Das **Dynamic Configuration Protocol** (DCP) hat die Aufgabe, die Adressen und Namen in einem PROFINET IO System den einzelnen Teilnehmern zu verteilen.

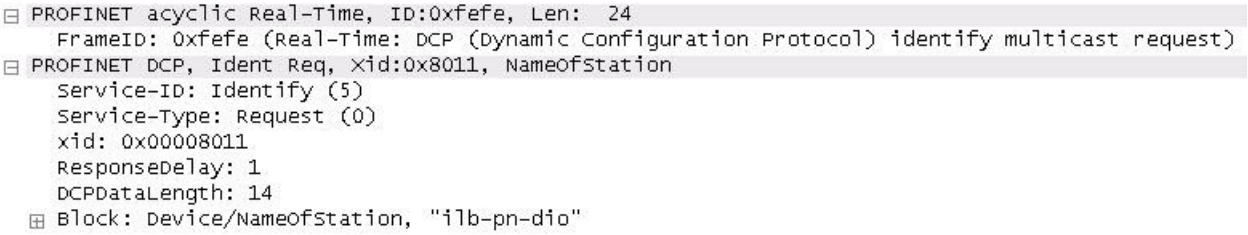
Das DCP Protokoll ist in der Norm IEC 61158 festgelegt.

Sequenzen von Telegrammen und die Dienste werden im Abschnitt der Adressverwaltung erläutert.

Die DCP Telegramme hat den Ethernet TYPE 0x8892 = PROFINET und die folgende Struktur:

DCP Telegramm	**Umfang**	**Werte**	**Bedeutung**
FrameID	2 Byte	0xfefd - 0xfeff	Reservierter Bereich für das DCP
ServiceID	1 Byte	1 = Get 2 = Set 5 = Identify	
Service-Type	1 Byte		
xid	4 Bytes		Identifikation der Transaktion
ResponseDelay	2 Bytes		
DCPDataLength	2 Bytes		Anzahl der Bytes in den nachfolgenden Blöcken
Block	x Bytes		

Ein DCP Telegramm kann bei Wireshark so aussehen:



```
⊟ PROFINET acyclic Real-Time, ID:0xfefe, Len:  24
    FrameID: 0xfefe (Real-Time: DCP (Dynamic Configuration Protocol) identify multicast request)
⊟ PROFINET DCP, Ident Req, Xid:0x8011, NameOfStation
    Service-ID: Identify (5)
    Service-Type: Request (0)
    xid: 0x00008011
    ResponseDelay: 1
    DCPDataLength: 14
  ⊞ Block: Device/NameOfStation, "ilb-pn-dio"
```

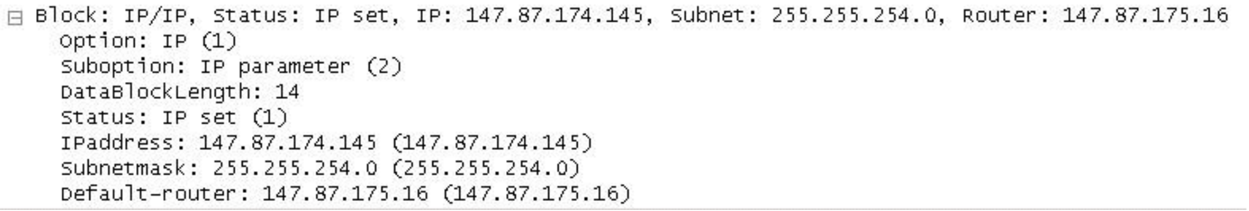
Der **Service-Type** wird wie folgt kodiert:

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Service-Type**
0	0	0	0	0				
					x			0 = success (Erfolgreich)
						x	x	0 = Request (Anforderung) 1 = Response (Antwort)

Jeder **Block** in einem DCP Telegramm beschreibt nun spezielle Eigenschaften:

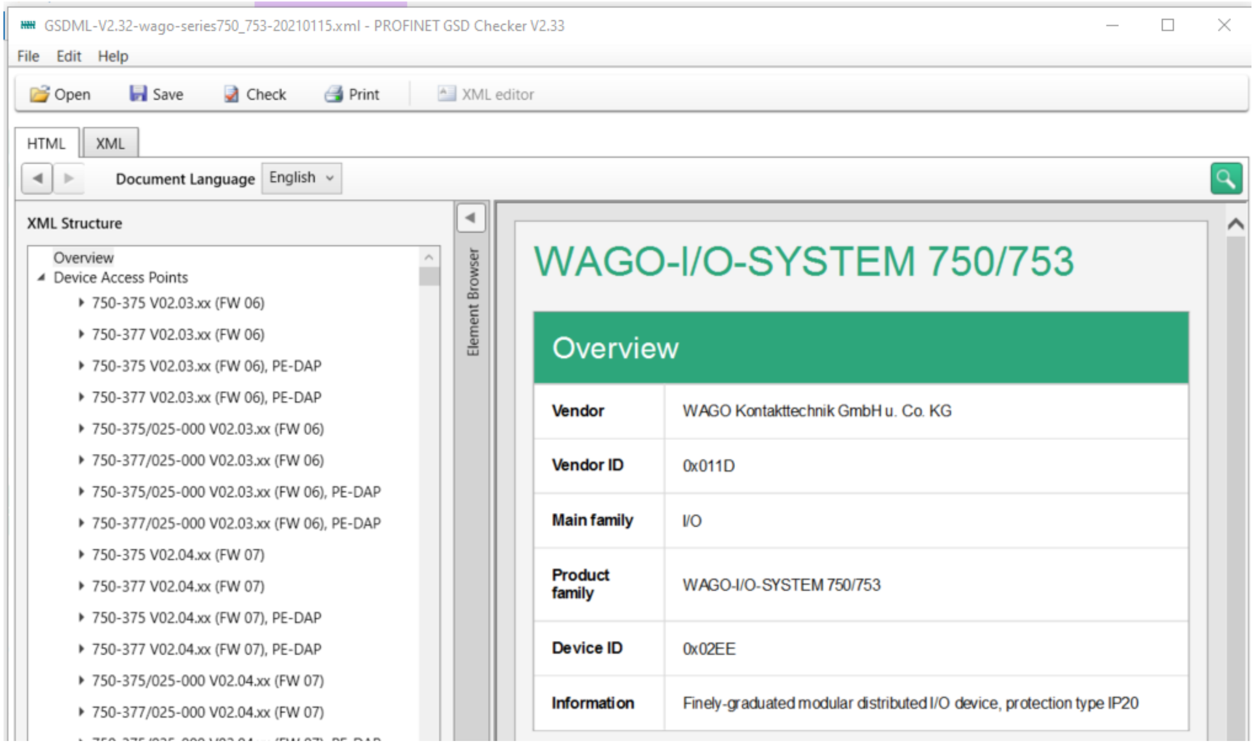
Option	Suboption	len	Status	Data
1 Byte	1 Byte	2 Bytes	2 Bytes	len Bytes
1	1 = MAC Address	6		
	2 = IP Address	14		
2	1 = Manufacturer Specific			
	2 = Name of Station			
	3 = Device ID			
	4 = Device	4		
	5 = Device options	1		
3	All DHCP options			
4	LLDP options			
5	1 = Start Transaction	0		
	2 = End Transaction	0		

Ein DCP-Block zum setzen einer IP Adresse kann bei Wireshark so aussehen:



```
⊟ Block: IP/IP, Status: IP set, IP: 147.87.174.145, Subnet: 255.255.254.0, Router: 147.87.175.16
    Option: IP (1)
    Suboption: IP parameter (2)
    DataBlockLength: 14
    Status: IP set (1)
    IPaddress: 147.87.174.145 (147.87.174.145)
    Subnetmask: 255.255.254.0 (255.255.254.0)
    Default-router: 147.87.175.16 (147.87.175.16)
```

It is often important for the user to look at the functionality of a PROFINET field device without having detailed knowledge about the contents of the description file. This is done by the GSD Checker available at www.profinet.com. The GSD Checker can be used to view GSD files for PROFINET. The GSD checker interprets such a file and is able to check it for correctness. With a simple mouse click, the syntax of a GSD file can be checked:



Overview
Device Access Points
- 750-375 V02.03.xx (FW 06)
- 750-377 V02.03.xx (FW 06)
- 750-375 V02.03.xx (FW 06), PE-DAP
- 750-377 V02.03.xx (FW 06), PE-DAP
- 750-375/025-000 V02.03.xx (FW 06)
- 750-377/025-000 V02.03.xx (FW 06)
- 750-375/025-000 V02.03.xx (FW 06), PE-DAP
- 750-377/025-000 V02.03.xx (FW 06), PE-DAP
- 750-375 V02.04.xx (FW 07)
- 750-377 V02.04.xx (FW 07)
- 750-375 V02.04.xx (FW 07), PE-DAP
- 750-377 V02.04.xx (FW 07), PE-DAP
- 750-375/025-000 V02.04.xx (FW 07)
- 750-377/025-000 V02.04.xx (FW 07)

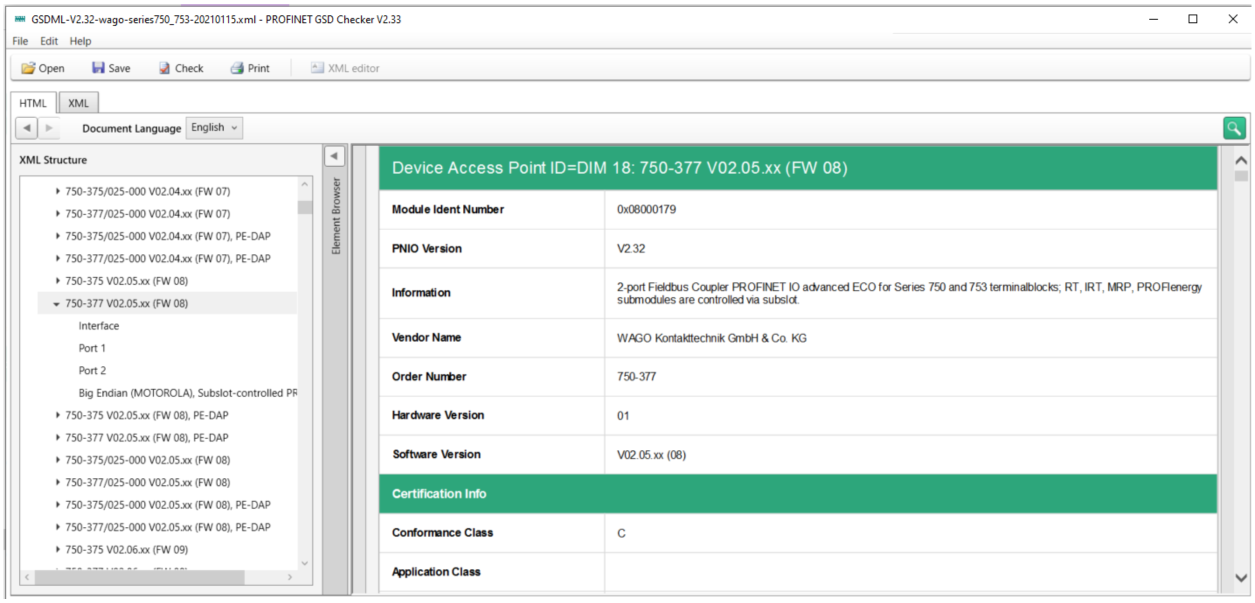
WAGO-I/O-SYSTEM 750/753

Overview

Vendor	WAGO Kontakttechnik GmbH u. Co. KG
Vendor ID	0x011D
Main family	I/O
Product family	WAGO-I/O-SYSTEM 750/753
Device ID	0x02EE
Information	Finely-graduated modular distributed I/O device, protection type IP20

Figure 68: GSD Checker for displaying the GSD file

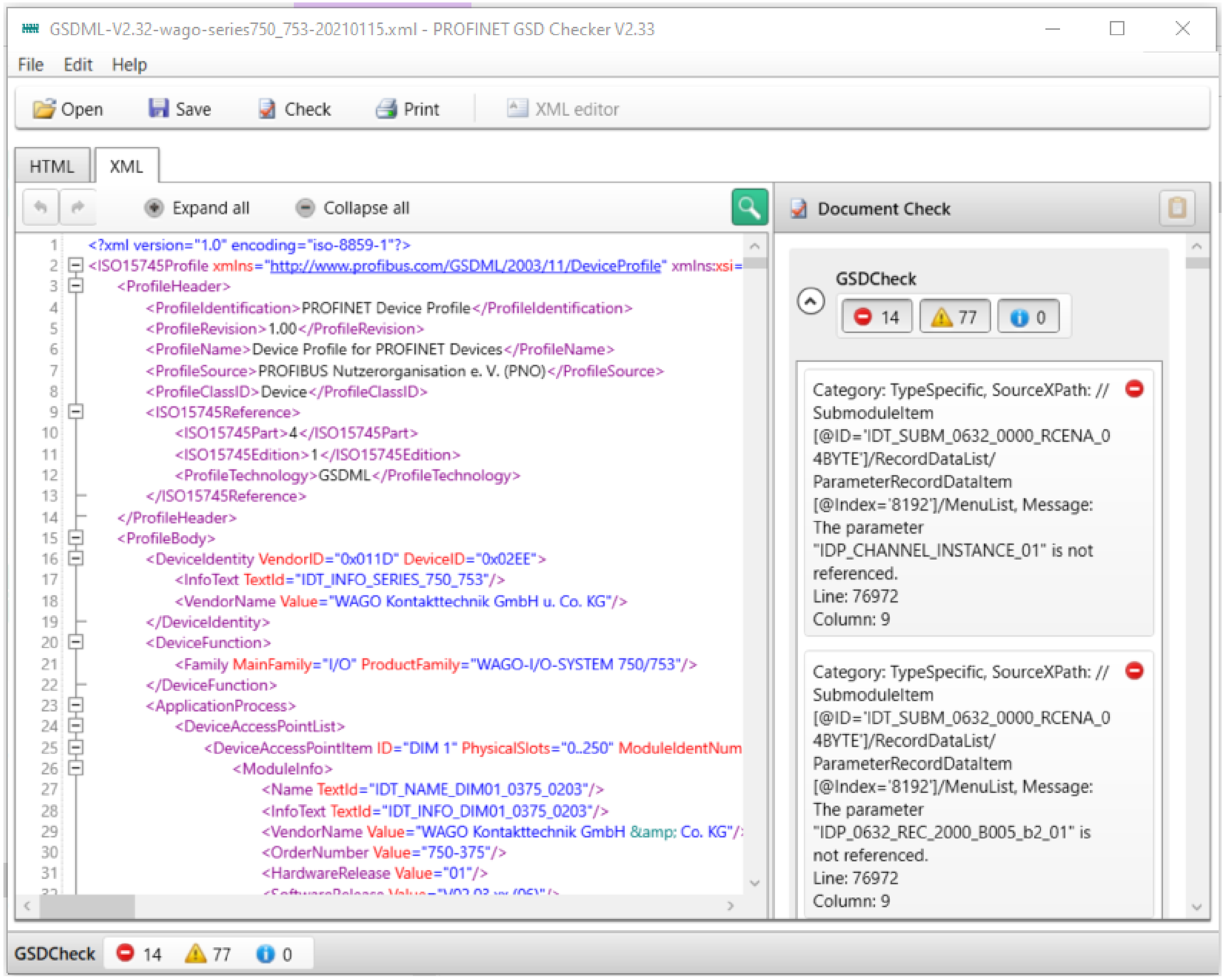
With a browser display, the contents of the GSD file can be displayed in a readable and printable form:


GSDML-V2.32-wago-series750_753-20210115.xml - PROFINET GSD Checker V2.33
File
Edit
Help
Open
Save
Check
Print
XML editor
HTML
XML
Document Language
English
XML Structure
750-375/025-000 V02.04.xx (FW 07)
750-377/025-000 V02.04.xx (FW 07)
750-375/025-000 V02.04.xx (FW 07), PE-DAP
750-377/025-000 V02.04.xx (FW 07), PE-DAP
750-375 V02.05.xx (FW 08)
750-377 V02.05.xx (FW 08)
Interface
Port 1
Port 2
Big Endian (MOTOROLA), Subslot-controlled PR
750-375 V02.05.xx (FW 08), PE-DAP
750-377 V02.05.xx (FW 08), PE-DAP
750-375/025-000 V02.05.xx (FW 08)
750-377/025-000 V02.05.xx (FW 08)
750-375/025-000 V02.05.xx (FW 08), PE-DAP
750-377/025-000 V02.05.xx (FW 08), PE-DAP
750-375 V02.06.xx (FW 09)
Element Browser

