# DNP3 Router

## Micro800 RTU Outstation

Technical Application Note

A-DNP3R

Document No. D109-014
Document Revision 1.3
05/2020

aparian

# CONTENTS

1. Preface ..... 3
1.1. Purpose of this Document ..... 3
1.2. Additional Information ..... 3
1.3. Support ..... 3
2. Application Description ..... 4
3. Setup – Micro800 ..... 5
3.1. Mapping Rationale ..... 9
3.2. Analog Input Example ..... 11
3.3. Binary Output Command Example ..... 13
4. Setup – DNP3 Router ..... 15
4.1. DNP3 Device Attributes ..... 16
4.2. DNP3 Events ..... 16
4.2.1. Event Variation ..... 18
4.2.2. Deadband ..... 18
4.2.3. Unsolicited Responses ..... 18
5. Monitoring ..... 19
5.1. Configuration ..... 19
5.2. Operation ..... 20

# 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 Aparian's DNP3 Router.

## 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 |
| DNP3 Router User Manual<br>DNP3 Router Datasheet<br>Example Code & UDTs | http://www.aparian.com/products/dnp3router |
| 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 |
| Slate Installation | http://www.aparian.com/software/slate |

## 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 | www.aparian.com/contact-us |
| Support email | support@aparian.com |

# 2. APPLICATION DESCRIPTION

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

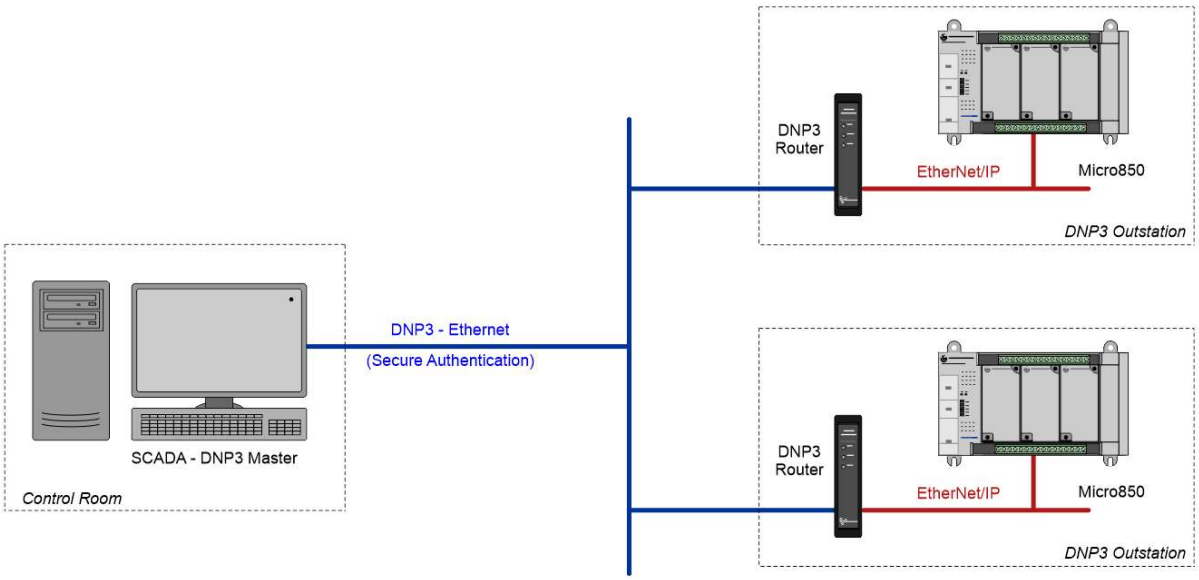



FIGURE 1 - EXAMPLE 1 - TYPICAL NETWORK SETUP

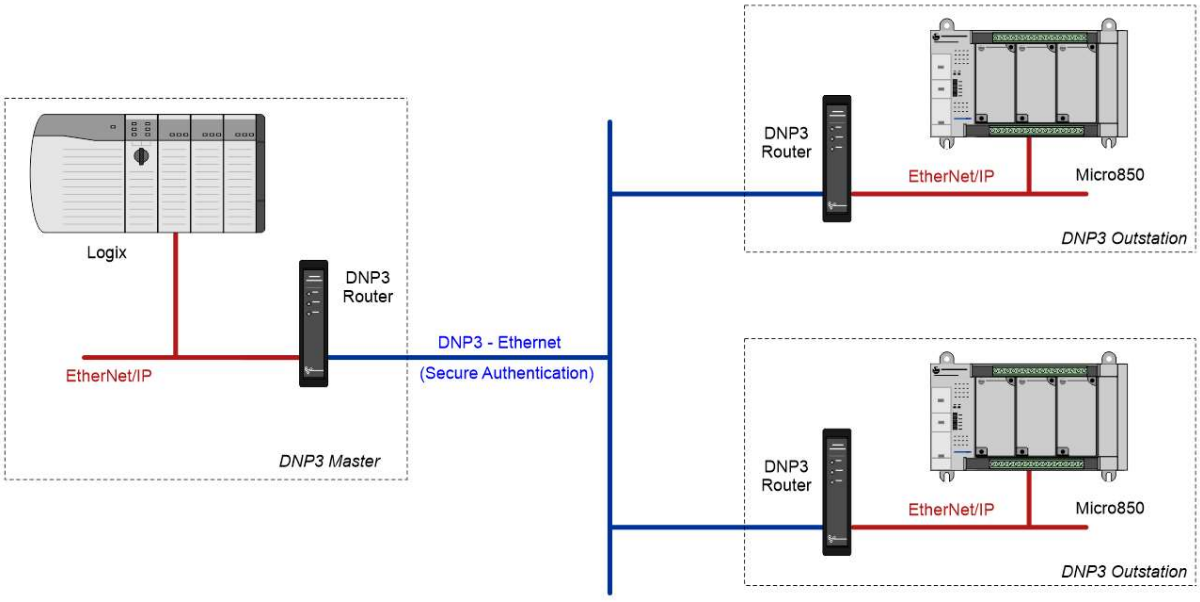



FIGURE 2 - EXAMPLE 2 - TYPICAL NETWORK SETUP

# 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 on the Aparian website which contains all the necessary DNP3 data type and mapping function blocks.

Importing this project (***DNP3OutstationExampleR2.ccwarc***) should be used 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.

