



## **DF PROFINET IO**

### **GettingStarted**

V1.0/18.07.2016

## Änderungshistorie

Version	Datum	Beschreibungen	Verantw.
V1.0	18.07.2016	Initial Version	

Version	Produktmanager	Projektleiter Software
V1.0	Joachim Kurpat	Christoph Vogt

**KUNBUS** GmbH  
Heerweg 15c  
73770 Denkendorf  
Phone +49 711 300 20 676  
Fax +49 711 300 20 677

Copyright © 2016 by **KUNBUS** GmbH

### Business Confidential/KUNBUS Proprietary

This document includes data that shall not be duplicated, used, or disclosed - in whole or in part - for any purpose other than to evaluate this document. If, however, a contract with a customer is in force, the customer shall have the right to duplicate, use, or disclose the data to the extent provided in this contract. This restriction does not limit the customer's right to use the data in this document if it can also be obtained from another source without restriction. The data subject to this restriction are confidential in all pages of this document.

# Inhalt

1	Installation .....	1
2	Betrieb als PROFINET IO-Controller.....	2
2.1	PROFINET IO-Konfiguration.....	2
2.2	Beschreibung der LED's .....	3
2.3	PROFINET IO Controller C und C++ Beispiel .....	4
3	Betrieb als PROFINET IO-Device .....	9
3.1	PROFINET IO Device C und C++ Beispiel.....	10

## Abbildungsverzeichnis

Abbildung 1: Hardware Installation.....	1
Abbildung 2: <i>Configurator III</i> PROFINET IO Konfigurations-Tool .....	2
Abbildung 3: <i>Configurator III</i> PROFINET IO Online-Betriebsart.....	3
Abbildung 4: Beispielprogramm PN IO Controller - Initialisierung.....	4
Abbildung 5: Beispielprogramm PN IO Controller - Kommandoübersicht.....	5
Abbildung 6: Beispielprogramm PN IO Controller - Identify .....	5
Abbildung 7: Beispielprogramm PN IO Controller - Cyclic Transfer.....	6
Abbildung 8: Beispielprogramm PN IO Controller - Cyclic Transfer .....	7
Abbildung 9: Beispielprogramm PN IO Controller - Cyclic Transfer.....	7
Abbildung 10: Beispielprogramm PN IO Controller - Acyclic Transfer .....	8
Abbildung 11: Beispielprogramm PN IO Device - Initialisierung .....	10
Abbildung 12: Beispielprogramm PN IO Device – Kommandoübersicht.....	11
Abbildung 13: Zustand des PN IO Device am angeschlossenen PN IO Controller vor der Aktivierung.....	12
Abbildung 14: Beispielprogramm PN IO Device – Start des Profinet.....	12
Abbildung 15: Zustand des PN IO Device am angeschlossenen PN IO Controller nach der Aktivierung.....	13
Abbildung 16: Beispielprogramm PN IO Device – Identifizierung der verbundenen PN IO Controller .....	13
Abbildung 17: Beispielprogramm PN IO Device - Start PN IO Device und Lesen/Schreiben der zyklischen Prozessdaten .....	14
Abbildung 18: Geänderte Output-Daten am PROFINET IO Controller .....	14
Abbildung 19: Beispielprogramm PN IO Device - Wiederholtes Lesen/Schreiben der zyklischen Prozessdaten und Anzeige der am PROFINET IO Controller geänderten Output-Daten .....	15
Abbildung 20: Beispielprogramm PN IO Device - Command 4- Pull/Plug Module .....	15
Abbildung 21: Anzeige der vom PROFINET IO Device gesendeten Pull / Plug – Alarme am PROFINET IO Controller.....	16
Abbildung 22: Beispielprogramm PN IO Device - Command 5- Senden eines Process- Alarms .....	16
Abbildung 23: Anzeige des vom PROFINET IO Device gesendeten Process – Alarms am PROFINET IO Controller.....	17
Abbildung 24: Beispielprogramm PN IO Device - Command 6- Setzen/Rücksetzen einer Diagnosemeldung .....	17
Abbildung 25: Anzeige der vom PROFINET IO Device gesendeten Diagnosemeldung am PROFINET IO Controller.....	18

Leerseite



# 1 Installation

- Installieren Sie die DF PROFINET IO PCI/CPCI/PCIe - Baugruppe in Ihrem PC-System.

