# BECKHOFF DNP3 Master/Slave Driver User Manual

## V3 RC23

## Feb 2020


www.pbsControl.com

| | | | |
|---|---|---|---|
| 1 | Kamjoo bayat | Create Ver 1.0 document | Q1 - 2014 |
| 2 | Kamjoo Bayat | Adding SYS , WDT Tags to kernel and make it non block operation Ver 1.3 | Q4 - 2014 |
| 3 | Kamjoo bayat | Adding Multi Master | May - 2015 |

| | | | |
|---|---|---|---|
| | | Functionality to driver , adding driver boot delay | |
| 4 | Kamjoo Bayat | Adding RTU Restart when No communication between TC and DNP Kernel | August 2016 |
| 5 | Kamjoo Bayat | Adding Some notes about TC Routers setting in Manual | March 2017 |
| 6 | Kamjoo Bayat | Adding DNP3 Master Functionality Release of V3 RC5 | Jan 2019 |
| 7 | Kamjoo Bayat | Adding Data logging based on SQLite database – Adding Time Label and Flag to Signals , increasing number of DNP slave to 7 instance , RC15 | March 2019 |
| 8 | Kamjoo Bayat | Adding DNP Tag Status to DNP3 Slave Driver , DNP Master Enhancement , Fixing ADS Lost Connection with Reconnecting to ADS Server | August 2019 |
| 9 | Kamjoo Bayat | Adding Direct DNP Tag Mapping from Master to Slave Kernels | Sep 2019 |
| 10 | Kamjoo Bayat | Adding mse event gathering from ADS | Oct 2019 |
| 11 | Kamjoo Bayat | Multi drop RS485 master driver mode is included | Nov 2019 |
| 12 | Kamjoo Bayat | Adding Multifragment to APP Layer | FEB2020 |
| | | | |
| | | | |
| | | | |
| | | | |

Table of content

# 1 – Introduction

pbsControl company developed DNP3 Master/Slave driver for BECKHOFF CX Series Controllers . DNP3 Driver is compatible with WinCE5.0, WinCE6.0 and Win32 .Please refer to www.beckhoff.com for detail information about CX Series Controllers.

DNP3 Driver is developed by C Language and gets small resources of Controller. So it is running smoothly on CX8000 and CX9000 controllers.

For technical support please contact kb@pbscontrol.com

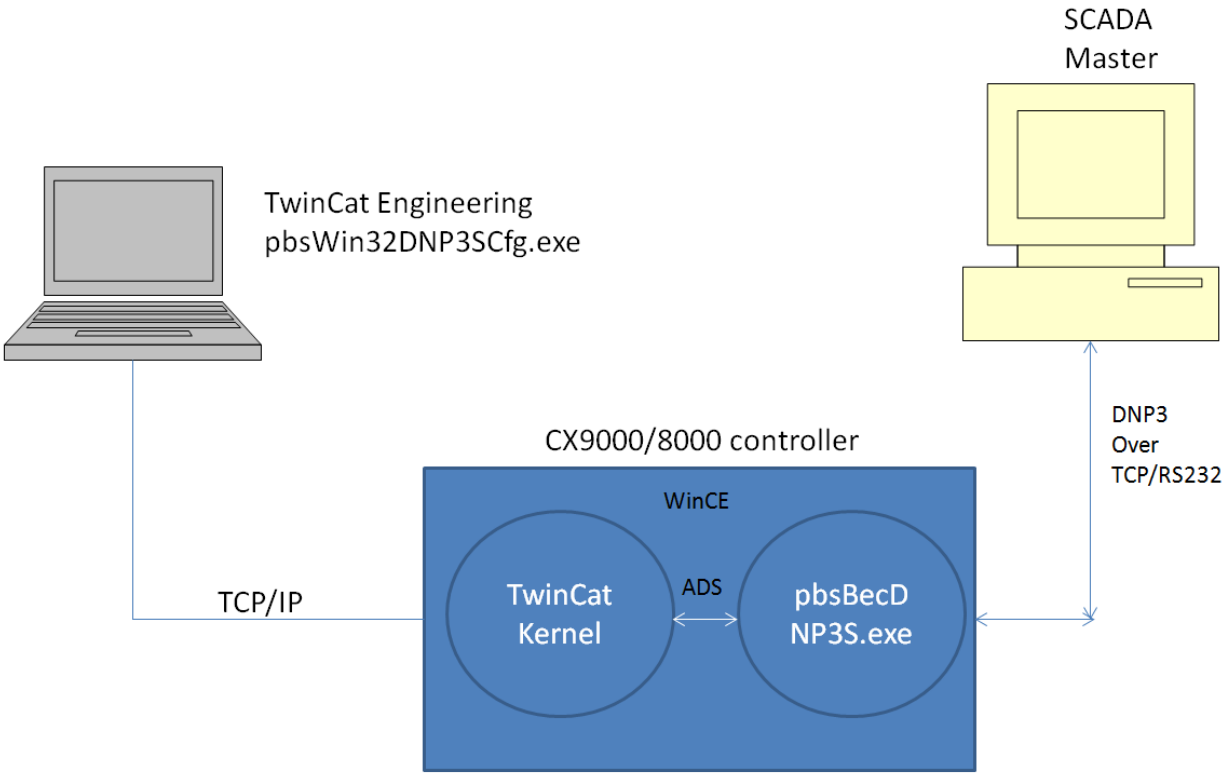
Driver license is linked to MACID of controller. Driver is running for 30 min without license as demo purpose. For each Controller you need to purchase separate license. License files for Slave and Master are separate.

Driver is tested with TwinCat 2 and TwinCat 3 .

# 2 – Driver Structure and installation

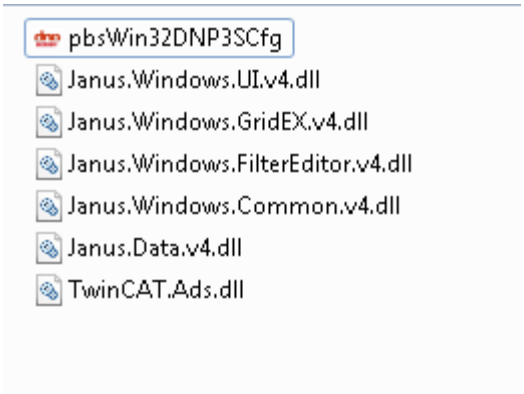
DNP3 Driver has following components:

- Configurator utility which is running on windows operating system. Configurator is developed by DotNet4. So you need to have DotNet 4 runtime engine on Configuration PC. (pbsWin32DNP3SCfg.exe)
- DNP3 Runtime kernel is running on CX9000/CX8000/CX1000 Controllers. (pbsBecDNP3SV3.exe) .Runtime kernel is developed by C for WindowsCE/win32 operating system and should run in CX Controller. Runtime Kernel is communicating with Twincat Kernel With ADS protocol for reading/Writing TwinCat variables and from the other side is communicating with DNP3 Master/Slave Driver.




Download DNP3 Slave Configurator utility from [www.pbscontrol.com](http://www.pbscontrol.com) and unzip it at any path in configuration PC.

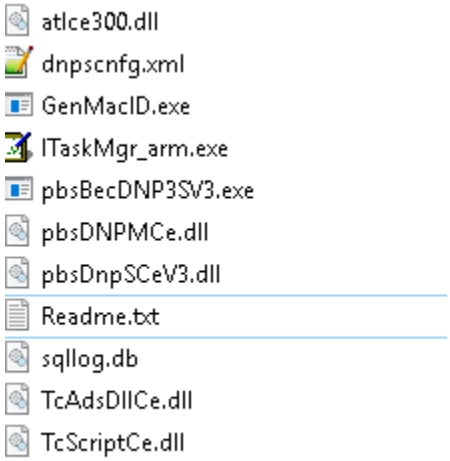
You can see following files:




- pbsWin32DNP3SCfg.exe main utility for DNP3 slave Configuration.
- JanusXXXX user interface library ( MUST be here)
- TwinCat.Ads.dll TwinCat Library for reading tags from CX Controller ( MUST be here)

Download and unzip runtime kernel for CX9000/8000/1000/Win32 from www.pbscontrol.com

Make a new directory at CX controller and copy runtime kernel to this directory.




Put pbsBecDNP3sV3.exe at WinCE Startup. Search “Automatic Start of CE Applications” in Beckhoff Information system.

For win32 runtime kernel put pbsBecDNP3SWin32V3.exe kernel at windows auto startup folder.

For proper running of kernel you should have following files at root of driver folder :

- pbsBecDnp3sV3.exe main application
- pbsDnpMCe.dll DNP3 Master library
- pbsDnpSCeV3.dll DNP3 Slave Library
- sqllog.db SQLite data base for logging DNP3 Master driver Tags
- dnpscfg.xml configuration file .

other folders are made automatically based on your configuration on the RTU by pbsBecDnp3sv3.exe application .

Run “Beckhoff CX Configuration Tool“ inside controller and do following settings at FTP tab:

Server Active: checked

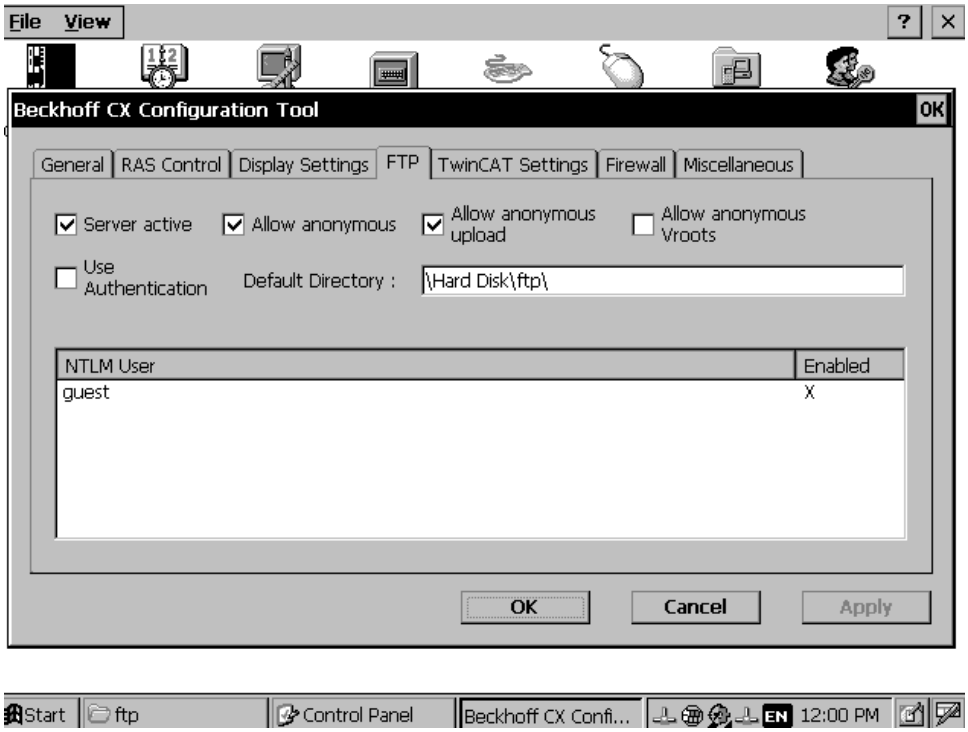
Allow anonymous: checked

Allow anonymous upload: checked

Use Authentication: Uncheck

Default Directory: Set to DNP3 slave runtime kernel directory

Restart controller. Now you are ready for DNP3 configuration.




We are using FTP for transferring Configuration and license file to controller. If you transfer configuration and license files by simple copy /Paste, it will work.

Dnpsi1 : first instance of DNP3 slave driver .

Dnpsi2 : second instance of DNP3 slave driver .

Dnpsi7 : seventh instance of DNP3 slave driver .

You can configure maximum 7 dnps slave instance on CX controller for communication with 7 DNP3 Master Software at the same time.

Dnpscfg.xml is configuration file which is making by configurator utility.if you want to copy /paste configuration file manually , you need to change name of your configuration ( which is made by pbsWin32DNP3SCfg.exe) to dnpscfg.xml

License.lic driver license file which is linked to mac ID of controller. Without license file, driver will work for 30 Min for test and demonstration.

Licensem.lic license file for DNP3 Master Driver .

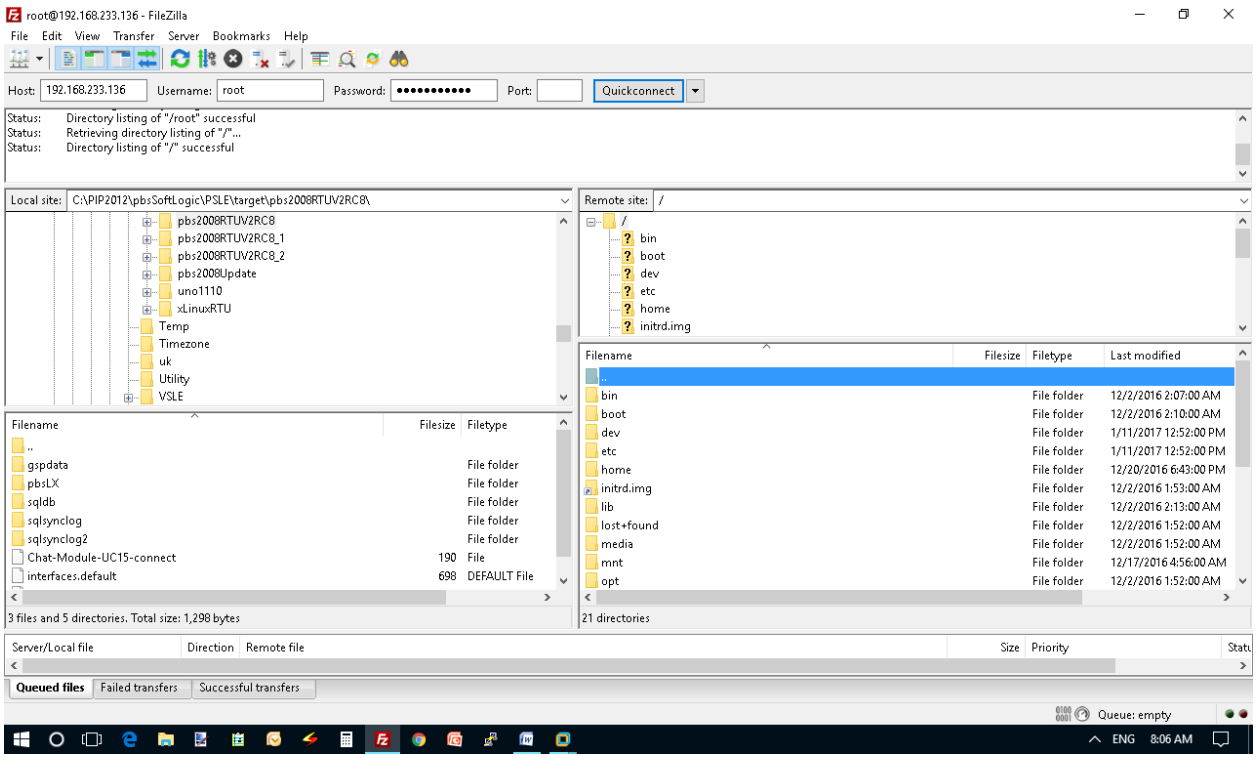
You can easily transfer files and Edit files by Filezilla client and Notepad++ utility. Please download and install them on Eng PC.

## Working with FileZilla

You can use FileZilla client utility to explore and edit RTU Files and directories.

Download filezilla from https://filezilla-project.org/

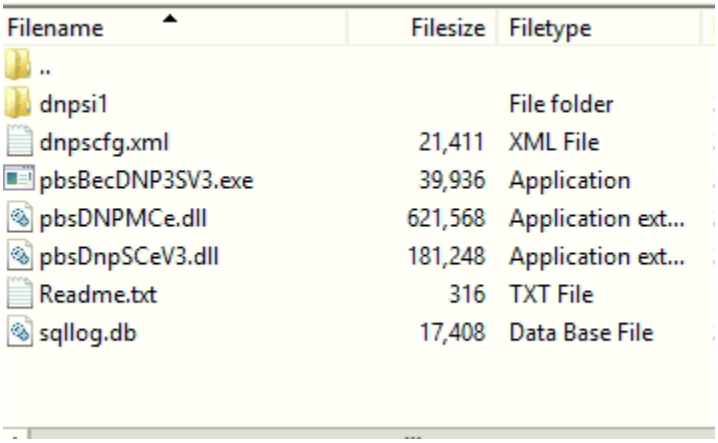
Run filezilla client you will see following page :



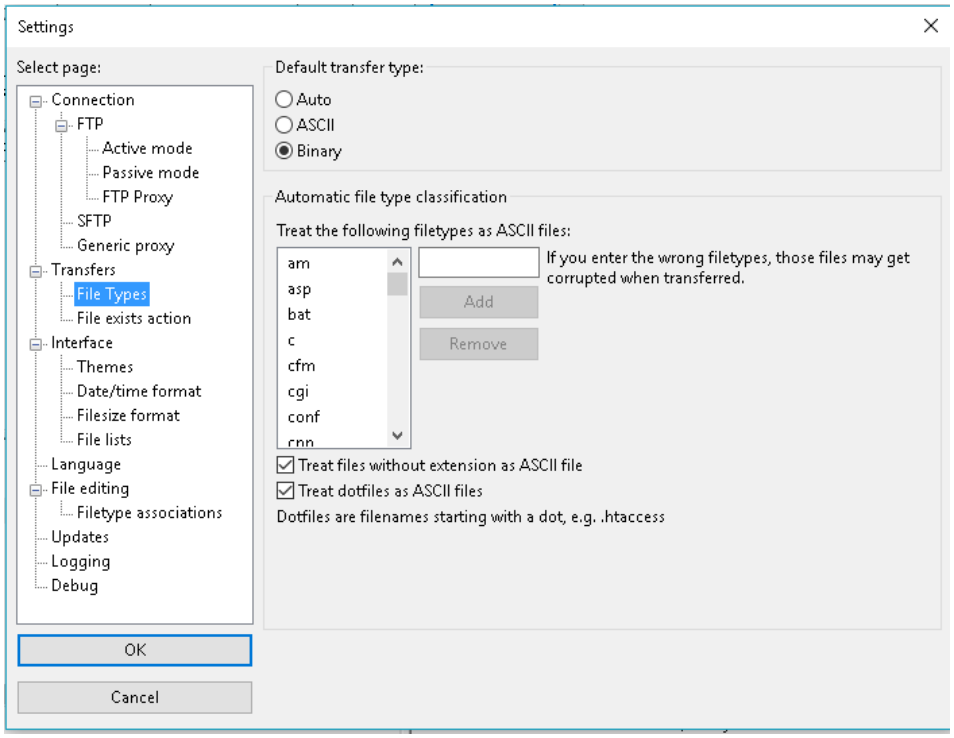

Type RTU IP at host field. Type root and root password in user name and password fields .for beckhoff RTU put user name and password blank and just click on quickconnect button .

RTU directories are showing at right panels and your PC directories at left s panels.

By default Filezilla is showing CX FTP folder that is set before to DNP kernel folder at right panel .




Note : for transferring files between Windows and Linux /WinCE Systems , always set Transfer File Type to Binary. you can find this option in Edit Menu , Setting menu and Transfers Segment .



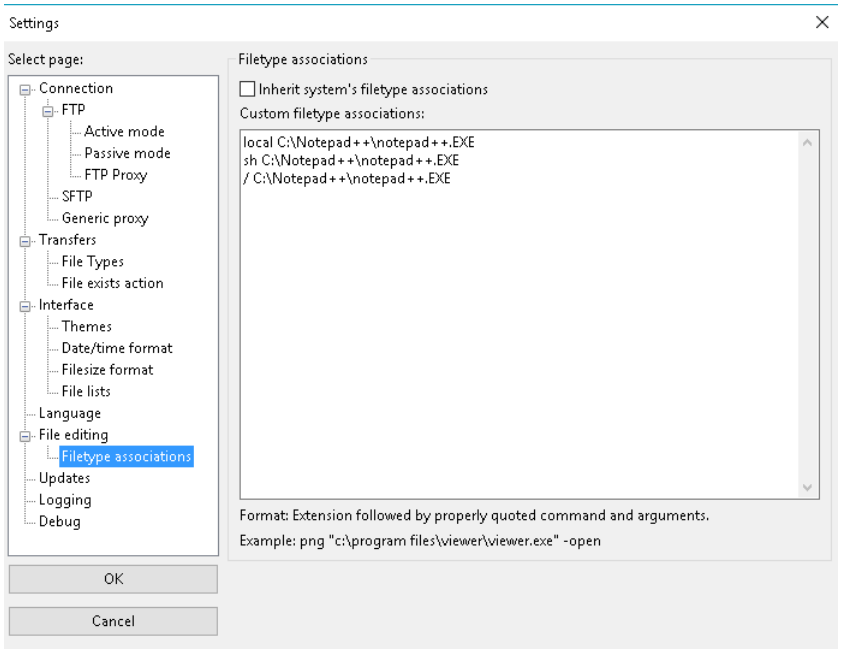

By default it is set to "Auto" that is damaging Linux files at transfer time from Windows to RTU.

For editing RTU configuration files in windows you need to use NotePad++ Editor to not damage Text file format when transfer to windows System .

Install NotePad++ utility from https://notepad-plus-plus.org/

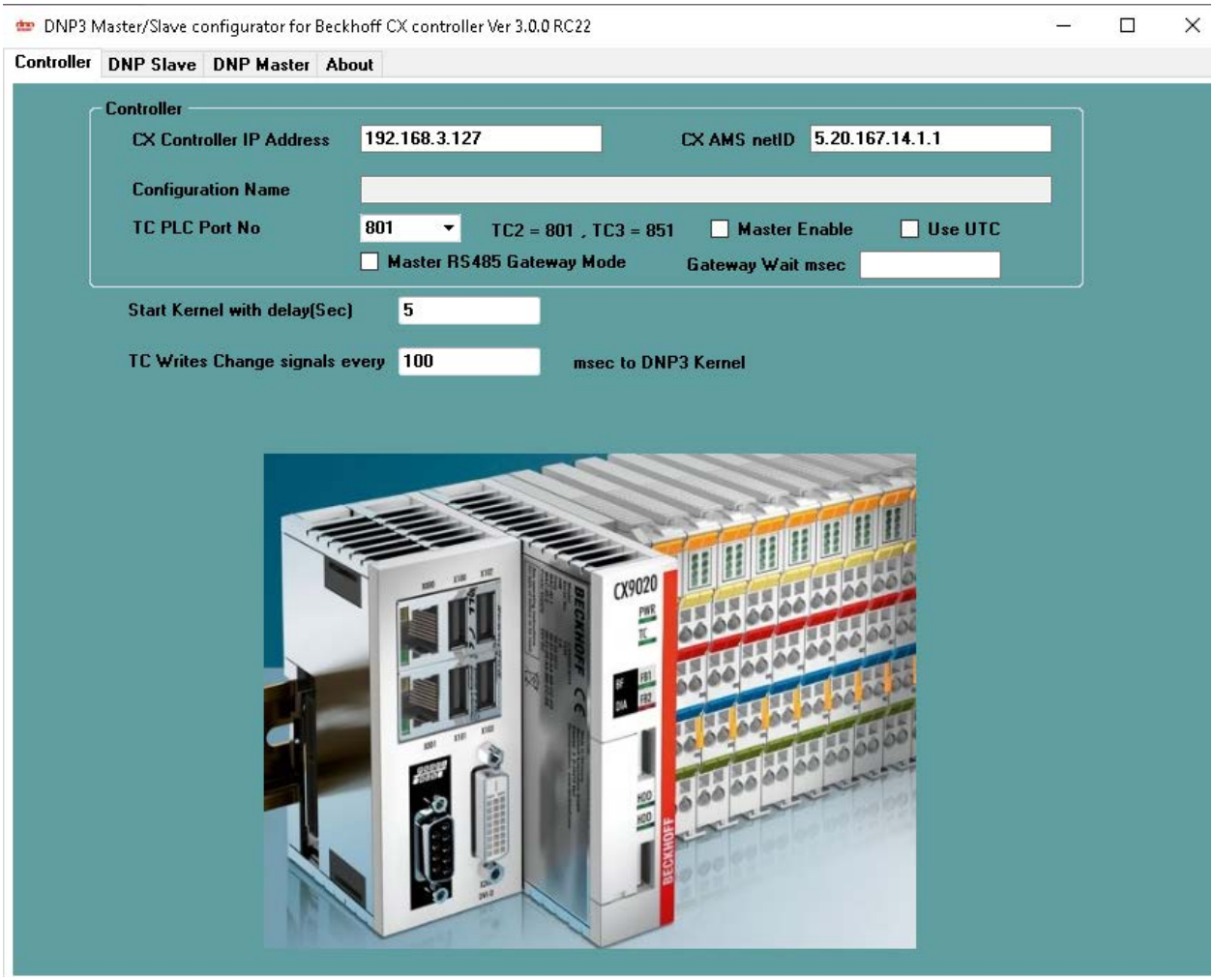
At first time that you View/Edit any Linux Configuration file , Filezilla will ask you for Custom Editor .