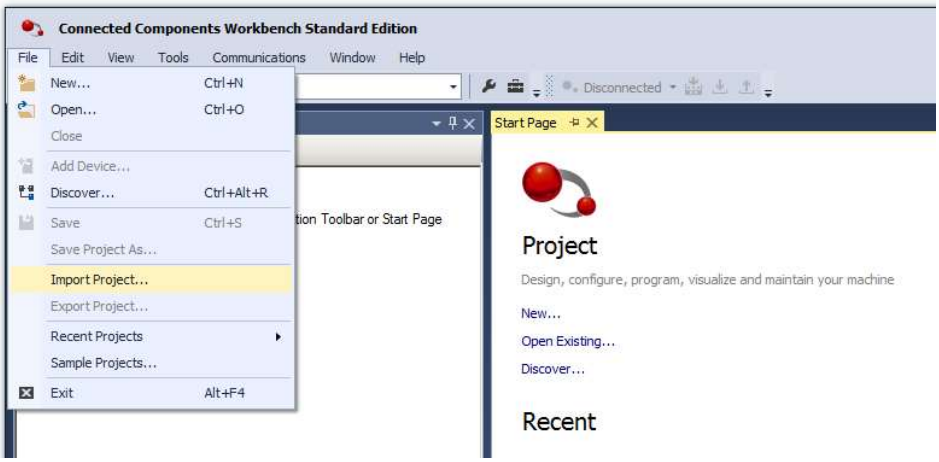



FIGURE 3 - CCW - IMPORT EXAMPLE PROJECT

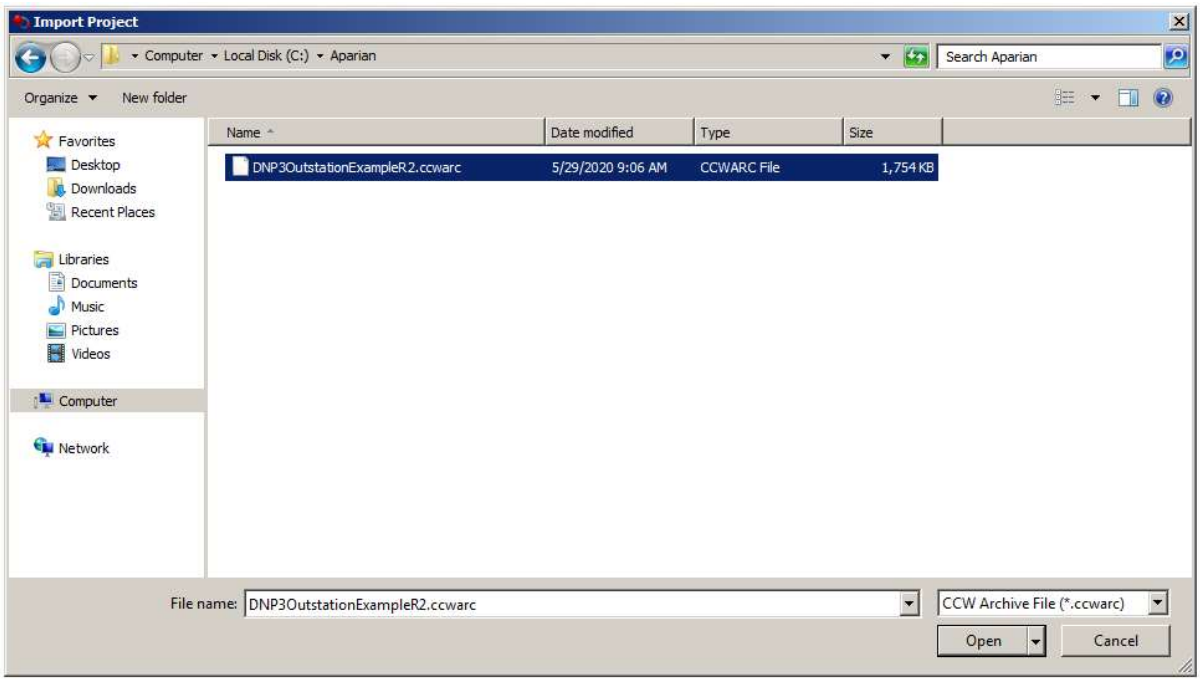



FIGURE 4 - SELECTING EXAMPLE PROJECT

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)

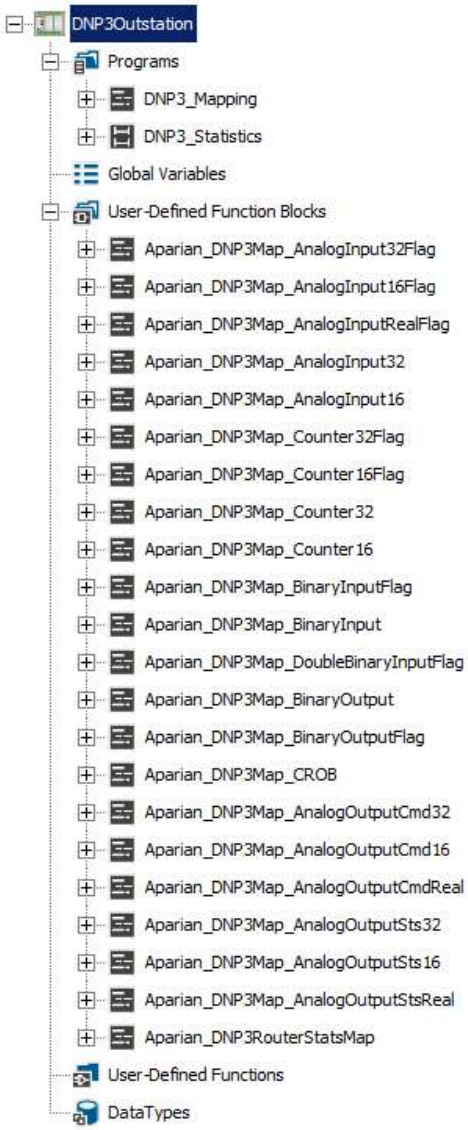



FIGURE 5 - PROJECT EXPLORER

Arrays | Structures | Defined Words

| Name | Data Type | String Size | Comment |
|---|---|---|---|
| Aparian | | | |
| AparianDNP3AnalogInput16Flag | AparianDNP3AnalogInput16Flag | | DNP3 Group 30 Variant 2 |
| AparianDNP3AnalogInput32Flag | AparianDNP3AnalogInput32Flag | | DNP3 Group 30 Variant 1 |
| AparianDNP3AnalogInputRealFlag | AparianDNP3AnalogInputRealFlag | | DNP3 Group 30 Variant 5 |
| AparianDNP3AnalogOutput16 | AparianDNP3AnalogOutput16 | | DNP3 Group 41 Variant 2 |
| AparianDNP3AnalogOutput32 | AparianDNP3AnalogOutput32 | | DNP3 Group 41 Variant 1 |
| AparianDNP3AnalogOutputReal | AparianDNP3AnalogOutputReal | | DNP3 Group 41 Variant 3 |
