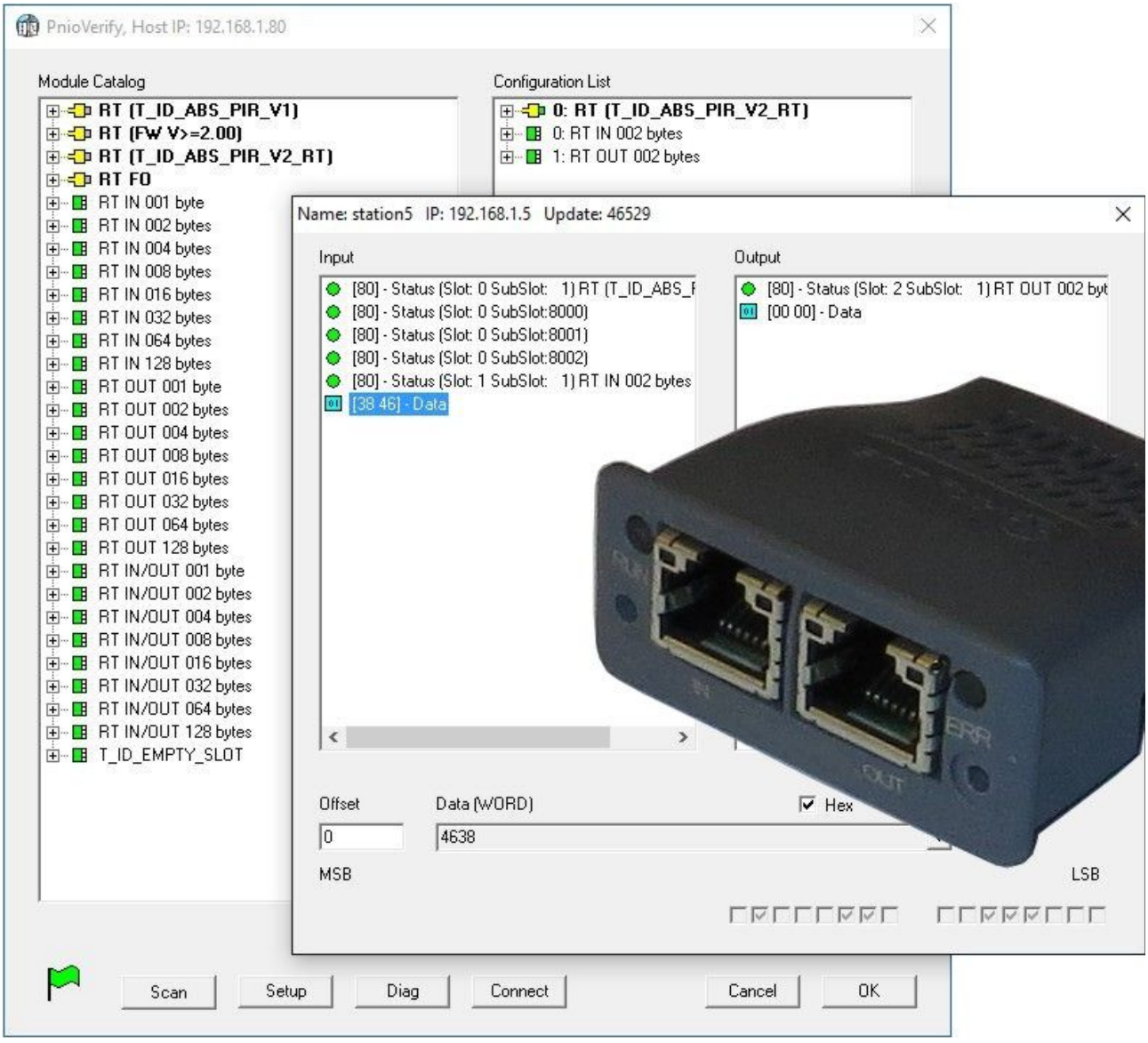
PnioVerify
ProfiNET Master Simulator
Documentation

Date: Feb,6.2024

1 Introduction

The SYBERA ProfiNET Master Simulator allows the control of ProfiNET IO devices at realtime, on a standard PC, without the need of additional controller hardware. The PROFINET control can be realized simply by the PC with a standard Ethernet adapter. The Master Simulator is based on the high precision XRealtime Engine for sending and receiving of ProfinetIO frames at realtime. The Master Simulator allows the handling of ProfiNET-IO data without the need of any complex ProfinetIO management. The Master Simulator scans the bus for Accesspoints, and offers simply configuration of functional modules. Also diagnostics can be handled easily.



1.1 Product Features

- ProfiNET Station Management
- Station Realtime Sampling-Cycles upto 100 µsec
- ProfiNET Diagnostics
- ProfiNET acyclic Service Interface (Read / Write)
- ARP, DCP, RPC, LLDP Implementation
- ProfiNET Cyclic Data Exchange
- Sequence Log

1.2 Supported OS

- Windows 7
- Windows 8
- Windows 10
- Windows 11

1.3 Reference Devices

- HMS Anybus-S Module (T_ID_DAP)
- HMS Anybus-S Module (T_ID_ABS_PIR)
- HMS Anybus-S Module (T_ID_ABS_PRT)
- Phoenix ILB 24 DI16 DIO16 – TX2
- Phoenix FL IL 24 BK-PN-PAC
- Deutschmann Unigate CL

2 Installation

For installation of the PnioVerify software following steps are required:

2.1 Preparation

1. Provide a PC with INTEL or REALTEK Ethernet adapter and Windows operating system
2. Check the installed Ethernet adapter has given a correct IP address

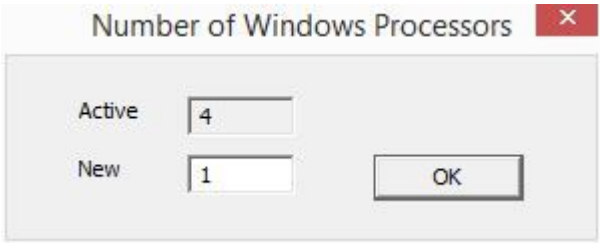
2.2 Installation

3. Install the MFC redistributables DLLs (in Folder MISC \ REDIST)
4. Install Fonts (in Folder MISC \ FONTS)
5. Install MSXML redistributables (in Folder MISC \ MSXML)
6. Next run SYSETUP64 with administrator privileges
   (make sure the directory path has no space characters)
7. On Installation the PEC information (PID, SERNUM and KEYCODE) must be
   entered. The SERNUM for the evaluation version is: 12345678,
   the KEYCODE is: 00001111-22223333
8. Select Network card
   Optional: Check license with SYLICENCECHECK64.EXE
9. Reboot the System