In this Stage set Nodepad++ as default editor in Filezilla . This will change File Editing Option in Setting page as following :



# 3- Driver Configuration (Slave)

Run pbsWin32DNP3SCfg.exe utility. You will see following page:




Set CX controller IP address and AMS netID for communication with CX Controller. IP address is using for transferring configuration file to controller by FTP and AMS netID is used for reading twinCat Tags from controller.

TC PLC Port number : TwinCat Port number for communication. For TC Version 2 , use 801 and for TwinCat Version 3 use 851 .

Master Enable :for Enabling DNP3 Master Driver . If DNP3 Master is Enable you need to have License file for Master Driver . “Licensem.lic” otherwise kernel will stop after 30 Min

Use UTC :Time zone of DNP3 Master and CX controller should be same or both using UTC . Otherwise you will not get events and only class0 tags ( current value of tags without time label) will move to DNP3 Master .

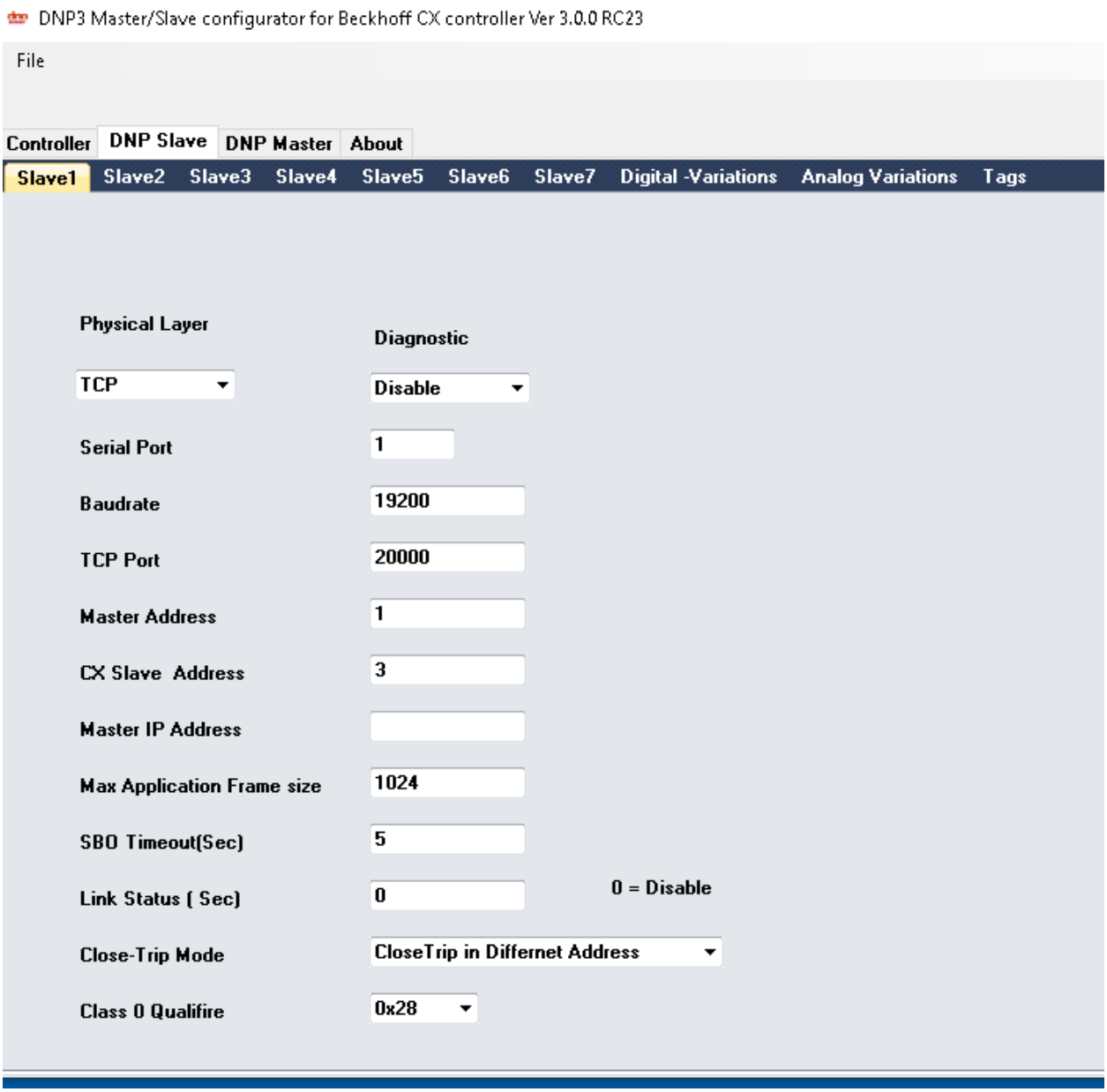
Start Kernel with Delay( Sec) : When Driver is starting , it will wait for this time ( normally considered for proper loading of TwinCat Kernel) before connect to TwinCat Kernel .

TC Write Change Signals every msec to DNP3 Kernel : This parameter shows TwinCat time resolution for writing changes to DNP3 Kernel .

Master RS485 Gateway Mode: RS485 Gateway Mode is used for multi drop RS485 master driver. Please refer to DNP Master configuration for detail information.

DNP Slave Configuration:

You can connect CX RTU to 7 master SCADA at the same time. For each Instance you can set following parameters. First Instance (Slave1) is always enabling. But you can Enable/Disable other instances. Select DNP Slave Tab. you will see following page:




Select physical layer TCP or RS232. You can select TCP or RS232 for each instance.

Serial Port: Port Number for communication in RS232 Mode .No affect in TCP Mode.

Baudrate : communication Baud Rate for RS232 Mode.

TCP Port : TCP Communication port number for TCP Port. By Default DNP3 port number is 20000.you should set different TCP Port number for each DNP Slave driver .

Master Address: DNP3 master SCADA address .Ask from master SCADA Administrator.

CX Slave address: DNP3 slave Address for CX Controller.

Master IP Address: DNP3 master SCADA IP address . If set to blank any IP can be connect to RTU .

Maximum application Frame size: Based on DNP3 protocol, maximum application frame size is 2048 bytes, but some masters could not handle 2048 bytes. Ask Master SCADA administrator for this parameter. We advise to use fewer values than maximum application frame size for this parameter. For example if Master supports 1024 bytes as maximum application frame size, set 1000 for this parameter. From version 3.RC23 Multi fragment application layer is supported and by default DNP Slave driver can transfer up to 2048 DNP Tags

SBO timeout (Sec): select before Operation time out for AOB and DOB tags.

Link Status (sec): This parameter is used in TCP /RS232 Mode. If master is not sending any request to RTU before No Communication timeout, then DNP3 driver will close connection and wait for a new master SCADA Connection. Link

Status is linked to keep alive timer in master SCADA. Suppose Keep alive timer is 60 sec in master SCADA, then Master SCADA will send Link status command to RTU every 60 sec .In DNP3 slave driver you should set Link Status more than keep alive timer .For above sample you can set 80 sec.

If Link Status is set to 0, Link Status functionality is disabling and DNP3 Driver is not checking master communication request timeout.

Diagnostic : If you want to see DNP Frame on Beckhoff Screen for diagnostic purpose , you can use this parameter . you can see physical layer , Data Link Layer , Transport Layer and application layer messages by this parameter . Set this Parameter to disable in normal operation of RTU.

Close –Trip Mode : When DNP Master Send Trip or close command to RTU , for protection purpose , you can consider separate Digital Output Commands for Close and Trip operation . When you use different address for close –trip commands , then DNP slave driver is only accept Close command on even DNP Address and Trip on Odd DNP Address .

At Close-Trip in Same Address Mode , DNP Slave driver is not checking Even or Odd for DNP Address . So Master can send Trip and close command on Same DNP3 Address .

Class0 Qualifier: You can Set DNP3 Qualifier for answering to class0 by this option. This is used only for class0 reading .

Qualifier 0x00 : one byte is used for start address , one byte is used for end address and after that Group Data with variation without address is transferred . For example suppose RTU wants to transfer 10 DI from address 100 to 110 by Q 0x00 , then data will transfer as following :

1-2-100 – 110- DI Value 100 - …… DIValue110

In this mode , there is no any address with Data Points .

For Q 0x00 because one byte is considered for point address , so you can only transfer tags with dnp address less than 255 .

Qualifier 0x01 :Two bytes is used for start address , two bytes is used for end address and after that Group Data with variation without address is transferred . For example suppose RTU wants to transfer 10 DI from address 100 to 110 by Q 0x01 , then data will transfer as following :

1 -2-100-0–110-0-DI Value 100 - …… DIValue110

In this mode , there is no any address with Data Points .

For Q 0x01 because two bytes is considered for point address , so you can transfer tags address up to DNP Standard ( 65535)

Qualifier 0x17 : one byte is used for Number of Tags , and each data point has one byte for DNP Address . For example suppose RTU wants to transfer 10 DI from address 100 to 110 by Q 0x17 , then data will transfer as following :

1-2-10–100- DI Value 100 -101- DI Value 101- …… -110-DIValue110

Qualifier 0x28 : two bytes is used for Number of Tags , and each data point has two bytes for DNP Address . For example suppose RTU wants to transfer 10 DI from address 100 to 110 by Q 0x17 , then data will transfer as following :

1-2-10–0-100-0- DI Value 100 -101-0- DI Value 101- …… -110-0-DIValue110

Note : if your DNP address are in different range it is better to use Q 0x17 or 0x28 .

If you have different address range for Tags and you must use Q0x00 or 0x01 then DNP Slave driver is automatically detected tags with same address range and make grouping based on tags address range .

For events only Q 0x28 is used for transferring events and unsolicited communication .

## Basic DNP3 Concepts

Before describing Other DNP3 Slave parameters we need to review some DNP Concepts:

In DNP3 main data types are Digital(Binary) Input, Analog Input, Double Bit Binary Input, Counter, Frozen Counter, Digital(Binary) Outputs Status, Analog Output Status, Digital Output Command and Analog Output Command.

Any DNP3 Tag has current value (Class0) and buffered value (Class 1 , 2, 3) . Based on DNP3 Standard there is no priority for different class 1 , 2 and 3 .

When a DNP3 tag has class0 , it means tag value changes will not buffer with proper time label .

When DNP3 tag has class 1 , 2 or 3 , it mean tag value changes will buffer inside RTU with proper time of change .

In DNP3 for separating Static value of tags and event value (Buffered Data), Tag Group concepts will use.

The most common Groups are as following:

Group 1 is Static value of Digital Input Tag

Group 2 is event buffer of Digital Input Tag

Notice that these two groups refer to same DI tag. Group 1 shows current value and Group 2 shows event buffered.

Group 3 is Double Bit Binary Input Current value

Group 4 is Event Buffer of Double Bit Binary Input.

Double Bit Binary Input from DNP3 Document:

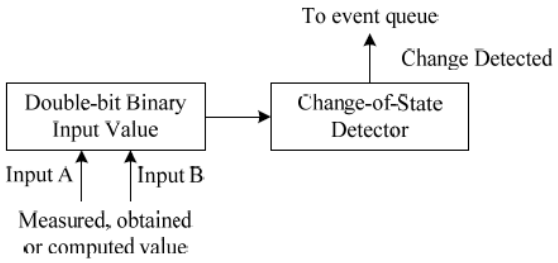



**Figure 11-8—Double-bit binary input model**

A double-bit binary input point is modeled as two single-bit binary inputs that are indivisibly linked together.

The four input states are defined in **Table 11-14**.

**Table 11-14—Double-bit binary input states**

| State value (UINT2) | State name | Description | Corresponding state if a single-bit binary input were used instead |
|---|---|---|---|
| 0 | INTERMEDIATE | Transitioning between end conditions | — |
| 1 | DETERMINED_OFF | End condition, determined to be OFF | 0 |
| 2 | DETERMINED_ON | End condition, determined to be ON | 1 |
| 3 | INDETERMINATE | Abnormal or custom condition | — |

Group 10 is Digital Output Status current value.

Group 11 is Digital Output Event Buffer.

Group 12 is Digital Output Command.

Group 20 is Counter static value.

Group 21 is Frozen Counter Static Value.

Group 22: event buffered of Counters

Group 23: Event buffered of Frozen Counters.

Group 30 is Static value of Analog Input

Group 32 is Event Buffer of Analog Input

Group 40 is Static value of Analog Outputs Status

Group 41 is Analog Output Command.

Group 42 Event Buffer of Analog Output

Group 50 is Data Time.

Group 80 is Internal Indication.

Internal Indication Concept: at any DNP3 Frame from Slave to master direction, There are 2 Bytes called Internal indication. (IIN) IIN is not in frames from Master to salve direction.

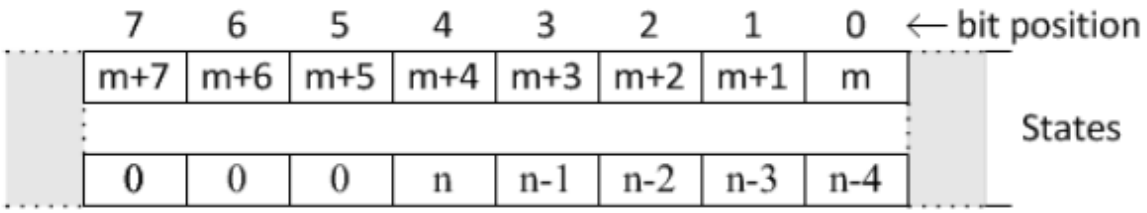
IIN definition is as following:

| Index | Bit | Name |
|---|---|---|
| 0 | IIN1.0 | BROADCAST |
| 1 | IIN1.1 | CLASS_1_EVENTS |
| 2 | IIN1.2 | CLASS_2_EVENTS |
| 3 | IIN1.3 | CLASS_3_EVENTS |
| 4 | IIN1.4 | NEED_TIME |
| 5 | IIN1.5 | LOCAL_CONTROL |
| 6 | IIN1.6 | DEVICE_TROUBLE |
| 7 | IIN1.7 | DEVICE_RESTART |
| 8 | IIN2.0 | NO_FUNC_CODE_SUPPORT |
| 9 | IIN2.1 | OBJECT_UNKNOWN |
| 10 | IIN2.2 | PARAMETER_ERROR |
| 11 | IIN2.3 | EVENT_BUFFER_OVERFLOW |
| 12 | IIN2.4 | ALREADY_EXECUTING |
| 13 | IIN2.5 | CONFIG_CORRUPT |
| 14 | IIN2.6 | RESERVED_1 |
| 15 | IIN2.7 | RESERVED_2 |

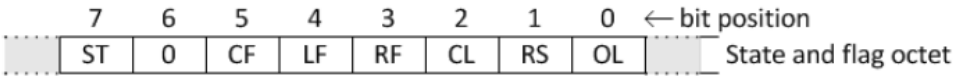
Tag Variation : in DNP3 Master can read data from RTU by different presentations . Suppose in first read ,master wants to read analog Input from address 1 to 100 by 32bit format and in second scan master read same analog input in 16 bit format . So Variations are refer to same tag Group.

Each Group has different variations.

Group 1, Static value of Digital Input, Variation is packed format.




Variation 2 is with flag:




**A.2.2.2.2 Formal structure**

**BSTR8:** Flag Octet

| | |
|---|---|
| Bit 0: | ONLINE |
| Bit 1: | RESTART |
| Bit 2: | COMM_LOST |
| Bit 3: | REMOTE_FORCED |
| Bit 4: | LOCAL_FORCED |
| Bit 5: | CHATTER_FILTER |
| Bit 6: | Reserved, always 0 |
| Bit 7: | STATE—Has a value of 0 or 1, representing the state of the physical or logical input. |

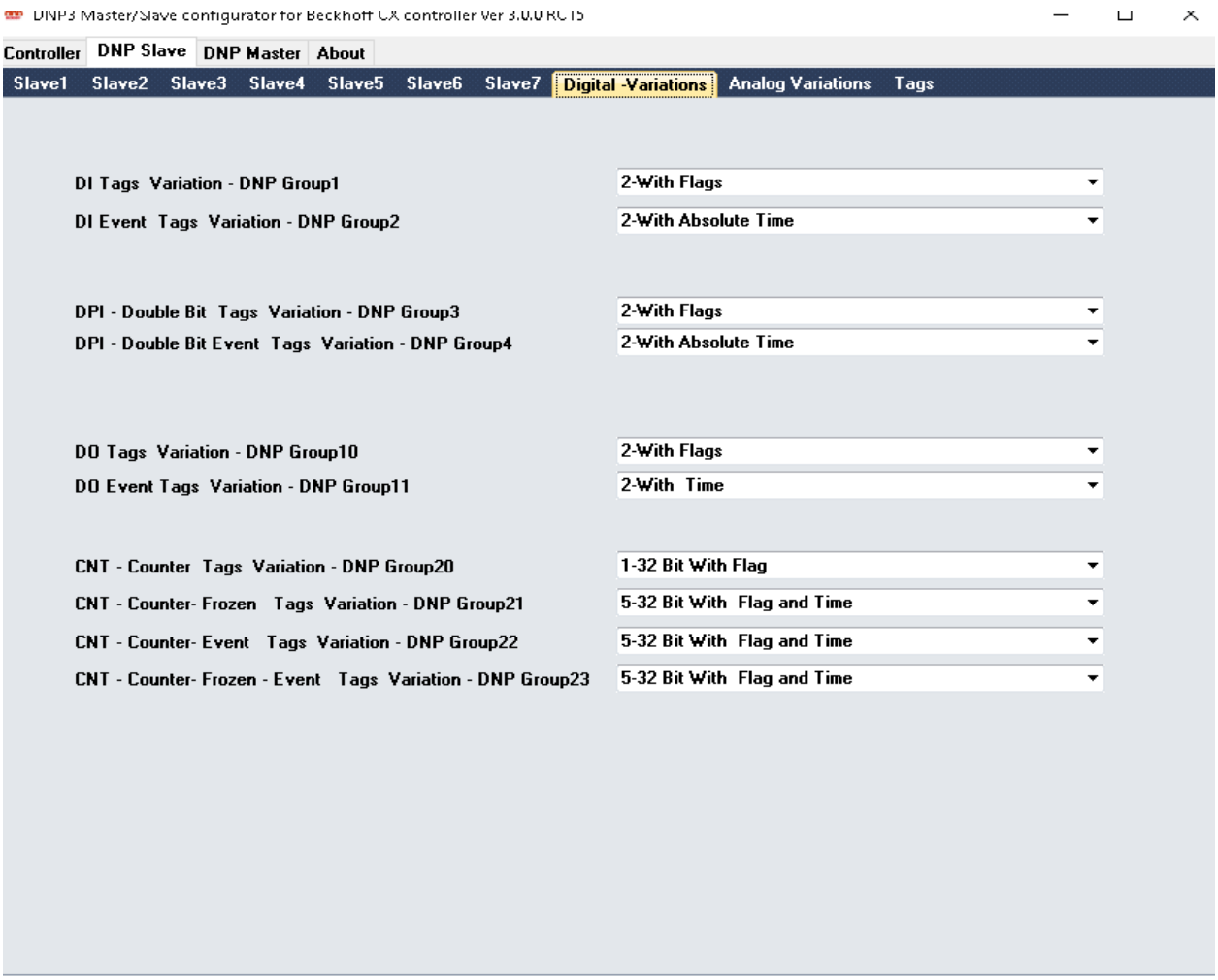
What is concept of Default variation?

Suppose master wants to read all tags in class 1.in this query master is not determine variation for Slave. Only master is asking for class1 tags. For this type of Master request, default variation should be set in RTU.

For following command default variation will use:

- Master is asking for class 0 tags
- Master is asking for class 1 tags
- Master is asking for class 2 tags
- Master is asking for class 3 tags
- Master is Sending Integrity poll (IP) Means master is asking class 0 , 1, 2 and 3 in one request
- Master is Sending RBE (Read Events) means Master is asking class 1, 2 and 3 in one request. Master wants to read only tags changes, not static value of all tags.
- Unsolicited communication. When Slave is sending tag changes to master without master request.
-

In DNP3 Configurator you can set default variation for Slave driver from digital-variation and analog –variation page.