Device Access Point ID=DIM 18: 750-377 V02.05.xx (FW 08)	
Module Ident Number	0x08000179
PNIO Version	V2.32
Information	2-port Fieldbus Coupler PROFINET IO advanced ECO for Series 750 and 753 terminalblocks; RT, IRT, MRP, PROFIenergy submodules are controlled via subslot.
Vendor Name	WAGO Kontakttechnik GmbH & Co. KG
Order Number	750-377
Hardware Version	01
Software Version	V02.05.xx (08)
Certification Info	
Conformance Class	C
Application Class	

Figure 69: Representation in the GSD Checker

Any commercially available XML editor or a normal text editor can be used to create a GSD file. With the functionality "Check" the syntax and part of the sematic is verified and errors and warnings are displayed.



```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<ISO15745Profile xmlns="http://www.profibus.com/GSDML/2003/11/DeviceProfile" xmlns:xsi=
    <ProfileHeader>
        <ProfileIdentification>PROFINET Device Profile</ProfileIdentification>
        <ProfileRevision>1.00</ProfileRevision>
        <ProfileName>Device Profile for PROFINET Devices</ProfileName>
        <ProfileSource>PROFIBUS Nutzerorganisation e. V. (PNO)</ProfileSource>
        <ProfileClassID>Device</ProfileClassID>
        <ISO15745Reference>
            <ISO15745Part>4</ISO15745Part>
            <ISO15745Edition>1</ISO15745Edition>
            <ProfileTechnology>GSDML</ProfileTechnology>
        </ISO15745Reference>
    </ProfileHeader>
    <ProfileBody>
        <DeviceIdentity VendorID="0x011D" DeviceID="0x02EE">
            <InfoText TextId="IDT_INFO_SERIES_750_753"/>
            <VendorName Value="WAGO Kontakttechnik GmbH u. Co. KG"/>
        </DeviceIdentity>
        <DeviceFunction>
            <Family MainFamily="I/O" ProductFamily="WAGO-I/O-SYSTEM 750/753"/>
        </DeviceFunction>
        <ApplicationProcess>
            <DeviceAccessPointList>
                <DeviceAccessPointItem ID="DIM 1" PhysicalSlots="0..250" ModuleIdentNum
                    <ModuleInfo>
                        <Name TextId="IDT_NAME_DIM01_0375_0203"/>
                        <InfoText TextId="IDT_INFO_DIM01_0375_0203"/>
                        <VendorName Value="WAGO Kontakttechnik GmbH &amp; Co. KG"/
                        <OrderNumber Value="750-375"/>
                        <HardwareRelease Value="01"/>
                        <SoftwareRelease Value="V02.03 xx (06)"/>
```

Category: TypeSpecific, SourceXPath: //SubmoduleItem[@ID='IDT_SUBM_0632_0000_RCENA_04BYTE']/RecordDataList/ParameterRecordDataItem[@Index='8192']/MenuList, Message: The parameter "IDP_CHANNEL_INSTANCE_01" is not referenced.
Line: 76972
Column: 9

Category: TypeSpecific, SourceXPath: //SubmoduleItem[@ID='IDT_SUBM_0632_0000_RCENA_04BYTE']/RecordDataList/ParameterRecordDataItem[@Index='8192']/MenuList, Message: The parameter "IDP_0632_REC_2000_B005_b2_01" is not referenced.
Line: 76972
Column: 9

GSDCheck 14 77 0

Figure 70: Representation as XML file and result of a check

In this chapter the configuration of a PROFINET IO-System with the CODESYS Engineering Tool (ET) is showed using a Raspberry Pi as a Controller.

First we have to download and install the CODESYS and Raspberry Pi runtime on your development PC

Download and install the free **CODESYS Development System** (https://store.codesys.com/engineering/codesys.html).

Download the **CODESYS Control for Raspberry Pi SL** (https://store.codesys.com/codesys-control-for-raspberry-pi-sl.html). Without a license this runtime will run for two hours. After two hours a restart is required. For demonstrations and training this is acceptable. A license can always be added later if required.

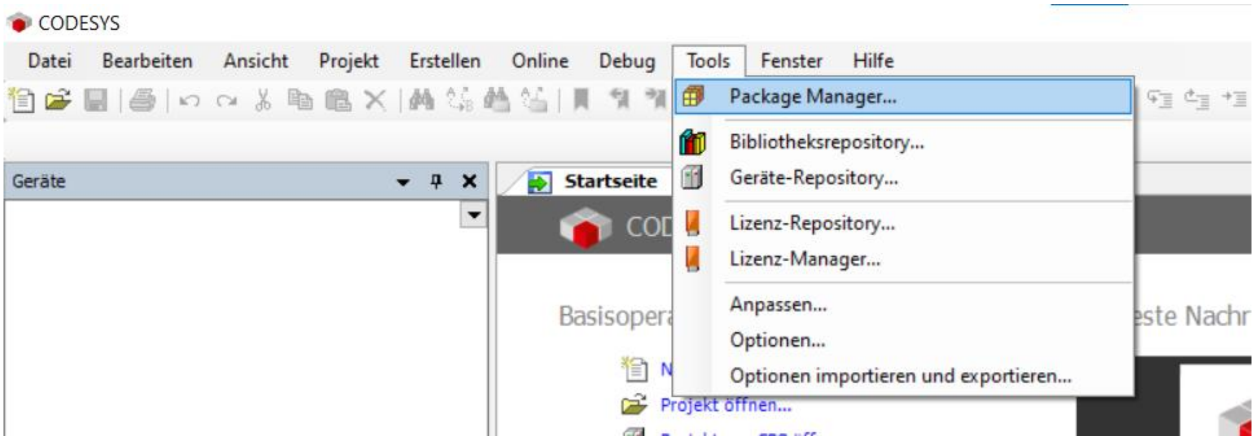
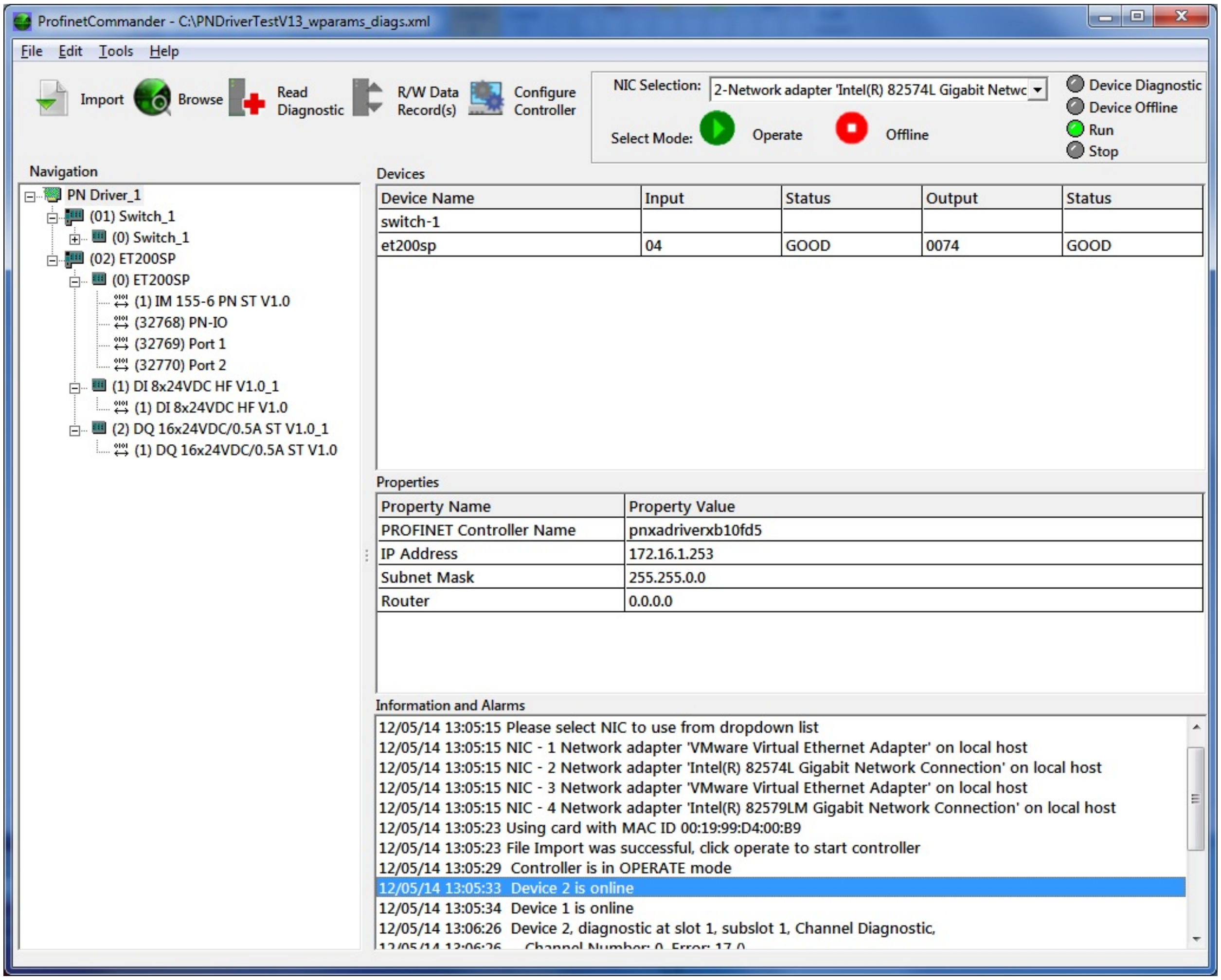