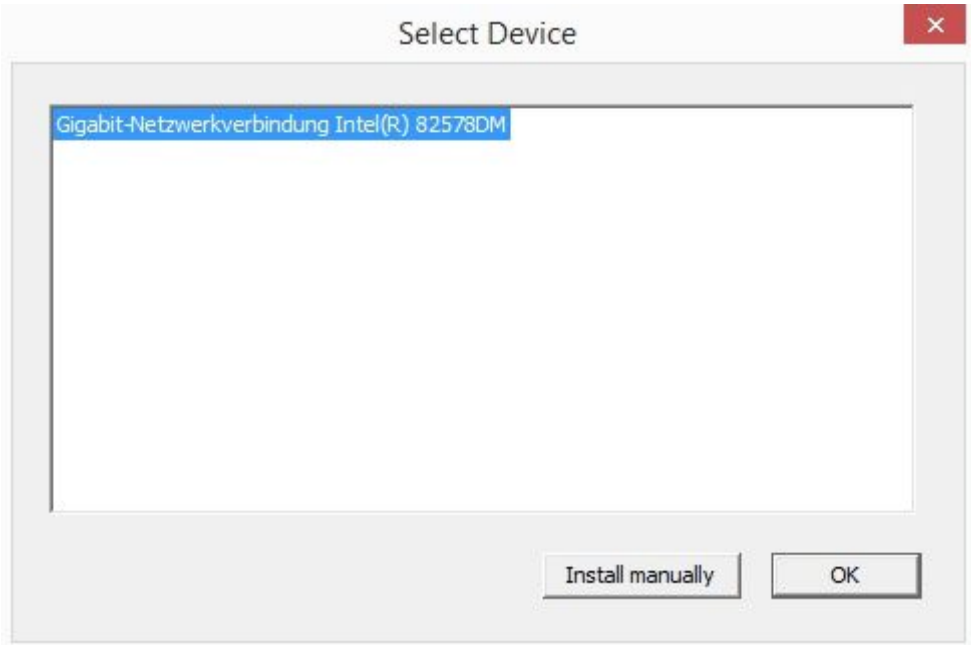
2.3 Operation

10. Run PnioVerify

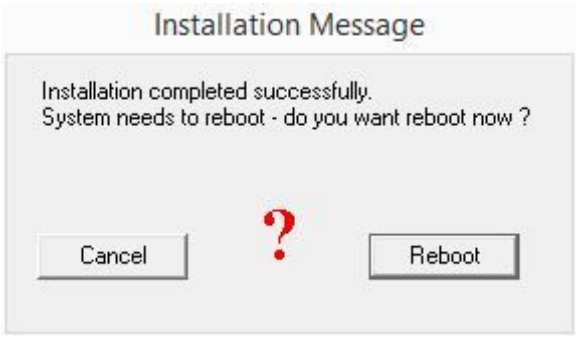
Select **Windows** processor count, typically [Active – 1]



Select ethernet device and quit with button [OK]

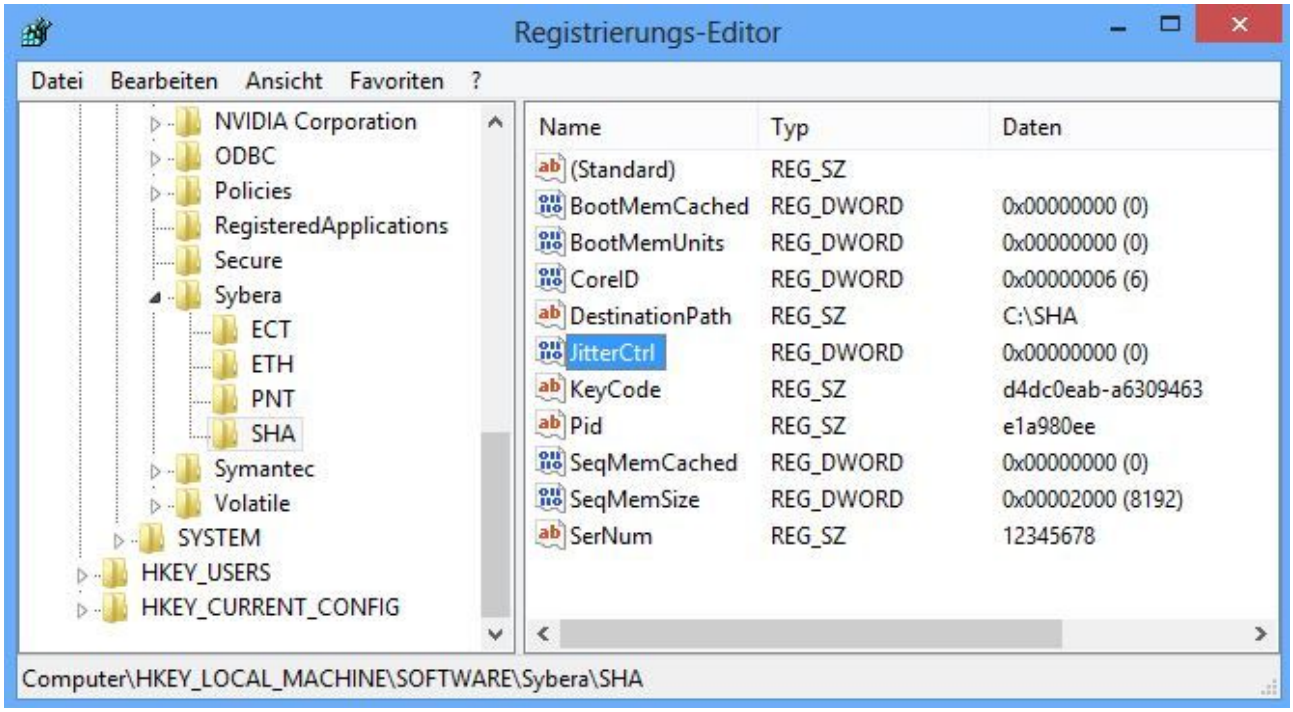


Reboot the system



2.4 Jitter Control (optional)

Since a notebook has a quiet different jitter behaviour than desktop systems, an enhanced jitter control mechanism is required. Therefore SYBERA provides a registry entry called “JitterCtrl”. This entry allows an adaptive iteration to the best jitter behaviour of the notebook.

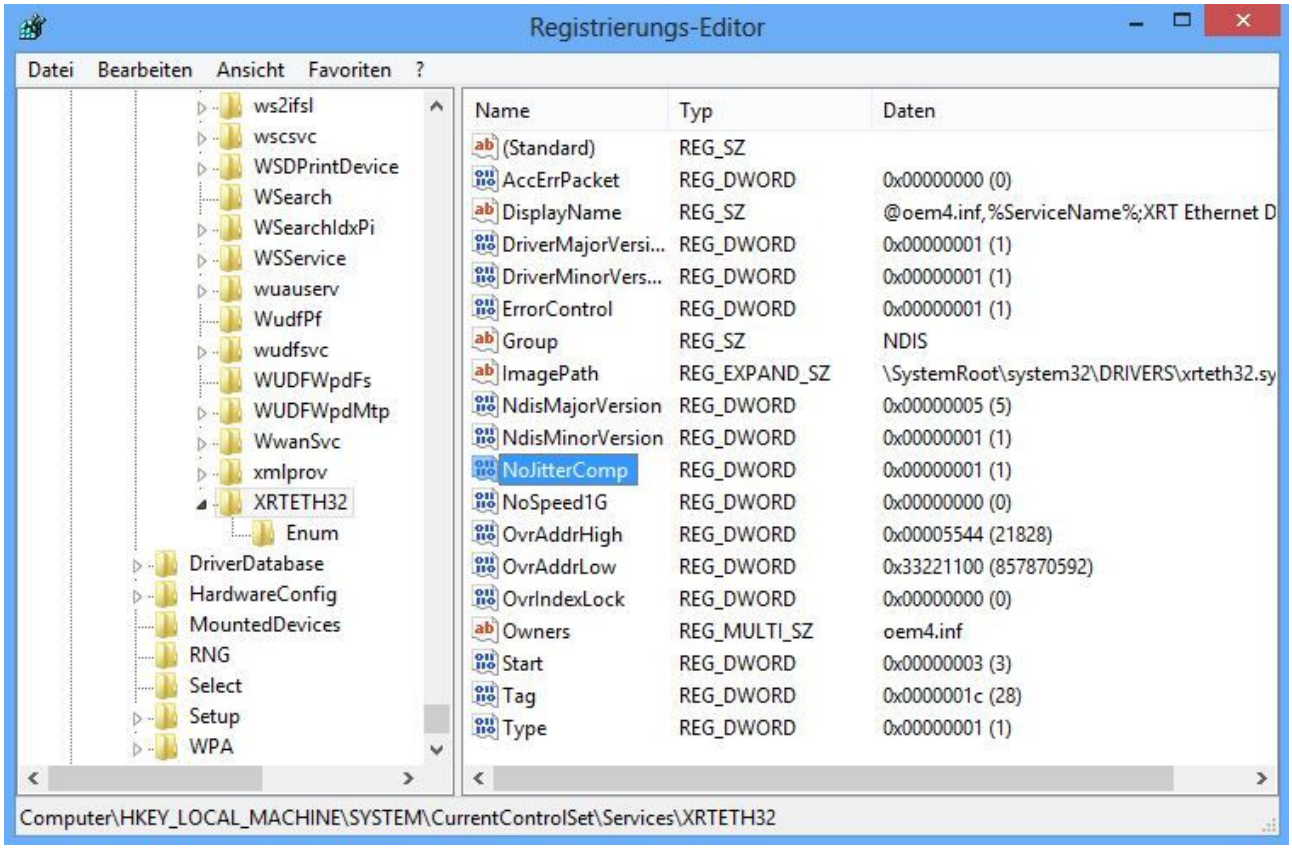


Following values are valid:

0: No enhanced jitter control
1: Enhaced Jitter Control, Step 1 (first choice together with BIOS settings)
2: Enhaced Jitter Control, Step 2 (for INTEL platforms only)
3: Enhaced Jitter Control, Step 3 (for INTEL platforms only, together with BIOS settings)

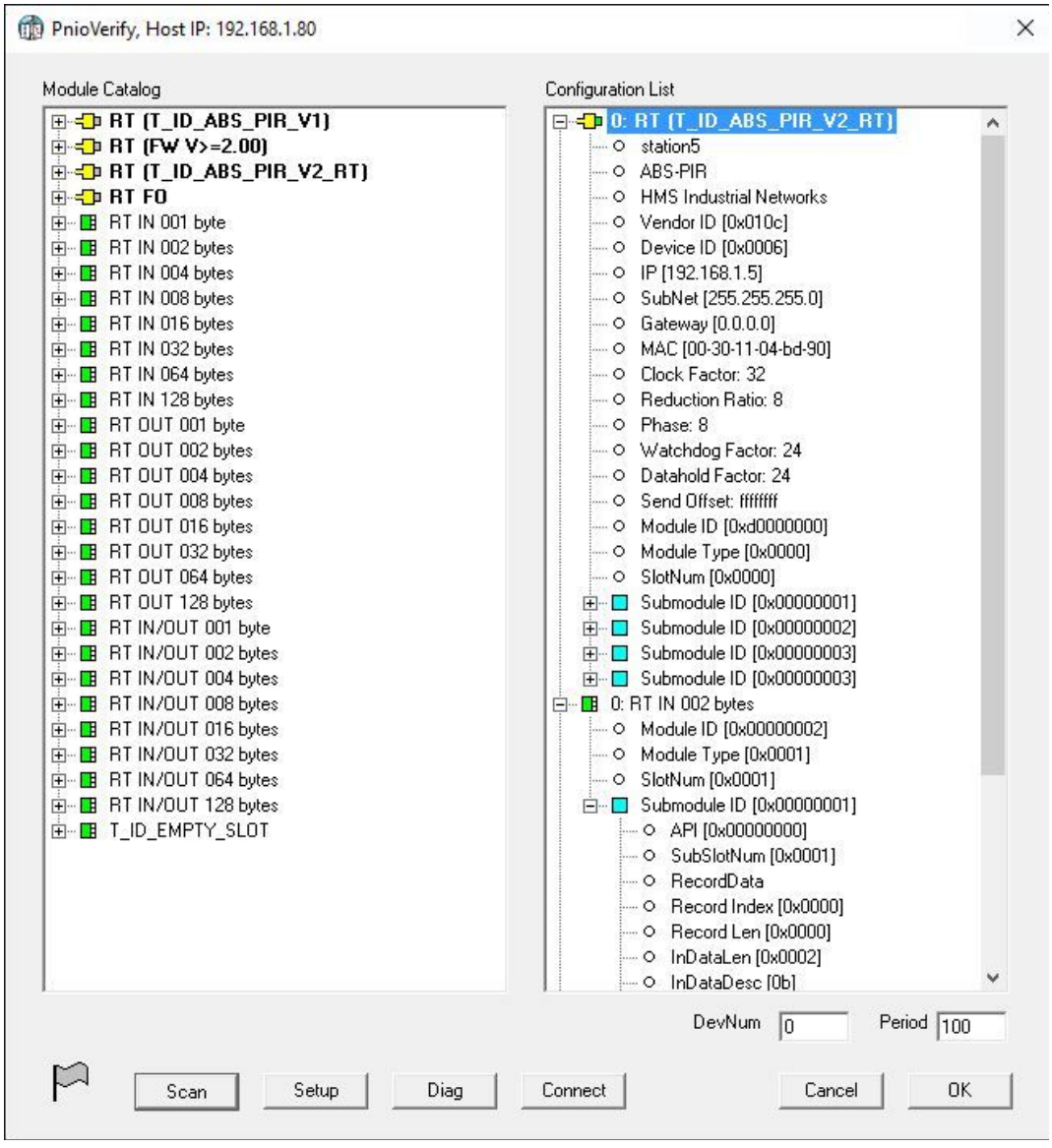
2.5 Dynamic Jitter Compensation (optional)

SYBERA uses the procedure "Dynamic Jitter Compensation" with active and passive feedback compensation within the realtime engine. Although the X-Real time engine of SYBERA allows a native maximum Jitter of approx. 15 µ sec (according to hardware platform), this behaviour may be reduced below 3 µsec by the dynamic jitter compensation.
For compatibility reason on some platforms it may be required to disable the dynamic jitter compensation. Therefore the registry value “NoJitterComp” has to be set to 1



3 Creating a Stationlist

A ProfinetIO fieldbus system consists of several station devices (typically buscoupler devices). A station consists at least of one module (SLOT) and a module consists at least of one submodule (SUBSLOT). For proper operation the ProfinetIO devices needs first to be configured (by Station Name and IP) and a native STATIONLIST for operating the ProfiNET realtime library has to be created. Therefore SYBERA provides a program called PNIOVERIFY64.EXE.



Note: Make shure a valid IP address is provided for the network connection.

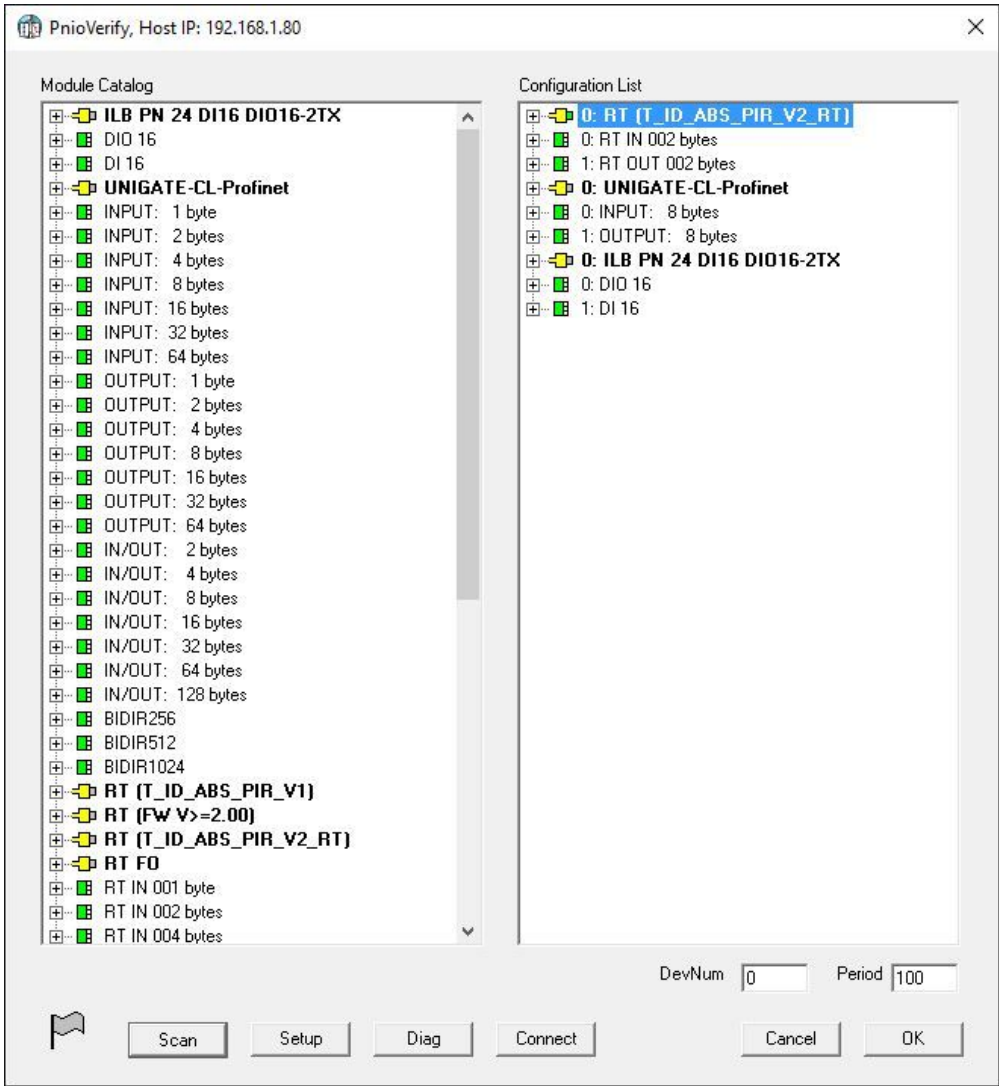
Note: If the application fails to run, check if the lastest Microsoft XML Parser has been installed. If not, install in the directory \APP\MSXML\MSXML4