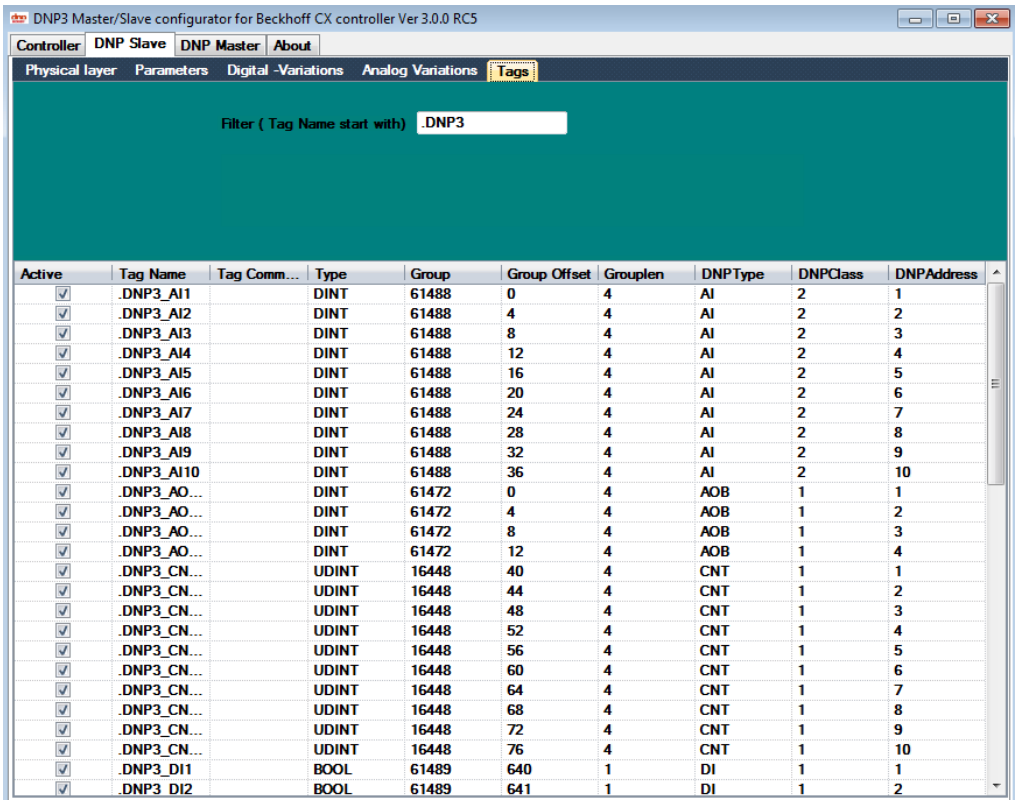

DNP3 Master/Slave configurator for Beckhoff CX controller Ver 3.0.0 RC15
Controller
DNP Slave
DNP Master
About
Slave1
Slave2
Slave3
Slave4
Slave5
Slave6
Slave7
Digital -Variations
Analog Variations
Tags
AI Tags Variation - DNP Group30
2-16 Bit With Flag
AI Event Tags Variation - DNP Group32
4-16 Bit With Time
FI Tags Variation - DNP Group30
5-Floating Point with Flag
FI Event Tags Variation - DNP Group32
7-Floating Point with Time
AO Tags Variation - DNP Group40
1-32 Bit With Flag
AO Event Tags Variation - DNP Group42
4-16 Bit With Time

## Reading Tags from RTU :

For Reading Tags from TC ,you should do following tasks :

- Connect and define controller to TwinCAT System manager.
- TwinCat Runtime kernel is in runtime mode ( Not configuration Mode)
- Write TwinCat PLC program and define DNP3 Variables.
- Compile and load twinCat Program to controller.

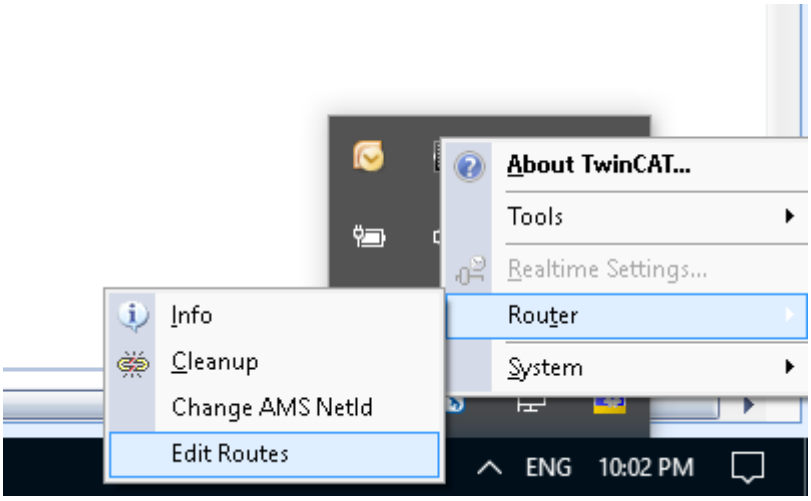
For reading TwinCat Tags from controller and configuration of tags for DNP3 protocol, you should use Tags Page.



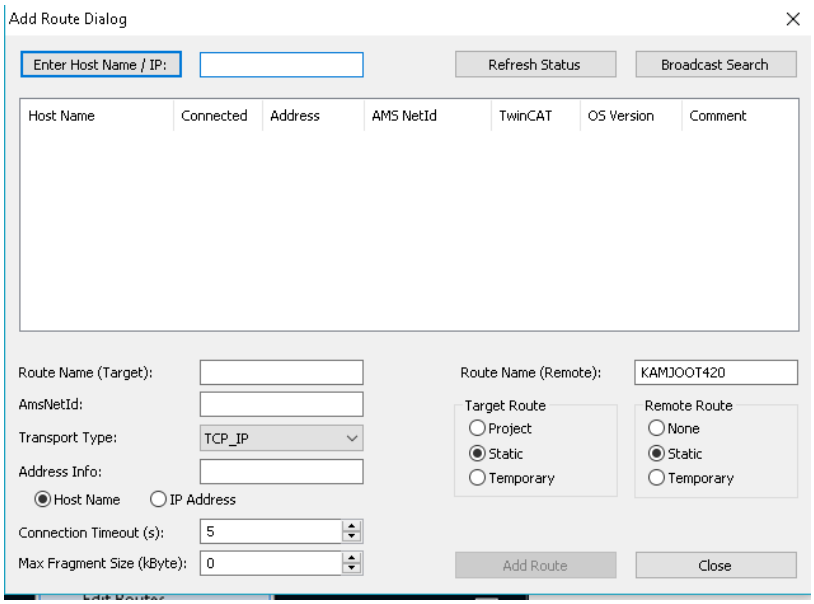

| Active | Tag Name | Tag Comm... | Type | Group | Group Offset | Grouplen | DNPType | DNPClass | DNPAddress |
|---|---|---|---|---|---|---|---|---|---|
| ☑ | .DNP3_AI1 | | DINT | 61488 | 0 | 4 | AI | 2 | 1 |
| ☑ | .DNP3_AI2 | | DINT | 61488 | 4 | 4 | AI | 2 | 2 |
| ☑ | .DNP3_AI3 | | DINT | 61488 | 8 | 4 | AI | 2 | 3 |
| ☑ | .DNP3_AI4 | | DINT | 61488 | 12 | 4 | AI | 2 | 4 |
| ☑ | .DNP3_AI5 | | DINT | 61488 | 16 | 4 | AI | 2 | 5 |
| ☑ | .DNP3_AI6 | | DINT | 61488 | 20 | 4 | AI | 2 | 6 |
| ☑ | .DNP3_AI7 | | DINT | 61488 | 24 | 4 | AI | 2 | 7 |
| ☑ | .DNP3_AI8 | | DINT | 61488 | 28 | 4 | AI | 2 | 8 |
| ☑ | .DNP3_AI9 | | DINT | 61488 | 32 | 4 | AI | 2 | 9 |
| ☑ | .DNP3_AI10 | | DINT | 61488 | 36 | 4 | AI | 2 | 10 |
| ☑ | .DNP3_AO... | | DINT | 61472 | 0 | 4 | AOB | 1 | 1 |
| ☑ | .DNP3_AO... | | DINT | 61472 | 4 | 4 | AOB | 1 | 2 |
| ☑ | .DNP3_AO... | | DINT | 61472 | 8 | 4 | AOB | 1 | 3 |
| ☑ | .DNP3_AO... | | DINT | 61472 | 12 | 4 | AOB | 1 | 4 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 40 | 4 | CNT | 1 | 1 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 44 | 4 | CNT | 1 | 2 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 48 | 4 | CNT | 1 | 3 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 52 | 4 | CNT | 1 | 4 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 56 | 4 | CNT | 1 | 5 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 60 | 4 | CNT | 1 | 6 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 64 | 4 | CNT | 1 | 7 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 68 | 4 | CNT | 1 | 8 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 72 | 4 | CNT | 1 | 9 |
| ☑ | .DNP3_CN... | | UDINT | 16448 | 76 | 4 | CNT | 1 | 10 |
| ☑ | .DNP3_DI1 | | BOOL | 61489 | 640 | 1 | DI | 1 | 1 |
| ☑ | .DNP3_DI2 | | BOOL | 61489 | 641 | 1 | DI | 1 | 2 |

***For reading tags from CX Controller, right click on Tags page and execute read tags from controller***. It will read all TwinCat Tags from controller that start with Filter field. Because there are many systematic tags inside CX Controllers, so it is better to define DNP3 tags for your logic and start them with DNP3 name, then you can read just DNP3 tags by configurator and make setting for each tag.

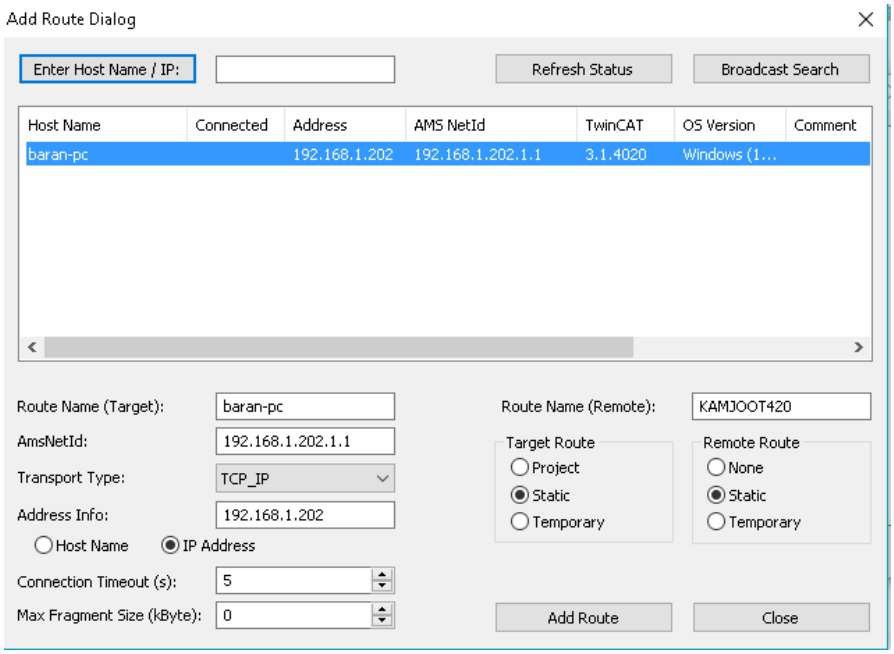
Important Point: you should Install TC ADS Utility in engineering station and add Controller AMSID in to router table of engineering station. Otherwise you will get Timeout when you want to read tags from controller.




When you run Edit Routers and click on add Button you will see following figure :

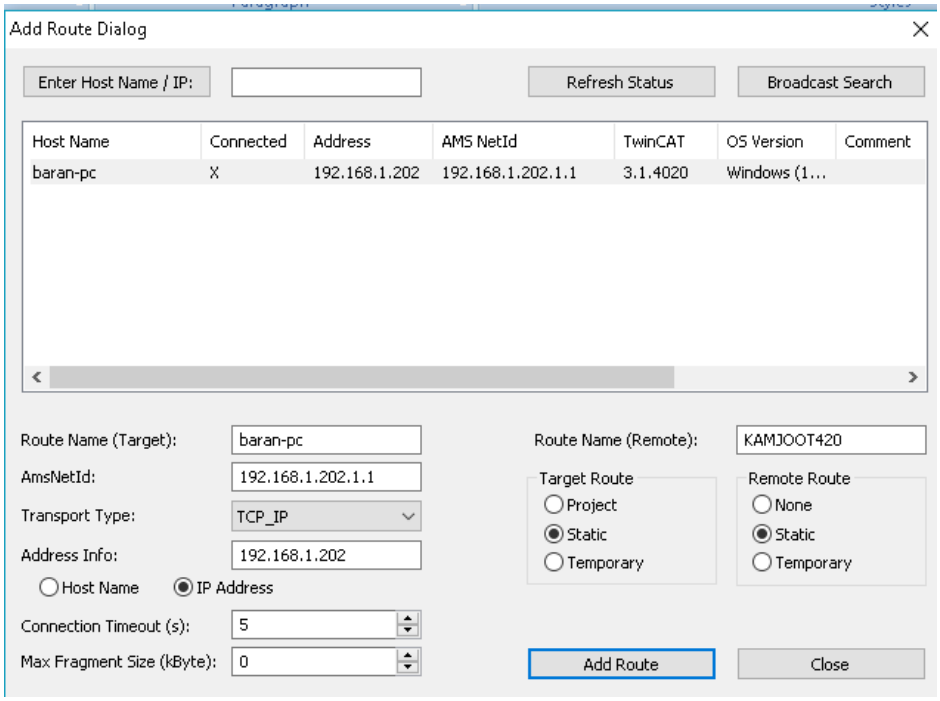



Normally it is better to set Address Info on IP Address and click on Broadcast search as following :




Click on Add Route, you need to enter User name and password of the target with administrator right:

If route is add properly, you will see X sign in Connected part:




Now you can run DNP3 Configurator and read tags from Controller .

DNP3 Tag Type: We support following types:

- DI : Digital input Read By Master with different variations , DNP Group1 , 2
- AI : Analog input Read By Master with different variations , DNP Group 30,31,32,33
- CNT : Counter Read By Master with different variations DNP Group 20,21,22,23
- FI : Float Input : DNP Group 32 , Variation 5
- DOB : Digital Output Block Write by master with different mode DNP Group 12 ,13
- AOB : Analog Output Block Write by master with different mode , DNP Group 41
- DO : DO Status Read By Master with different variations , DNP Group 10,11
- AO : AO Status Read By Master with different variations , DNP Group 40
- DPI : Double Bit Binary Read By Master with different variations , DNP Group 3,4
- FOB : Float Output Block , Group 41
- SYS: System Tag for monitoring Master Connection status for TCP connection. When you have two master , first SYS tag will link to first driver instance and second SYS tag will link to second driver instance.SYS Tags are using same DNP Address Space like AOB Tags . So you should not have any AOB Tags with Same DNP Address for SYS Tags .

- WDT: System Tag for making watch dog signal for TwinCat runtime.

Class: Based on DNP3 Standard we have class 0 ,1,2,3

Class 0 means current value of tags without event buffering. So if you put class 0 for a tag, RTU is not buffering tag changes and every time master read tag , RTU will send current value .

Class 1,2,3 there is no different or priority between different classes. So if you put class 1,2,3 for a tag RTU will buffer all tag changes with time and will report to Master SCADA .

There is a cyclic buffer with 10,000 events for each DNP Type in RTU.

Address: DNP3 tag address. AI and FI are using same address range.

DNP3 function codes which are implemented:

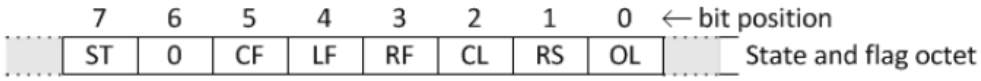
- Read class 0,1,2,3
- Integrity command
- Read Event by exception ( RBE)
- Time synchronization
- Enable /Disable unsolicited communications ( Transfer data from RTU to Master SCADA )
- Dynamic Class assign

- Freezing counters
- Write

DNP Tag Status:

In DNP3 Standard , any tag has status flag byte. Bit 0 shows Tag is online or not . Value 1 shows tag is online ad Value 0 Shows tag is offline . Other bits are as following:

DI Tag Status Flag:




**A.2.2.2.2 Formal structure**

**BSTR8:** Flag Octet

| | |
|---|---|
| Bit 0: | ONLINE |
| Bit 1: | RESTART |
| Bit 2: | COMM_LOST |
| Bit 3: | REMOTE_FORCED |
| Bit 4: | LOCAL_FORCED |
| Bit 5: | CHATTER_FILTER |
| Bit 6: | Reserved, always 0 |
| Bit 7: | STATE—Has a value of 0 or 1, representing the state of the physical or logical input. |

If you want to set DNP Tag Status Flag for driver you should define DIS , AIS , FIS , CNTS and DPIS Tags .

If did not define Status tag for a DNP Tag , Its Flag value is always 1 .

Suppose you define DNP Tag with type DI and address 10 , then if you define DIS Tag with same DNP Address 10 , then DIS Tag Value will use as Status flag of DI Tag .

Look at following code in TC :

```
0001 VAR_GLOBAL
0002
0003     DNP3_AI1 AT %Q* : DINT ;
0004     DNP3_AI2 AT %Q* : DINT ;
0005     DNP3_AI3 AT %Q* : DINT ;
0006     DNP3_AI4 AT %Q* : DINT ;
0007     DNP3_AI5 AT %Q* : DINT ;
0008     DNP3_AI6 AT %Q* : DINT ;
0009     DNP3_AI7 AT %Q* : DINT ;
0010     DNP3_AI8 AT %Q* : DINT ;
0011
0012     DNP3_AI9 AT %Q* : DINT ;
0013     DNP3_AI10 AT %Q* : DINT ;
0014
0015
0016     DNP3_AI1Status AT %Q* : INT ;
0017     DNP3_AI9Status AT %Q* : INT ;
0018
```

Two Status Tag for DNP3_AI1 and DNP3AI9 is defined in above sample . DNP Status Flag always has type int for DIS , AIS , FIS , CNTS, DPIS .

In DNP3 Configurator we will do following configuration for Status Tags:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ☑ | .DNP3_AI1STATUS | | INT16 | 16448 | 1690 | 2 | AIS | 1 | 1 |
| ☑ | .DNP3_AI9STATUS | | INT16 | 16448 | 1692 | 2 | AIS | 1 | 9 |

And Following configuration for DNP Tags :

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ☑ | .DNP3_AI1 | | DINT | 61488 | 0 | 4 | AI | 1 | 1 |
| ☑ | .DNP3_AI10 | | DINT | 61488 | 36 | 4 | AI | 1 | 10 |
| ☑ | .DNP3_AI2 | | DINT | 61488 | 4 | 4 | AI | 1 | 2 |
| ☑ | .DNP3_AI3 | | DINT | 61488 | 8 | 4 | AI | 1 | 3 |
| ☑ | .DNP3_AI4 | | DINT | 61488 | 12 | 4 | AI | 1 | 4 |
| ☑ | .DNP3_AI5 | | DINT | 61488 | 16 | 4 | AI | 1 | 5 |
| ☑ | .DNP3_AI6 | | DINT | 61488 | 20 | 4 | AI | 1 | 6 |
| ☑ | .DNP3_AI7 | | DINT | 61488 | 24 | 4 | AI | 1 | 7 |
| ☑ | .DNP3_AI8 | | DINT | 61488 | 28 | 4 | AI | 1 | 8 |
| ☑ | .DNP3_AI9 | | DINT | 61488 | 32 | 4 | AI | 1 | 9 |

As it is clear , Status Tags and DNP Tags has same DNP Address .

In TC when you change value of DNP3_AI1Status , this value will pass to DNP Kernel and will assign for DNP_AI1 Tag as status flag . When value of Tags ( DNP3_AI1Status or DNP3_AI1 ) is changed , DNP kernel will make an Event in Kernel .

So when Tag status changed , tag value with new status will report to master SCADA .

## DNP3 Slave driver Operation:

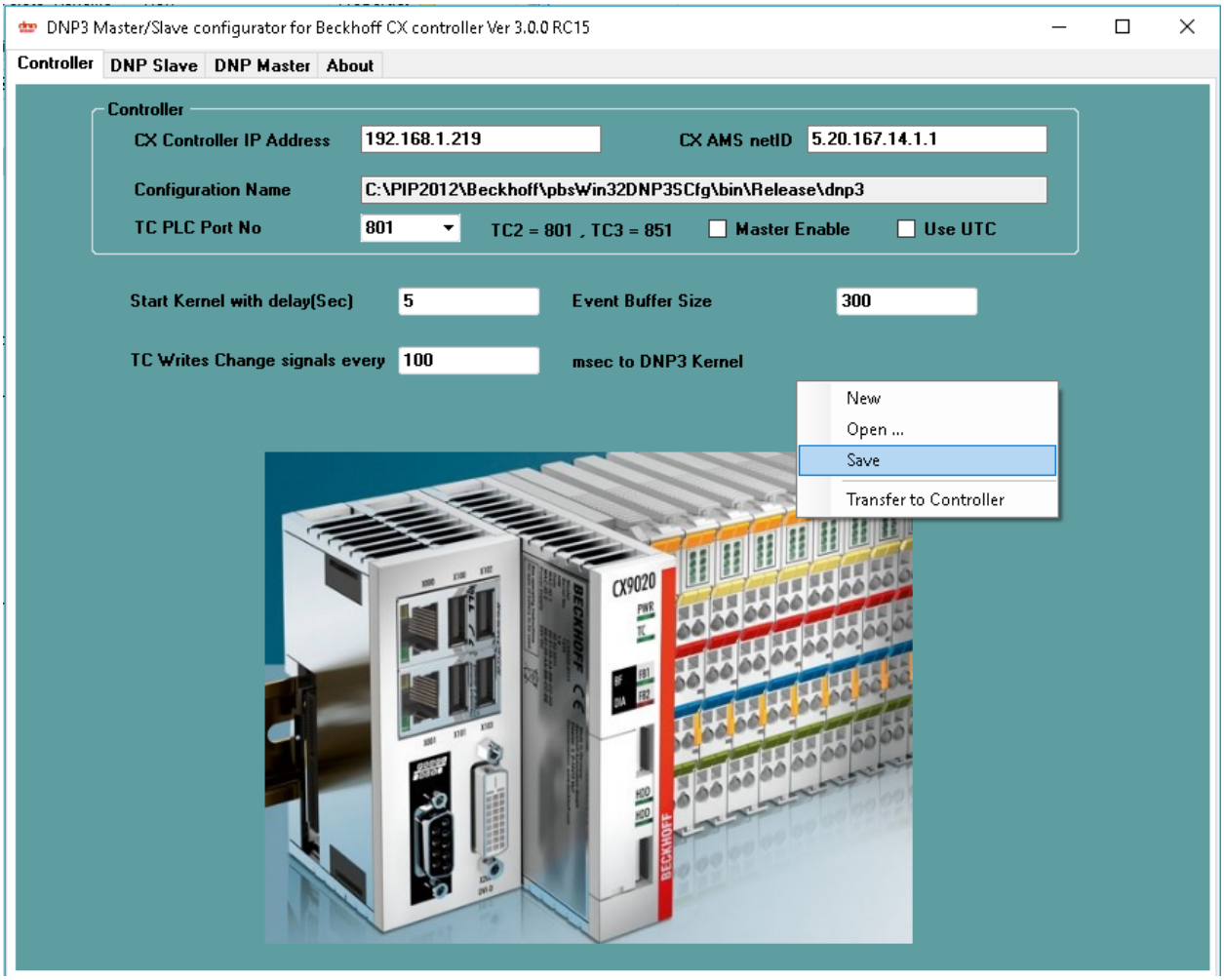
1 - Master SCADA will read all Input Signals ( DI , AI , FI , DO , AO , DPI)

- You need to write all Input Signals in your logic by TC

2 – Master SCADA will write Output Signals (DOB , AOB , FOB )

- You need to read all Output Tags in your logic by TC

For saving configuration file to a local file on your PC, select Controller page, and right click. You can see following menu:



Open …: open offline configuration file for editing.

Save: save opened configuration file.

Transfer to Controller: Transfer configuration file to controller.