**Bitte beachten Sie, falls Sie eine DF PROFINET IO CPCI Baugruppe verwenden, dass diese kein Hot Plugging unterstützt, d.h. sie darf nur in ausgeschalteten und von der Versorgungsspannung getrennten Compact PCI-Systemen ein- und ausgebaut werden.**

- Schalten Sie das PC-System ein.
- Wenn der Windows „Neue Hardware“ – Dialog, welcher automatisch durch die Installation der Baugruppe geöffnet wird, erscheint, brauchen Sie nicht darauf zu reagieren. Der Treiber wird automatisch geladen.

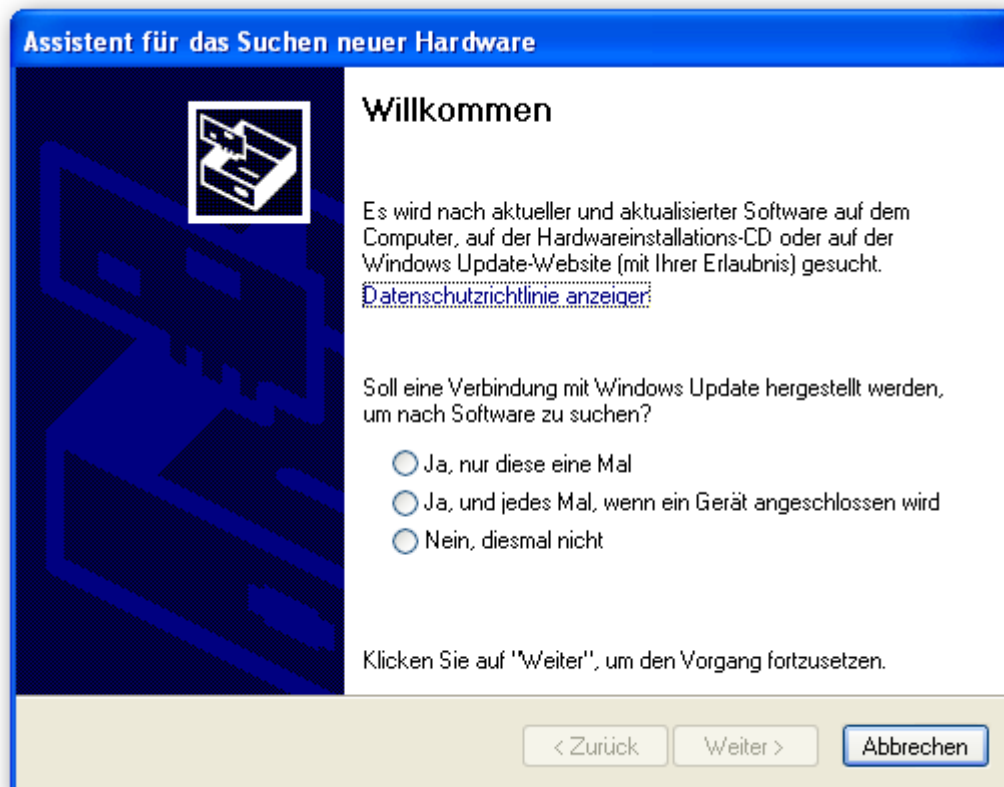


Abbildung 1: Hardware Installation

- Starten Sie das Setup von der **KUNBUS** Treiber - CD, welche sich im Lieferumfang befindet. Dieses Setup installiert alle notwendige Software einschließlich Treiber.