PNIOVERIFY allows creating a native stationlist by selecting modules from a module catalog (leftside view). The catalog get its entries by the provides GSDML files which must be present in the same directory as PNIOVERIFY. A module is inserted to the station list configuration (rightside view) by a DRAG and DROP operation (just drag a module from the catalog to the station list configuration). There are two types of modules:

	Accesspoint Module	(SLOT 0)
	Functional Module	(SLOT 1 .. n)

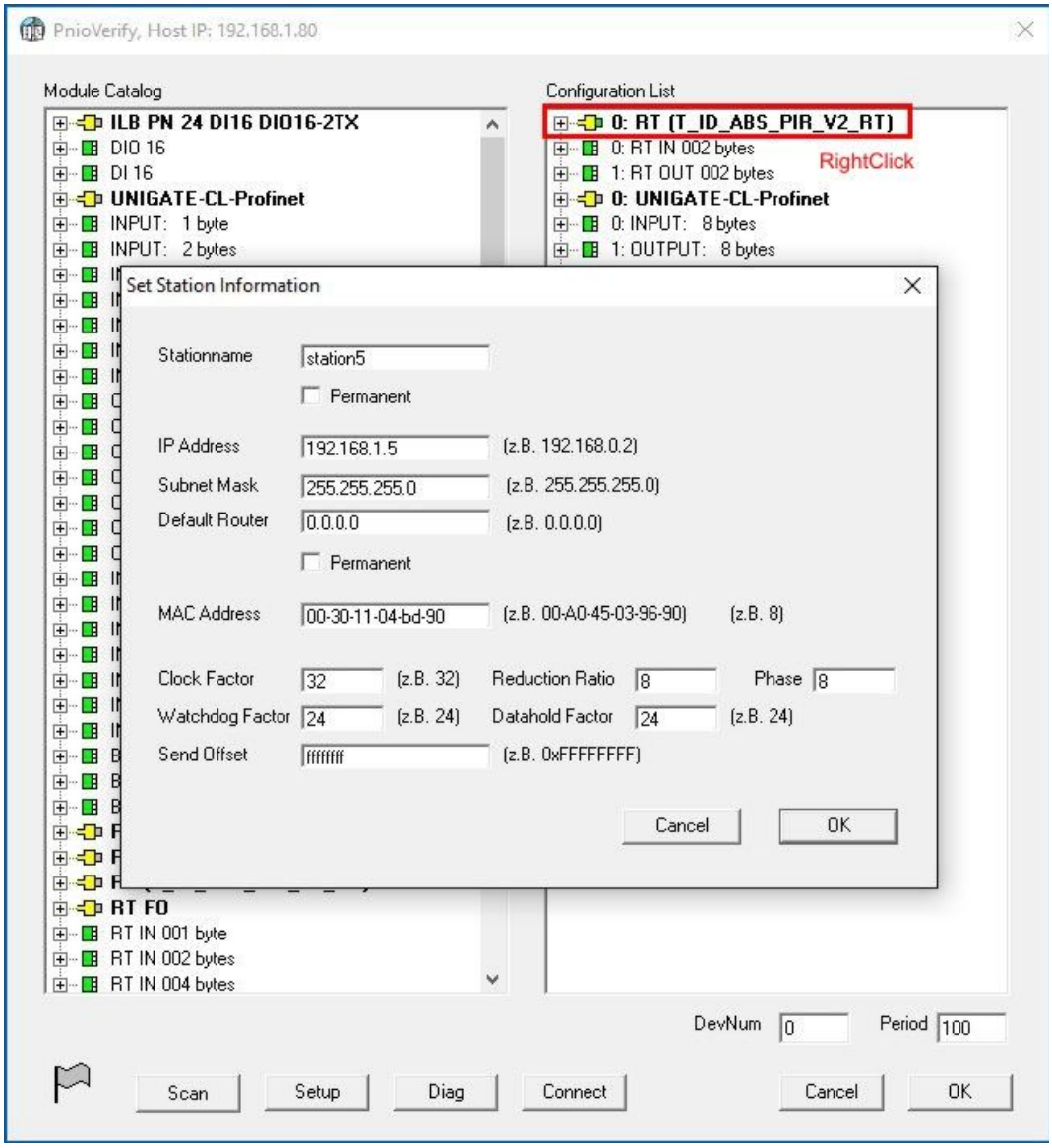
3.1 Accesspoint Module

The accesspoint module keeps all information required for connecting to the fieldbus, as station name, IP parameters, MAC address, timing parameters. Therefore first task is to collect information about the ProfinetIO configuration by scanning the bus.



3.2 Station Settings

The scan gets information about manufacturer name and MAC address. Now individual assignment must set (e.g. IP address, station name, timings). On a right button click at the accesspoint module the Set Station Information dialog appears.



3.2.1 ProfiNET Timing

The timing settings of each station are based on a clock unit of 31,25 µsec. The synchronisation period is calculated as follow:

SyncTime = 31,25 µsec * ClockFactor * ReductionRatio

(e.g. 31,25 µsec * 32 * 8 = 8000 µsec = 8 msec)