## 4 – TwinCat programming

You need to define DNP3 tags in TwinCat programming environment as following sample. (DNP Slave Tags)

```
0001 VAR_GLOBAL
0002 (* AI, DI , FI , CNT Tag Definition*)
0003     DNP3_AI1 AT %Q* : DINT;
0004     DNP3_AI2 AT %Q* : DINT;
0005     DNP3_AI3 AT %Q* : DINT;
0006
0007     DNP3_FI4 AT %Q* : REAL;
0008     DNP3_FI5 AT %Q* : REAL;
0009     DNP3_FI6 AT %Q* : REAL;
0010
0011     DNP3_DI1 AT %Q* : BOOL;
0012     DNP3_DI2 AT %Q* : BOOL;
0013     DNP3_DI3 AT %Q* : BOOL;
0014
0015     DNP3_DO1 AT %Q* : BOOL;
0016     DNP3_DO2 AT %Q* : BOOL;
0017     DNP3_DO3 AT %Q* : BOOL;
0018
0019     DNP3_DPI1 AT %Q* : INT;
0020     DNP3_DPI2 AT %Q* : INT;
0021     DNP3_DPI3 AT %Q* : INT;
0022
0023     DNP3_CNT1 AT %Q* : UDINT;
0024     DNP3_CNT2 AT %Q* : UDINT;
0025     DNP3_CNT3 AT %Q* : UDINT;
0026
0027     DNP3_AO1 AT %Q* : DINT;
0028     DNP3_AO2 AT %Q* : DINT;
0029     DNP3_AO3 AT %Q* : DINT;
0030
0031 (*AOB , DOB Tag Definition*)
0032
0033     DNP3_SP1_W AT %I* : DINT;
0034     DNP3_SP2_W AT %I* : DINT;
0035     DNP3_SP3_W AT %I* : DINT;
0036     DNP3_SP4_W AT %I* : DINT;
0037
0038     DNP3_DOB1_W AT %I* : BOOL;
0039     DNP3_DOB2_W AT %I* : BOOL;
0040     DNP3_DOB3_W AT %I* : BOOL;
0041     DNP3_DOB4_W AT %I* : BOOL;
0042 END_VAR
```

Rule 1: You should update (Write) DI, AI, FI, CNT and DPI Tags in your logic. DNP3 Inputs Tags (DI, AI, FI, CNT, and DPI) should be defining as output variables in TwinCat.

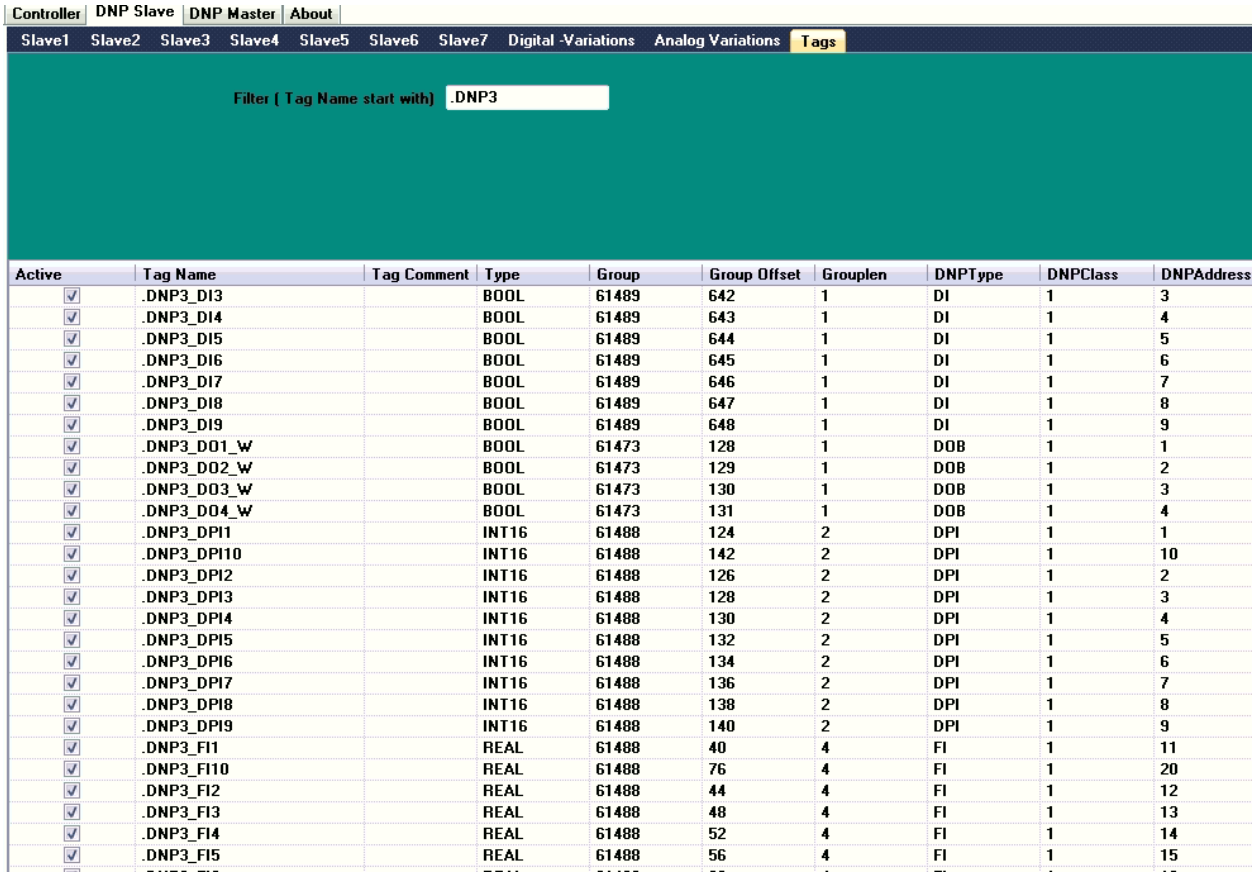
Rule 2: You should read AOB ,DOB,FOB tags in your logic.

AOB is analog Output Block and DOB is digital Output Block. FOB is Float Output Block. AOB and FOB use same address space . When SCADA Master wants to write AO,FO and DO tags to Controller, you should define DNP3 AOB,FOB and DOB Tag type. In TwinCat AOB,FOB and DOB tags should define as input Variables.

Rule 3: It is advised to define DNP3 Variables in TwinCat by DNP3 prefix. Then you can easily read DNP3 variables by DNP3 configurator utility.

Rule 4: you can define maximum 1024 DNP Tags

Rule 5: You should read TwinCat DNP3 Variables by DNP3 configurator utility and set DNP3 Type, Class and address. Please look at following figure as an example.




| Active | Tag Name | Tag Comment | Type | Group | Group Offset | Grouplen | DNPType | DNPClass | DNPAddress |
|---|---|---|---|---|---|---|---|---|---|
| ☑ | .DNP3_DI3 | | BOOL | 61489 | 642 | 1 | DI | 1 | 3 |
| ☑ | .DNP3_DI4 | | BOOL | 61489 | 643 | 1 | DI | 1 | 4 |
| ☑ | .DNP3_DI5 | | BOOL | 61489 | 644 | 1 | DI | 1 | 5 |
| ☑ | .DNP3_DI6 | | BOOL | 61489 | 645 | 1 | DI | 1 | 6 |
| ☑ | .DNP3_DI7 | | BOOL | 61489 | 646 | 1 | DI | 1 | 7 |
| ☑ | .DNP3_DI8 | | BOOL | 61489 | 647 | 1 | DI | 1 | 8 |
| ☑ | .DNP3_DI9 | | BOOL | 61489 | 648 | 1 | DI | 1 | 9 |
| ☑ | .DNP3_DO1_W | | BOOL | 61473 | 128 | 1 | DOB | 1 | 1 |
| ☑ | .DNP3_DO2_W | | BOOL | 61473 | 129 | 1 | DOB | 1 | 2 |
| ☑ | .DNP3_DO3_W | | BOOL | 61473 | 130 | 1 | DOB | 1 | 3 |
| ☑ | .DNP3_DO4_W | | BOOL | 61473 | 131 | 1 | DOB | 1 | 4 |
| ☑ | .DNP3_DPI1 | | INT16 | 61488 | 124 | 2 | DPI | 1 | 1 |
| ☑ | .DNP3_DPI10 | | INT16 | 61488 | 142 | 2 | DPI | 1 | 10 |
| ☑ | .DNP3_DPI2 | | INT16 | 61488 | 126 | 2 | DPI | 1 | 2 |
| ☑ | .DNP3_DPI3 | | INT16 | 61488 | 128 | 2 | DPI | 1 | 3 |
| ☑ | .DNP3_DPI4 | | INT16 | 61488 | 130 | 2 | DPI | 1 | 4 |
| ☑ | .DNP3_DPI5 | | INT16 | 61488 | 132 | 2 | DPI | 1 | 5 |
| ☑ | .DNP3_DPI6 | | INT16 | 61488 | 134 | 2 | DPI | 1 | 6 |
| ☑ | .DNP3_DPI7 | | INT16 | 61488 | 136 | 2 | DPI | 1 | 7 |
| ☑ | .DNP3_DPI8 | | INT16 | 61488 | 138 | 2 | DPI | 1 | 8 |
| ☑ | .DNP3_DPI9 | | INT16 | 61488 | 140 | 2 | DPI | 1 | 9 |
| ☑ | .DNP3_FI1 | | REAL | 61488 | 40 | 4 | FI | 1 | 11 |
| ☑ | .DNP3_FI10 | | REAL | 61488 | 76 | 4 | FI | 1 | 20 |
| ☑ | .DNP3_FI2 | | REAL | 61488 | 44 | 4 | FI | 1 | 12 |
| ☑ | .DNP3_FI3 | | REAL | 61488 | 48 | 4 | FI | 1 | 13 |
| ☑ | .DNP3_FI4 | | REAL | 61488 | 52 | 4 | FI | 1 | 14 |
| ☑ | .DNP3_FI5 | | REAL | 61488 | 56 | 4 | FI | 1 | 15 |

NOTE : When you change Variables in TC ( Add or delete or change Data type ) , sometimes TC is changing Tags Group – Group Offset and Group Len internally . So you need to refresh Tags in Configurator for proper running of DNP Kernel. Otherwise DNP Tags value will not update in TC .

# 5 – System Tags

SYS: From Version 3 RC23 , we changed mechanism for SYS and WDT Tags . SYS tag is showing Connection Status with master SCADA. When Master is connected to Driver, SYS tag value is 1

and when it is disconnect, SYS Tag Value will change to 0.

By SYS tag, twinCat Runtime kernel will find status of connection. When you have two DNP3 master SCADA , you need to define two SYS tag with different DNP3 Address .

DNP3 Address for Instance 1 of DNP Slave driver MUST be 1.

DNP3 Address for Instance 2 of DNP Slave driver MUST be 2.

DNP3 Address for Instance 3 of DNP Slave driver MUST be 3.

DNP3 Address for Instance 4 of DNP Slave driver MUST be 4.

DNP3 Address for Instance 5 of DNP Slave driver MUST be 5.

DNP3 Address for Instance 6 of DNP Slave driver MUST be 6.

DNP3 Address for Instance 7 of DNP Slave driver MUST be 7.

Define an INT Tag in Twincat for SYS as following:

```
DNP3_SYS AT %I* : DINT;
DNP3_WDT AT %I* :DINT;
```

After transferring configuration to Controller, and restart it, you can see status of DNP3 master connection in DNP3_SYS tag in Twincat runtime.

WDT: Watch Dog Timer Tag. With WDT Tag, TwinCat Runtime will find healthy status of Driver. WDT Tag makes a permanent 10 Sec Pulse for TwinCat. As TwinCat Runtime is detecting 10 Sec Pulse, DNP3 Driver is working properly. When TwinCat is not getting 10 Sec Pulse Train, Something is happened in Driver and needs to restart Controller by TwinCAT.

WDT Tag is working in TCP and Serial Mode communication and it is independent of Master SCADA Connection.

WDT Tag DNP Address can be any value expect 0 . if it is 0 it is disabled .

Define an INT Tag in Twincat for WDT as following:

```
DNP3_SYS AT %I* : DINT;
DNP3_WDT AT %I* :DINT;
```

After transferring configuration to Controller, and restart it, you can see DNP3 Driver start to write a 10 sec permanent pulse train to TwinCAT.

DNP Group 50 ( Date Time) , 60 ( Class) , 80 ( IIN) are supported in Driver but no need to do any configuration for this Groups .

## ADS Reconnecting

It is possible that ADS connection with DNP Kernel is lost inside CX controller. DNP Kernel will reconnect to ADS at following condition:

- DNP Kernel writes to TC by ADS and it makes error at write time.
- There is no any write from TC to DNP kernel for 3600 Sec.

So we advise to define WDT tag for DNP Kernel for not reconnecting to ADS after 3600 Sec if there is really no write from TC to DNP Kernel.

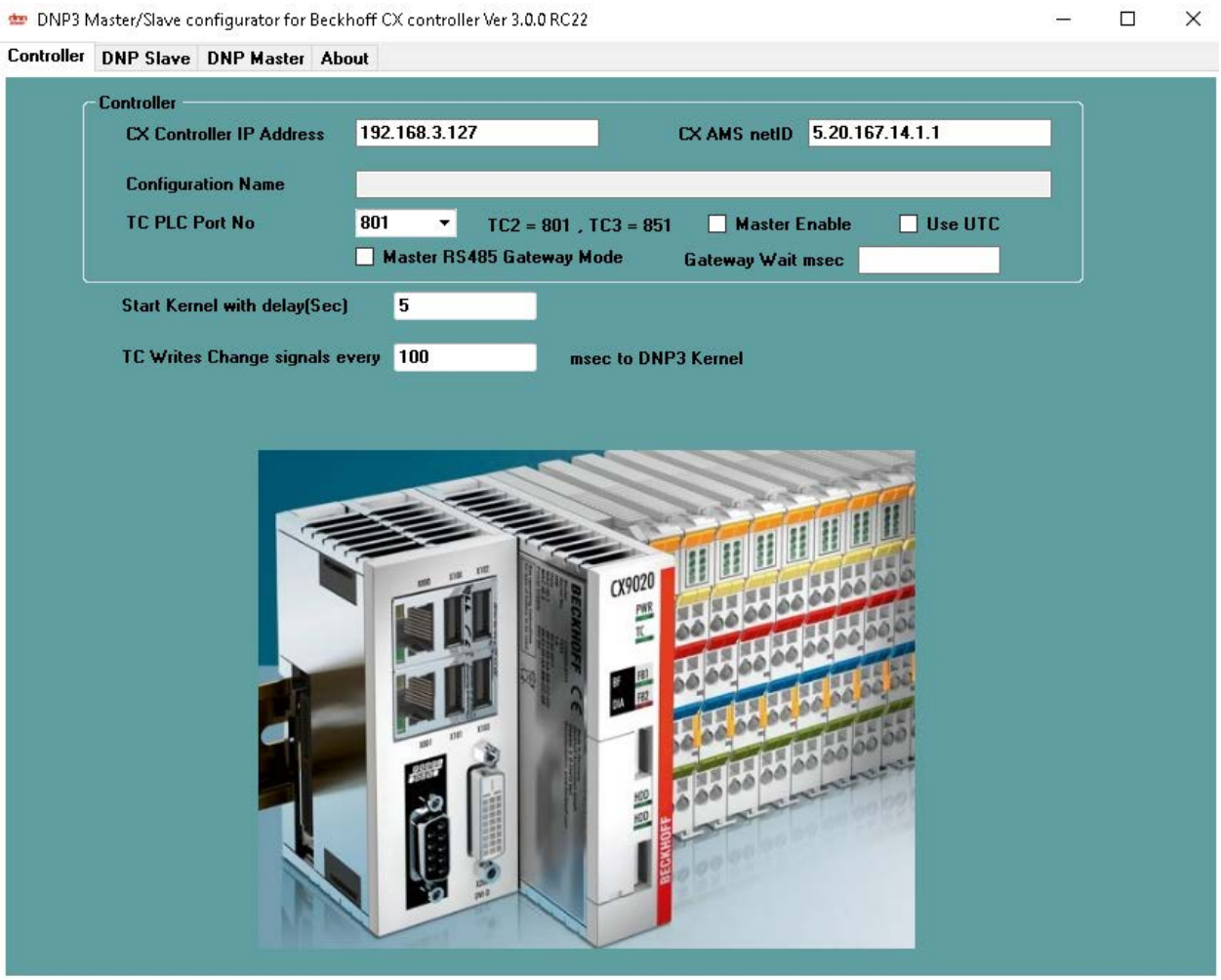
In Some projects Tags are not changing frequently, so maybe there is no any write from TC to DNP kernel, but ADS connection is not lost with DNP Kernel. For these projects using WDT signal is necessary to stop frequent reconnecting of ADS link with DNP Kernel.

# 6 – DNP Master Configuration

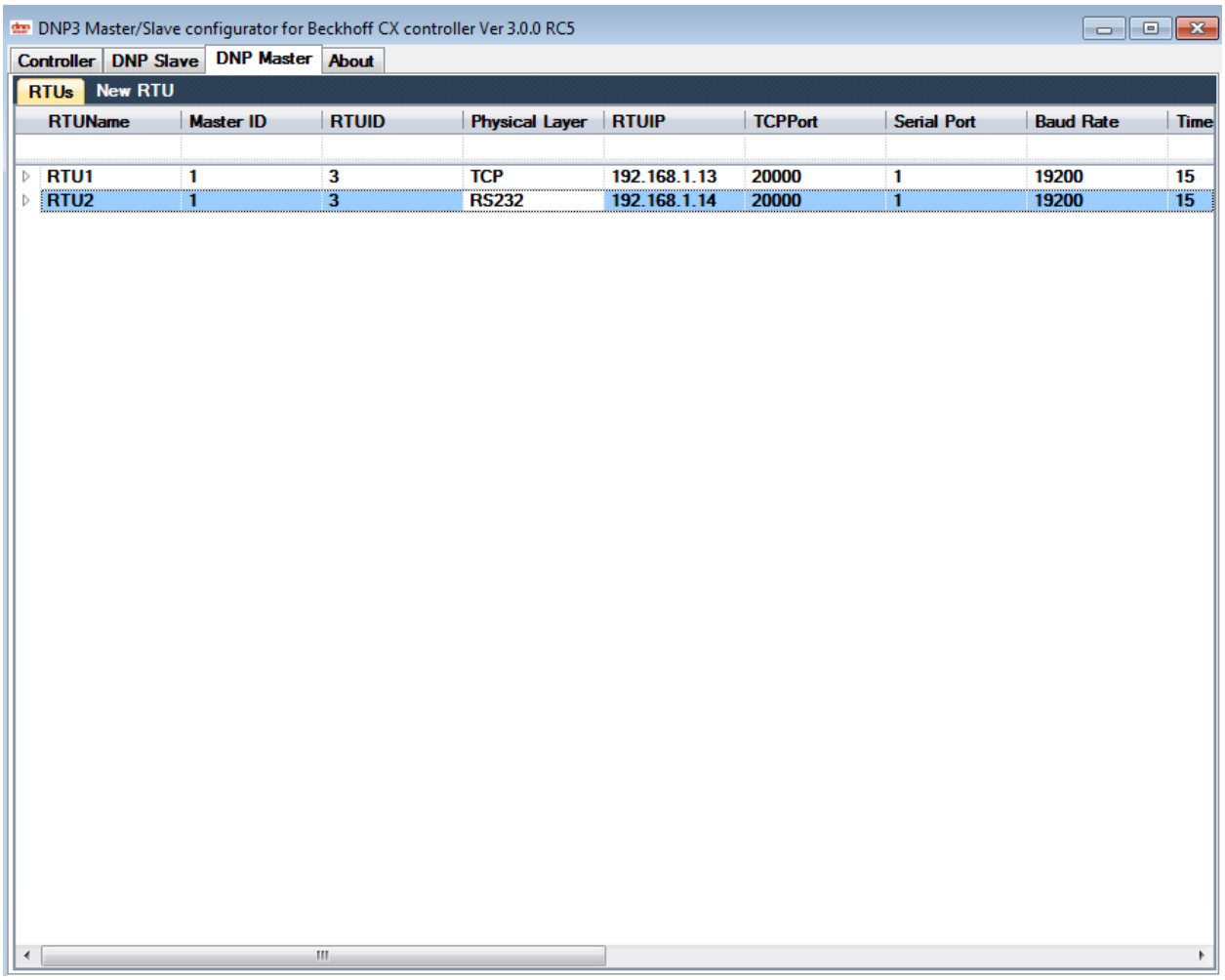
From Version 3 RC5 DNP3 Kernel supports DNP3 Master Protocol. You can connect to DNP3 Slave RTUS over RS232 or TCP and read /Write tags to external RTUS.

Suppose there is an old RTU with DNP3 Slave Functionality over RS232, then you can use CX controller as a gateway to convert it to DNP over TCP.

At Controller page, Enable Master Functionality:



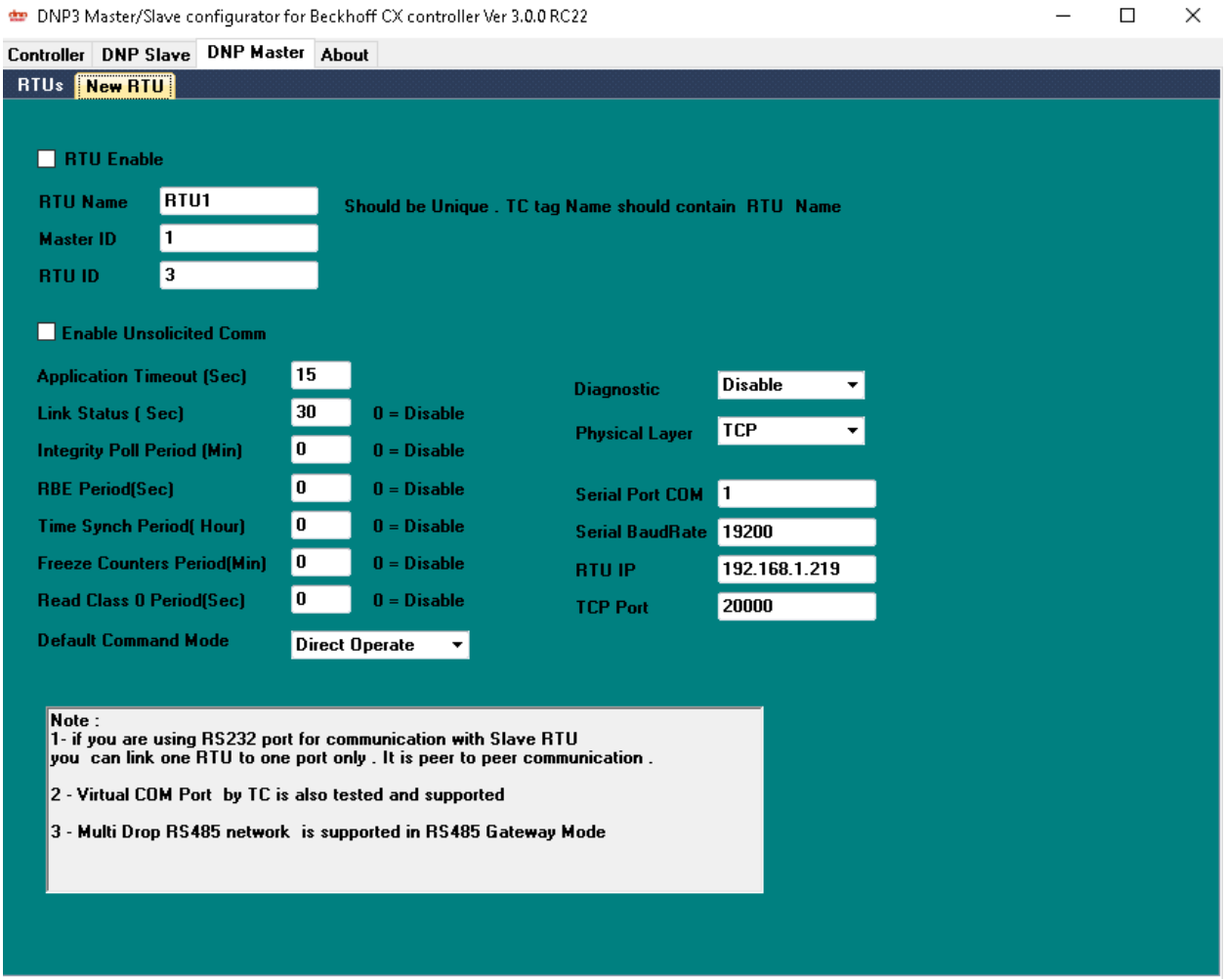
In DNP Master page you can see list of defined DNP3 Slave RTUS :




You can edit all RTU parameters from List .