Figure 71: Select the Packer Manager to add the Raspberry Pi Controller

Open the Package Manager and install the **CODESYS Control for Raspberry Pi SL.**

PROFINET Commander runs as a PROFINET I/O controller (conformance class B, real time – RT) on a PC with an easy to use graphical user interface. With PROFINET Commander, users can test and build a PROFINET network and quickly connect I/O Devices without PLC programming.



NIC Selection: 2-Network adapter 'Intel(R) 82574L Gigabit Netwc

Select Mode: Operate

Devices

Device Name	Input	Status	Output	Status
switch-1				
et200sp	04	GOOD	0074	GOOD

Properties

Property Name	Property Value
PROFINET Controller Name	pnxadriverxb10fd5
IP Address	172.16.1.253
Subnet Mask	255.255.0.0
Router	0.0.0.0

Information and Alarms

```
12/05/14 13:05:15 Please select NIC to use from dropdown list
12/05/14 13:05:15 NIC - 1 Network adapter 'VMware Virtual Ethernet Adapter' on local host
12/05/14 13:05:15 NIC - 2 Network adapter 'Intel(R) 82574L Gigabit Network Connection' on local host
12/05/14 13:05:15 NIC - 3 Network adapter 'VMware Virtual Ethernet Adapter' on local host
12/05/14 13:05:15 NIC - 4 Network adapter 'Intel(R) 82579LM Gigabit Network Connection' on local host
12/05/14 13:05:23 Using card with MAC ID 00:19:99:D4:00:B9
12/05/14 13:05:23 File Import was successful, click operate to start controller
12/05/14 13:05:29  Controller is in OPERATE mode
12/05/14 13:05:33  Device 2 is online
12/05/14 13:05:34  Device 1 is online
12/05/14 13:06:26  Device 2, diagnostic at slot 1, subslot 1, Channel Diagnostic,
12/05/14 13:06:26     Channel Number: 0, Error: 17 0
```

Figure 72: Controlling device without PLC programming

Users gain the ability to test and set up a PROFINET IO Device, test wiring (I/O Checking) and/or systems from their PC before putting into production. For developers of PROFINET I/O devices, the added benefit of using the tool is simple testing of their product for correct operation and diagnostic functionality prior to PROFINET certification testing and final product release.

For more information and download the product see PROFINET Commander.

The PROFINET Test Bundle supports the development of PROFINET interfaces for field devices and controllers. It can be used as a means for preparation of the mandatory certification for PROFINET devices. The goal of this test system is to have all the necessary documents and test systems for RT and IRT and Security Level 1 (NetLoad) combined in one bundle with all the electronic test cases to be performed during a certification test.

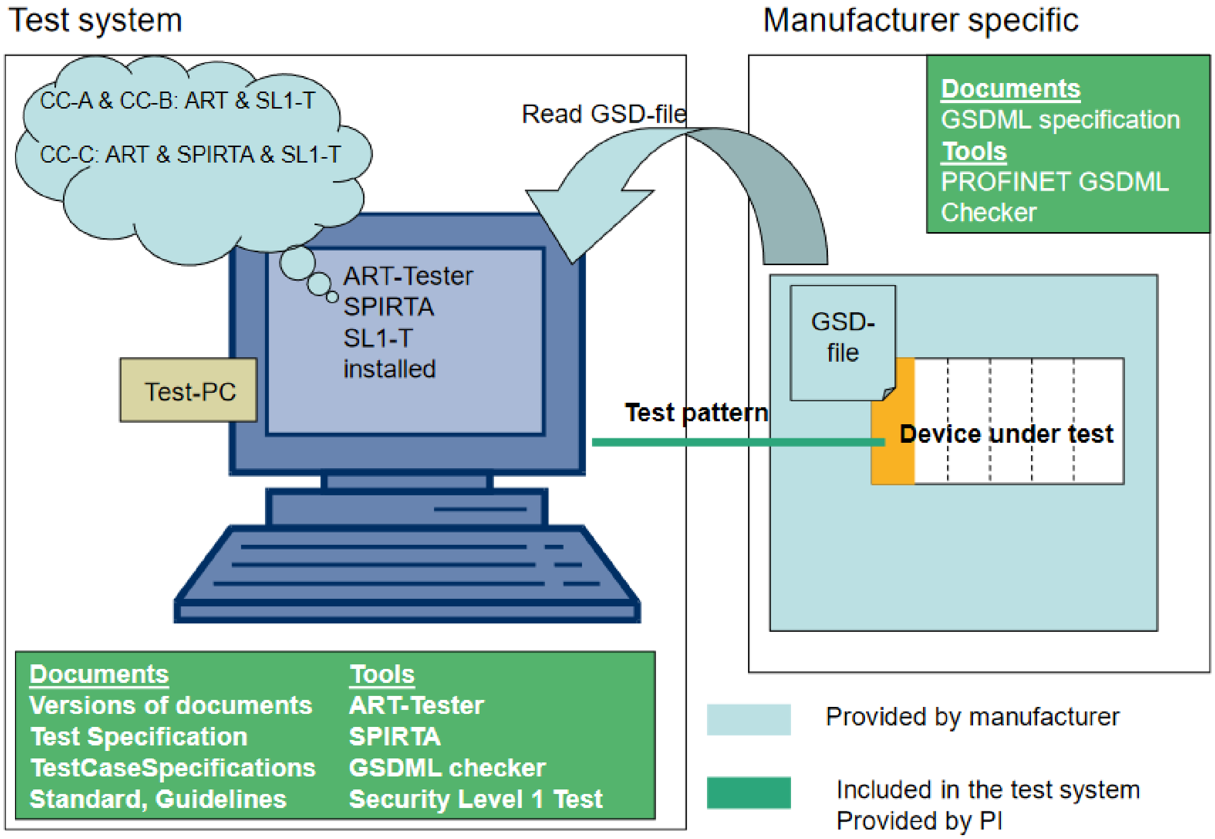


Figure 73: PROFINET Test Bundle usage

This PROFINET Test Bundle can be downloaded for free for PI Members only at PROFINET Test Bundle.

Wireshark is an open-source programme for recording Ethernet and PROFINET telegrams

With Wireshark every PC is upgraded to a Ethernet analysing device without any additional cost.