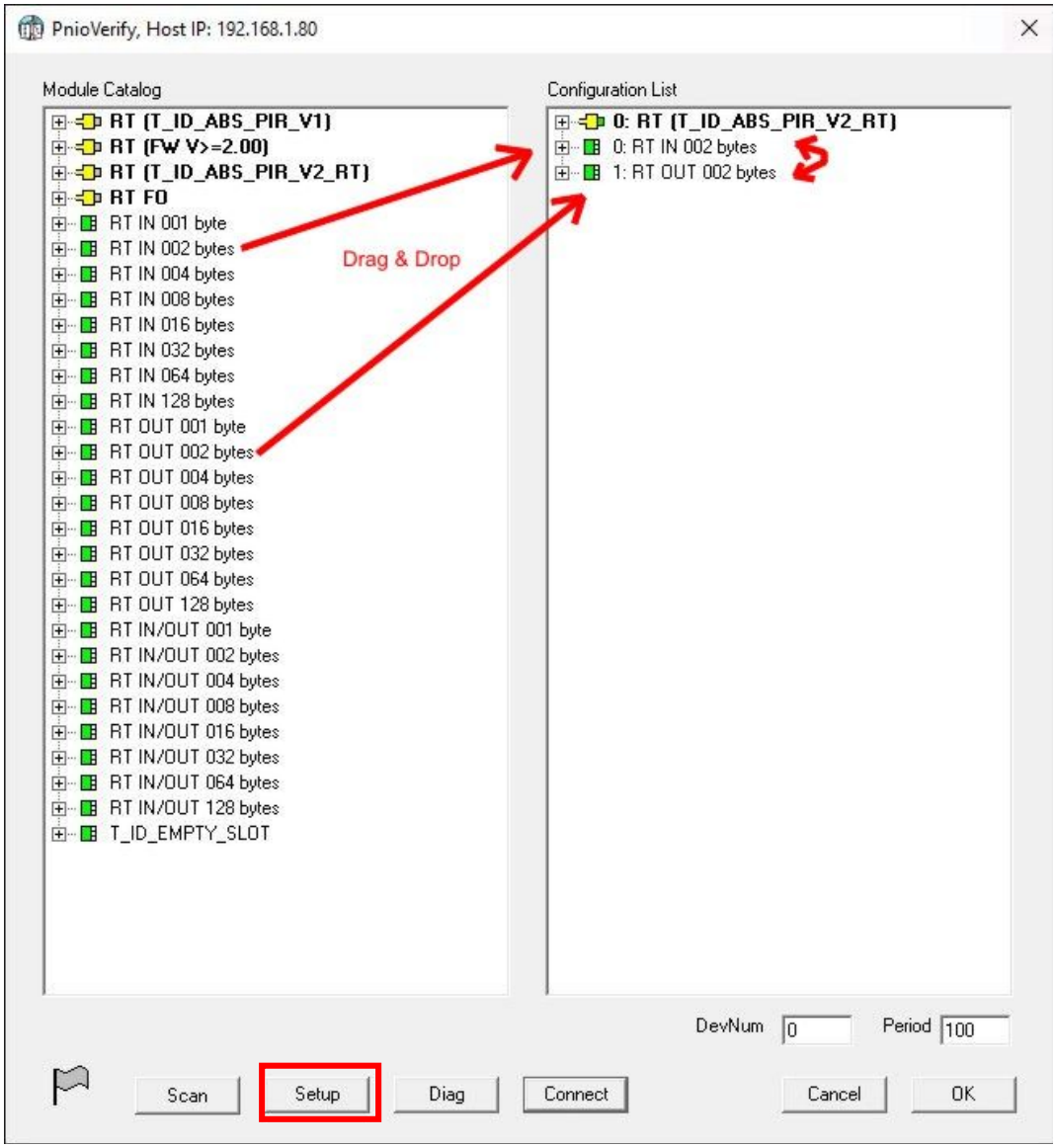
WatchdogTime = SyncTime * WatchdogFactor

(e.g. 8 msec * 24 = 192 msec)

The SendOffset must be set to 0xFFFFFFFF

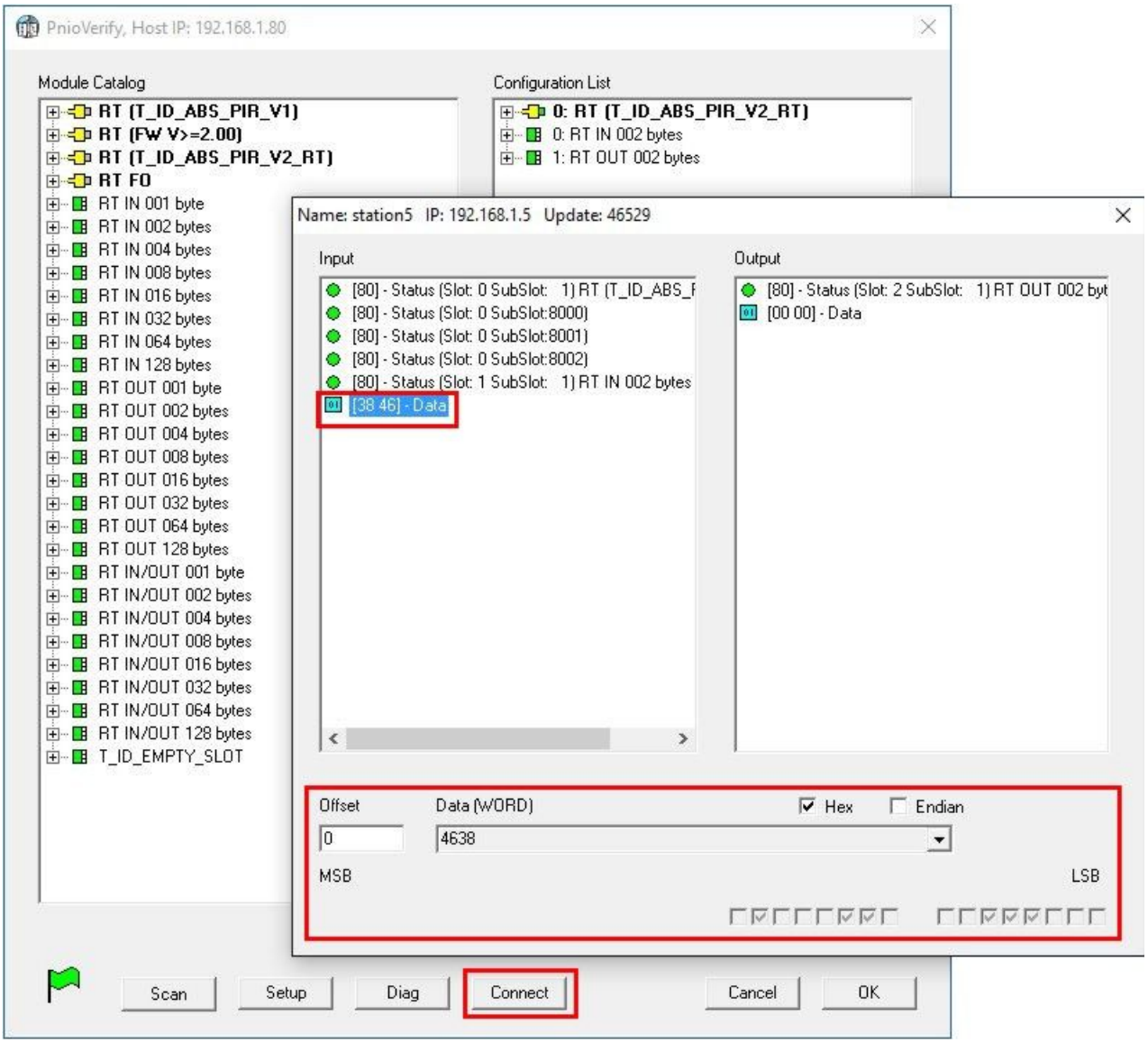
3.3 Functional Module

Each station typically consists of multiple functional modules (SLOT 1..n). Function Modules have to be inserted from the catalog by DRAG and DROP operations. As well the nmodules may be sorted below the AccessPoint. A station configuration should contain all functional modules (in the order these modules are physically connected). When inserting a new module from the catalog, after dropping, it appears at the end of the configuration list and may be pushed to the correct slot location.