By right click on list you can switch between list or card view for Slave RTU parameters.

For defining New RTU , it is more easy to go to New RTU Page :




RTU Name : Select a unique Name for RTU . Suppose you select RTU3. Then All Tags in TC for RTU3 should be start with RTU3 . If you are using TC3 , then configurator will look for all TC tags that contains RTU Name .

Master ID : DNP3 Master ID

RTU ID : DNP3 Slave ID

Enable Unsolicited COMM : If checked when Master Driver Connect to RTU and after Reading Tags by Integrity poll , Will Enable Unsolicited communication in Slave RTU . After that Slave RTU will send all DNP Tag Changes to Master Without Master Request.

If unchecked Master Will Disable Unsolicited Communication in Slave RTU .

Application Timeout ( Sec) : When Master is in Solicited Mode – Send Request to Slave and wait for answer - Master Driver will wait for above time to get answer , otherwise will move to Idle State and will be ready to handle next request .

Link Status ( Sec) : When there is no communication with RTU for Link Status time , Master Will Send Link Status Command to RTU to check status of connection . If not get ACK , Master will disconnect connection with Slave and reconnect again . If 0 , it is Disabled .

Integrity Poll Period (Min) : Shows Sending IP command period to Slave in min . If 0 , it is Disable . If value is -1 , Master Driver is not send IP to Slave RTU at connection time

RBE Period ( Sec) : Shows Sending RBE Command period to RTU in Sec . If 0, it is Disable .

Time Synch Period (Hour): Shows Sending Time Synch Command to RTU in Hour. If 0, it is Disabled .if -1 , DNP Master is not send Time Synch to RTU at connection time .

Freeze Counter Period (Min): Shows Sending Freeze Counters Command period to Slave RTU in Min. If 0, it is Disable .

Default Command Mode: Shows Default mode for DOB and AOB commands. Direct Operate or Select Before Operate .

Physical layer: Select TCP or RS232 . This is physical layer connection with Slave RTU.

For RS232 Connection you can only link one RTU to one Serial Port. Multi drop RS485 is supported but you need to Enable Master RS485 Gateway Mode.

Serial Port (COM) : This is COM number on Beckhoff CPU Module . Serial Ports that are connected by KBUS or EBUS are supported if you use proper TC driver for Virtual COM Port . Any COM port that is operating as Standard Serial port in WinCE can be used .

Baud rate : Serial Baud Rate for connection with RTU in RS232 Mode .

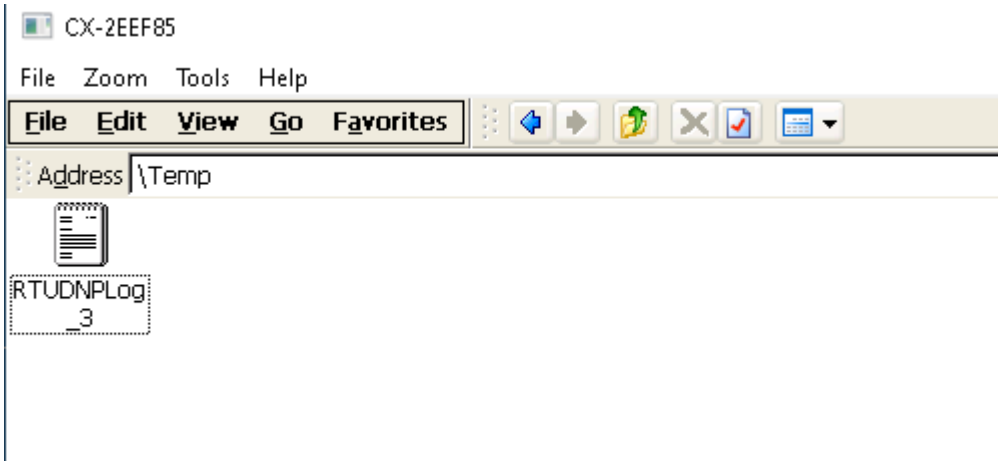
RTU IP : if you select TCP as communication layer you need to put RTU IP here .

TCP Port : if you select TCP as communication layer you need to put RTU TCP Port here . DNP Port is 20000

Diagnostic : if you want to analyze DNP Frames between CX Controller and Slave RTU , you can use this parameter . you can select from Disable , Physical Layer , Data Link Layer , Transport Layer and Application Layer options.

When Disable no DNP log File will generate.

When Set to Physical layer: only Physical layer Frames will log in a text file in CX \Temp \ Folder with name RTUDnpLog_{RTUID}



In following figure you can see one sample DNP Log file that is generated by DNP Master Driver:

```
 physcial layer :Write  To RTU:2019-12-24 3:58:59
,5 ,64 ,5 ,c0 ,3 ,0 ,1 ,0 ,76 ,9b

 physcial layer :Read  From RTU:2019-12-24 3:58:59
,5 ,64 ,5 ,0 ,1 ,0 ,3 ,0 ,f4 ,19

  DLL  Layer :GET from Physical:2019-12-24 3:58:59
5 64 5 0 1 0 3 0 f4 19

 Transport Layer :GET From APP:2019-12-24 3:58:59
c0 2 32 1 7 1 fa d6 e 36 6f 1

  DLL  Layer :GET From Transport:2019-12-24 3:58:59
c0 c0 2 32 1 7 1 fa d6 e 36 6f 1

 physcial layer :Write  To RTU:2019-12-24 3:58:59
,5 ,64 ,12 ,c4 ,3 ,0 ,1 ,0 ,91 ,b1 ,c0 ,c0 ,2 ,32 ,1 ,7 ,1 ,fa ,d6 ,e ,36 ,6f ,1 ,47 ,7e

 physcial layer :Read  From RTU:2019-12-24 3:59:0
,5 ,64 ,a ,44 ,1 ,0 ,3 ,0 ,b4 ,e1 ,c0 ,c0 ,81 ,2 ,0 ,79 ,f3

  DLL  Layer :GET from Physical:2019-12-24 3:59:0
5 64 a 44 1 0 3 0 b4 e1 c0 c0 81 2 0 79 f3

  Transport Layer :GET From DLL:2019-12-24 3:59:0
c0 c0 81 2 0
```

When you set Diagnostic to Data link layer , it will log physical and data link layer frames .

When you set Diagnostic to Transport layer , it will log physical , data link and transport layer frames .

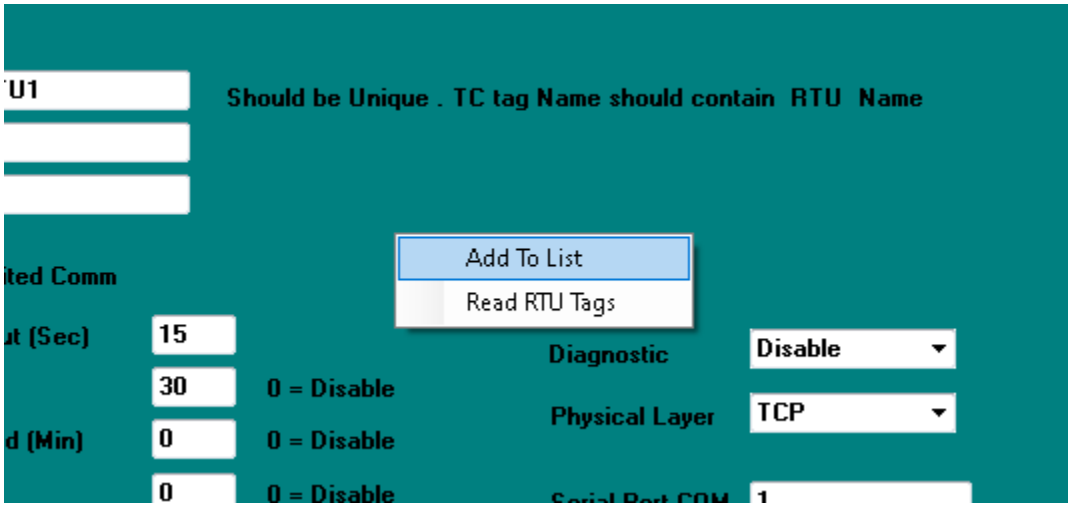
When you set Diagnostic to Application layer , it will log all frames and all tags that is gathered from Application layer frame.

In following sample you can see frames with application layer Diagnostic option:

```
physcial layer :Read  From RTU:2019-12-24 4:21:36
,0 ,0 ,0 ,7 ,0 ,1 ,0 ,0 ,97 ,3c ,0 ,0 ,8 ,0 ,1 ,0 ,0 ,0 ,0 ,0 ,9 ,0 ,1 ,0 ,0 ,0 ,0 ,67 ,b4 ,14 ,1 ,28 ,a ,0 ,1 ,0 ,1 ,0 ,0 ,0 ,0 ,2 ,0 ,1 ,0 ,e8 ,f7 ,0 ,0 ,0 ,3 ,0 ,1 ,0 ,0 ,0

  DLL  Layer :GET from Physical:2019-12-24 4:21:36
0 0 0 7 0 1 0 0 97 3c 0 0 8 0 1 0 0 0 0 0 9 0 1 0 0 0 0 0 67 b4 14 1 28 a 0 1 0 1 0 0 0 0 0 2 0 1 0 e8 f7 0 0 0 3 0 1 0 0 0 0 4 0 1 0 0 0 ba 87 0 5 0 1 0 0 0 0 0 6 0 1 0 0 0 0 0 7 d4

  Transport Layer :GET From DLL:2019-12-24 4:21:36
40 e3 81 0 0 2 2 28 3 0 1 0 81 44 12 9b 38 6f 1 1 0 1 ed 25 9b 38 6f 1 1 0 81 95 39 9b 38 6f 1 1 2 28 1d 0 1 0 81 2 0 1 3 0 1 4 0 1 5 0 1 6 0 1 7 0 1 8 0 1 9 0 1 a 0 1 b 0 1

  Transport Layer :GET From DLL:2019-12-24 4:21:36
81 1 0 0 0 0 4 0 1 0 0 0 0 5 0 1 0 0 0 0 6 0 1 0 0 0 0 7 0 1 0 0 0 0 8 0 1 0 0 0 0 9 0 1 0 0 0 0 14 1 28 a 0 1 0 1 0 0 0 0 2 0 1 0 0 0 0 3 0 1 0 0 0 0 4 0 1 0 0 0 0 5 0 1 0

 App Layer :GET From Transport:2019-12-24 4:21:36
e3 81 0 0 2 2 28 3 0 1 0 81 44 12 9b 38 6f 1 1 0 1 ed 25 9b 38 6f 1 1 0 81 95 39 9b 38 6f 1 1 2 28 1d 0 1 0 81 2 0 1 3 0 1 4 0 1 5 0 1 6 0 1 7 0 1 8 0 1 9 0 1 a 0 1 b 0 1 c

 Transport Layer :GET From APP:2019-12-24 4:21:36
c3 0

 TmpTagType = 1 ,Gr = 2 ,Var=2  ,Add =1 ,Val=1.000000 ,TL=1577202684 ,TLms=484

 TmpTagType = 1 ,Gr = 2 ,Var=2  ,Add =1 ,Val=0.000000 ,TL=1577202689 ,TLms=517

 TmpTagType = 1 ,Gr = 2 ,Var=2  ,Add =1 ,Val=1.000000 ,TL=1577202694 ,TLms=549

 TmpTagType = 1 ,Gr = 1 ,Var=2  ,Add =1 ,Val=1.000000 ,TL=0 ,TLms=0

 TmpTagType = 1 ,Gr = 1 ,Var=2  ,Add =2 ,Val=0.000000 ,TL=0 ,TLms=0

 TmpTagType = 1 ,Gr = 1 ,Var=2  ,Add =3 ,Val=0.000000 ,TL=0 ,TLms=0

 TmpTagType = 1 ,Gr = 1 ,Var=2  ,Add =4 ,Val=0.000000 ,TL=0 ,TLms=0

 TmpTagType = 1 ,Gr = 1 ,Var=2  ,Add =5 ,Val=0.000000 ,TL=0 ,TLms=0
```

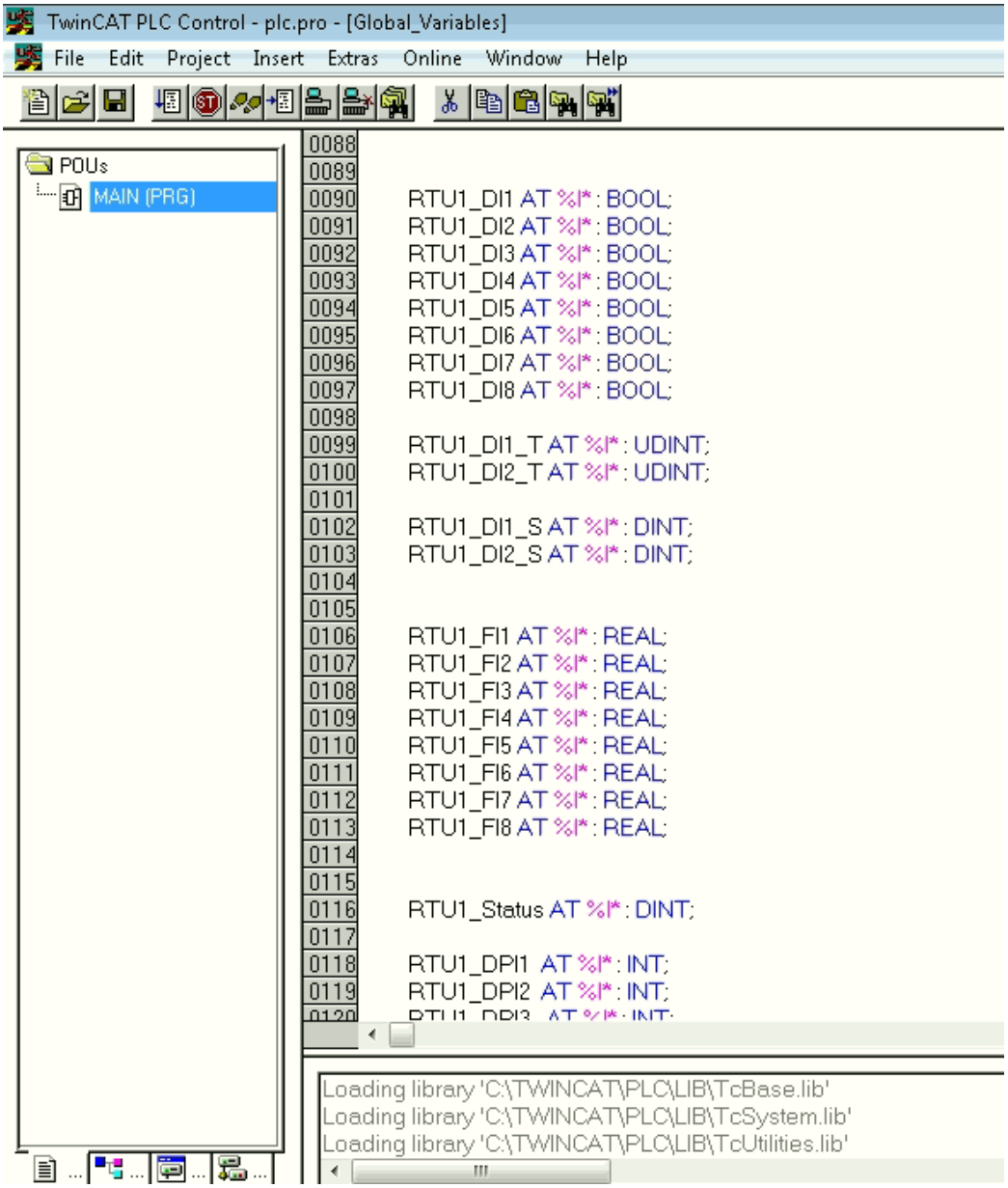
Always set Diagnostic to Disable for normal operation of Master Driver.

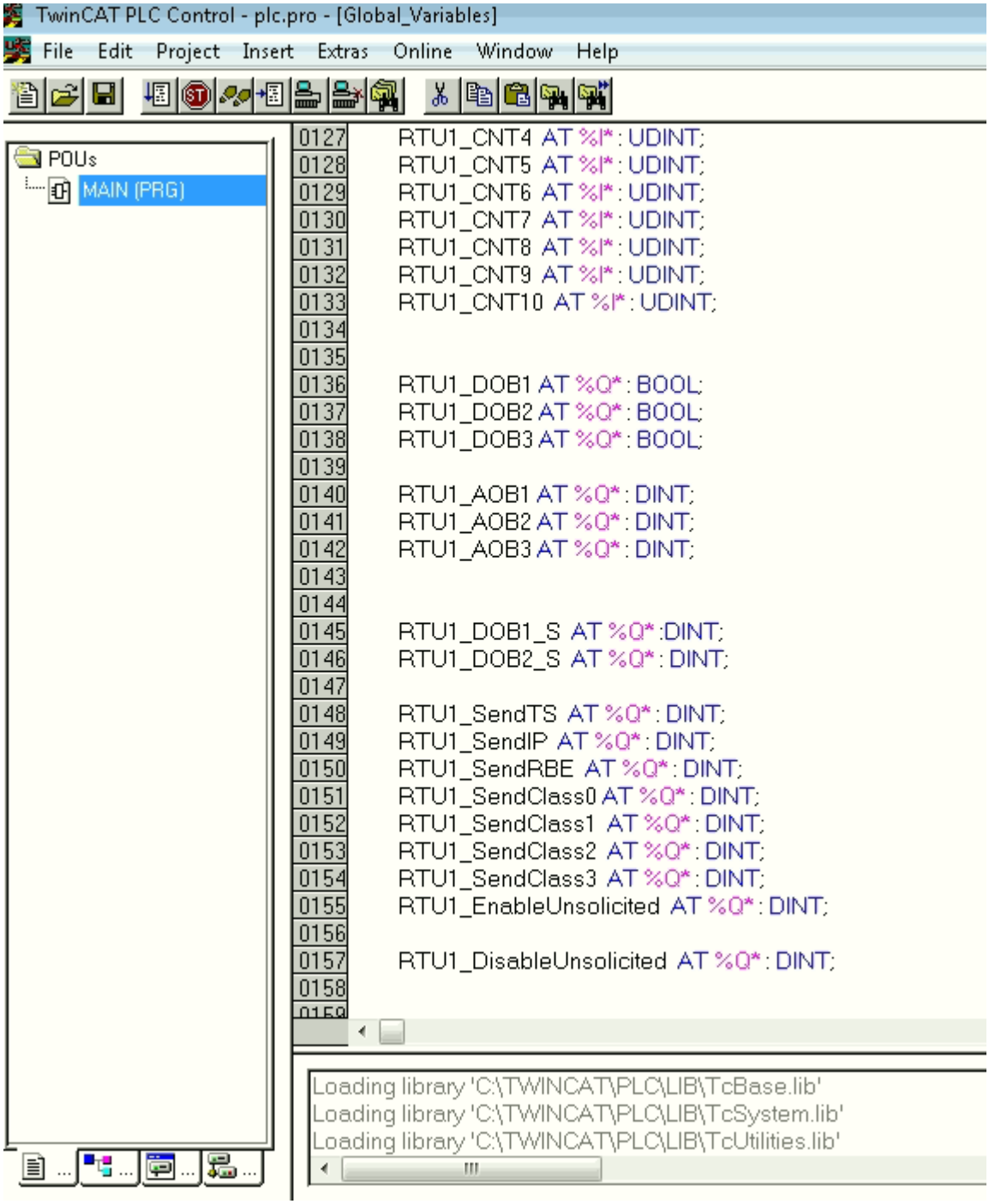
After setting above parameter right click on page and add RTU to list .



After adding RTU to List, if name of RTU is unique , it will add to RTU list . If you want to change above parameter ,directly change on List.

RTU Tags in TC : you should define DNP3 master RTU tags in TC as following Sample :





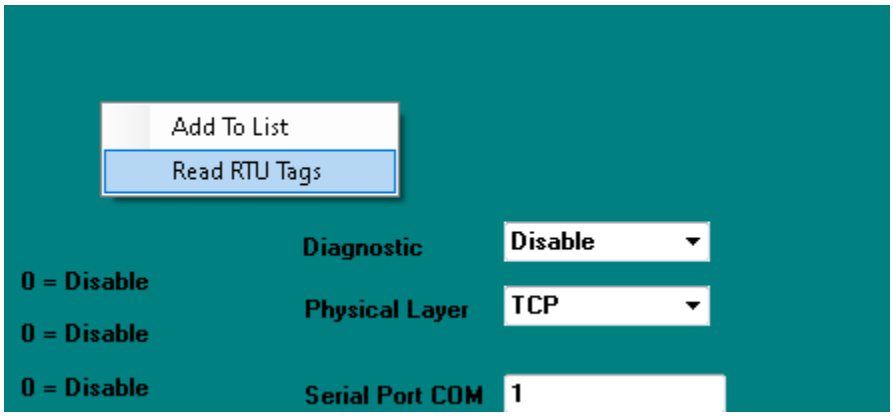

Define DI , AI , FI , DO , AO and DPI Tags as Input Tags to TC .

Define DOB and AOB as Output Tags.

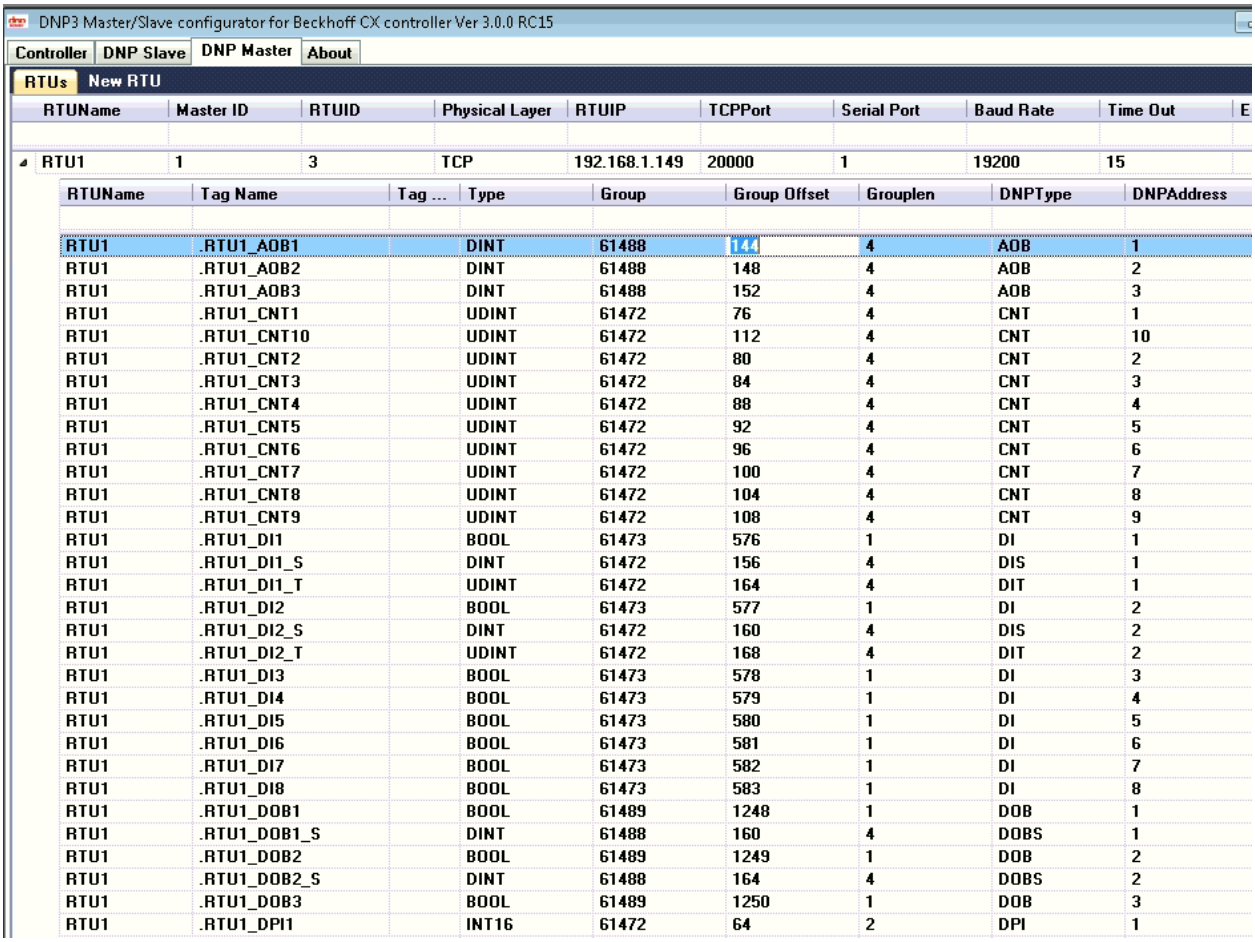
After transferring Tags to CX Controller , you can read tags by DNP3 Configurator .