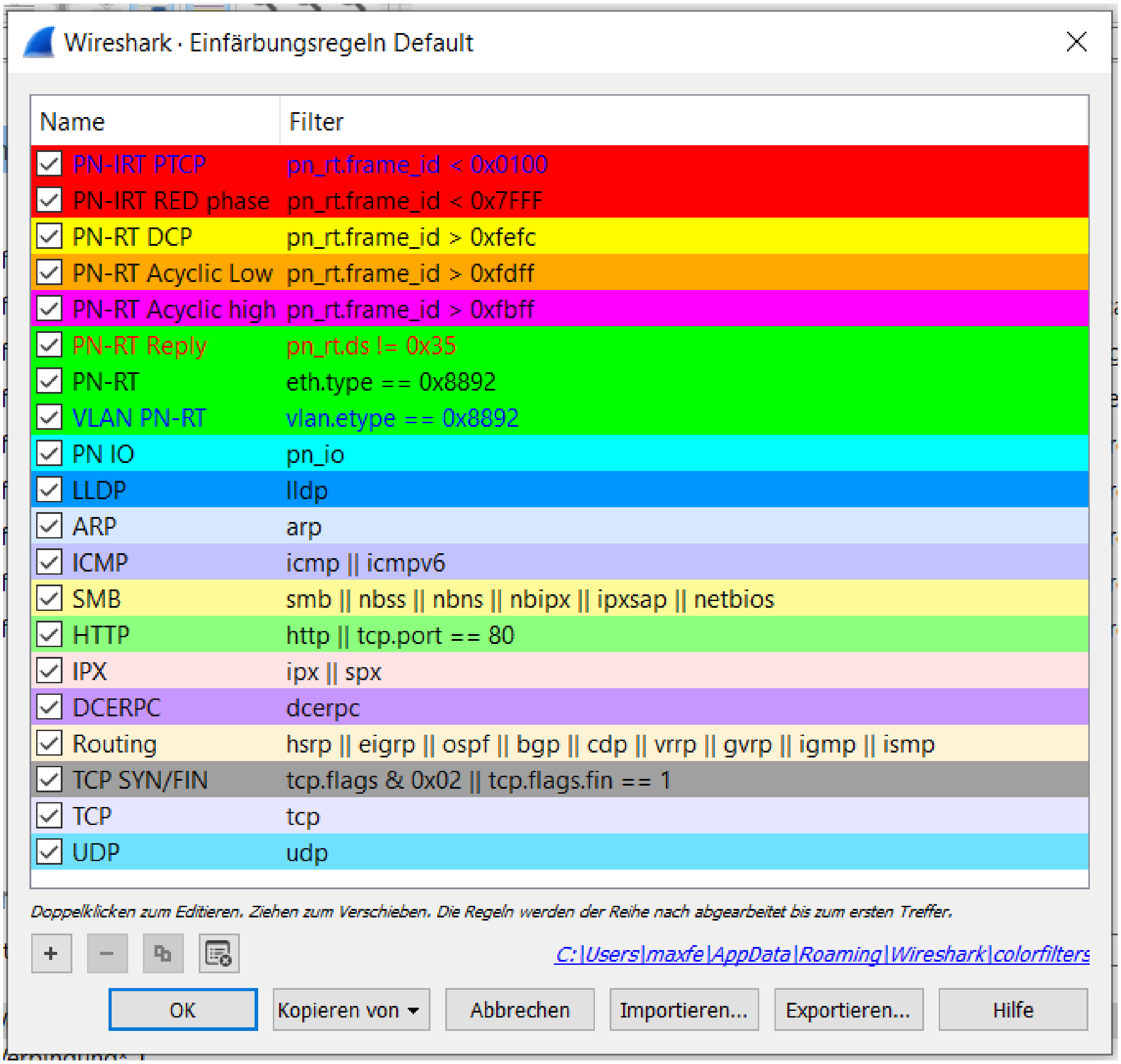
Attention: in some enterprises the installation of such a powerful analysing tool is considered as a first step for hacking and requires a special allowance.

Installation of Wireshark

1.Download the latest version of the software for your operating system from the website www.wireshark.org.

2.Install the Wireshark on your PC. Typically the driver for your Ethernet interface is included.

3.Go to coloring rules and replace the color scheme with the one for PROFINET: PROFINET IRT Colors Wireshark.ini



Wireshark · Einfärbungsregeln Default

Name	Filter
☑ PN-IRT PTCP	pn_rt.frame_id < 0x0100
☑ PN-IRT RED phase	pn_rt.frame_id < 0x7FFF
☑ PN-RT DCP	pn_rt.frame_id > 0xfefc
☑ PN-RT Acyclic Low	pn_rt.frame_id > 0xfdff
☑ PN-RT Acyclic high	pn_rt.frame_id > 0xfbff
☑ PN-RT Reply	pn_rt.ds != 0x35
☑ PN-RT	eth.type == 0x8892
☑ VLAN PN-RT	vlan.etype == 0x8892
☑ PN IO	pn_io
☑ LLDP	lldp
☑ ARP	arp
☑ ICMP	icmp || icmpv6
☑ SMB	smb || nbss || nbns || nbipx || ipxsap || netbios
☑ HTTP	http || tcp.port == 80
☑ IPX	ipx || spx
☑ DCERPC	dcerpc
☑ Routing	hsrp || eigrp || ospf || bgp || cdp || vrrp || gvrp || igmp || ismp
☑ TCP SYN/FIN	tcp.flags & 0x02 || tcp.flags.fin == 1
☑ TCP	tcp
☑ UDP	udp

Doppelklicken zum Editieren. Ziehen zum Verschieben. Die Regeln werden der Reihe nach abgearbeitet bis zum ersten Treffer.

C:\Users\maxfe\AppData\Roaming\Wireshark\colorfilters

Figure 74: Wireshark coloring rukes for PROFINET

The Software has the following functions.

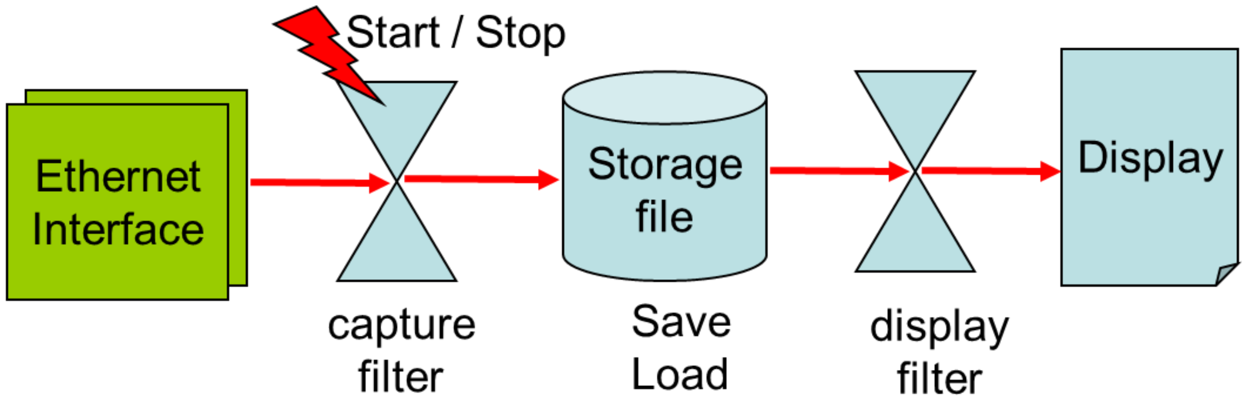
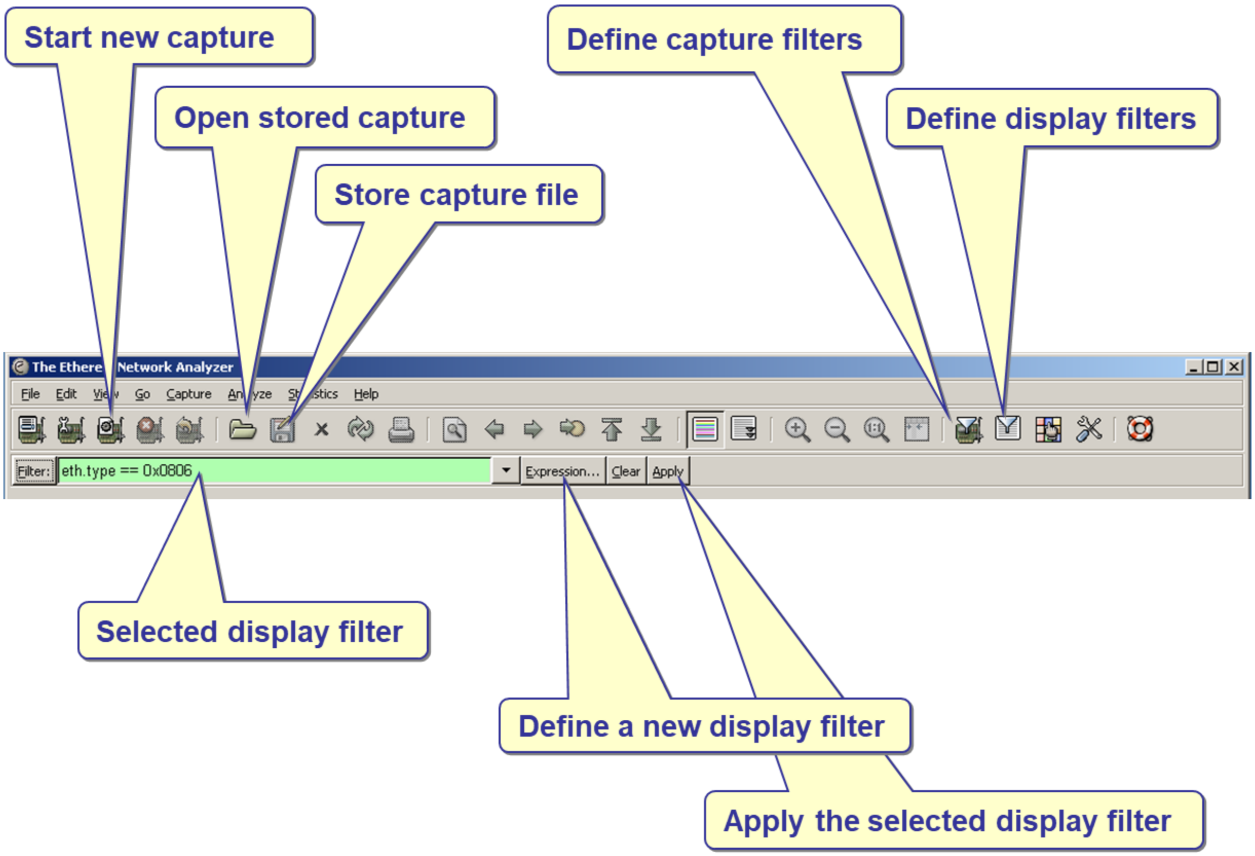