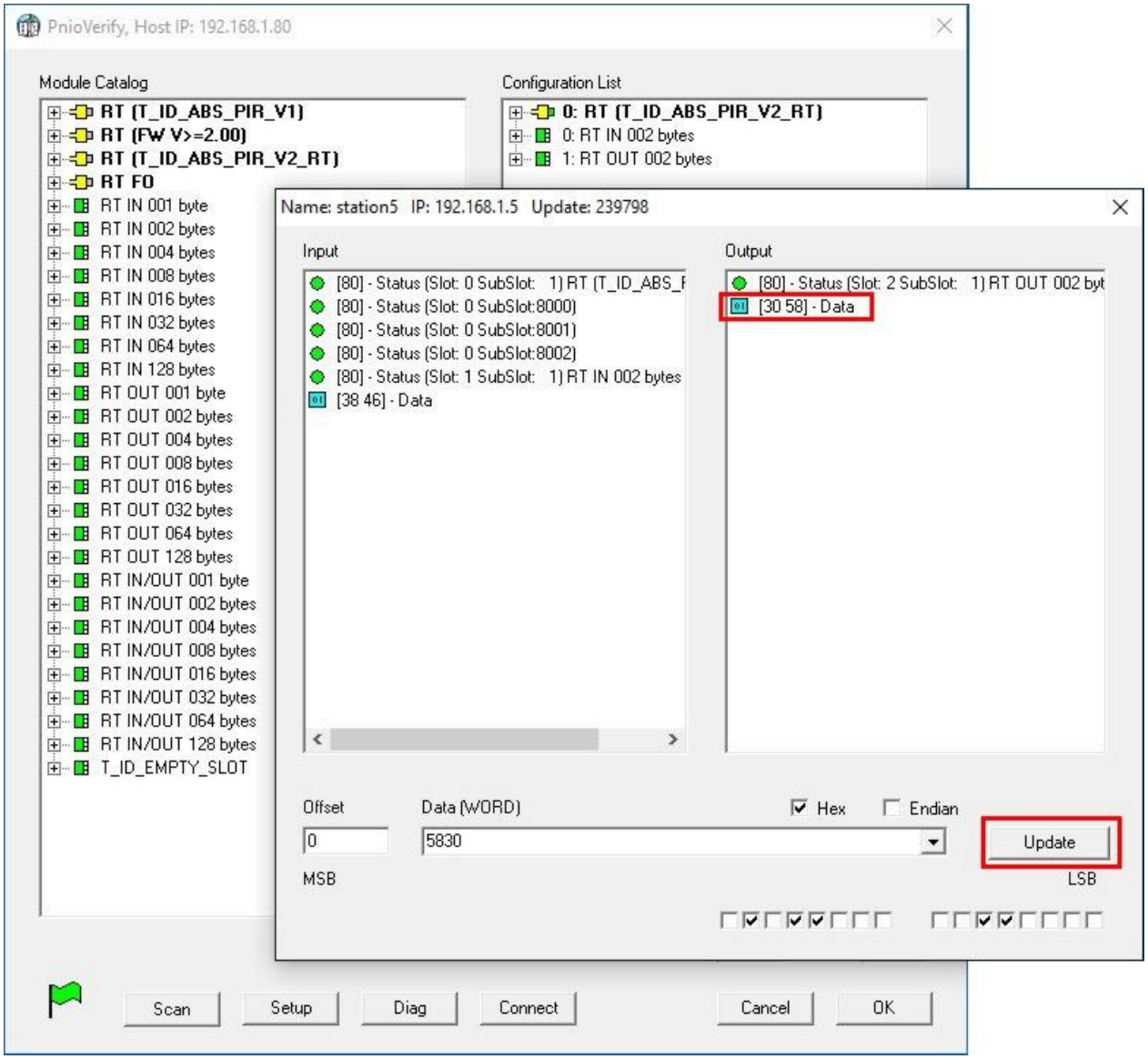


3.4 Cyclic Operation

For cyclic data exchange, select an access point and press the button [Connect]. On the left side you will see the input submodules, on the right side the output submodules with status and data fields. When selecting a data field, the value is displayed below with several selectable options (hex, endian, bitwise …). Each value may displayed and evaluated by a maximum size of 4 bytes. At larger data fields, an offset must be used to display and evaluate the next data bytes.

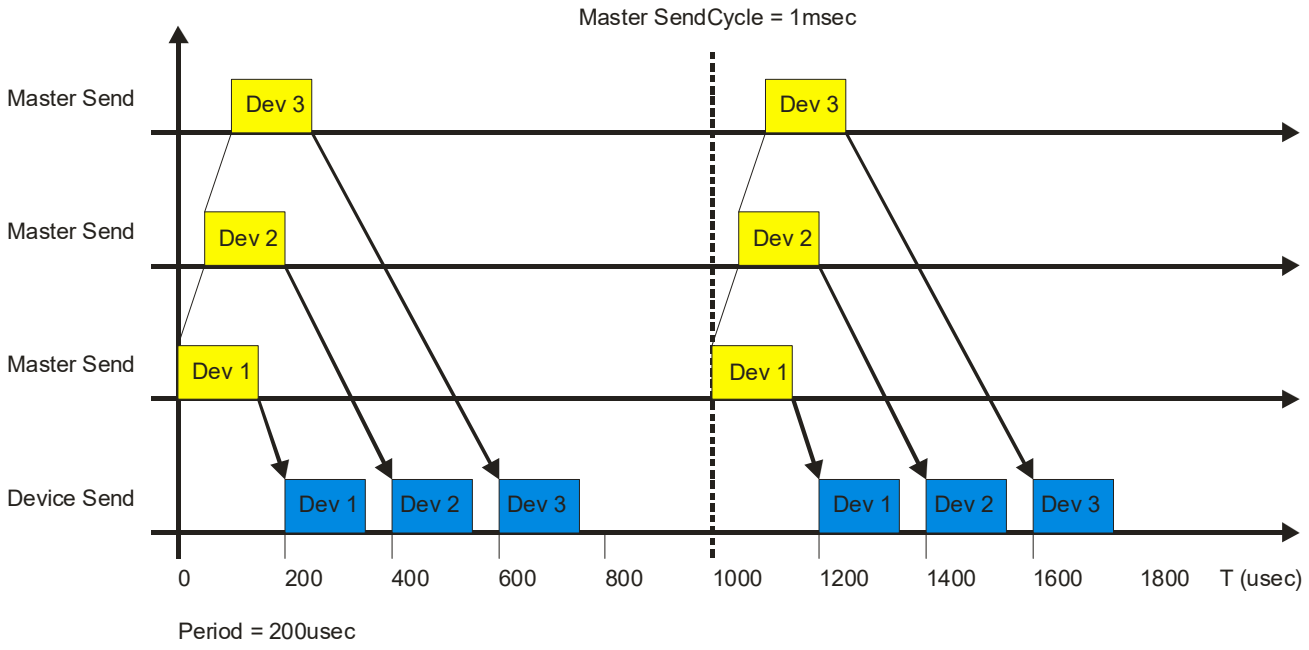


An output value for a output data field my be written by pressing the button [Update].



3.5 PLL Send Mode

With the PLL send mode, a station is bound to the send timing of the master. The device will send its frame, when receiving a master frame.



Registry:

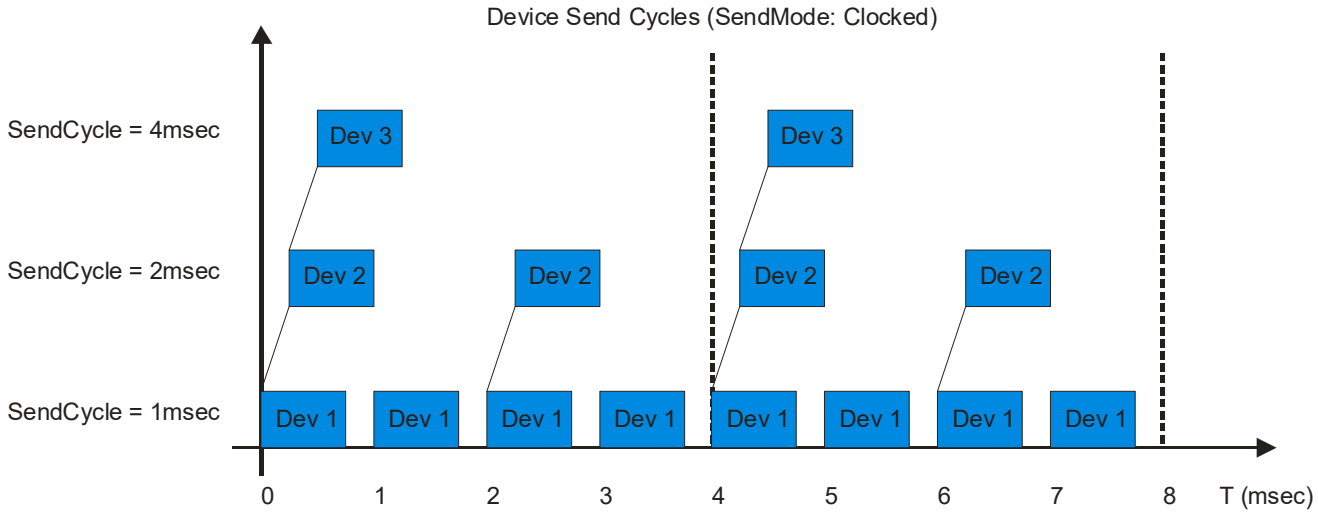
```
HKEY_LOCAL_MACHINE\SOFTWARE\Sybera\PNS\SendMode       0
```