As you can see , Tag Names for one RTU ,starts with Same name that we put in Configurator for RTU .

For reading Tags in Configurator, go to New RTU Page, Write RTU name, Right Click and Execute “Read RTU Tags”




After Reading Tags , you can see all tags in RTU List .




| RTUName | Master ID | RTUID | Physical Layer | RTUIP | TCPPort | Serial Port | Baud Rate | Time Out |
|---|---|---|---|---|---|---|---|---|
| RTU1 | 1 | 3 | TCP | 192.168.1.149 | 20000 | 1 | 19200 | 15 |

| RTUName | Tag Name | Tag ... | Type | Group | Group Offset | Grouplen | DNPType | DNPAddress |
|---|---|---|---|---|---|---|---|---|
| RTU1 | .RTU1_AOB1 | | DINT | 61488 | 144 | 4 | AOB | 1 |
| RTU1 | .RTU1_AOB2 | | DINT | 61488 | 148 | 4 | AOB | 2 |
| RTU1 | .RTU1_AOB3 | | DINT | 61488 | 152 | 4 | AOB | 3 |
| RTU1 | .RTU1_CNT1 | | UDINT | 61472 | 76 | 4 | CNT | 1 |
| RTU1 | .RTU1_CNT10 | | UDINT | 61472 | 112 | 4 | CNT | 10 |
| RTU1 | .RTU1_CNT2 | | UDINT | 61472 | 80 | 4 | CNT | 2 |
| RTU1 | .RTU1_CNT3 | | UDINT | 61472 | 84 | 4 | CNT | 3 |
| RTU1 | .RTU1_CNT4 | | UDINT | 61472 | 88 | 4 | CNT | 4 |
| RTU1 | .RTU1_CNT5 | | UDINT | 61472 | 92 | 4 | CNT | 5 |
| RTU1 | .RTU1_CNT6 | | UDINT | 61472 | 96 | 4 | CNT | 6 |
| RTU1 | .RTU1_CNT7 | | UDINT | 61472 | 100 | 4 | CNT | 7 |
| RTU1 | .RTU1_CNT8 | | UDINT | 61472 | 104 | 4 | CNT | 8 |
| RTU1 | .RTU1_CNT9 | | UDINT | 61472 | 108 | 4 | CNT | 9 |
| RTU1 | .RTU1_DI1 | | BOOL | 61473 | 576 | 1 | DI | 1 |
| RTU1 | .RTU1_DI1_S | | DINT | 61472 | 156 | 4 | DIS | 1 |
| RTU1 | .RTU1_DI1_T | | UDINT | 61472 | 164 | 4 | DIT | 1 |
| RTU1 | .RTU1_DI2 | | BOOL | 61473 | 577 | 1 | DI | 2 |
| RTU1 | .RTU1_DI2_S | | DINT | 61472 | 160 | 4 | DIS | 2 |
| RTU1 | .RTU1_DI2_T | | UDINT | 61472 | 168 | 4 | DIT | 2 |
| RTU1 | .RTU1_DI3 | | BOOL | 61473 | 578 | 1 | DI | 3 |
| RTU1 | .RTU1_DI4 | | BOOL | 61473 | 579 | 1 | DI | 4 |
| RTU1 | .RTU1_DI5 | | BOOL | 61473 | 580 | 1 | DI | 5 |
| RTU1 | .RTU1_DI6 | | BOOL | 61473 | 581 | 1 | DI | 6 |
| RTU1 | .RTU1_DI7 | | BOOL | 61473 | 582 | 1 | DI | 7 |
| RTU1 | .RTU1_DI8 | | BOOL | 61473 | 583 | 1 | DI | 8 |
| RTU1 | .RTU1_DOB1 | | BOOL | 61489 | 1248 | 1 | DOB | 1 |
| RTU1 | .RTU1_DOB1_S | | DINT | 61488 | 160 | 4 | DOBS | 1 |
| RTU1 | .RTU1_DOB2 | | BOOL | 61489 | 1249 | 1 | DOB | 2 |
| RTU1 | .RTU1_DOB2_S | | DINT | 61488 | 164 | 4 | DOBS | 2 |
| RTU1 | .RTU1_DOB3 | | BOOL | 61489 | 1250 | 1 | DOB | 3 |
| RTU1 | .RTU1_DPI1 | | INT16 | 61472 | 64 | 2 | DPI | 1 |

You should set DNP Type and DNP Address for each tag .

DI = Digital input

AI = Analog Input

FI = Float Input

DPI = Double Bit

DO = Digital Output Status

AO : Analog Output Status

DOB : Digital Output Block ( Write Command )

AOB : Analog Output Block ( Write Command )

System Tags with help of system tags you can manage RTU by DNP3 commands from TC .

SYS : System Tag

- SYS with DNP Address 1 = RTU Connection status
  It is showing Connection Status with Slave RTU (Offline, Online, Connecting,… ) value of this tag is 10 ( connecting ) , 11 ( connected ) , 20 ( Send Time synch ) , 30 ( request data at startup ) and finally 100 shows driver got dnp tags from salve RTU and Driver is online .

- SYS with DNP Address 2 = Send Time Synch Command to RTU
- SYS with DNP Address 3 = Send integrity Poll (IP) to RTU
- SYS with DNP Address 4 = Send RBE to RTU
- SYS with DNP Address 5 = Read tags with Class 0 from RTU
- SYS with DNP Address 6 = Read tags with Class 1 from RTU
- SYS with DNP Address 7 = Read tags with Class 2 from RTU
- SYS with DNP Address 8 = Read tags with Class 3 from RTU
- SYS with DNP Address 9 = Enable Unsolicited communication in RTU
- SYS with DNP Address 10 = Disable Unsolicited communication in RTU

- SYS with DNP Address 11 = Change Default Command Type to Direct Operate
- SYS with DNP Address 12 = Change Default Command Type to Select before Operate
-

SYS tag with Address 1 is read Only , but you should write other SYS tags .

Suppose you want to send Time synch from TC to RTU. Then you should write a pulse with width 5 Sec to SYS.SendTS signal (address 2) Driver will send Time Synch command when it detect rising Edge in System Tag value with Address 2.

Define system tags in TC as following sample :

```
RTU1_SendTS AT %Q* : DINT;
RTU1_SendIP AT %Q* : DINT;
RTU1_SendRBE AT %Q* : DINT;
RTU1_SendClass0 AT %Q* : DINT;
RTU1_SendClass1 AT %Q* : DINT;
RTU1_SendClass2 AT %Q* : DINT;
RTU1_SendClass3 AT %Q* : DINT;
RTU1_EnableUnsolicited AT %Q* : DINT;
RTU1_DisableUnsolicited AT %Q* : DINT;
```

Then link them in DNP Configurator as following:

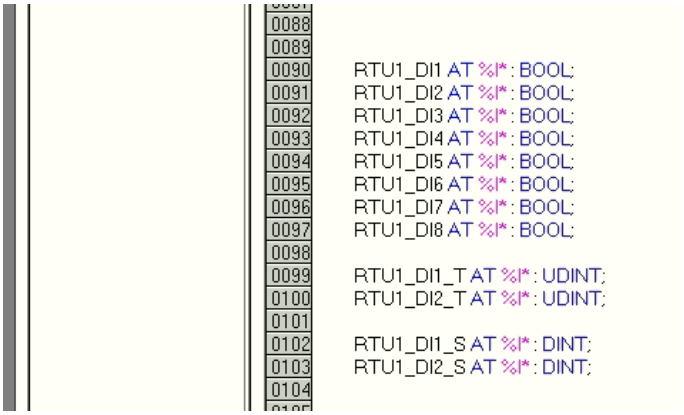
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RTU1 | .RTU1_STATUS | | DINT | 61472 | 60 | 4 | SYS | 1 |
| RTU1 | .RTU1_DISABLEUNSOLICITED | | DINT | 61488 | 200 | 4 | SYS | 10 |
| RTU1 | .RTU1_ENABLEUNSOLICITED | | DINT | 61488 | 196 | 4 | SYS | 9 |
| RTU1 | .RTU1_SENDCLASS0 | | DINT | 61488 | 180 | 4 | SYS | 5 |
| RTU1 | .RTU1_SENDCLASS1 | | DINT | 61488 | 184 | 4 | SYS | 6 |
| RTU1 | .RTU1_SENDCLASS2 | | DINT | 61488 | 188 | 4 | SYS | 7 |
| RTU1 | .RTU1_SENDCLASS3 | | DINT | 61488 | 192 | 4 | SYS | 8 |
| RTU1 | .RTU1_SENDIP | | DINT | 61488 | 172 | 4 | SYS | 3 |
| RTU1 | .RTU1_SENDRBE | | DINT | 61488 | 176 | 4 | SYS | 4 |
| RTU1 | .RTU1_SENDTS | | DINT | 61488 | 168 | 4 | SYS | 2 |

## Defining Tag Flag and Time Label:

You can read DNP Master tags Flag and Time label in TC by {X}S and {X}T Data type ( DIS , AIS , DPIS , FIS ,CNTS, DIT, AIT , DPIT , FIT ,CNTT)

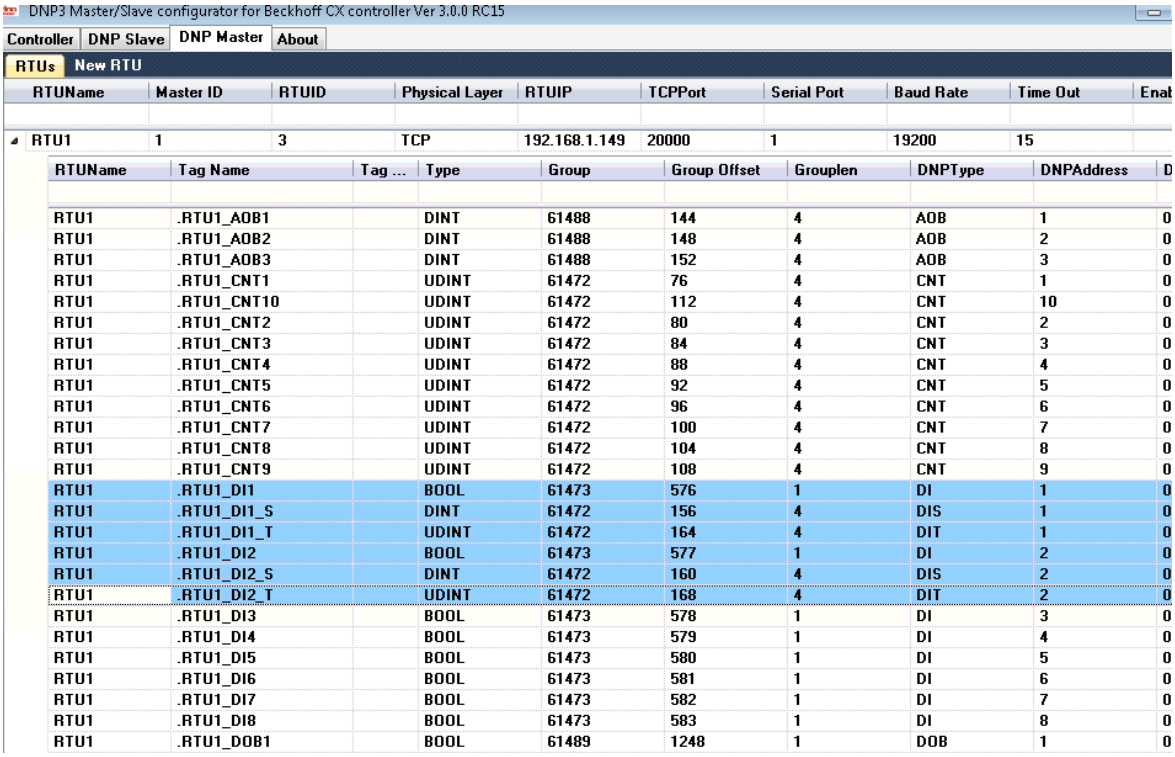
DNP Address of Tag Should be same as {X}S /{X}T Tag. Look at following sample :

In TC we defined following tags :




In configurator we will link same DNP Address for RTU_DI1 ( Signal Value) , RTU_DI1_T ( time Label) and RTU_DI_S ( Flag) signals .

Time Label Tags are defined based on DNP Standard which is Linux epoch Time , msec from 1/1/1970 .




| RTUName | Master ID | RTUID | Physical Layer | RTUIP | TCPPort | Serial Port | Baud Rate | Time Out | Enab |
|---|---|---|---|---|---|---|---|---|---|
| RTU1 | 1 | 3 | TCP | 192.168.1.149 | 20000 | 1 | 19200 | 15 | |

| RTUName | Tag Name | Tag ... | Type | Group | Group Offset | Grouplen | DNPType | DNPAddress | D |
|---|---|---|---|---|---|---|---|---|---|
| RTU1 | .RTU1_AOB1 | | DINT | 61488 | 144 | 4 | AOB | 1 | 0 |
| RTU1 | .RTU1_AOB2 | | DINT | 61488 | 148 | 4 | AOB | 2 | 0 |
| RTU1 | .RTU1_AOB3 | | DINT | 61488 | 152 | 4 | AOB | 3 | 0 |
| RTU1 | .RTU1_CNT1 | | UDINT | 61472 | 76 | 4 | CNT | 1 | 0 |
| RTU1 | .RTU1_CNT10 | | UDINT | 61472 | 112 | 4 | CNT | 10 | 0 |
| RTU1 | .RTU1_CNT2 | | UDINT | 61472 | 80 | 4 | CNT | 2 | 0 |
| RTU1 | .RTU1_CNT3 | | UDINT | 61472 | 84 | 4 | CNT | 3 | 0 |
| RTU1 | .RTU1_CNT4 | | UDINT | 61472 | 88 | 4 | CNT | 4 | 0 |
| RTU1 | .RTU1_CNT5 | | UDINT | 61472 | 92 | 4 | CNT | 5 | 0 |
| RTU1 | .RTU1_CNT6 | | UDINT | 61472 | 96 | 4 | CNT | 6 | 0 |
| RTU1 | .RTU1_CNT7 | | UDINT | 61472 | 100 | 4 | CNT | 7 | 0 |
| RTU1 | .RTU1_CNT8 | | UDINT | 61472 | 104 | 4 | CNT | 8 | 0 |
| RTU1 | .RTU1_CNT9 | | UDINT | 61472 | 108 | 4 | CNT | 9 | 0 |
| RTU1 | .RTU1_DI1 | | BOOL | 61473 | 576 | 1 | DI | 1 | 0 |
| RTU1 | .RTU1_DI1_S | | DINT | 61472 | 156 | 4 | DIS | 1 | 0 |
| RTU1 | .RTU1_DI1_T | | UDINT | 61472 | 164 | 4 | DIT | 1 | 0 |
| RTU1 | .RTU1_DI2 | | BOOL | 61473 | 577 | 1 | DI | 2 | 0 |
| RTU1 | .RTU1_DI2_S | | DINT | 61472 | 160 | 4 | DIS | 2 | 0 |
| RTU1 | .RTU1_DI2_T | | UDINT | 61472 | 168 | 4 | DIT | 2 | 0 |
| RTU1 | .RTU1_DI3 | | BOOL | 61473 | 578 | 1 | DI | 3 | 0 |
| RTU1 | .RTU1_DI4 | | BOOL | 61473 | 579 | 1 | DI | 4 | 0 |
| RTU1 | .RTU1_DI5 | | BOOL | 61473 | 580 | 1 | DI | 5 | 0 |
| RTU1 | .RTU1_DI6 | | BOOL | 61473 | 581 | 1 | DI | 6 | 0 |
| RTU1 | .RTU1_DI7 | | BOOL | 61473 | 582 | 1 | DI | 7 | 0 |
| RTU1 | .RTU1_DI8 | | BOOL | 61473 | 583 | 1 | DI | 8 | 0 |
| RTU1 | .RTU1_DOB1 | | BOOL | 61489 | 1248 | 1 | DOB | 1 | 0 |

# In DNP3 Standard we have following Tag Flags :

# Digital Input

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ← bit position |
|---|---|---|---|---|---|---|---|---|
| ST | 0 | CF | LF | RF | CL | RS | OL | State and flag octet |

### A.3.1.2.2 Formal structure

**BSTR8:** Flag Octet

Bit 0: ONLINE

Bit 1: RESTART

Bit 2: COMM_LOST

Bit 3: REMOTE_FORCED

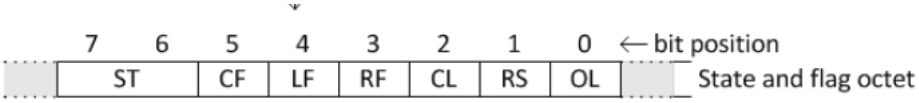
Bit 4: LOCAL_FORCED

Bit 5: CHATTER_FILTER

Bit 6: Reserved, always 0

Bit 7: STATE—Has a value of 0 or 1, representing the state of the physical or logical input.

# Double Bit Binary input :




**A.4.2.2.2 Formal structure**

**BSTR6:** Flags

| | |
|---|---|
| Bit 0: | ONLINE |
| Bit 1: | RESTART |
| Bit 2: | COMM_LOST |
| Bit 3: | REMOTE_FORCED |
| Bit 4: | LOCAL_FORCED |
| Bit 5: | CHATTER_FILTER |

**UINT2:** State.

This integer contains the state of the double-bit binary input as defined for a Double-bit Binary Input Point Type in **11.9.6**. These values represent states INTERMEDIATE, DETERMINED_OFF, DETERMINED_ON, and INDETERMINATE.

# Counters ( 16 Bit and 32 Bit Counter has identical Flag)


**A.10.1.2.2 Formal structure**

**BSTR8:** Flag Octet

| | |
|---|---|
| Bit 0: | ONLINE |
| Bit 1: | RESTART |
| Bit 2: | COMM_LOST |
| Bit 3: | REMOTE_FORCED |
| Bit 4: | LOCAL_FORCED |
| Bit 5: | ROLLOVER |
| Bit 6: | DISCONTINUITY |
| Bit 7: | Reserved, always 0. |

# Analog Input (Different variation has identical flag)


### A.14.1.2.2 Formal structure

**BSTR8:** Flag Octet

| | |
|---|---|
| Bit 0: | ONLINE |
| Bit 1: | RESTART |
| Bit 2: | COMM_LOST |
| Bit 3: | REMOTE_FORCED |
| Bit 4: | LOCAL_FORCED |
| Bit 5: | OVER_RANGE |
| Bit 6: | REFERENCE_ERR |
| Bit 7: | Reserved, always 0. |

## Direct Mapping Between DNP Master and Slave Kernel

If you want to transfer DNP3 Tags from DNP Master Driver with msec time label to DNP3 Slave Kernel then you should not use ADS for mapping Tags.

For this scenario you can directly map Master tags to Slave Kernels.

| 3 | TCP | 192.168.1.106 | 20000 | 1 | 19200 | 15 | ☑ | 20 |
|---|---|---|---|---|---|---|---|---|
| ame | Tag Comment | Type | Group | Group Offset | Grouplen | DNPType | DNPAddress | DNPAddress Slave |
| _AOB1 | | DINT | 61488 | 144 | 4 | | 1 | 0 |
| _AOB2 | | DINT | 61488 | 148 | 4 | | 1 | 0 |
| _AOB3 | | DINT | 61488 | 152 | 4 | | 1 | 0 |
| _CNT1 | | UDINT | 61472 | 76 | 4 | | 1 | 0 |
| _CNT10 | | UDINT | 61472 | 112 | 4 | | 1 | 0 |
| _CNT2 | | UDINT | 61472 | 80 | 4 | | 1 | 0 |
| _CNT3 | | UDINT | 61472 | 84 | 4 | | 1 | 0 |
| _CNT4 | | UDINT | 61472 | 88 | 4 | | 1 | 0 |
| _CNT5 | | UDINT | 61472 | 92 | 4 | | 1 | 0 |
| _CNT6 | | UDINT | 61472 | 96 | 4 | | 1 | 0 |
| _CNT7 | | UDINT | 61472 | 100 | 4 | | 1 | 0 |
| _CNT8 | | UDINT | 61472 | 104 | 4 | | 1 | 0 |
| _CNT9 | | UDINT | 61472 | 108 | 4 | | 1 | 0 |
| _DI1 | | BOOL | 61473 | 576 | 1 | DI | 1 | 1 |
| _DI1_S | | DINT | 61472 | 156 | 4 | DIS | 1 | 0 |
| _DI1_T | | UDINT | 61472 | 164 | 4 | DIT | 1 | 0 |
| _DI2 | | BOOL | 61473 | 577 | 1 | DI | 2 | 2 |
| _DI2_S | | DINT | 61472 | 160 | 4 | DIS | 2 | 0 |
| _DI2_T | | UDINT | 61472 | 168 | 4 | DIT | 2 | 0 |
| _DI3 | | BOOL | 61473 | 578 | 1 | DI | 3 | 0 |
| _DI4 | | BOOL | 61473 | 579 | 1 | DI | 4 | 0 |
| _DI5 | | BOOL | 61473 | 580 | 1 | | 1 | 0 |

When you define DNP Master tags in Configurator , you can make direct link between dnp master and slave tags by DNPAddressSlave address .

In above example we mapped DI Tag with address 1 ( Master Side) to DNP3 Slave tag with same Address 1 .