| AparianDNP3AnalogOutputStatus32Flag | AparianDNP3AnalogOutputStatus32Flag | | DNP3 Group 40 Variant 1 |
| AparianDNP3AnalogOutputStatusRealFlag | AparianDNP3AnalogOutputStatusRealFlag | | DNP3 Group 40 Variant 3 |
| AparianDNP3Counter16WithFlag | AparianDNP3Counter16WithFlag | | DNP3 Group 20 Variant 2 |
| AparianDNP3Counter32WithFlag | AparianDNP3Counter32WithFlag | | DNP3 Group 20 Variant 1 |
| AparianDNP3BinaryInputWithFlag | AparianDNP3BinaryInputWithFlag | | DNP3 Group 1 Variant 2 |
| AparianDNP3DoubleBinaryInputWithFlag | AparianDNP3DoubleBinaryInputWithFlag | | DNP3 Group 3 Variant 2 |
| AparianDNP3BinaryOutputWithFlag | AparianDNP3BinaryOutputWithFlag | | DNP3 Group 10 Variant 2 |
| AparianDNP3AnalogOutputStatus16Flag | AparianDNP3AnalogOutputStatus16Flag | | DNP3 Group 40 Variant 2 |
| AparianDNP3RouterDNP3Statistics | AparianDNP3RouterDNP3Statistics | | DNP3 Router DNP3 Statistics |