Wireshark:

```
2735 16.795502000 CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:28370 (Valid,Primary,Ok,Run)
2736 16.795850000 FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:28370 (Valid,Primary,Ok,Run)
2737 16.799501000 CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:28498 (Valid,Primary,Ok,Run)
2738 16.799851000 FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:28498 (Valid,Primary,Ok,Run)
2739 *REF*        CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:28627 (Valid,Primary,Ok,Run)
2740 0.000354000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:28627 (Valid,Primary,Ok,Run)
2741 0.004009000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:28755 (Valid,Primary,Ok,Run)
2742 0.004356000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:28755 (Valid,Primary,Ok,Run)
2743 0.008013000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:28883 (Valid,Primary,Ok,Run)
2744 0.008360000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:28883 (Valid,Primary,Ok,Run)
2745 0.012013000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:29011 (Valid,Primary,Ok,Run)
2746 0.012361000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:29011 (Valid,Primary,Ok,Run)
2747 0.016016000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:29139 (Valid,Primary,Ok,Run)
2748 0.016362000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:29139 (Valid,Primary,Ok,Run)
2749 0.020023000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:29267 (Valid,Primary,Ok,Run)
2750 0.020349000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:29267 (Valid,Primary,Ok,Run)
2751 0.024023000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:29395 (Valid,Primary,Ok,Run)
2752 0.024369000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:29395 (Valid,Primary,Ok,Run)
2753 0.028025000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:29523 (Valid,Primary,Ok,Run)
2754 0.028367000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:29523 (Valid,Primary,Ok,Run)
```

3.6 Clocked Send Mode

With the clocked send mode, a station is bound to a (master) specified send cycle, independently to the master cycle itself. This allows a different device send cycle to the master send cycle.



Registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Sybera\PNS\SendMode       1
```

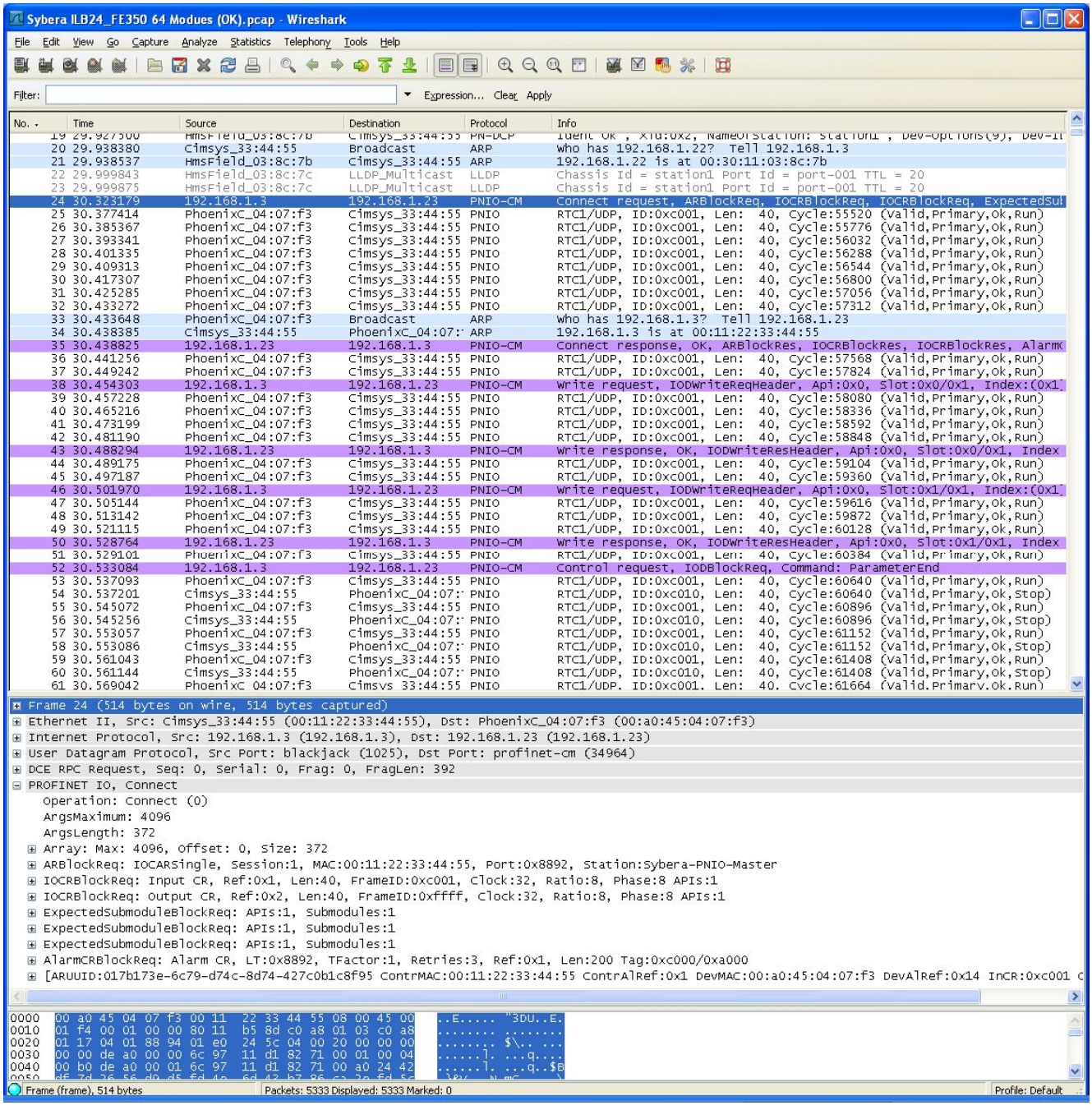
Wireshark:

```
3296 13.897668000 CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:62012 (Valid,Primary,Ok,Run)
3297 13.898353000 FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62012 (Valid,Primary,Ok,Run)
3298 13.900355000 FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62012 (Valid,Primary,Ok,Run)
3299 13.901645000 CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:62140 (Valid,Primary,Ok,Run)
3300 *REF*        FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62140 (Valid,Primary,Ok,Run)
3301 0.002011000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62140 (Valid,Primary,Ok,Run)
3302 0.003281000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:62268 (Valid,Primary,Ok,Run)
3303 0.004007000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62268 (Valid,Primary,Ok,Run)
3304 0.006002000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62268 (Valid,Primary,Ok,Run)
3305 0.007301000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:62396 (Valid,Primary,Ok,Run)
3306 0.008009000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62396 (Valid,Primary,Ok,Run)
3307 0.010007000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62396 (Valid,Primary,Ok,Run)
3308 0.011305000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:62525 (Valid,Primary,Ok,Run)
3309 0.011994000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62525 (Valid,Primary,Ok,Run)
3310 0.014011000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62525 (Valid,Primary,Ok,Run)
3311 0.015305000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:62653 (Valid,Primary,Ok,Run)
3312 0.016013000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62653 (Valid,Primary,Ok,Run)
3313 0.018024000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62653 (Valid,Primary,Ok,Run)
3314 0.019306000  CIMSYS_33:44:55          FritzKue_03:23:96    PNIO_PS    64 RTC1(legacy), ID:0xc002, Len:  40, Cycle:62781 (Valid,Primary,Ok,Run)
3315 0.019996000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62781 (Valid,Primary,Ok,Run)
3316 0.022010000  FritzKue_03:23:96        CIMSYS_33:44:55      PNIO_PS    64 RTC1(legacy), ID:0xc001, Len:  40, Cycle:62781 (Valid,Primary,Ok,Run)
```

Registry:

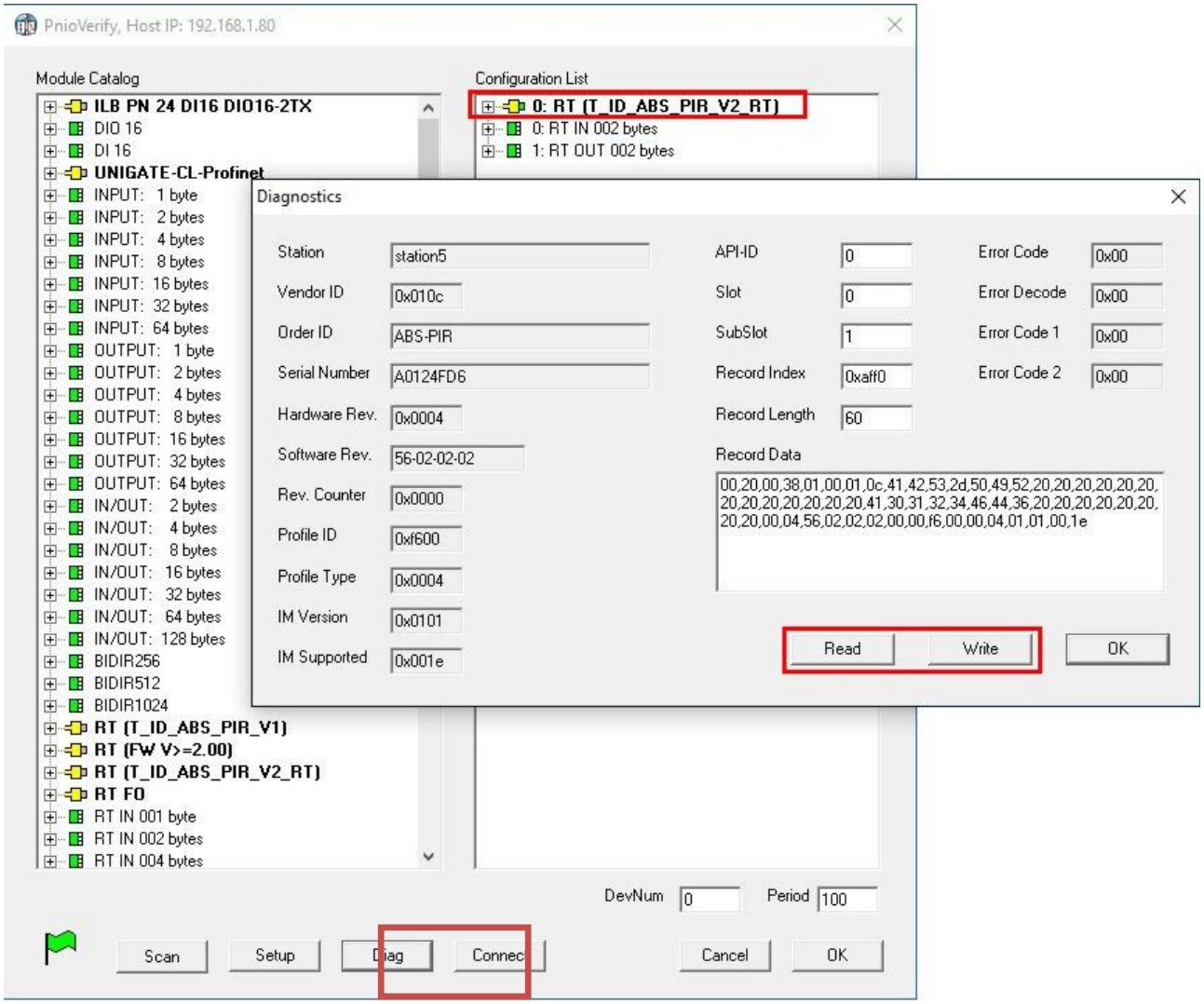
```
HKEY_LOCAL_MACHINE\SOFTWARE\Sybera\PNS\SendMode       1
```

Sample Startup Protocoll:



3.7 Station Diagnostics

The station diagnostics allows gathering of I&M data, as well as reading and writing acyclic information (by API, SlotNum, SubSlotNum and RecordIndex). Therefore select an AccessPoint and press button [Diag].



To read or write acyclic information, put in the API-ID, Slot, SubSlot and Record Index. If the function fails, you'll get the corresponding PNIO error code.

4 Save Configuration

The resulting stationlist is stored in a text file which must be provided to the ProfinetIO Master Library.

Sample:

```
> Station

[NAME]
station8
[MFG_NAME]
ABS-PIR
[VENDOR_NAME]
HMS Industrial Networks
[STATION_ID]
0c 01 06 00 01 00
[MAC_ADDR]
00 30 11 04 bd 90
[IP_PARAMS]
c0 a8 01 08 ff ff ff 00 00 00 00 00
[TI_PARAMS]
20 00 08 00 08 00 18 00 18 00 ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                                                  00

>> Module

[NAME]
T_ID_ABS_PIR_V2_RT
[MOD_ID]
00 00 00 d0
[MOD_TYPE]
00 00
[SLOT_NUM]
00 00

>>> Submodule

[SUBMOD_ID]
01 00 00 00
[SUBSLOT_NUM]
01 00
[OBJ_INPUT]
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                  00 00 00 00 01 00 00 00 01 00 00 00

>>> Submodule

[SUBMOD_ID]
02 00 00 00
[SUBSLOT_NUM]
00 80
[OBJ_INPUT]
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                  00 00 00 00 01 00 00 00 01 00 00 00

>>> Submodule

[SUBMOD_ID]
03 00 00 00
[SUBSLOT_NUM]
01 80
[OBJ_INPUT]
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                  00 00 00 00 01 00 00 00 01 00 00 00
```

```
>>> Submodule

[SUBMOD_ID]
03 00 00 00
[SUBSLOT_NUM]
02 80
[OBJ_INPUT]
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                  00 00 00 00 01 00 00 00 01 00 00 00

>> Module

[NAME]
T_ID_RT_IN2
[MOD_ID]
02 00 00 00
[MOD_TYPE]
01 00
[SLOT_NUM]
01 00

>>> Submodule

[SUBMOD_ID]
01 00 00 00
[SUBSLOT_NUM]
01 00
[OBJ_INPUT]
01 0b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                  02 00 00 00 01 00 00 00 01 00 00 00

>> Module

[NAME]
T_ID_RT_OUT2
[MOD_ID]
20 00 00 00
[MOD_TYPE]
01 00
[SLOT_NUM]
02 00

>>> Submodule

[SUBMOD_ID]
01 00 00 00
[SUBSLOT_NUM]
01 00
[OBJ_OUTPUT]
01 0b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                  02 00 00 00 01 00 00 00 01 00 00 00
```

5 Error Handling

The master library provides an error handling and tracing mechanism.

5.1 Debug LOG File

On execution the master library creates a sequence file PNTDBG.LOG in Text-Format

Note: This file is not accessible while the application is running

5.2 Event File

On execution the master library logs error event to the Windows Event Manager. The master library logs Application and System events. These events can be exported to a file and provided for support purposes.