputs (Group 30 Variant 5 – REAL Value + Flag) *)
PS_DNP3Map_AnalogInputRealFlag_00(BlockIndex := 0, DNP3AnalogInput := MyAnalogInputV500,
MapBlockIn := DNP3_AnalogInputBlockV5A, MapBlockOut => DNP3_AnalogInputBlockV5A);

PS_DNP3Map_AnalogInputRealFlag_01(BlockIndex := 1, DNP3AnalogInput := MyAnalogInputV501,
MapBlockIn := DNP3_AnalogInputBlockV5A, MapBlockOut => DNP3_AnalogInputBlockV5A);
```

NOTE: The *BlockIndex* is 0 and 1 for the two objects, respectively. Since the Analog Input block size is 8, the *BlockIndex* would need to be in the range 0-7.

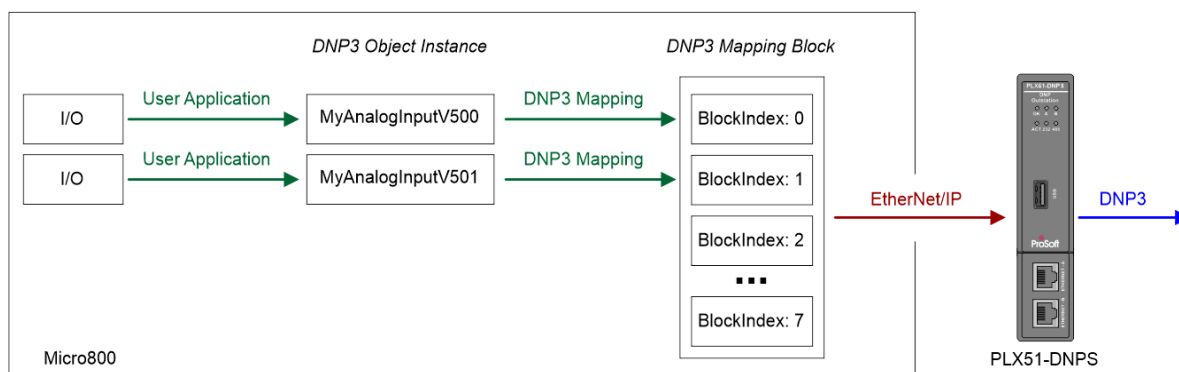The PLX51-DNPS then reads the entire mapping block and presents the DNP3 objects accordingly.

Figure 8 - Mapping Example - Analog Input

FIGURE 9 - ANALOG INPUT OBJECT

The PLX51-DNPS configuration would need to include the following mapped item.

Figure 10 – PLX51-DNPS Mapped Item

If more (than 8) Analog Input objects are required, then a second mapping block tag can be created, and so on. This would also require a second PLX51-DNPS mapped line item.

The following DNP3 objects (produced by the Micro800) would behave in a similar way:
- Binary Inputs
- Double Binary Inputs
- Binary Outputs (Status)
- Counters
- Analog Inputs (example above)
- Analog Outputs Status

NOTE: In order to save memory, it is important to delete all the tags, function blocks and mapping code associated with unused group-variation combinations.

3.3. BINARY OUTPUT COMMAND EXAMPLE

The diagram below illustrates the example mapping (and control) of a Binary Output Command (commonly referred as a CROB). The object instance is the **MyCROBV100** tag of type *BOOL*.