## 2 Betrieb als PROFINET IO-Controller

### 2.1 PROFINET IO-Konfiguration

Die PROFINET IO-Konfiguration wird mit dem **KUNBUS** Profibus Konfigurations-Tool *Configurator III* durchgeführt. *Configurator III* ist ein mächtiges Werkzeug um PROFINET IO - Konfigurationen zu erstellen, downzuloaden und online zu testen. Eine detaillierte Beschreibung finden Sie im On-Line-Hilfesystem von *Configurator III*.

- Erstellen Sie die PROFINET IO Konfiguration und laden Sie sie auf die DF PROFINET IO – Baugruppe.

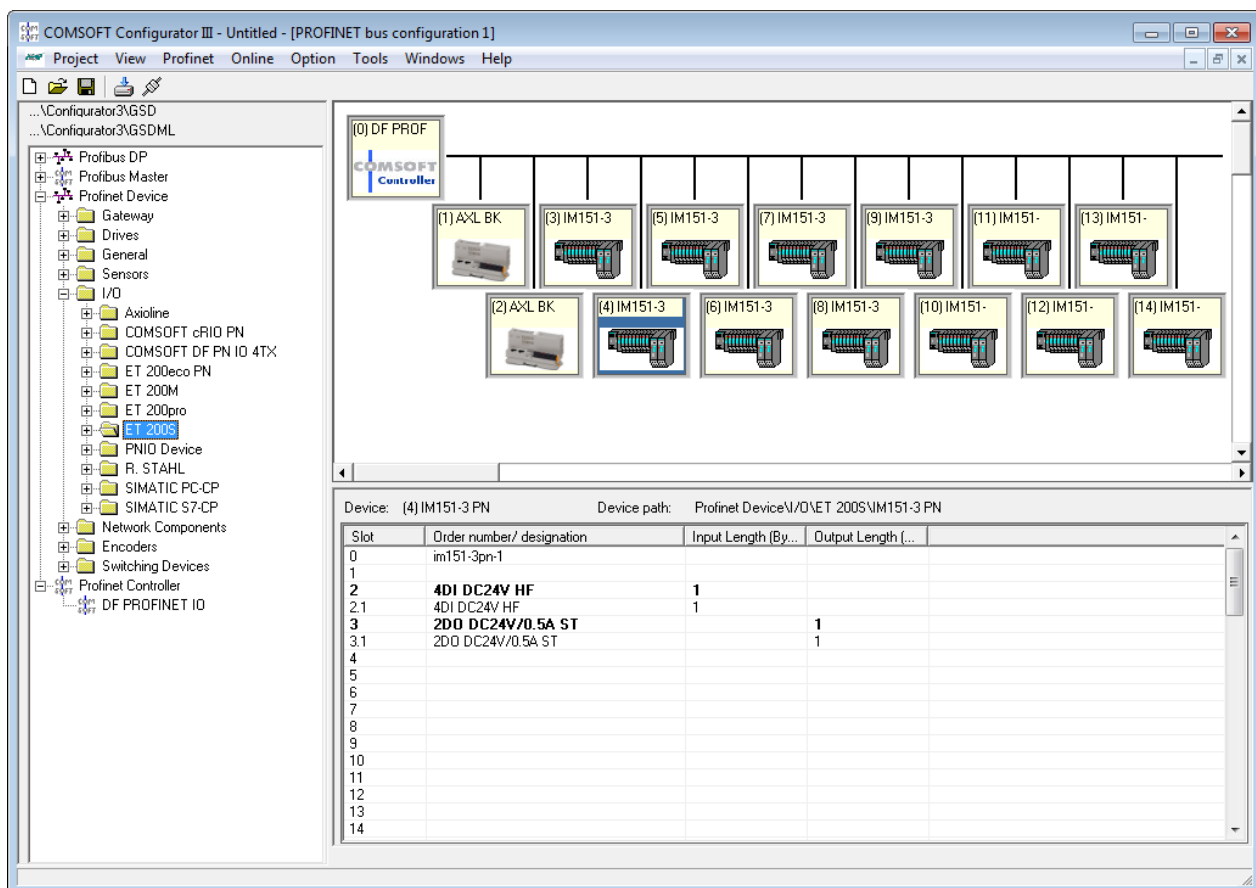


Abbildung 2: *Configurator III* PROFINET IO Konfigurations-Tool



Falls die PROFINET IO Devices angeschlossen sind, kann mit der Online-Betriebsart die Konfiguration sofort getestet werden und ev. Konfigurationsfehler beseitigt werden.