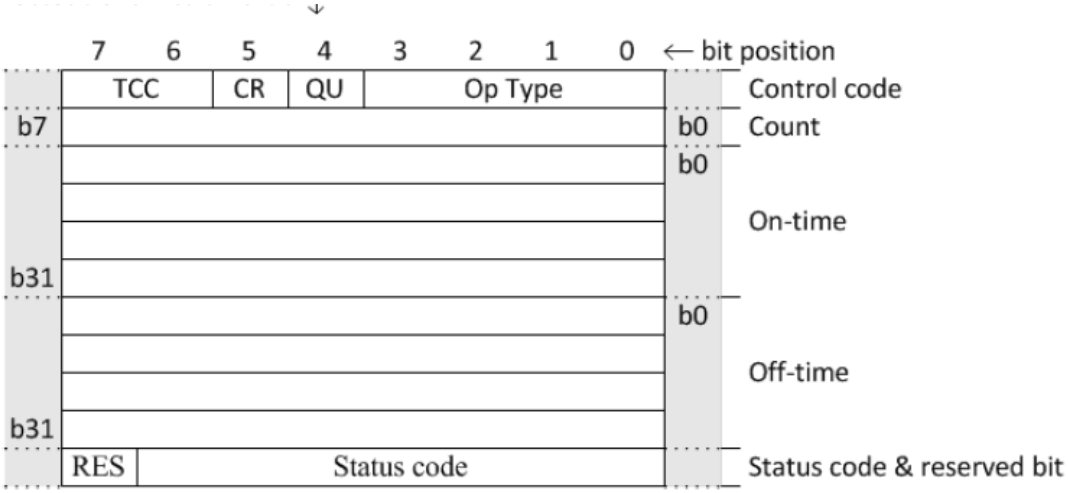
Then all SOE and changes for this tag will map directly to Slave Kernel .

There is no Direct map from Slave kernel to Master Kernel.

## DOBS Data Type

By help of DOBS datatype you can write with different configuration from TC to Slave RTU.

First will look at DNP3 Digital output Block Data Type from Standard:




**UINT4:** Control code, Operation Type field [Op Type].

This field is used in conjunction with the *TCC* field to specify a control operation. See operational functions (**A.8.1.3.2**) for additional details. The code names are:

0: NUL

1: PULSE_ON

2: PULSE_OFF

3: LATCH_ON

4: LATCH_OFF

5 to 15: Undefined

**BSTR1:** Control code, Queue field [QU].

This field is obsolete. The master shall always set this bit to 0. Outstations that receive a g12v1 object with this bit set shall return a status code NOT_SUPPORTED in the response.

**BSTR1:** Control code, Clear field [CR].

Support for this field is optional. If the device supports commands with the clear bit set:

— It shall support the case where the Op Type code is NUL, and may support the command for other Op Type values.

— When the clear bit is set, the device shall remove pending control commands for that index and stop any control operation that is in progress for that index. The indexed point shall go to the state that it would have if the command were allowed to complete normally. If a non-NUL Op Type code is requested, the new command shall be initiated immediately after the cancellation actions complete.

When the clear bit is set and the *TCC* - *Op Type* combination is not supported, the device shall return status code NOT_SUPPORTED in the response.

**UINT2:** Control code, Trip-Close Code field [TCC].

This field is used in conjunction with the *Op Type* field to specify a control operation. See operational functions for additional details. The code names are

0: NUL

1: CLOSE

2: TRIP

3: RESERVED

**UINT8:** Count.

This is the number of times the outstation shall execute the operation. Counts greater than 1 generate a series of pulses or repeated operations for the point. Both *On-time* and *Off-time* values are obeyed as illustrated in the figures under timing illustrations, subject to the comments regarding timing in interpreting the time fields.

Implementation of a zero-count functionality is optional. A count value of 0 indicates that the output operation shall not be executed. Setting the count value to 0 is a useful technique for testing communications without affecting an output. When the outstation receives a 0 value, it shall:

— Not change the output.

— Ignore the On-time and Off-time values.

— Return the same status code as if the execution had been attempted.

An outstation shall return status code NOT_SUPPORTED in the response when the count value is 0 in the request and the outstation does not implement the zero-count functionality.

**UINT32:** On-time.

This is the duration, expressed as the number of milliseconds, that the output drive remains active. See interpreting the time fields for more details.

**UINT32:** Off-time.

This is the duration, expressed as the number of milliseconds that the output drive remains non-active. See interpreting the time fields for more details.

**UINT7:** Status code.

This value shall be set to 0 in request messages.

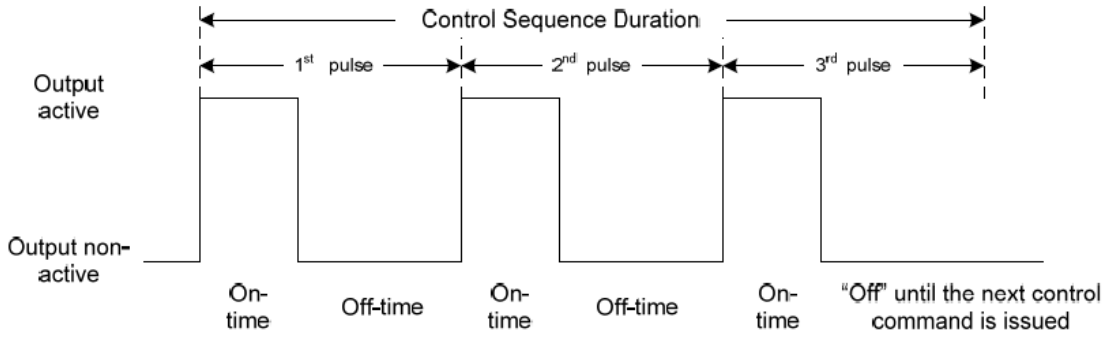
In response messages, this value represents the status of the selected or executed command. See **Table 11-7** for descriptions of control-related status codes.

**BSTR1:** Reserved [RES].

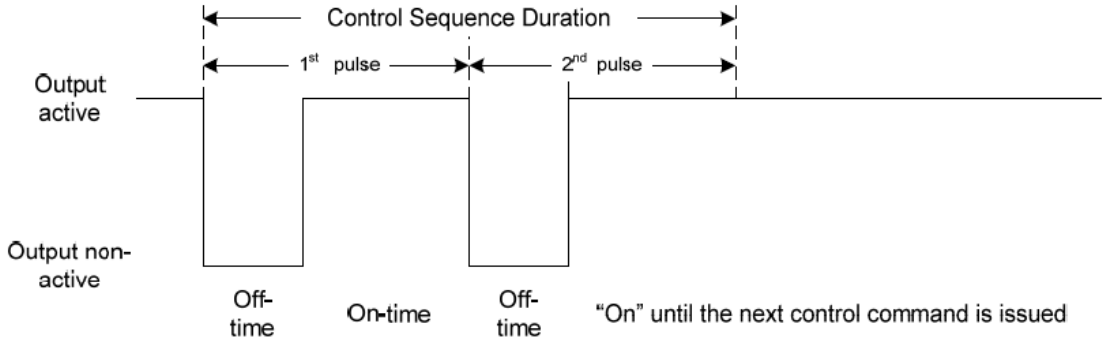
This bit is reserved. The master and outstation shall always set it to 0.

#### A.8.1.3.1 Timing illustrations

This subclause illustrates just two of the many timing possibilities that can appear in a g12v1 object.




**PULSE ON with Count = 3**




**PULSE OFF with Count = 2 (Not interoperable)**

#### A.8.1.3.2 Operational functions

#### A.8.1.3.2.1 Interoperable commands

Only a few of the 64 possible *TCC* and *Op Type* field bit combinations are interoperable. **Table A-1** indicates, for each point index, which commands are optional, preferred, not permitted, or not interoperable, depending on which point model is implemented.

## Table A-1—Interoperable control commands

| TCC field | Op Type field | Point model in outstation | Support requirements |
|---|---|---|---|
| NUL | NUL | All | Optional |
| NUL | PULSE_ON | Activation | May support |
| | | Complementary latch | Not permitted |
| | | Complementary two-output | Not permitted |
| NUL | LATCH_ON | Activation | May support |
| | | Complementary latch | Preferred support |
| | | Complementary two-output | May support |
| NUL | LATCH_OFF | Activation | May support |
| | | Complementary latch | Preferred support |
| | | Complementary two-output | May support |
| CLOSE | PULSE_ON | Activation | May support |
| | | Complementary latch | May support |
| | | Complementary two-output | Preferred support |
| TRIP | PULSE_ON | Activation | May support |
| | | Complementary latch | May support |
| | | Complementary two-output | Preferred support |
| All other combinations | | All | Not interoperable |

Table A-2 indicates for a single point index, what action the outstation performs based on the contents of the interoperable control codes and the point model implemented.

## Table A-2—Actions performed by outstation for interoperable commands

| Row # | Control code | TCC field | Op Type field | Clear field | Action |
|---|---|---|---|---|---|
| 1 | 0x00 | NUL | NUL | 0 | Does not initiate an action or change an in-progress or pending command. Values in *On-time* and *Off-time* fields are ignored. |
| 2 | 0x20 | NUL | NUL | 1 | Cancel in-progress and pending commands. Values in *On-time* and *Off-time* fields are ignored. |
| 3 | 0x01 | NUL | PULSE_ON | 0 | For activation model, set output to active for the duration of the *On-time*. For both complementary models, return NOT_SUPPORTED status. |

508

| Row # | Control code | TCC field | Op Type field | Clear field | Action |
|---|---|---|---|---|---|
| 4 | 0x21 | NUL | PULSE_ON | 1 | For activation model, cancel in-progress and pending commands and then set output to active for the duration of the *On-time*. For both complementary models, return NOT_SUPPORTED status. |
| 5 | 0x03 | NUL | LATCH_ON | 0 | For activation model, set output to active for the duration of the *On-time*. For complementary latch model, set the output to active. For complementary two-output model, set the close output to active for the duration of the *On-time*. |
| 6 | 0x23 | NUL | LATCH_ON | 1 | Cancel in-progress and pending commands. Afterwards, initiate the action specified in row 5. |
| 7 | 0x04 | NUL | LATCH_OFF | 0 | For activation model, set output to active for the duration of the *On-time*. For complementary latch model, set the output to inactive. For complementary two-output model, set the trip output to active for the duration of the *On-time*. |
| 8 | 0x24 | NUL | LATCH_OFF | 1 | Cancel in-progress and pending commands. Afterwards, initiate the action specified in row 7. |
| 9 | 0x41 | CLOSE | PULSE_ON | 0 | For activation model, set output to active for the duration of the *On-time*. For complementary latch model, set the output to active. For complementary two-output model, set the close output to active for the duration of the *On-time*. |
| 10 | 0x61 | CLOSE | PULSE_ON | 1 | Cancel in-progress and pending commands. Afterwards, initiate the action specified in row 9. |
| 11 | 0x81 | TRIP | PULSE_ON | 0 | For activation model, set output to active for the duration of the *On-time*. For complementary latch model, set the output to inactive. For complementary two-output model, set the trip output to active for the duration of the *On-time*. |
| 12 | 0xA1 | TRIP | PULSE_ON | 1 | Cancel in-progress and pending commands. Afterwards, initiate the action specified in row 11. |

DOBS is 32 bit value which is mapped to {control Code |count|OnTime|OffTim} fields.

DOBS Byte 1 = Control Code

DOBS Byte 2 = Count

DOBS Byte 3 = OnTime in Sec

DOBS Byte 4 = OffTime in Sec

In TC you need to merge 4 Bytes and make DOBS value. When Value of DOBS is changing, DNP3 Kernel will get then changes and Saves value of DOBS Tag.

When you change DOB Tag value from 0 to 1 ,( DOB Tag that has same address as DOBS ) , DNP Driver sends G12V1 With specified Flag that is saved in DOBS Tag to RTU .

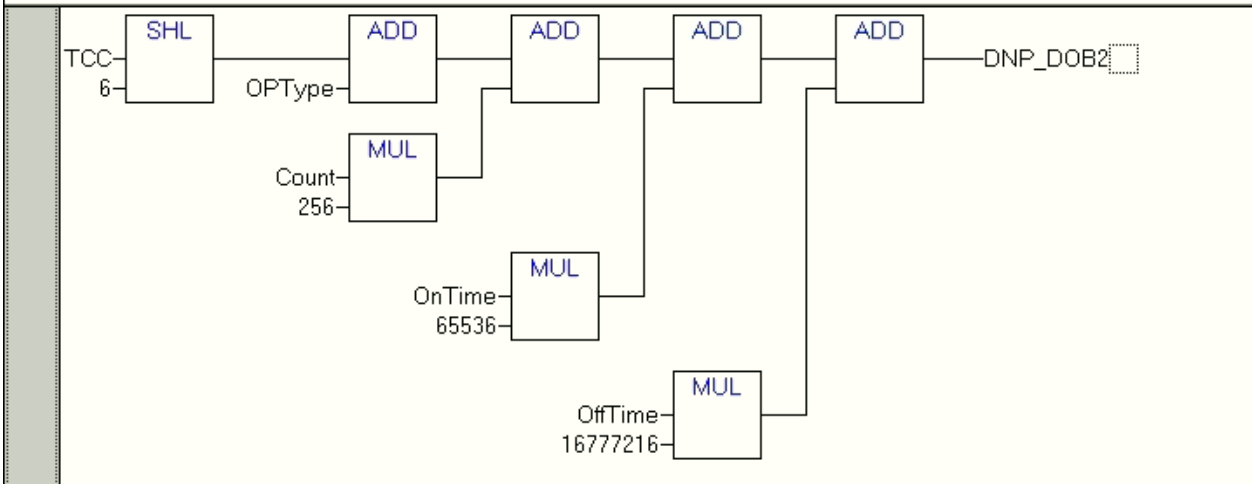
If DOBS Tag is not defined for DOB Tag , Then default operation ( Direct Operate or Select Before Operate ) Will execute when DOB Tag Value change from 0 to 1 .

Sequence of Sending DOB Command with defined DOBS Tag:

- Set Value of DOBS Tag and Write to Kernel
- Change DOB Tag with same address of DOBS from 0 to 1.

In TC with help of following function you can make DOBS tag value:

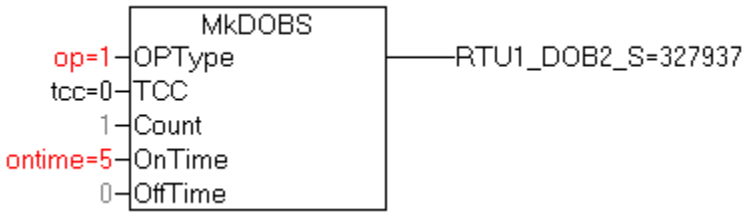
```
0001 FUNCTION DNP_DOB2 : UDINT
0002 VAR_INPUT
0003  OPType :BYTE;
0004       TCC : BYTE;
0005      Count : BYTE;
0006      OnTime:BYTE;
0007      OffTime:BYTE;
0008
0009 END_VAR
0010 VAR
0011 END_VAR
0012
0013
0014
```




DOBS Samples :

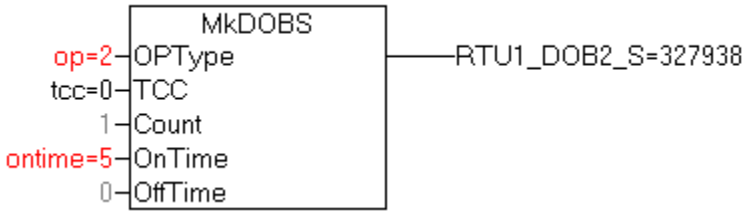
OpType = 1 (Pulse On),Count = 1 , On Time = 5 Sec , off Time = 0 , TCC = 0(Nul)

DOBS Value = 327937



OpType = 2 (Pulse Off),Count = 1 , On Time = 5 Sec , off Time = 0 , TCC = 0(Nul)

DOBS Value = 327938




OpType = 3 (Latch On),Count = 0 , On Time = 0 Sec , off Time = 0 , TCC = 0(Nul)

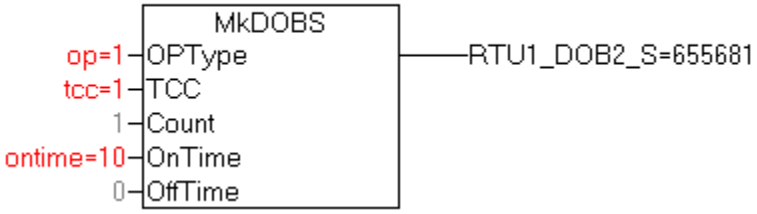
DOBS Value = 3

OpType = 4 (Latch Off),Count = 0 , On Time = 0 Sec , off Time = 0 , TCC = 0(Nul)

DOBS Value = 4

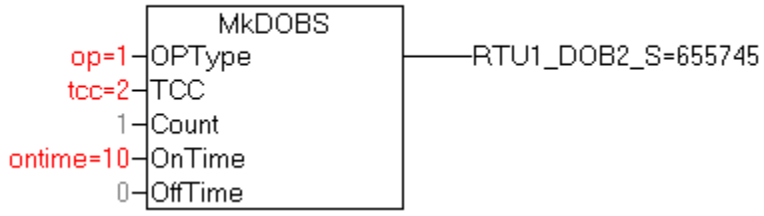
OpType = 1 (Pulse On),Count = 1 , On Time = 10 Sec , off Time = 0 , TCC = 1(Close)

DOBS Value = 655681



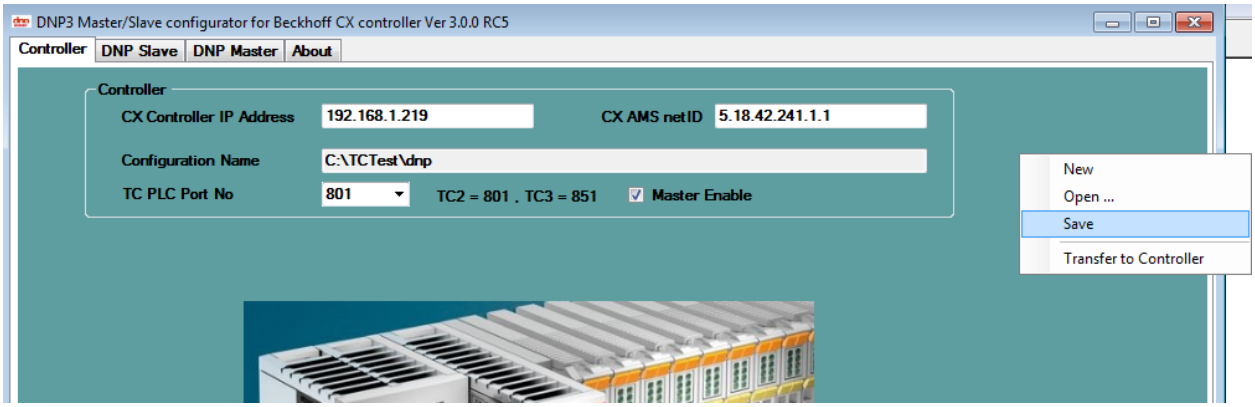
OpType = 1 (Pulse On),Count = 1 , On Time = 10 Sec , off Time = 0 , TCC = 2(Trip)

DOBS Value = 655745




## Saving configuration and transfer to Controller

After setting Tag types and Address , save configuration from Controller page . Right click on Controller page and Execute Save Item .

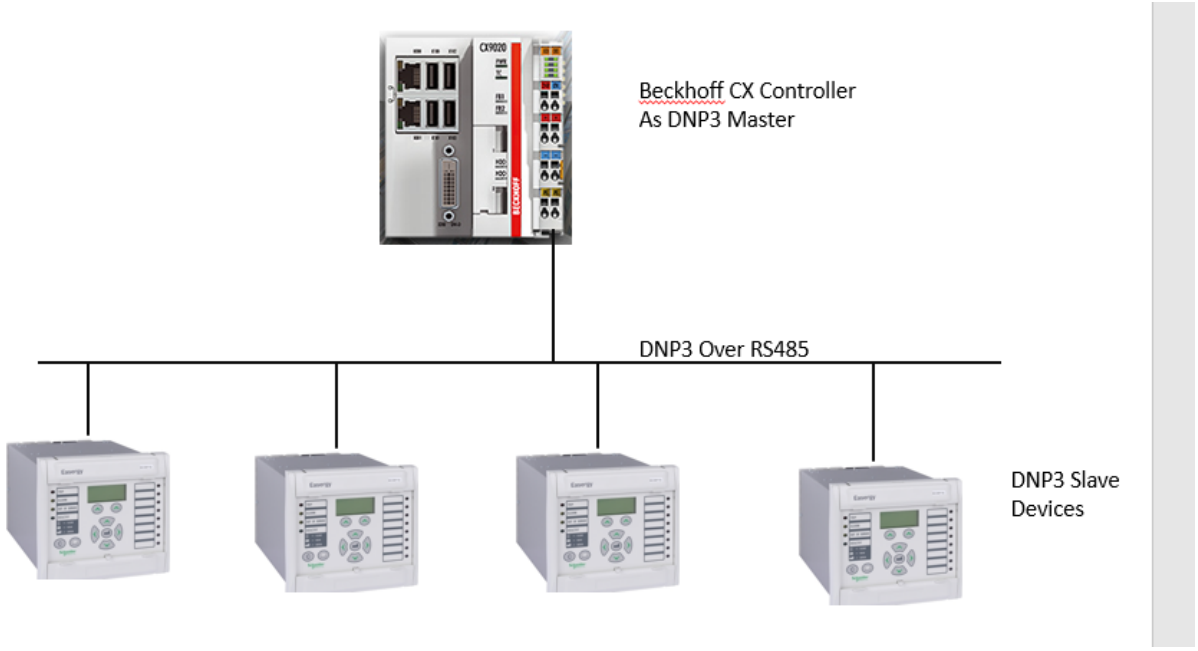



Transfer saved configuration to CX Controller by executing “Transfer to Controller” Item.

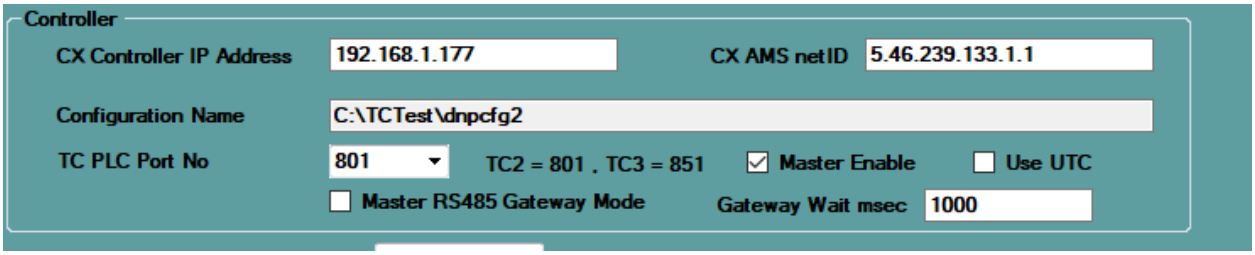
## Master RS485 Gateway Mode

If you want to connect CX controller to Slave Devices like protection relays by RS485 network and using multi drop connection, then you need to enable RS485 Gateway Mode.

Suppose you want to setup following configuration:

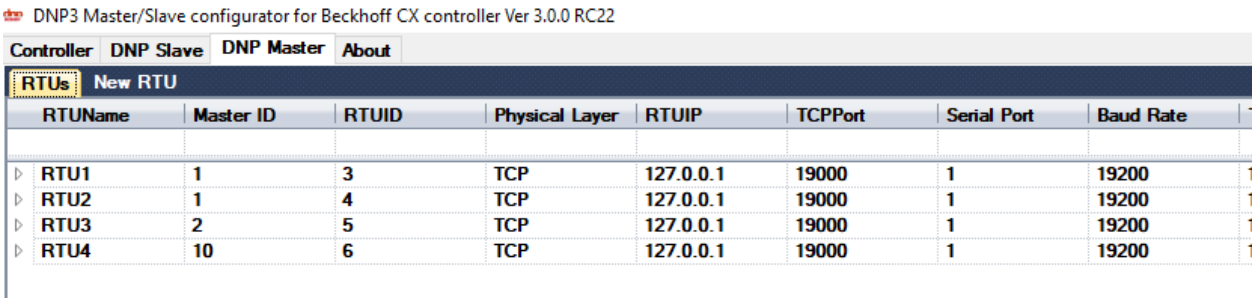



At controller page enable Master and RS485 Gateway Mode :




Add all RTUs as TCP RTU . Select an internal port Number for communication like 19000 and set RTU IP to 127.0.0.1 for all Slave RTUS.