FIGURE 6 - USER-DEFINED DATA STRUCTURES

The table below shows 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 Slate project (DNP3 Router configuration) includes only the recommended Variations of each group, but this 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 1 - 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 | AparianDNP3BinaryInputWithFlag | 8 | SINT [0..31] |
| | | | | | | |
| 3 | Double Binary Inputs | 2 | With Flags | AparianDNP3DoubleBinaryInputWithFlag | 8 | SINT [0..31] |
| | | | | | | |
| 10 | Binary Outputs | 1 | Packed Format | BOOL | 8 | SINT [0..1] |
| | | 2 | Output Status With Flags | AparianDNP3BinaryOutputWithFlag | 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 | AparianDNP3Counter32WithFlag | 8 | SINT [0..63] |
| | | 2 | 16-bit With Flag | AparianDNP3Counter16WithFlag | 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 | AparianDNP3AnalogInput32Flag | 8 | SINT [0..63] |
| | | 2 | 16-bit With Flag | AparianDNP3AnalogInput16Flag | 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 | AparianDNP3AnalogInputRealFlag | 8 | SINT [0..63] |
| | | | | | | |
| 40 | Analog Output Status | 1 | 32-bit With Flag | AparianDNP3AnalogOutputStatus32Flag | 8 | SINT [0..63] |
| | | 2 | 16-bit With Flag | AparianDNP3AnalogOutputStatus16Flag | 8 | SINT [0..31] |
| | | 3 | Single Floating Point With Flag | AparianDNP3AnalogOutputStatusRealFlag | 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 DNP3 Router. 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 DNP3 Router 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.

```
(* BINARY INPUTS *)

(* Map Binary Inputs (Group 1 Variant 1 - Packed) *)
Aparian_DNP3Map_BinaryInput_00(BlockIndex := 0, DNP3BinaryInput := MyBinaryInputV100, MapBlc
Aparian_DNP3Map_BinaryInput_01(BlockIndex := 1, DNP3BinaryInput := MyBinaryInputV101, MapBlc

(* Map Binary Inputs (Group 1 Variant 2 - With Flag) *)
Aparian_DNP3Map_BinaryInputFlag_00(BlockIndex := 0, DNP3BinaryInput := MyBinaryInputV200, Ma
Aparian_DNP3Map_BinaryInputFlag_01(BlockIndex := 1, DNP3BinaryInput := MyBinaryInputV201, Ma

(* DOUBLE BINARY INPUTS *)

(* Map Double Binary Inputs (Group 3 Variant 2 - With Flag) *)
Aparian_DNP3Map_DoubleBinaryInputFlag_00(BlockIndex := 0, DNP3DoubleBinaryInput := MyDoubleB
Aparian_DNP3Map_DoubleBinaryInputFlag_01(BlockIndex := 1, DNP3DoubleBinaryInput := MyDoubleB

(* BINARY OUTPUTS STATUS *)

(* Map Binary Output Status (Group 10 Variant 1 - Packed) *)
Aparian_DNP3Map_BinaryOutput_00(BlockIndex := 0, DNP3BinaryOutput := MyBinaryOutputV100, Map
Aparian_DNP3Map_BinaryOutput_01(BlockIndex := 1, DNP3BinaryOutput := MyBinaryOutputV101, Map

(* Map Binary Output Status (Group 10 Variant 2 - With Flag) *)
Aparian_DNP3Map_BinaryOutputFlag_00(BlockIndex := 0, DNP3BinaryOutput := MyBinaryOutputV200,
Aparian_DNP3Map_BinaryOutputFlag_01(BlockIndex := 1, DNP3BinaryOutput := MyBinaryOutputV201,

```

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 *AparianDNP3AnalogInputRealFlag*.

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) *)
Aparian_DNP3Map_AnalogInputRealFlag_00(BlockIndex := 0, DNP3AnalogInput := MyAnalogInputV500,
MapBlockIn := DNP3_AnalogInputBlockV5A, MapBlockOut => DNP3_AnalogInputBlockV5A);

Aparian_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 DNP3 Router then reads the entire mapping block and presents the DNP3 objects accordingly.