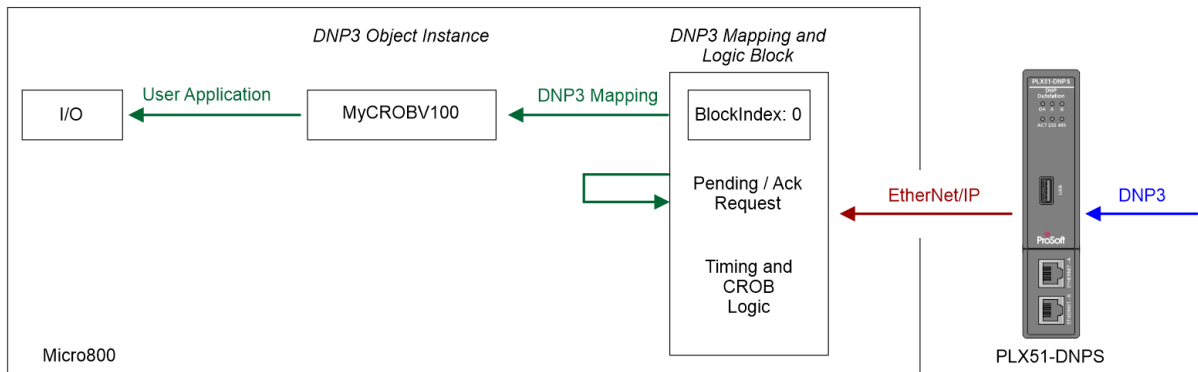

Figure 11 - Mapping and Control Example - CROB

When the PLX51-DNPS receives a valid command (Operate, Direct Operate etc.) from the DNP3 Master it will write to the corresponding mapping block tag DNP3_**CROBBlockV100** (of type SINT[0..19]).

The first bit of the first byte is the **RequestPending** flag, which will be cleared once the command has been processed. This ensures that the command is processed once, and only once.

The mapping function block not only extracts the relevant command information, such as command type, and On and Off timing information, but also executes the required logic.

The following CROB commands are processed by the function block:
- Pulse ON (Operation Type 1)
- Pulse OFF (Operation Type 2)
- Latch ON (Operation Type 3)
- Latch OFF (Operation Type 4)

For example, the DNP3 Master may send a single command to pulse the coil (**MyCROBV100)** *3 times, with an On-time of 700ms and an Off-time of 1000ms.*

The mapping and control code (Structured Text) for this CROB examples is as follows:

```
(* Process Binary Output Command (Group 12 Variant 1 - CROB) *)
PS_DNP3Map_CROB_00(MapBlockIn := DNP3_CROBBlockV100, DNP3CROBCoil => MyCROBV100);
IF(PS_DNP3Map_CROB_00.RequestPending) THEN
    DNP3_CROBBlockV100[0] := 0;
END_IF;
```

NOTE: There is no *BlockIndex* as only one CROB object is mapped per block.

The Analog Output Commands are processed using a similarly method, albeit without the complicated logic.

4. SETUP – PLX51-DNPS

The following sections describe the configuration of the PLX51-DNPS using the PLX50 Configuration Utility software.

NOTE: The example Slate project (***PLX51-DNP-Micro800-Example.psj***) is available on the ProSoft website.

NOTE: For specific details on how to configure the PLX51-DNPS this document should be read in conjunction with the **PLX51-DNPS User Manual**.

It is recommended the example PLX50 Configuration Utility project be used as a starting point, however a new project can also be used. In the case of the latter, a PLX51-DNPS should be added to the project.

In the PLX51-DNPS's General Configuration, ensure that the *Operating Interface* has been set to **EtherNet/IP Micro 800**.