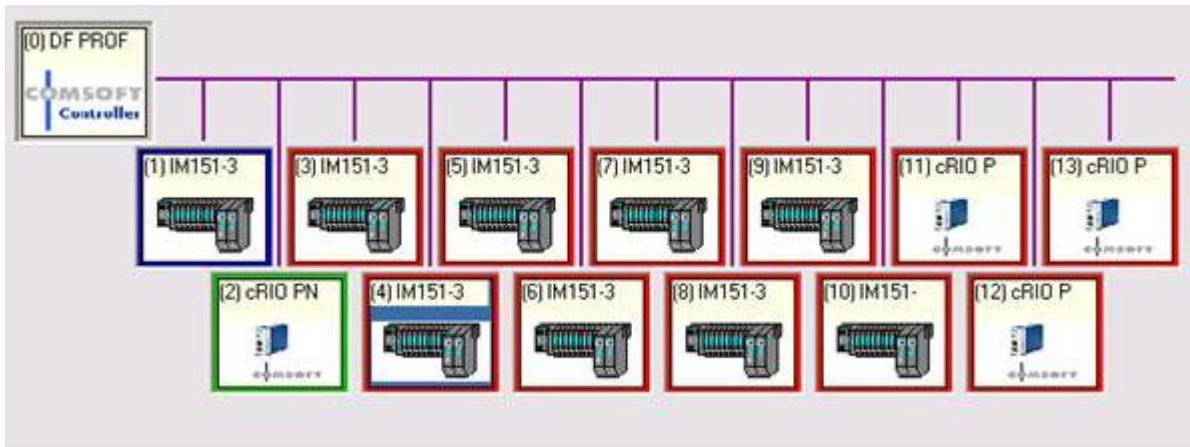


Abbildung 3: Configurator III PROFINET IO Online-Betriebsart

## 2.2 Beschreibung der LED's

● Grüne LED:

An: Firmware geladen und gestartet

Aus: Firmware nicht geladen

● Gelbe LED:

An: PROFINET IO gestartet

Aus: PROFINET IO gestoppt

● Rote LED:

An: Fehler im PROFINET IO Netzwerk (min. ein konfiguriertes PN IO Device antwortet nicht oder meldet Diagnose)

Aus: Kein Fehler im PROFINET IO Netzwerk

## 2.3 PROFINET IO Controller C und C++ Beispiel

Dieses Beispiel zeigt exemplarisch und auf einfache Weise wie das Interface der Baugruppe zu nutzen ist. Zuvor muss mit *Configurator III* eine Konfiguration auf die Karte geladen werden.