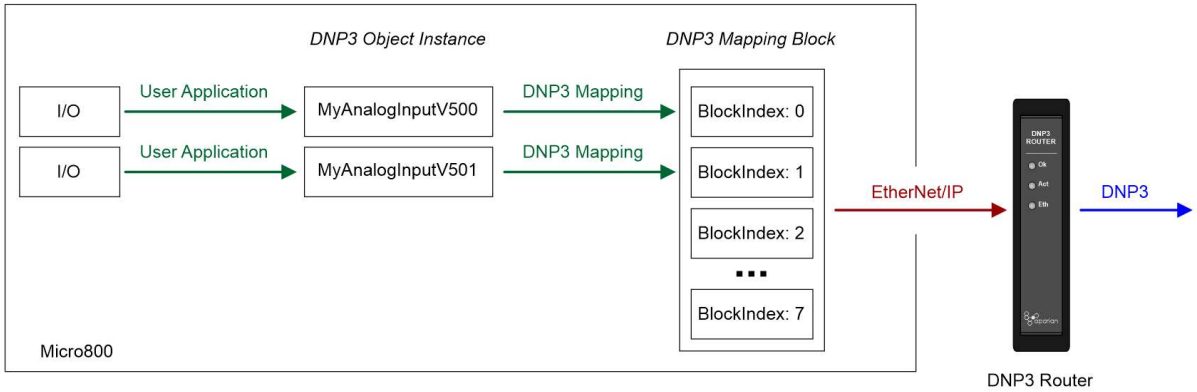



FIGURE 8 - MAPPING EXAMPLE - ANALOG INPUT

| MyAnalogInputV500 | ... | AparianDNP3AnalogInputRealFlag |
|---|---|---|
| MyAnalogInputV500.Online | ☑ | BOOL |
| MyAnalogInputV500.Restart | ☐ | BOOL |
| MyAnalogInputV500.CommLost | ☐ | BOOL |
| MyAnalogInputV500.RemoteForced | ☐ | BOOL |
| MyAnalogInputV500.LocalForced | ☐ | BOOL |
| MyAnalogInputV500.OverRange | ☐ | BOOL |
| MyAnalogInputV500.ReferenceError | ☐ | BOOL |
| MyAnalogInputV500.EventTrigger | ☐ | BOOL |
| MyAnalogInputV500.Value | 23.6 | REAL |

FIGURE 9 - ANALOG INPUT OBJECT

The DNP3 Router configuration would need to include the following mapped item.

| Group | Variation | Index Start | Index Count | Target Name | Target Tag |
|---|---|---|---|---|---|
| G030 AnalogInputs | V05 Single Float + Flag | 0 | 2 | Micro800 | DNP3_AnalogInputBlockV5A[0] |

FIGURE 10 – DNP3 ROUTER 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 DNP3 Router 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*.

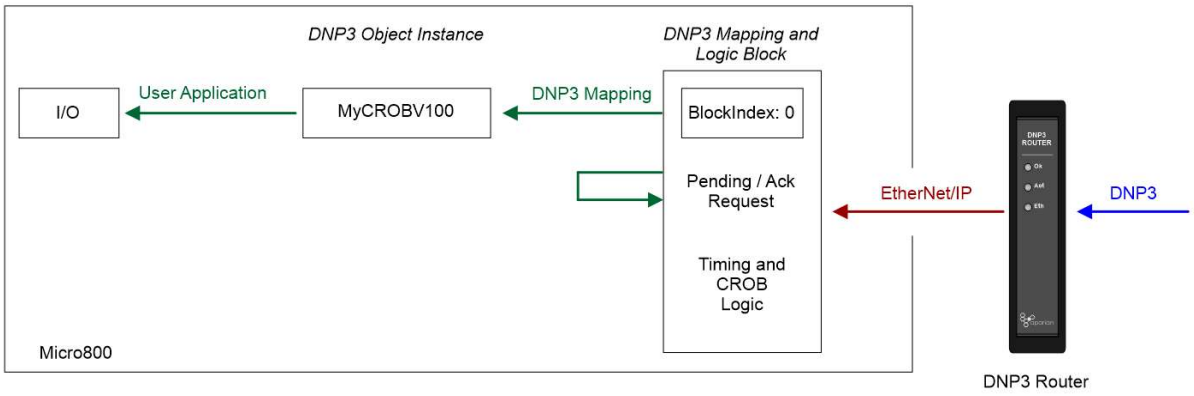



FIGURE 11 - MAPPING AND CONTROL EXAMPLE - CROB

When the DNP3 Router 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) *)
Aparian_DNP3Map_CROB_00(MapBlockIn := DNP3_CROBBlockV100, DNP3CROBCoil => MyCROBV100);
IF(Aparian_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 – DNP3 ROUTER

The following sections describe the configuration of the DNP3 Router using the Aparian Slate configuration software.