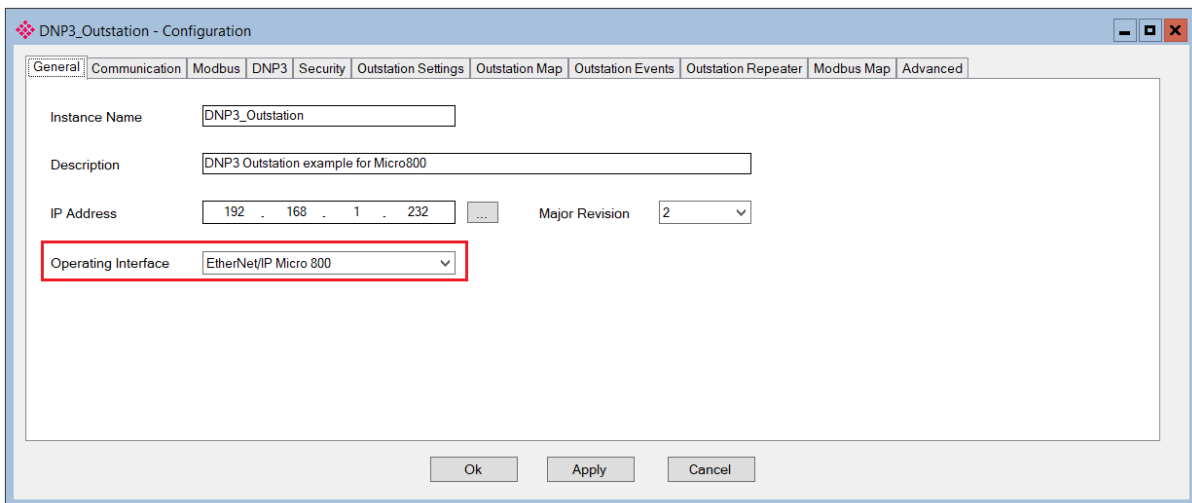

Figure 12 - General Configuration

In the Outstation Map tab, configure the appropriate mapped items. The example project provides the recommended choice of variants for each group, but this should be adjusted to suite the required application.

NOTE: When changing a variation, a Target tag of a matching structure must be selected.

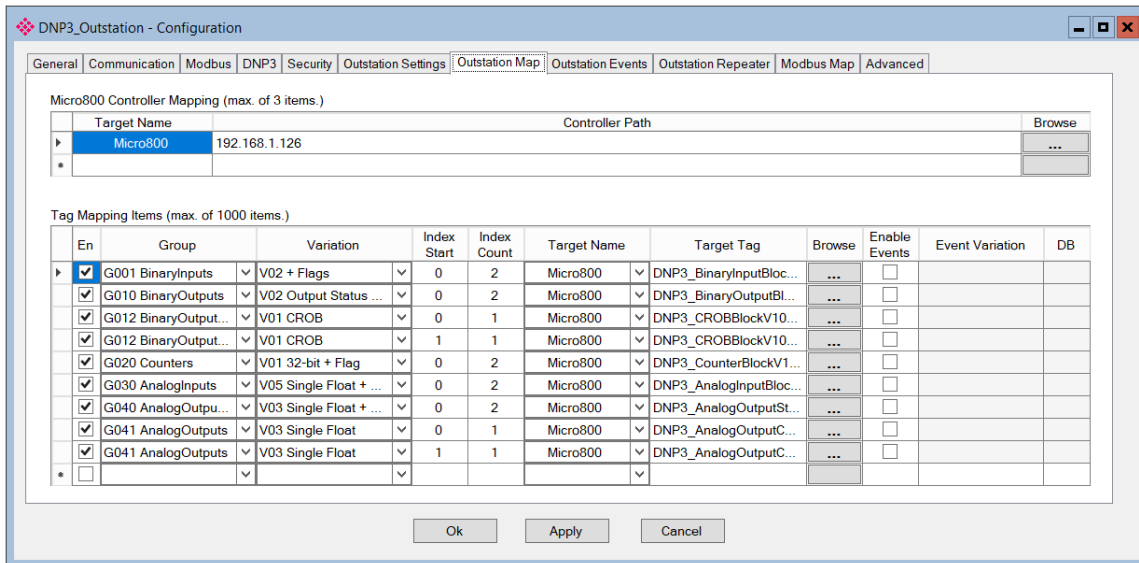Refer to Table 2 in the previous section.

Figure 13 - PLX51-DNPS Outstation Mapping

4.1. DNP3 DEVICE ATTRIBUTES

The PLX51-DNPS will respond to Device Attribute requests from the DNP3 Master which will provide the DNP3 Master with various identity and capability information. Certain attributes can be configured via the ProSoft Configuration Utility to provide a custom identity for a specific Micro800 Controller.

4.2. DNP3 EVENTS

The PLX51-DNPS provides the ability to log various DNP3 elements and commands. The DNP3 module will manage all the time stamping, and logging of the event data. The following parameters are required for DNP3 Events.