Das Beispielprogramm befindet sich im Verzeichnis

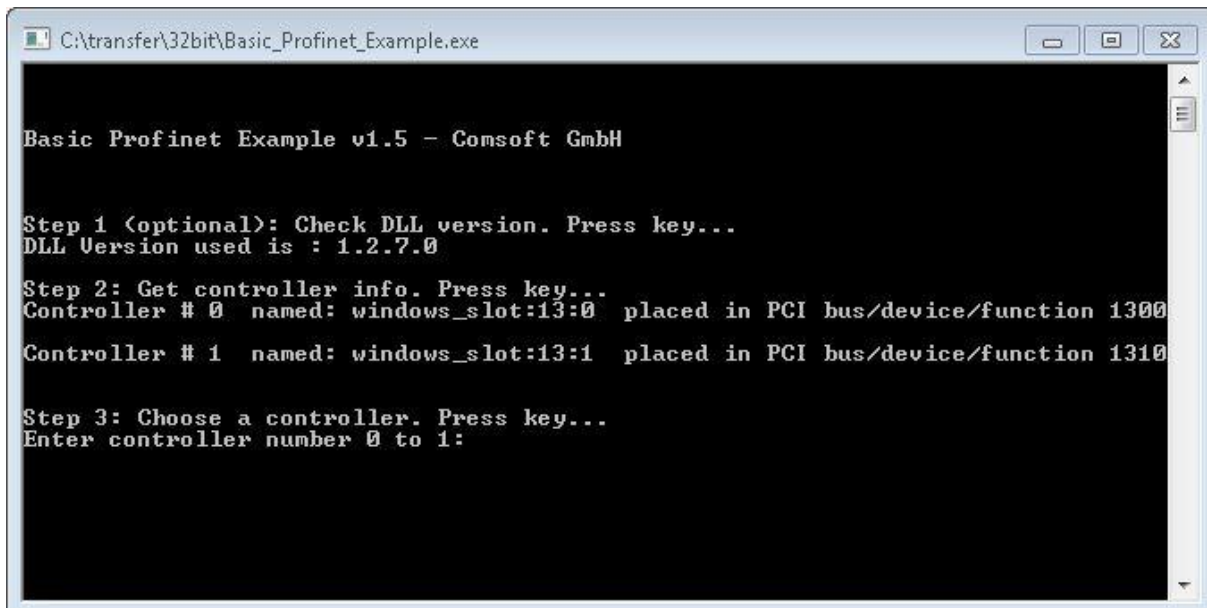
"C:\Programme\KUNBUS GmbH\Profinet IO Controller\Basic\_Profinet\_Example".

oder

"C:\Programme (x86)\KUNBUS GmbH\Profinet IO Controller\Basic\_Profinet\_Example".

und kann ab Visual C++ 2008 direkt editiert und übersetzt werden. Es handelt sich um eine 32-Bit Konsolapplikation, welche die einzelnen notwendigen Schritte nacheinander exemplarisch abarbeitet. Weitere Informationen über das Programm finden Sie im Sourcecode selbst. Der C-Code ist im File "Basic\_Profinet\_Example.cpp" enthalten. Für Test und Inbetriebnahme des Profibus-Netzwerkes bietet der *Configurator III* eine sehr leistungsfähige Online Betriebsart mit vollgraphischer Oberfläche und vielen Funktionen für Datenaustausch und Klartextdiagnose.

Die bereits mitgelieferte "Basic\_Profinet\_Example.exe" kann direkt verwendet werden:



```
C:\transfer\32bit\Basic_Profinet_Example.exe

Basic Profinet Example v1.5 - Comsoft GmbH

Step 1 (optional): Check DLL version. Press key...
DLL Version used is : 1.2.7.0

Step 2: Get controller info. Press key...
Controller # 0 named: windows_slot:13:0 placed in PCI bus/device/function 1300
Controller # 1 named: windows_slot:13:1 placed in PCI bus/device/function 1310

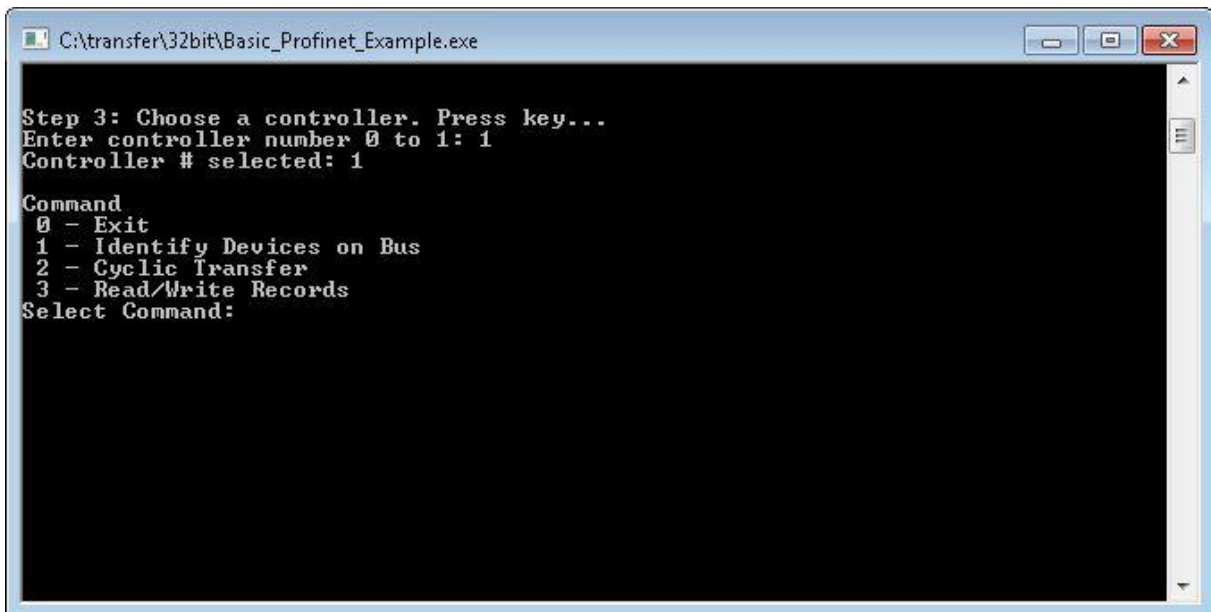
Step 3: Choose a controller. Press key...
Enter controller number 0 to 1:
```