**NOTE:** The example Slate project (***DNP3_Micro800_Example.spj***) is available on the Aparian website.

**NOTE:** For specific details on how to configure the DNP3 Router this document should be read in conjunction with the **DNP3 Router User Manual**.

**NOTE:** The Micro800 functionality requires Slate version 1.030 or higher.

It is recommended the example Slate project be used as a starting point, however a new project can also be used. In the case of the latter, a DNP3 Router should be added to the project.

In the DNP3 Router's General Configuration, ensure that the Controller Type has been set to Micro800 and that the Reactive Mode has been selected.

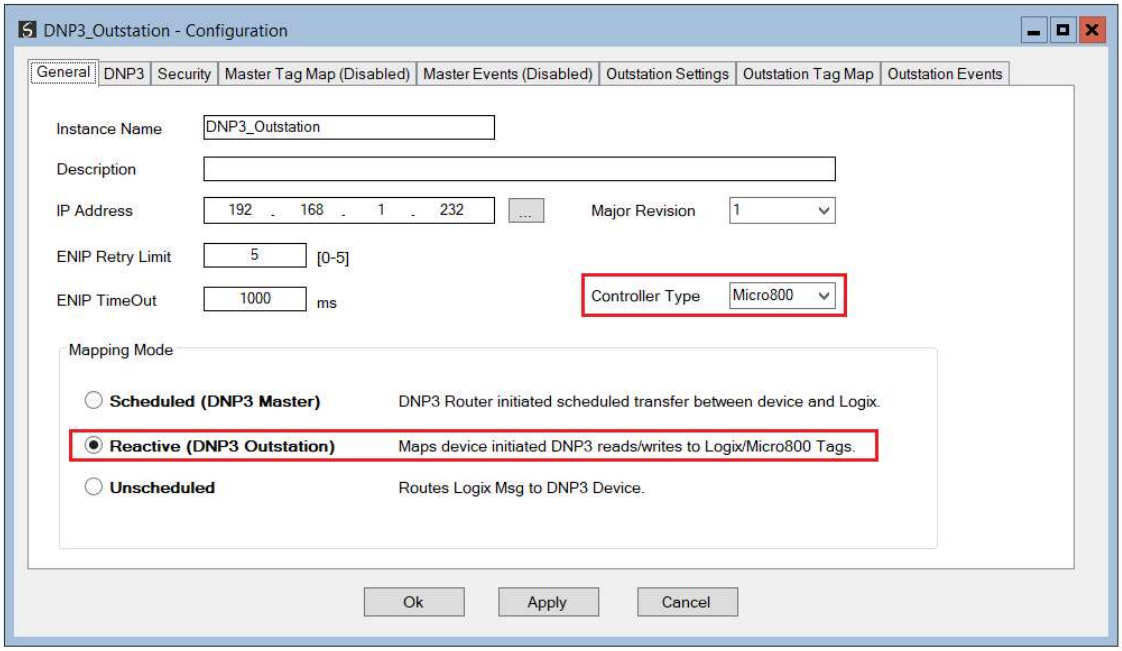



FIGURE 12 - GENERAL CONFIGURATION

In the Outstation Tag Mapping 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.