Parameter	Description
Enable Events	This parameter will enable or disable the DNP3 Event function for a specific set of DNP3 elements (configured in the same line item)
Event Variation	For each DNP3 Group there are various combinations of DNP3 Event variations that can selected. These event variations will be used to log the default event format for when default event requests are used.
Deadband (DB)	The deadband is used to determine when to log an event for Counters, Analog Inputs, and Analog Outputs Status. If the value changes by more than the deadband an event is logged with the specific DNP3 element's value.

Table 4 - Event logging parameters

NOTE: DNP3 elements will always be logged when any status bit changes (e.g. Online, Restart, etc.).

NOTE: When using Binary Groups (e.g. Binary Inputs, Double Binary Inputs, and Binary Output Status elements will be logged whenever there is a change in any value (status or process value).

NOTE: When using DNP3 Commands (e.g. Control Relay Output Block – CROB, or Analog Outputs) events will be logged when commands are received, rendering the deadband irrelevant.
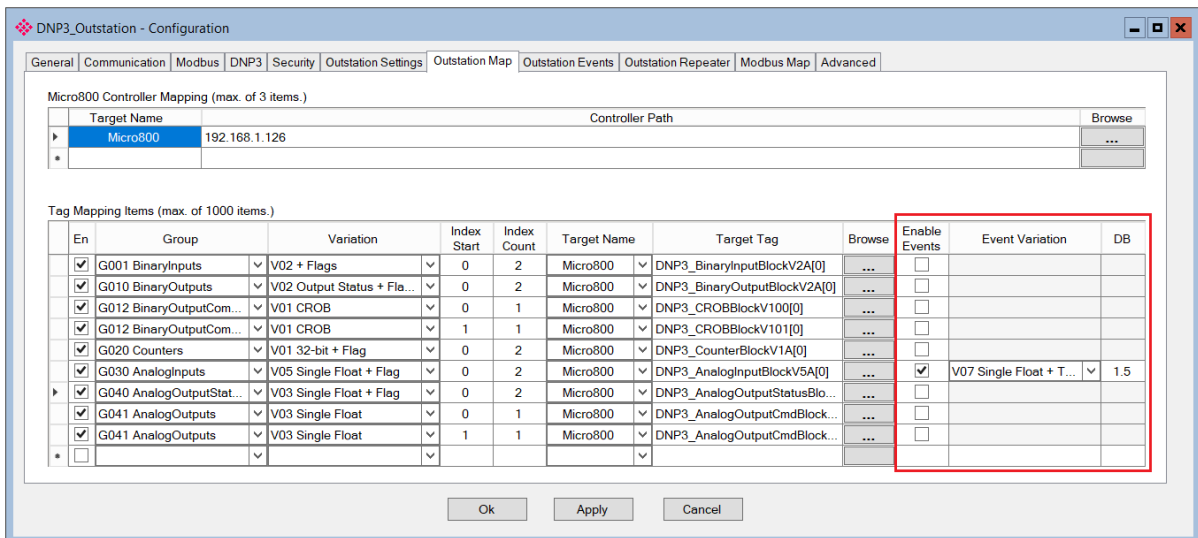

Figure 14 - DNP3 Event selection

Each mapped line item will be read every 10ms from the Micro800 controller. If a mapped line item is sufficiently big the reading of all the elements in the mapped line item will be spread over multiple Micro800 controller reads.

A maximum of 100,000 events can be logged per DNP3 Group.

4.2.1. EVENT VARIATION

For each DNP3 Group there are various combinations of DNP3 Event variations that can be selected. These event variations will be used to log the default event format for when default event requests are used.