Abbildung 4: Beispielprogramm PN IO Controller - Initialisierung

Das Programm läuft Schritt für Schritt ab. Mit beliebigem Tastendruck wird weitersprungen. Folgende Schritte sind verfügbar:

- Step 1: *Check DLL version*: Initialisierung der DFX-DLL und Generierung eines File Handles für den Baugruppenzugriff
- Step 2: *Get controller info*: Lesen der Installierten Baugruppenkonfiguration

- Step 3: *Choose a controller*: Auswahl einer installierten Baugruppe
- LED Grün = An, LED Gelb = Aus, LED Rot = Aus



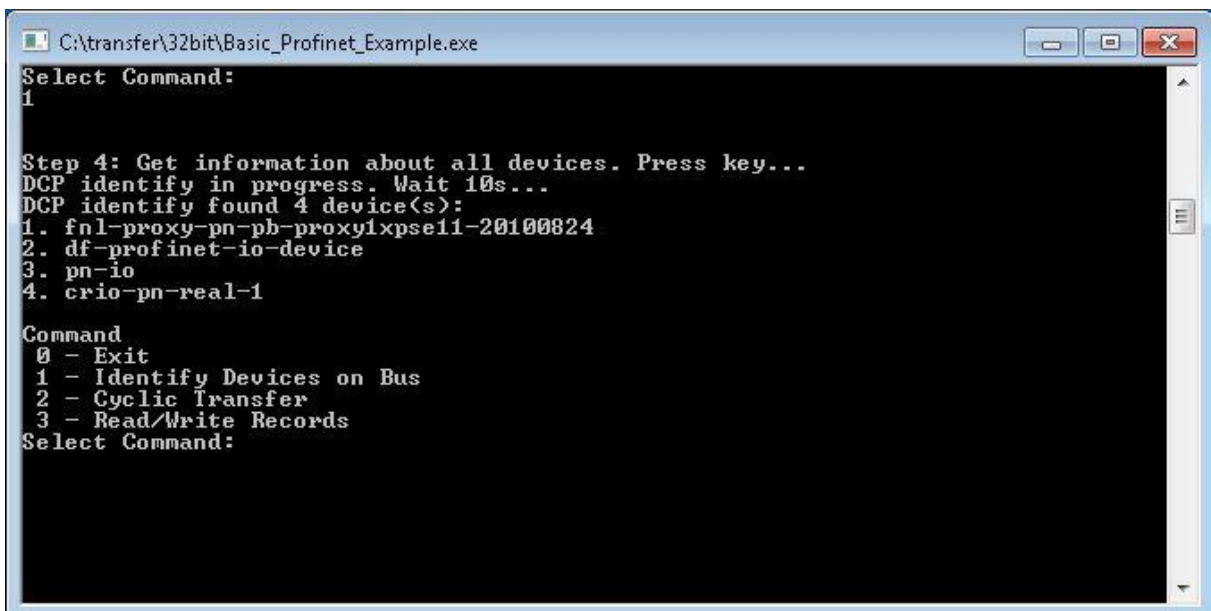
```
C:\transfer\32bit\Basic_Profinet_Example.exe

Step 3: Choose a controller. Press key...
Enter controller number 0 to 1: 1
Controller # selected: 1

Command
0 - Exit
1 - Identify Devices on Bus
2 - Cyclic Transfer
3 - Read/Write Records
Select Command:
```