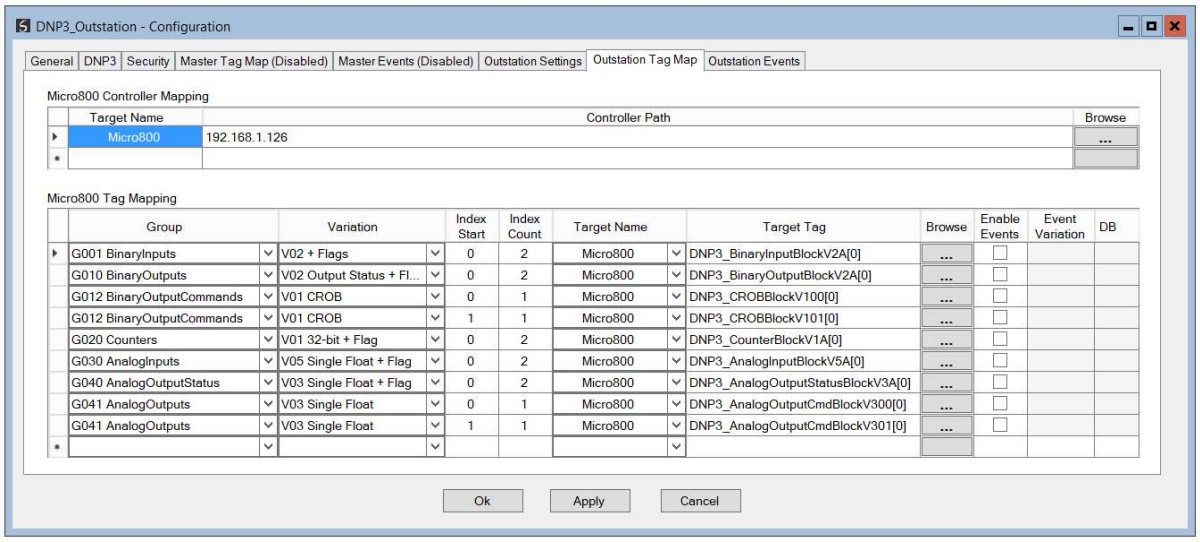



FIGURE 13 - DNP3 ROUTER OUTSTATION MAPPING

## 4.1. DNP3 DEVICE ATTRIBUTES

The DNP3 Router 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 Aparian's Slate application to provide a custom identity for a specific Micro800 Controller.

## 4.2. DNP3 EVENTS

The DNP3 Router 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 2 - 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.

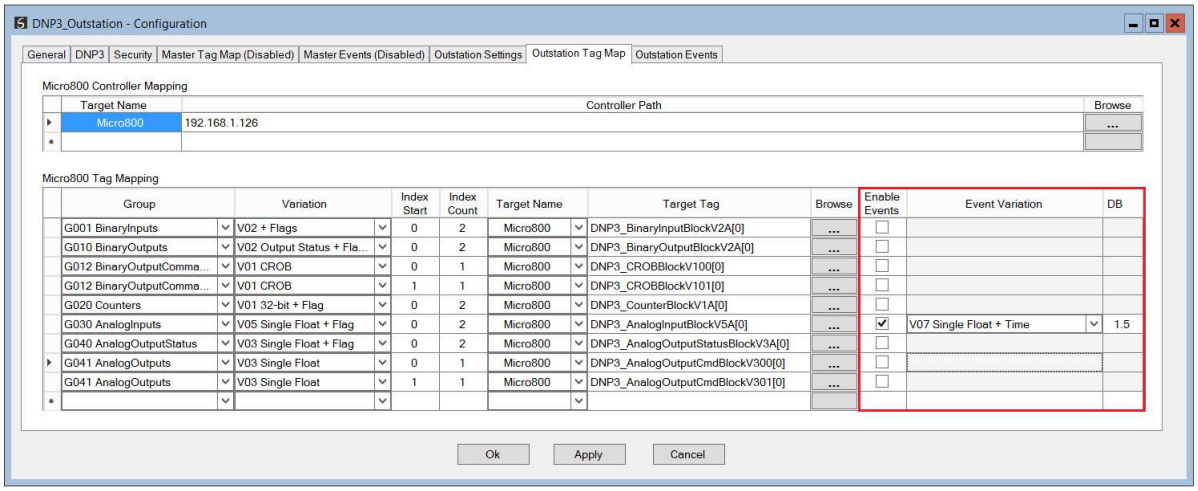



FIGURE 14 - DNP3 EVENT SELECTION

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.

Micro800 Tag Mapping

| | 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 BinaryOutputComma... | V01 CROB | 0 | 1 | Micro800 | DNP3_CROBBlockV100[0] | ... | ☐ | | |
| | G012 BinaryOutputComma... | 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 AnalogOutputStatus | V03 Single Float + Flag | 0 | 2 | Micro800 | DNP3_AnalogOutputStatusBlockV3A[0] | ... | ☐ | V05 Single Float No Time | |
| | G041 AnalogOutputs | V03 Single Float | 0 | 1 | Micro800 | DNP3_AnalogOutputCmdBlockV300[0] | ... | ☐ | V07 Single Float + Time | |
| | G041 AnalogOutputs | V03 Single Float | 1 | 1 | Micro800 | DNP3_AnalogOutputCmdBlockV301[0] | ... | ☐ | | |
| * | | | | | | | | | | |

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 DNP3 Router 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 DNP3 Router using CIP explicit messaging.

A 1 second timer is used to trigger the CIPGeneric read message which fetches the statistics from the DNP3 Router. The response is then mapped into the tag DNP3_RouterStatistics which is a user defined data structure.