	En	Group	Variation	Index Start	Index Count	Target Name	Target Tag	Browse	Enable Events	Event Variation	DB
	✓	G001 BinaryInputs	V02 + Flags	0	2	Micro800	DNP3_BinaryInputBlockV2A[0]	...	☐		
	✓	G010 BinaryOutputs	V02 Output Status + Fla...	0	2	Micro800	DNP3_BinaryOutputBlockV2A[0]	...	☐		
	✓	G012 BinaryOutputCom...	V01 CROB	0	1	Micro800	DNP3_CROBBlockV100[0]	...	☐		
	✓	G012 BinaryOutputCom...	V01 CROB	1	1	Micro800	DNP3_CROBBlockV101[0]	...	☐		
	✓	G020 Counters	V01 32-bit + Flag	0	2	Micro800	DNP3_CounterBlockV1A[0]	...	☐		
▶	✓	G030 AnalogInputs	V05 Single Float + Flag	0	2	Micro800	DNP3_AnalogInputBlockV5A[0]	...	✓	V07 Single Float + Time ⌄	1.5
	✓	G040 AnalogOutputStat...	V03 Single Float + Flag	0	2	Micro800	DNP3_AnalogOutputStatusBlo...	...	☐	V05 Single Float No Time	
	✓	G041 AnalogOutputs	V03 Single Float	0	1	Micro800	DNP3_AnalogOutputCmdBlock...	...	☐	V07 Single Float + Time	
	✓	G041 AnalogOutputs	V03 Single Float	1	1	Micro800	DNP3_AnalogOutputCmdBlock...	...	☐		
*	☐										

Figure 15 - DNP3 Event Variation selection

4.2.2. DEADBAND

The deadband is used to determine when to log an event for Counters, Analog Inputs, and Analog Outputs Status. If the value changes by more than the deadband an event is logged with the specific DNP3 element's value.

4.2.3. UNSOLICITED RESPONSES

The PLX51-DNPS supports Unsolicited Responses for DNP3 Events. The module will default to Unsolicited Responses disabled after power up. Unsolicited Responses can be enabled by the DNP3 Master.

5. MONITORING

The Micro800 example project contains a program "DNP3_Statistics" that continuously reads the DNP3 statistics from the PLX51-DNPS using CIP explicit messaging.

A 1 second timer is used to trigger the CIPGeneric read message which fetches the statistics from the PLX51-DNPS. The response is then mapped into the tag DNP3_RouterStatistics which is a user defined data structure.
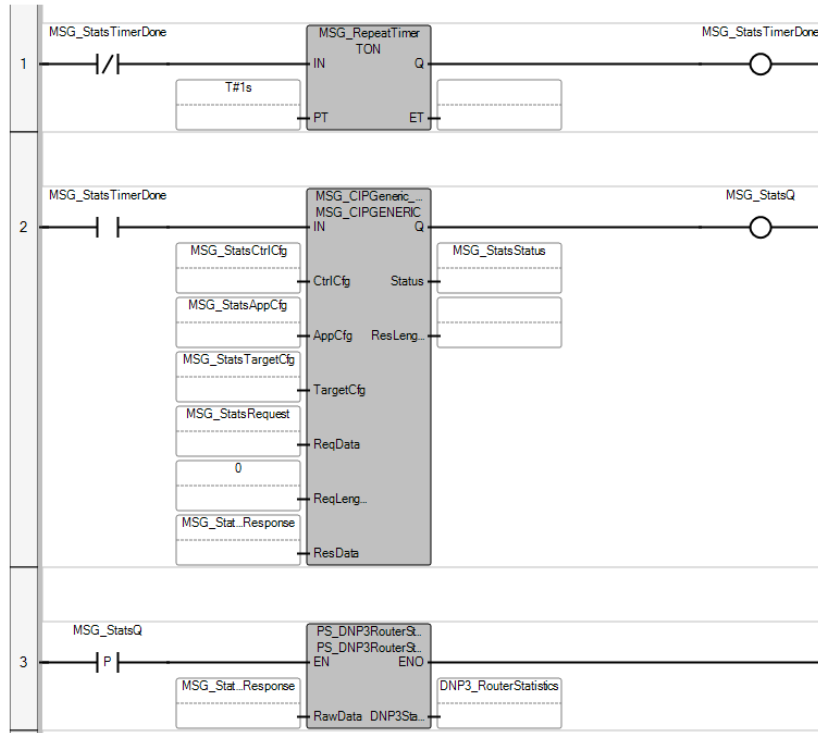

Figure 16 - DNP3 Statistics Program

5.1. CONFIGURATION

The CIP path contained within the local variable **MSG_StatsTargetCfg** must be set for the PLX51-DNPS.