Abbildung 5: Beispielprogramm PN IO Controller - Kommandoübersicht

- Command 0: *Exit*: Beispielprogramm wird beendet
- Command 1: *Identify Devices on Bus*: Anzeige aller erreichbaren PN IO Devices im Netzwerk



```
C:\transfer\32bit\Basic_Profinet_Example.exe

Select Command:
1

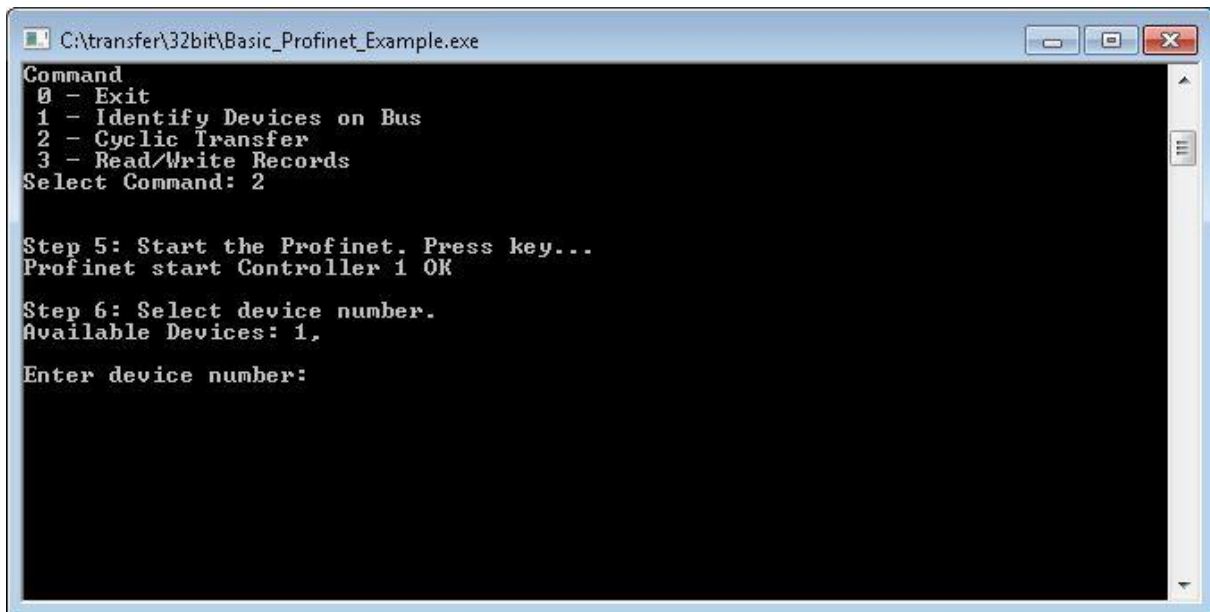
Step 4: Get information about all devices. Press key...
DCP identify in progress. Wait 10s...
DCP identify found 4 device(s):
1. fnl-proxy-pn-pb-proxy1xpse11-20100824
2. df-profinet-io-device
3. pn-io
4. crio-pn-real-1

Command
0 - Exit
1 - Identify Devices on Bus
2 - Cyclic Transfer
3 - Read/Write Records
Select Command:
```