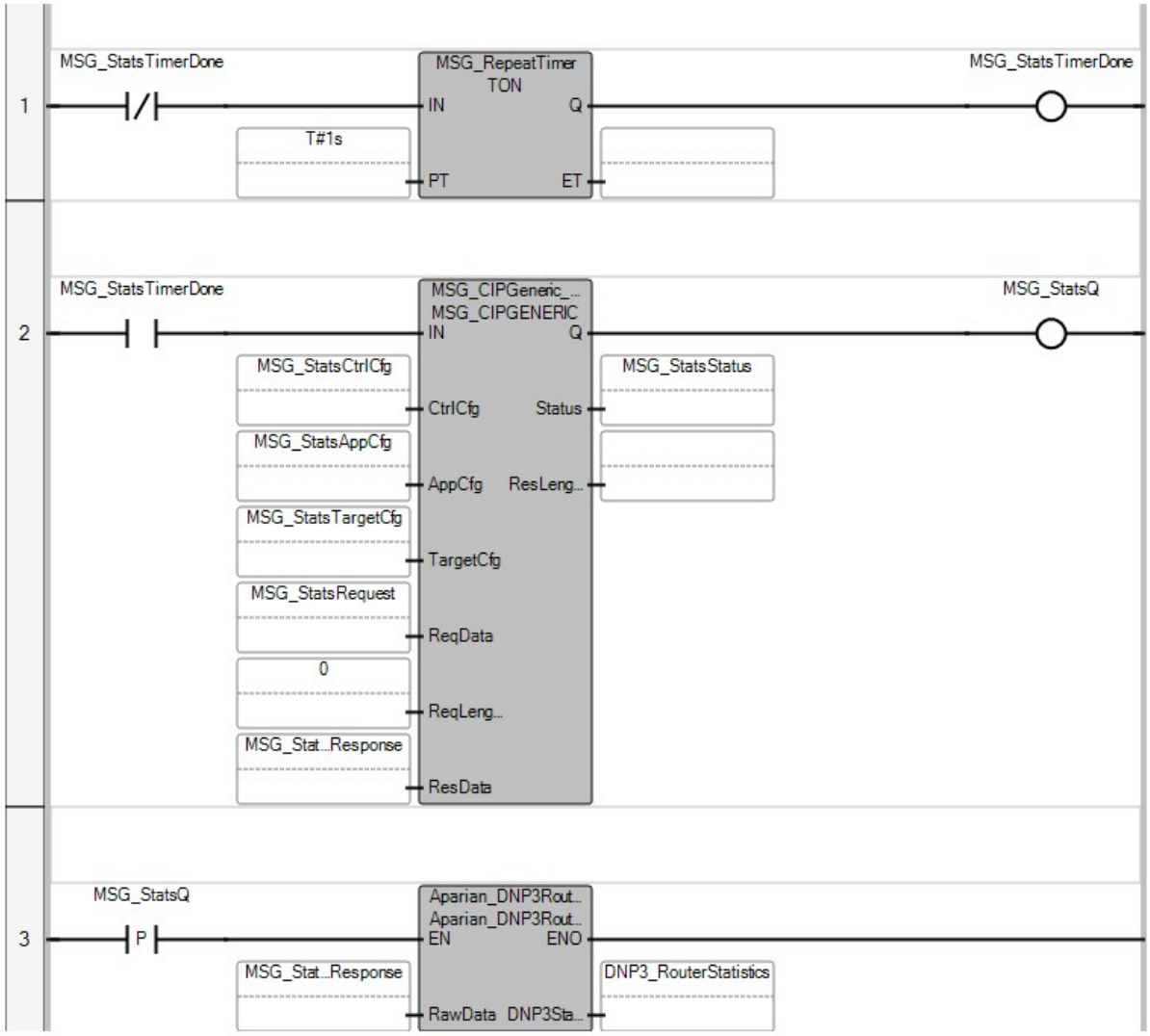



FIGURE 16 - DNP3 STATISTICS PROGRAM

## 5.1. CONFIGURATION

The CIP path contained within the local variable ***MSG_StatsTargetCfg*** must be set for the DNP3 Router.

| MSG_StatsTargetCfg | CIPTARGETCFG | | ... | |
|---|---|---|---|---|
| MSG_StatsTargetCfg.Path | STRING | | '4,192.168.1.232' | CIP destination path |
| MSG_StatsTargetCfg.CipConnMode | USINT | | 0 | 0 - Unconnected, 1 - Class3 connection |
| MSG_StatsTargetCfg.UcmmTimeout | UDINT | | | Unconnected message time out. |
| MSG_StatsTargetCfg.ConnMsgTimeout | UDINT | | | Connected message time out. |
| MSG_StatsTargetCfg.ConnClose | BOOL | | FALSE | TRUE: Close CIP connection upon message completion |

FIGURE 17 - DNP3 ROUTER 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 DNP3 Router.

## 5.2. OPERATION

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

| DNP3_RouterStatistics | ... | AparianDNP3RouterDNP3Statistics |
|---|---|---|
| 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 - DNP3 ROUTER STATISTICS TAG

The last item in the structure "***Online***", indicates that the CIP message to the DNP3 Router was successful. All the values in the structure are valid only if the ***Online*** is True.