Figure 17 - PLX51-DNPS CIP Path Configuration

Change the Initial Value of the .*Path* to **4,[IPAddress],** where:
The 4 indicates the Micro800's Ethernet port, and the [IPAddress] is the IP address of the PLX51-DNPS.

5.2. OPERATION

When program is running, the **DNP3_RouterStatistics** tag will be updated every second.

DNP3_RouterStatistics	...	PSDNP3RouterDNP3Statistics
DNP3_RouterStatistics.AppMsgTx	171764	UDINT
DNP3_RouterStatistics.AppMsgRx	166685	UDINT
DNP3_RouterStatistics.CriticalMsgTx	0	UDINT
DNP3_RouterStatistics.CriticalMsgRx	0	UDINT
DNP3_RouterStatistics.DiscardedMsg	0	UDINT
DNP3_RouterStatistics.ErrorMsgTx	0	UDINT
DNP3_RouterStatistics.ErrorMsgRx	0	UDINT
DNP3_RouterStatistics.ChecksumErrors	0	UDINT
DNP3_RouterStatistics.Timeouts	293	UDINT
DNP3_RouterStatistics.AssignClassReceived	0	UDINT
DNP3_RouterStatistics.TimeUpdateReceived	1	UDINT
DNP3_RouterStatistics.DeadbandUpdateRx	0	UDINT
DNP3_RouterStatistics.BroadcastMessageRx	0	UDINT
DNP3_RouterStatistics.UnsolicitedEventRx	80863	UDINT
DNP3_RouterStatistics.SuccessAuthentication	0	UDINT
DNP3_RouterStatistics.SessionKeyChanges	0	UDINT
DNP3_RouterStatistics.FailedSessionKeyChanges	0	UDINT
DNP3_RouterStatistics.UpdateKeyChanges	0	UDINT
DNP3_RouterStatistics.AuthenticationFailure	0	UDINT
DNP3_RouterStatistics.AuthorizationFailure	0	UDINT
DNP3_RouterStatistics.UnexpectedResponse	0	UDINT
DNP3_RouterStatistics.NoResponse	0	UDINT
o o o		
DNP3_RouterStatistics.NodeNumberMismatch	0	UDINT
DNP3_RouterStatistics.SelectSizeTooLarge	0	UDINT
DNP3_RouterStatistics.SelectOperateDataMismatch	0	UDINT
DNP3_RouterStatistics.ParityError	0	UDINT
DNP3_RouterStatistics.TransferKeyChanges	0	UDINT
DNP3_RouterStatistics.Online	✓	BOOL

Figure 18 - PLX51-DNPS Statistics Tag

The last item in the structure "**Online**", indicates that the CIP message to the PLX51-DNPS was successful. All the values in the structure are valid only if the **Online** is True.

6. SUPPORT, SERVICE & WARRANTY

6.1. CONTACTING TECHNICAL SUPPORT

ProSoft Technology, Inc. is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

1 Product Version Number
2 System architecture
3 Network details

If the issue is hardware related, we will also need information regarding:

1 Module configuration and associated ladder files, if any
2 Module operation and any unusual behavior
3 Configuration/Debug status information
4 LED patterns
5 Details about the interfaced serial, Ethernet or Fieldbus devices

Note: *For technical support calls within the United States, ProSoft's 24/7 after-hours phone support is available for urgent plant-down issues.*

North America (Corporate Location)	Europe / Middle East / Africa Regional Office
Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com	Phone: +33.(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com
Latin America Regional Office	**Asia Pacific Regional Office**
Phone: +52.222.264.1814 latinam@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com

For additional ProSoft Technology contacts in your area, please visit:

https://www.prosoft-technology.com/About-Us/Contact-Us.

6.2. WARRANTY INFORMATION

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS, please see the documents at: www.prosoft-technology/legal