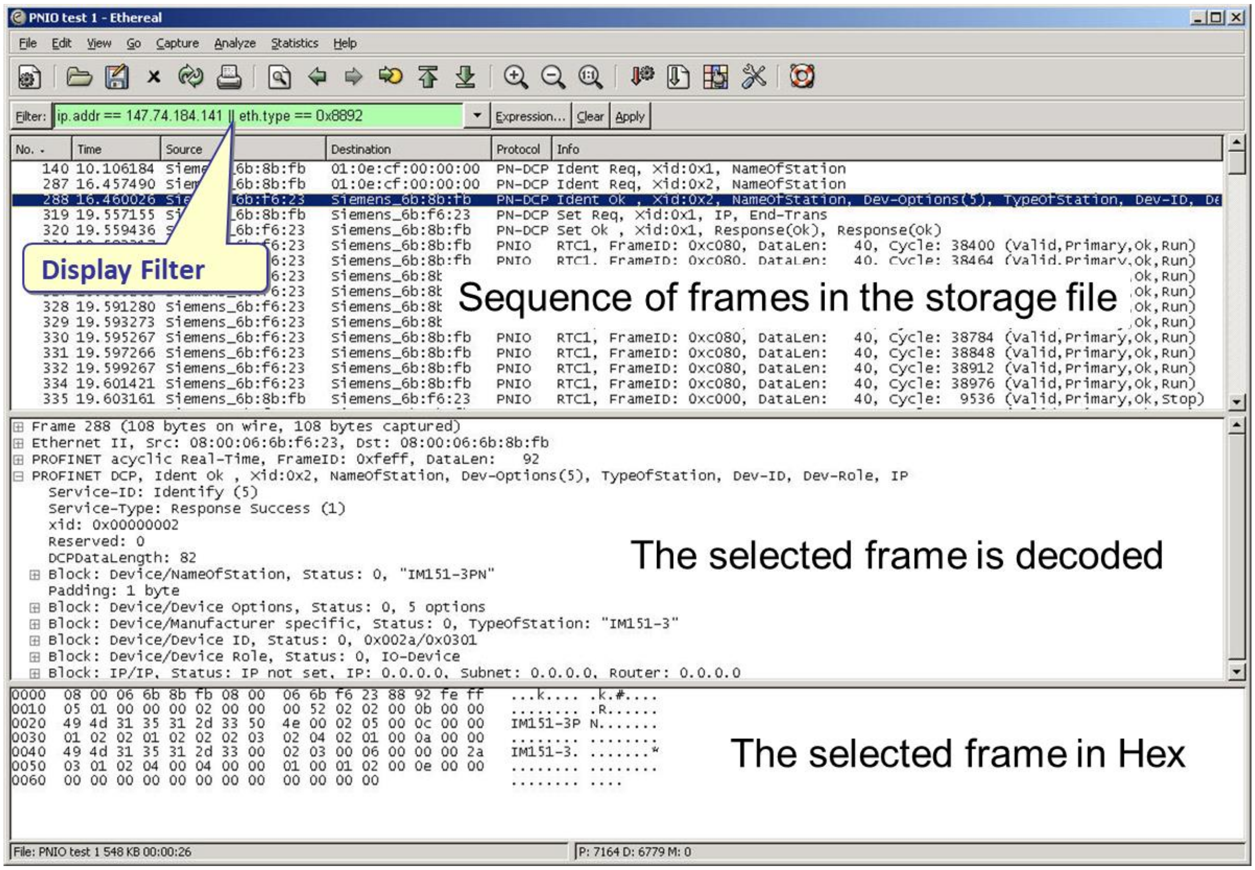
Figure 75: Functionalities of Wireshark



Filter: eth.type == 0x0806

Figure 76: Functions of Wireshark

And you see the following windows:


PNIO test 1 - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
ip.addr == 147.74.184.141 || eth.type == 0x8892
```

```
No. -  Time      Source            Destination        Protocol Info
   140 10.106184 Sieme   6b:8b:fb  01:0e:cf:00:00:00  PN-DCP  Ident Req, Xid:0x1, NameOfStation
   287 16.457490 Siem    6b:8b:fb  01:0e:cf:00:00:00  PN-DCP  Ident Req, Xid:0x2, NameOfStation
   288 16.460026 Sie     6b:f6:23  Siemens_6b:8b:fb   PN-DCP  Ident Ok , Xid:0x2, NameOfStation, Dev-Options(5), TypeOfStation, Dev-ID, De
   319 19.557155 Si      6b:8b:fb  Siemens_6b:f6:23   PN-DCP  Set Req, Xid:0x1, IP, End-Trans
   320 19.559436 S       6b:f6:23  Siemens_6b:8b:fb   PN-DCP  Set Ok , Xid:0x1, Response(Ok), Response(Ok)
                         6:23      Siemens_6b:8b:fb   PNIO    RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38400 (Valid,Primary,Ok,Run)
                         6:23      Siemens_6b:8b:fb   PNIO    RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38464 (Valid,Primary,Ok,Run)
                         6:23      Siemens_6b:8k                                                                      ,Ok,Run)
                         6:23      Siemens_6b:8k                                                                      ,Ok,Run)
   328 19.591280 Siemens_6b:f6:23  Siemens_6b:8k                                                                      ,Ok,Run)
   329 19.593273 Siemens_6b:f6:23  Siemens_6b:8k                                                                      ,Ok,Run)
   330 19.595267 Siemens_6b:f6:23  Siemens_6b:8b:fb   PNIO    RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38784 (Valid,Primary,Ok,Run)
   331 19.597266 Siemens_6b:f6:23  Siemens_6b:8b:fb   PNIO    RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38848 (Valid,Primary,Ok,Run)
   332 19.599267 Siemens_6b:f6:23  Siemens_6b:8b:fb   PNIO    RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38912 (Valid,Primary,Ok,Run)
   334 19.601421 Siemens_6b:f6:23  Siemens_6b:8b:fb   PNIO    RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38976 (Valid,Primary,Ok,Run)
   335 19.603161 Siemens_6b:8b:fb  Siemens_6b:f6:23   PNIO    RTC1, FrameID: 0xc000, DataLen:   40, Cycle:  9536 (Valid,Primary,Ok,Stop)
```

```
⊞ Frame 288 (108 bytes on wire, 108 bytes captured)
⊞ Ethernet II, Src: 08:00:06:6b:f6:23, Dst: 08:00:06:6b:8b:fb
⊞ PROFINET acyclic Real-Time, FrameID: 0xfeff, DataLen:   92
⊟ PROFINET DCP, Ident Ok , Xid:0x2, NameOfStation, Dev-Options(5), TypeOfStation, Dev-ID, Dev-Role, IP
    Service-ID: Identify (5)
    Service-Type: Response Success (1)
    Xid: 0x00000002
    Reserved: 0
    DCPDataLength: 82
  ⊞ Block: Device/NameOfStation, Status: 0, "IM151-3PN"
    Padding: 1 byte
  ⊞ Block: Device/Device Options, Status: 0, 5 options
  ⊞ Block: Device/Manufacturer specific, Status: 0, TypeOfStation: "IM151-3"
  ⊞ Block: Device/Device ID, Status: 0, 0x002a/0x0301
  ⊞ Block: Device/Device Role, Status: 0, IO-Device
  ⊞ Block: IP/IP, Status: IP not set, IP: 0.0.0.0, Subnet: 0.0.0.0, Router: 0.0.0.0
```