Important Note: In Multi drop Mode Disable Unsolicited communication .




| RTUName | Master ID | RTUID | Physical Layer | RTUIP | TCPPort | Serial Port | Baud Rate |
|---|---|---|---|---|---|---|---|
| RTU1 | 1 | 3 | TCP | 127.0.0.1 | 19000 | 1 | 19200 |
| RTU2 | 1 | 4 | TCP | 127.0.0.1 | 19000 | 1 | 19200 |
| RTU3 | 2 | 5 | TCP | 127.0.0.1 | 19000 | 1 | 19200 |
| RTU4 | 10 | 6 | TCP | 127.0.0.1 | 19000 | 1 | 19200 |

In driver folder you can see pbsDnp2Rs485Gw.dll

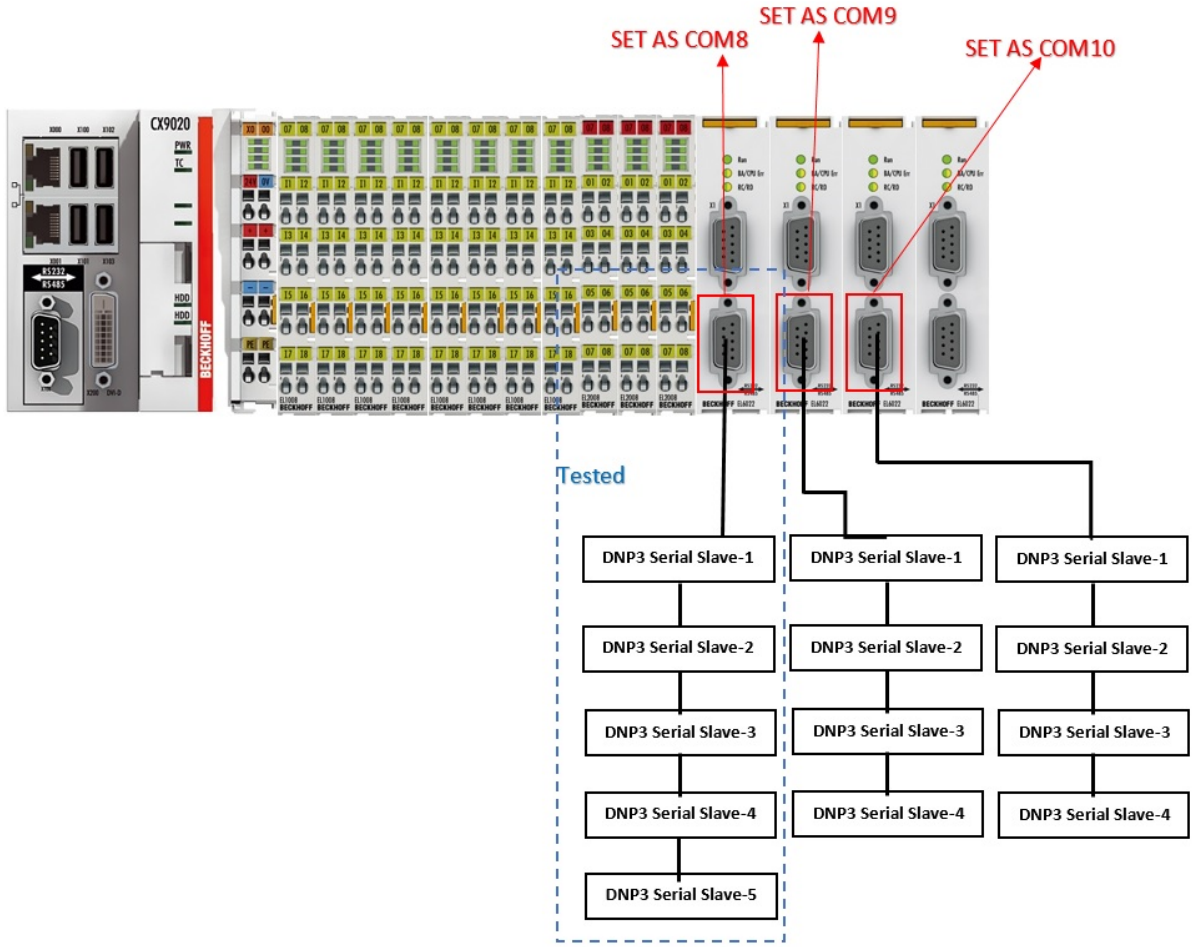
When you enable RS485 gateway Mode , above pbsDnp2Rs485Gw .dll will load automatically . This Dll is responsible for converting TCP to RS485 communication . pbsDnp2Rs485Gw is listening to Port TCPPort ( you need to set same port for all Multi drop RTUs in one network like 19000 in above configuration ) and all RTUS should have IP 127.0.0.1

Serial port and Baudrate should be same for all RTUS in one network . All other setting are same as TCP RTU connection .

Gateway Wait msec : because Gateway Mode is Master /Slave and every request is started by Master and Slave devices should answer to Master , so Slave device answer should not be In multi-Application fragment . Buffer size for serial communication is set to 2048 Bytes in pbsDnp2Rs485Gw.dll so your device answer should not be more than 2048 bytes for any Master request . Gateway Wait time is a time that driver is waiting for getting answer from Serial Slave Devices .

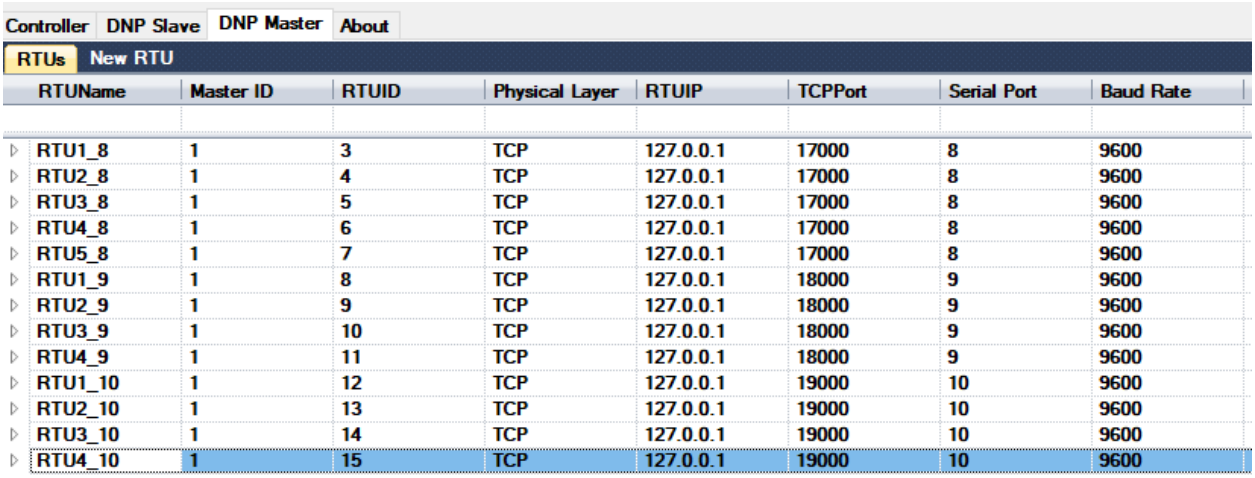
If your devices status are 31 in Master controller and not moving to 100 which is online , you need to increase Gateway Wait time .

Suppose you want to configure following system :




Note : all RTUS that connect to one RS485 Port , Should have same internal TCP Port for communication and all RTU should have IP 127.0.0.1

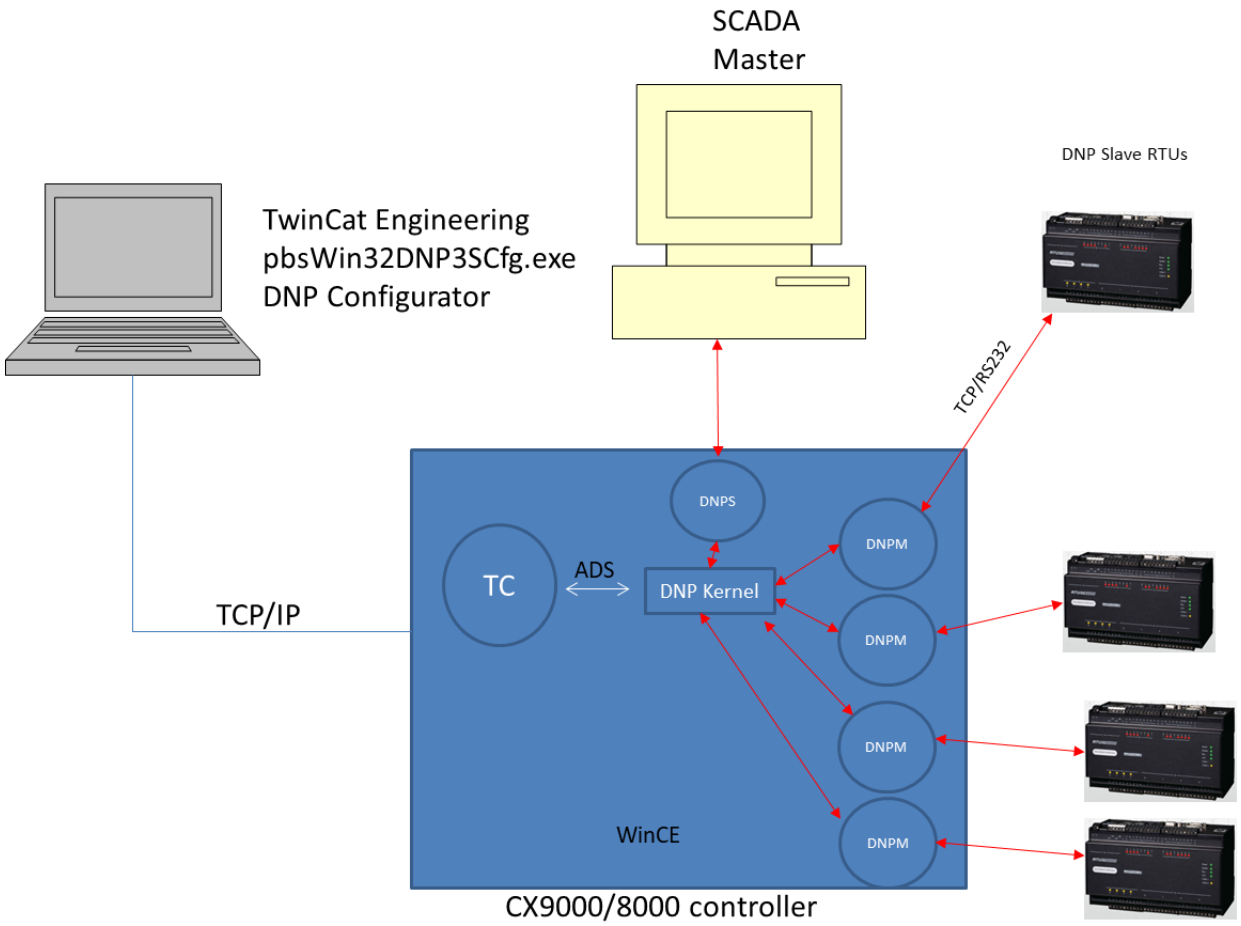
In following figure you can see configuration file for above setup . TCP port 17000 Is set for COM8 , 18000 is set for COM9 and 19000 is set for COM10 .




| RTUName | Master ID | RTUID | Physical Layer | RTUIP | TCPPort | Serial Port | Baud Rate |
|---|---|---|---|---|---|---|---|
| RTU1_8 | 1 | 3 | TCP | 127.0.0.1 | 17000 | 8 | 9600 |
| RTU2_8 | 1 | 4 | TCP | 127.0.0.1 | 17000 | 8 | 9600 |
| RTU3_8 | 1 | 5 | TCP | 127.0.0.1 | 17000 | 8 | 9600 |
| RTU4_8 | 1 | 6 | TCP | 127.0.0.1 | 17000 | 8 | 9600 |
| RTU5_8 | 1 | 7 | TCP | 127.0.0.1 | 17000 | 8 | 9600 |
| RTU1_9 | 1 | 8 | TCP | 127.0.0.1 | 18000 | 9 | 9600 |
| RTU2_9 | 1 | 9 | TCP | 127.0.0.1 | 18000 | 9 | 9600 |
| RTU3_9 | 1 | 10 | TCP | 127.0.0.1 | 18000 | 9 | 9600 |
| RTU4_9 | 1 | 11 | TCP | 127.0.0.1 | 18000 | 9 | 9600 |
| RTU1_10 | 1 | 12 | TCP | 127.0.0.1 | 19000 | 10 | 9600 |
| RTU2_10 | 1 | 13 | TCP | 127.0.0.1 | 19000 | 10 | 9600 |
| RTU3_10 | 1 | 14 | TCP | 127.0.0.1 | 19000 | 10 | 9600 |
| RTU4_10 | 1 | 15 | TCP | 127.0.0.1 | 19000 | 10 | 9600 |

You can define maximum 8 separate RS485 network and at each Network 10 RTU can be defined.

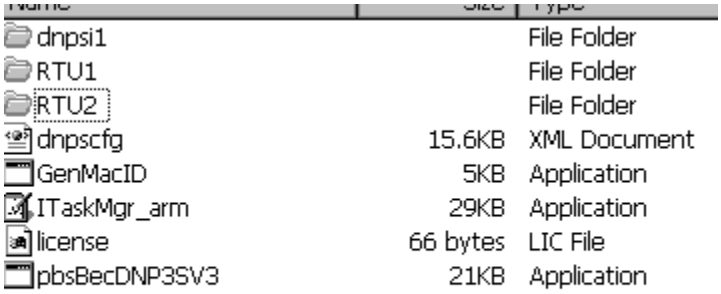
# 7 -DNP Master Operation




When DNP Kernel (pbsBecDnp3SV3.exe) is booting , it will read Configuration from dnpscfg.xml file . DNP3 Master /Slave configuration is in dnpscfg.xml.

DNP kernel will load DNP Master Library for each RTU that is defined in configuration . Suppose you define 4 RTU as above figure . then DNP Kernel will load 4 Separate DNP Master Library to communicate with RTUS.

If communication is RS232 , then you can link one COM port of CX to one RTU .




You should have one RTUX folder in runtime folder (inside CX Controller) for each RTU . means if you define 10 DNP Slave RTU , then you should have RTU1 , RTU2,… RTU10 folder in Main DNP Folder . inside RTUX folder you should have pbsDNPMCeiX.dll library . in above figure DNP Master Library (DNPM) is equal to pbsDNPMCeiX.dll library .

DNP Kernel ( pbsBecDNP3SV3.exe ) is reading Input Tags from pbsDNPMCeiX.dll library and Write User commands to them .

DNP Master License has different version based on Number of DNP Slave RTUS that you want to connect to CX Controller .

DNP Master Library Sequence :

When DNP Master Library is loading to communicate with one RTU , It will do following sequence :

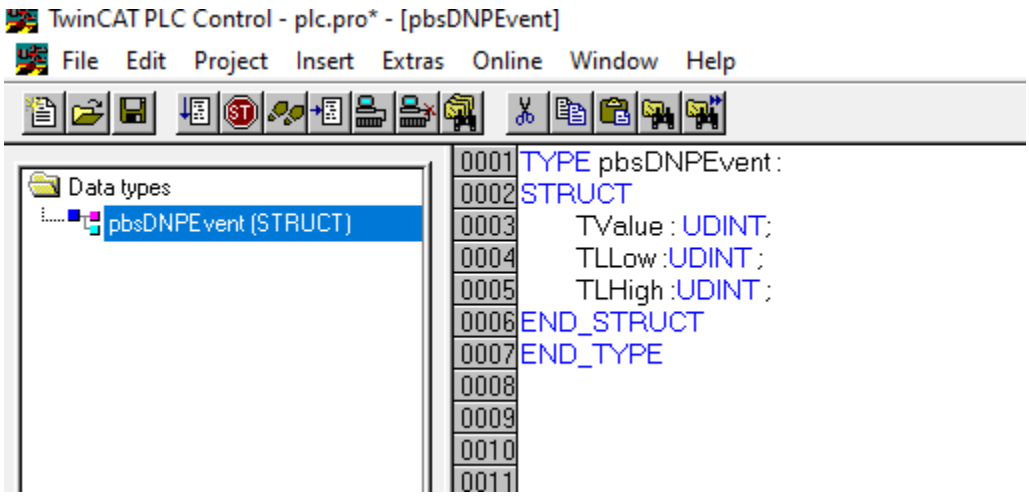
( TCP Mode)

- Try to Connect to RTU
- If connect , It will send Time Synch to RTU
- Send IP ( Integrity Poll ) command to RTU and read all status of Tags and buffers from RTU
- If Enable Unsolicited is enabled : Send Enable Unsolicited command to RTU
- If Enable Unsolicited is Disable : Send Disable Unsolicited command to RTU
- If UNS is Disable , Send RBE based on Configuration periodically to RTU and read Changes from RTU .
- Send IP to RTU to read all tags periodically .
- Send Time Synch Periodically.
- If there is no communication with RTU , Send Link Status to check connection status
- If communication is disconnected , Try to connect again to RTU .

## 8 – msec Digital Input Gathering from ADS

From V3 RC20 you can transfer Digital Input Buffers with msec time resolution to DNP Kernel.

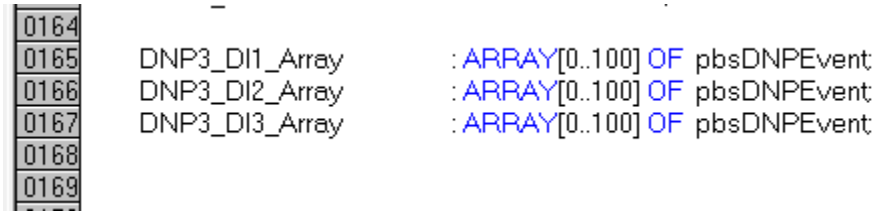
1- Define pbsDNPEvent as following :



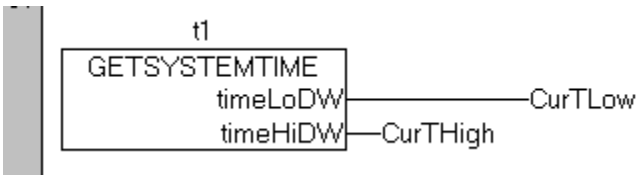

TValue : Tag Value . 0 or 1

TLLOW , TLLHigh : Low and High Value of Time Label .

2 – Define one Array for each DNP DI tag that you want to transfer its event to DNP Kernel by ADS. Suppose you have 3 DI Tags with DNP Address 1 , 2, and 3 then define 3 Array as following in TC :




3 – Use GETSYSTEMTIME FB in TC and get current time label for your Tag and fill above Arrays.

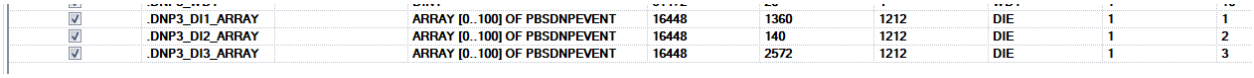



4 – Index 0 of Arrays is used for synchronizing between ADS and DNP Kernel. Do not use index 0 for event.

Index 0, TagValue shows number of Events in Array

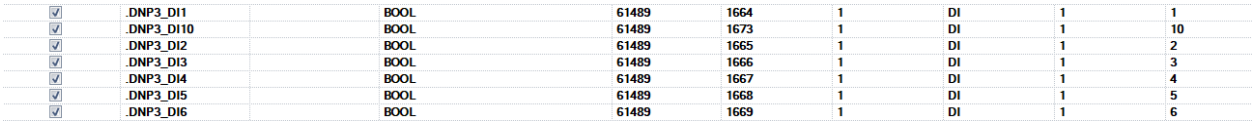
Index 0, TLHigh is a Seq Number . Every time you want to transfer buffer to DNP Kernel increase it by 1. When you reach to 32000 then again set it to 0.

5 – in DNP Configurator you can see following tags :



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| .DNP3_DI1_ARRAY | ARRAY [0..100] OF PBSDNPEVENT | 16448 | 1360 | 1212 | DIE | 1 | 1 |
| .DNP3_DI2_ARRAY | ARRAY [0..100] OF PBSDNPEVENT | 16448 | 140 | 1212 | DIE | 1 | 2 |
| .DNP3_DI3_ARRAY | ARRAY [0..100] OF PBSDNPEVENT | 16448 | 2572 | 1212 | DIE | 1 | 3 |

Set DNP Type for Arrays to DIE .please notice that you should Defined before DI tags with DNP address 1, 2, 3, in TC and configurator.



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| .DNP3_DI1 | BOOL | 61489 | 1664 | 1 | DI | 1 | 1 |
| .DNP3_DI10 | BOOL | 61489 | 1673 | 1 | DI | 1 | 10 |
| .DNP3_DI2 | BOOL | 61489 | 1665 | 1 | DI | 1 | 2 |
| .DNP3_DI3 | BOOL | 61489 | 1666 | 1 | DI | 1 | 3 |
| .DNP3_DI4 | BOOL | 61489 | 1667 | 1 | DI | 1 | 4 |
| .DNP3_DI5 | BOOL | 61489 | 1668 | 1 | DI | 1 | 5 |
| .DNP3_DI6 | BOOL | 61489 | 1669 | 1 | DI | 1 | 6 |

Set Class and DNP Address of DIE tags same as main DI Tags.

## 9 – DNP Master Specification

- Following DNP Functions are not Supported in V3 RC5 : File Transfer , Group 0 , Data Set , Virtual Terminal
- RS485 Mode is not supported. ( no Collision Avoidance )

- Receiver Inter character Timeout : Not Checked
- UDP Datagram is not supported
- Confirmation in Data Link Layer is always Off .
- Group 12 (DOB) and 41 (AOB) Tags are sending one tag Per frame . Multiple Command in one Frame is not supported.
- Supported Qualifier :
    - Prefix 0 Range 0,1,2
    - Prefix 1 Range 7,8,9
    - Prefix 2 Range 7,8,9
    - Prefix 3 Range 7,8,9
- Supported Groups :
    - DI Group 1 , Variation 1,2,3
    - DI Group 2 : Variation 1 ,2
    - DPI Group 3 : Variation 2
    - DPI Group 4 : Variation 1 , 2
    - DO Group 10 : Variation 1,2
    - DO Group 11 : Variation 1 , 2
    - DOB Group 12 : Variation 1
    - CNT Group 20 : Variation 1 , 2, 5,6
    - CNT Group 21 : Variation 1,2,5,6,9,10
    - CNT Group 22 : Variation 1 , 2, 5, 6
    - CNT Group 23 : Variation 1 , 2, 5, 6
    - AI Group 30 : Variation 1 , 2, 3, 4, 5, 6
    - AI Group 32 : Variation 1,2,3,4,5,6,7,8

- AO Group40 : Variation 1,2,3,4
- AOB Group 41 : Variation 1
- AO Group 42 : Variation 1,2,3,4,5,6,7,8

EOD