```
0000  08 00 06 6b 8b fb 08 00  06 6b f6 23 88 92 fe ff   ...k.... .k.#....
0010  05 01 00 00 00 02 00 00  00 52 02 02 00 0b 00 00   ........ .R......
0020  49 4d 31 35 31 2d 33 50  4e 00 02 05 00 0c 00 00   IM151-3P N.......
0030  01 02 02 01 02 02 02 03  02 04 02 01 00 0a 00 00   ........ ........
0040  49 4d 31 35 31 2d 33 00  02 03 00 06 00 00 00 2a   IM151-3. .......*
0050  03 01 02 04 00 04 00 00  01 00 01 02 00 0e 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

```
File: PNIO test 1 548 KB 00:00:26        P: 7164 D: 6779 M: 0
```

Display Filter

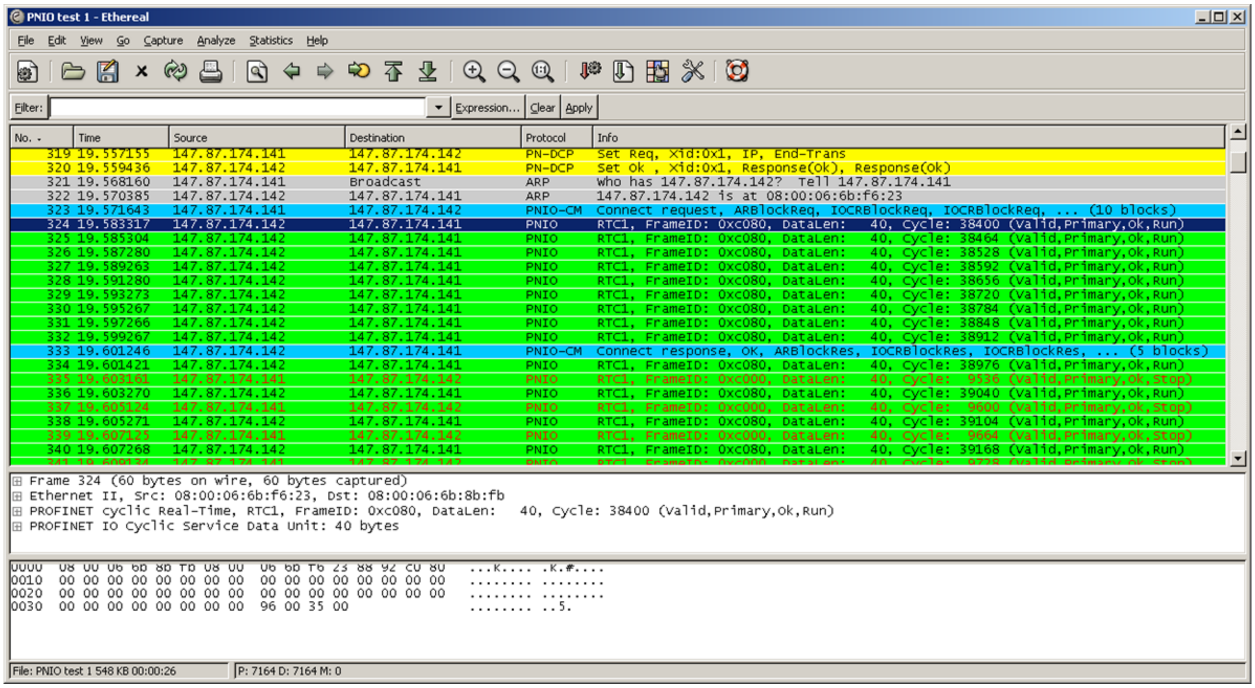
Sequence of frames in the storage file

The selected frame is decoded

The selected frame in Hex

Figure 77: Display of Wireshark

If you use the PROFINET colors the cyclic data will be green.


PNIO test 1 - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. -  Time      Source          Destination     Protocol Info
   319 19.557155 147.87.174.141  147.87.174.142  PN-DCP   Set Req, Xid:0x1, IP, End-Trans
   320 19.559436 147.87.174.142  147.87.174.141  PN-DCP   Set Ok , Xid:0x1, Response(Ok), Response(Ok)
   321 19.568160 147.87.174.141  Broadcast       ARP      Who has 147.87.174.142?  Tell 147.87.174.141
   322 19.570385 147.87.174.142  147.87.174.141  ARP      147.87.174.142 is at 08:00:06:6b:f6:23
   323 19.571643 147.87.174.141  147.87.174.142  PNIO-CM  Connect request, ARBlockReq, IOCRBlockReq, IOCRBlockReq, ... (10 blocks)
   324 19.583317 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38400 (Valid,Primary,Ok,Run)
   325 19.585304 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38464 (Valid,Primary,Ok,Run)
   326 19.587280 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38528 (Valid,Primary,Ok,Run)
   327 19.589263 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38592 (Valid,Primary,Ok,Run)
   328 19.591280 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38656 (Valid,Primary,Ok,Run)
   329 19.593273 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38720 (Valid,Primary,Ok,Run)
   330 19.595267 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38784 (Valid,Primary,Ok,Run)
   331 19.597266 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38848 (Valid,Primary,Ok,Run)
   332 19.599267 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38912 (Valid,Primary,Ok,Run)
   333 19.601246 147.87.174.142  147.87.174.141  PNIO-CM  Connect response, OK, ARBlockRes, IOCRBlockRes, IOCRBlockRes, ... (5 blocks)
   334 19.601421 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38976 (Valid,Primary,Ok,Run)
   335 19.603161 147.87.174.141  147.87.174.142  PNIO     RTC1, FrameID: 0xc000, DataLen:   40, Cycle:  9536 (Valid,Primary,Ok,Stop)
   336 19.603270 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 39040 (Valid,Primary,Ok,Run)
   337 19.605124 147.87.174.141  147.87.174.142  PNIO     RTC1, FrameID: 0xc000, DataLen:   40, Cycle:  9600 (Valid,Primary,Ok,Stop)
   338 19.605271 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 39104 (Valid,Primary,Ok,Run)
   339 19.607125 147.87.174.141  147.87.174.142  PNIO     RTC1, FrameID: 0xc000, DataLen:   40, Cycle:  9664 (Valid,Primary,Ok,Stop)
   340 19.607268 147.87.174.142  147.87.174.141  PNIO     RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 39168 (Valid,Primary,Ok,Run)
   341 19.609134 147.87.174.141  147.87.174.142  PNIO     RTC1, FrameID: 0xc000, DataLen:   40, Cycle:  9728 (Valid,Primary,Ok,Stop)
```

```
⊞ Frame 324 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 08:00:06:6b:f6:23, Dst: 08:00:06:6b:8b:fb
⊞ PROFINET cyclic Real-Time, RTC1, FrameID: 0xc080, DataLen:   40, Cycle: 38400 (Valid,Primary,Ok,Run)
⊞ PROFINET IO Cyclic Service Data Unit: 40 bytes
```

```
0000  08 00 06 6b 8b fb 08 00  06 6b f6 23 88 92 c0 80   ...k.... .k.#....
0010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  96 00 35 00               ........ ..5.
```

```
File: PNIO test 1 548 KB 00:00:26        P: 7164 D: 7164 M: 0
```

Figure 78: Frames are colored according the PROFINET Manual

Display and capture filters have not the same syntax!

•Only PROFINET frames:

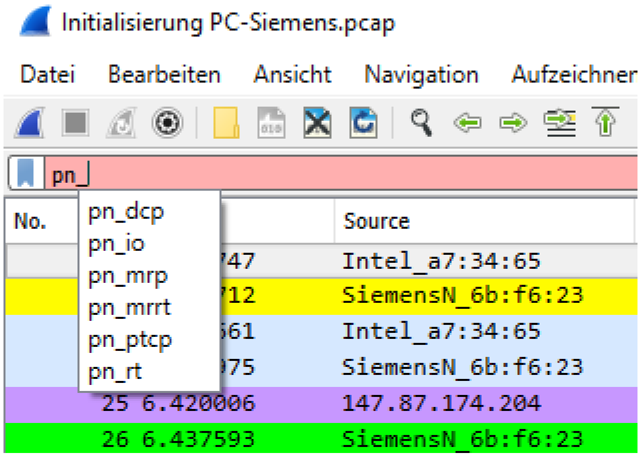
pn_rt (same as eth.type == 0x8892)

•Only frames related with the PROFINET IO protocol

pn_io (does not include e.g. the DCP protocol)

•All PROFINET frames

pn_rt or pn_io

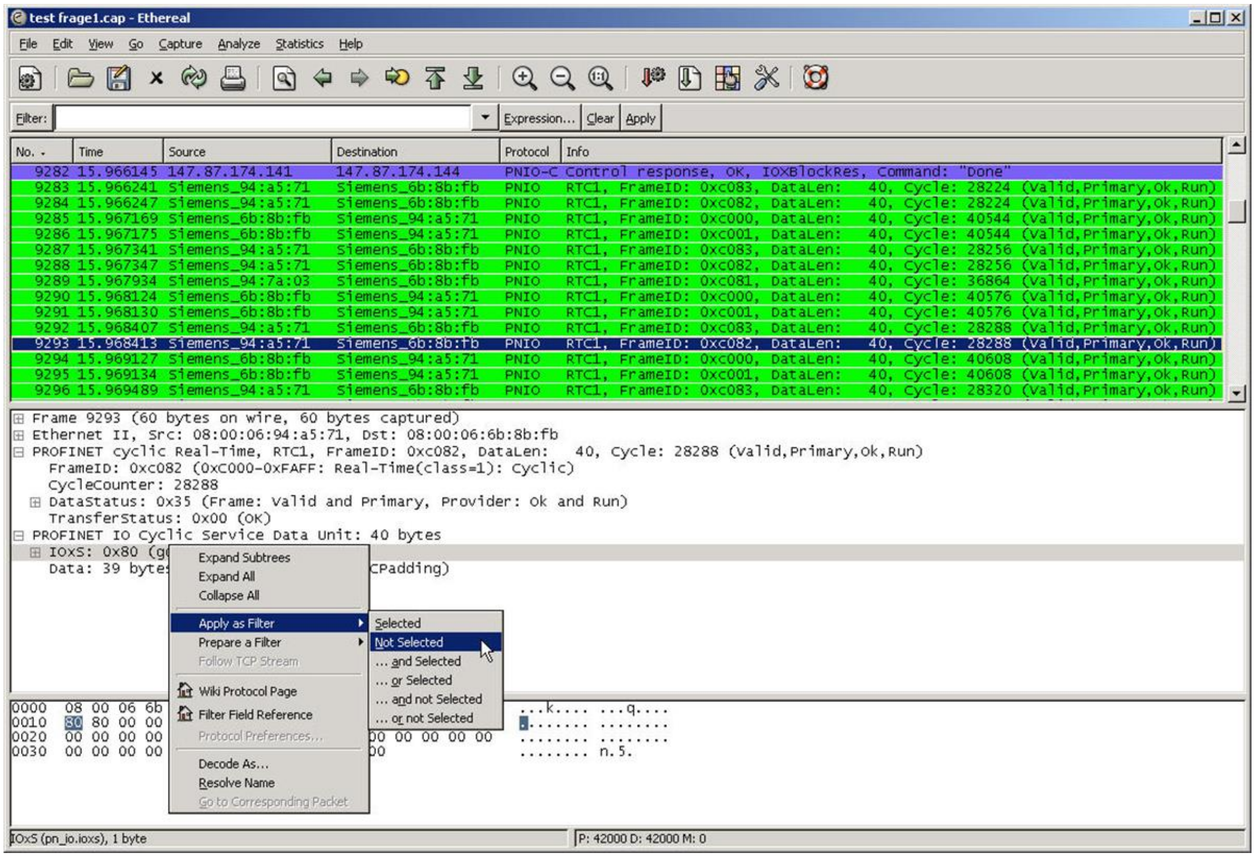


pn_

```
No.                       Source
                    747   Intel_a7:34:65
                    712   SiemensN_6b:f6:23
                    561   Intel_a7:34:65
                    975   SiemensN_6b:f6:23
 25 6.420006              147.87.174.204
 26 6.437593              SiemensN_6b:f6:23
```

Figure 79: Display filter of the PROFINET protocols

More specific filters can be set by selecting the protocol element.



```
No. -  Time      Source           Destination      Protocol Info
  9282 15.966145 147.87.174.141   147.87.174.144   PNIO-C Control response, OK, IOXBlockRes, Command: "Done"
  9283 15.966241 Siemens_94:a5:71 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc083, DataLen:   40, Cycle: 28224 (Valid,Primary,Ok,Run)
  9284 15.966247 Siemens_94:a5:71 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc082, DataLen:   40, Cycle: 28224 (Valid,Primary,Ok,Run)
  9285 15.967169 Siemens_6b:8b:fb Siemens_94:a5:71 PNIO   RTC1, FrameID: 0xc000, DataLen:   40, Cycle: 40544 (Valid,Primary,Ok,Run)
  9286 15.967175 Siemens_6b:8b:fb Siemens_94:a5:71 PNIO   RTC1, FrameID: 0xc001, DataLen:   40, Cycle: 40544 (Valid,Primary,Ok,Run)
  9287 15.967341 Siemens_94:a5:71 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc083, DataLen:   40, Cycle: 28256 (Valid,Primary,Ok,Run)
  9288 15.967347 Siemens_94:a5:71 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc082, DataLen:   40, Cycle: 28256 (Valid,Primary,Ok,Run)
  9289 15.967934 Siemens_94:7a:03 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc081, DataLen:   40, Cycle: 36864 (Valid,Primary,Ok,Run)
  9290 15.968124 Siemens_6b:8b:fb Siemens_94:a5:71 PNIO   RTC1, FrameID: 0xc000, DataLen:   40, Cycle: 40576 (Valid,Primary,Ok,Run)
  9291 15.968130 Siemens_6b:8b:fb Siemens_94:a5:71 PNIO   RTC1, FrameID: 0xc001, DataLen:   40, Cycle: 40576 (Valid,Primary,Ok,Run)
  9292 15.968407 Siemens_94:a5:71 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc083, DataLen:   40, Cycle: 28288 (Valid,Primary,Ok,Run)
  9293 15.968413 Siemens_94:a5:71 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc082, DataLen:   40, Cycle: 28288 (Valid,Primary,Ok,Run)
  9294 15.969127 Siemens_6b:8b:fb Siemens_94:a5:71 PNIO   RTC1, FrameID: 0xc000, DataLen:   40, Cycle: 40608 (Valid,Primary,Ok,Run)
  9295 15.969134 Siemens_6b:8b:fb Siemens_94:a5:71 PNIO   RTC1, FrameID: 0xc001, DataLen:   40, Cycle: 40608 (Valid,Primary,Ok,Run)
  9296 15.969489 Siemens_94:a5:71 Siemens_6b:8b:fb PNIO   RTC1, FrameID: 0xc083, DataLen:   40, Cycle: 28320 (Valid,Primary,Ok,Run)
```

```
⊞ Frame 9293 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 08:00:06:94:a5:71, Dst: 08:00:06:6b:8b:fb
⊟ PROFINET cyclic Real-Time, RTC1, FrameID: 0xc082, DataLen:   40, Cycle: 28288 (Valid,Primary,Ok,Run)
    FrameID: 0xc082 (0xC000-0xFAFF: Real-Time(class=1): Cyclic)
    CycleCounter: 28288
  ⊞ DataStatus: 0x35 (Frame: Valid and Primary, Provider: Ok and Run)
    TransferStatus: 0x00 (OK)
⊟ PROFINET IO Cyclic Service Data Unit: 40 bytes
  ⊞ IOxS: 0x80 (g
    Data: 39 bytes                     CPadding)
```

```
0000  08 00 06 6b                 00 00 00 00 00   ...k.... ...q....
0010  80 80 00 00                                  ........ ........
0020  00 00 00 00                                  ........ ........
0030  00 00 00 00                 00               ........ n.5.
```

```
IOxS (pn_io.ioxs), 1 byte                P: 42000 D: 42000 M: 0
```

Figure 80: Simple definition of display filters