Abbildung 6: Beispielprogramm PN IO Controller - Identify

- Step 4: *Get information about all devices*: DCP Service zur Identifizierung der angeschlossenen PN IO Devices mit Anzeige der PN IO spezifischen Namen

- Command 2: *Cyclic Transfer*: Start des zyklischen Datenverkehrs



```
C:\transfer\32bit\Basic_Profinet_Example.exe
Command
0 - Exit
1 - Identify Devices on Bus
2 - Cyclic Transfer
3 - Read/Write Records
Select Command: 2

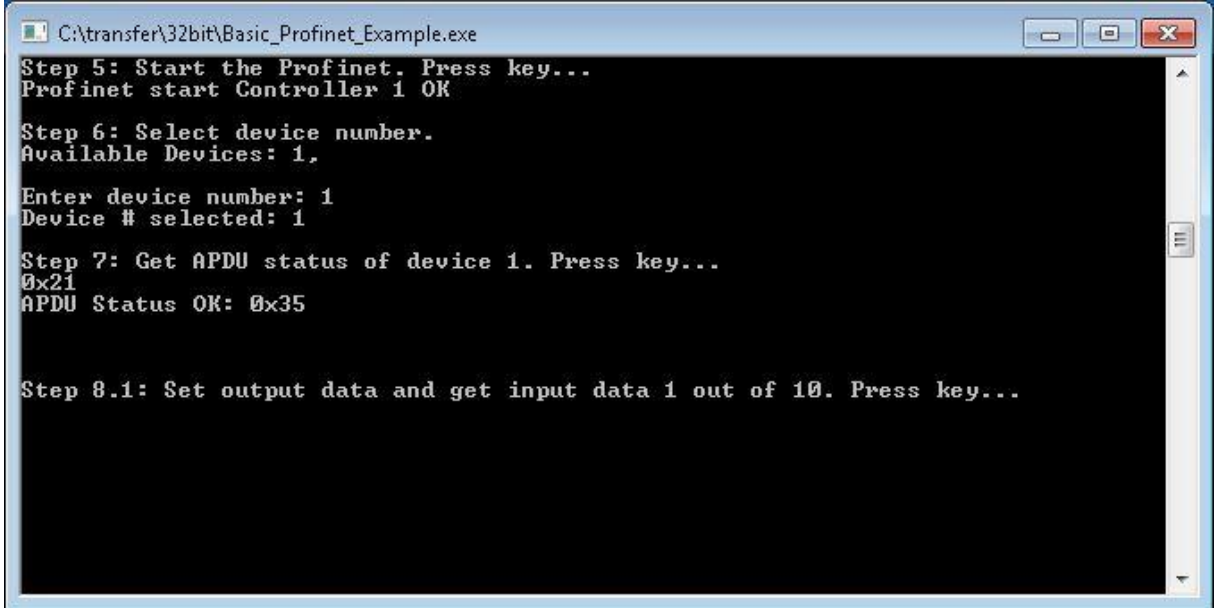
Step 5: Start the Profinet. Press key...
Profinet start Controller 1 OK

Step 6: Select device number.
Available Devices: 1.

Enter device number:
```

Abbildung 7: Beispielprogramm PN IO Controller - Cyclic Transfer

- Step 5: *Start the Profinet*: PROFINET IO wird gestartet
- LED Grün = AN, LED Gelb = An, LED Rot = Aus
  - Falls die LED Rot = An, die PN IO Konfiguration mit dem On Line-Mode von *Configurator III* auf Fehler überprüfen.
- Step 6: *Select device number*: Anzeige der verfügbaren PN IO Devices und Auswahl der PN IO Device-Nummer, mit welchem die zyklischen I/O-Daten visualisiert/manipuliert werden sollen. Die PN IO Device-Nummer kann der Projektierung in *Configurator III* entnommen werden.



```

C:\transfer\32bit\Basic_Profinet_Example.exe
Step 5: Start the Profinet. Press key...
Profinet start Controller 1 OK

Step 6: Select device number.
Available Devices: 1,

Enter device number: 1
Device # selected: 1

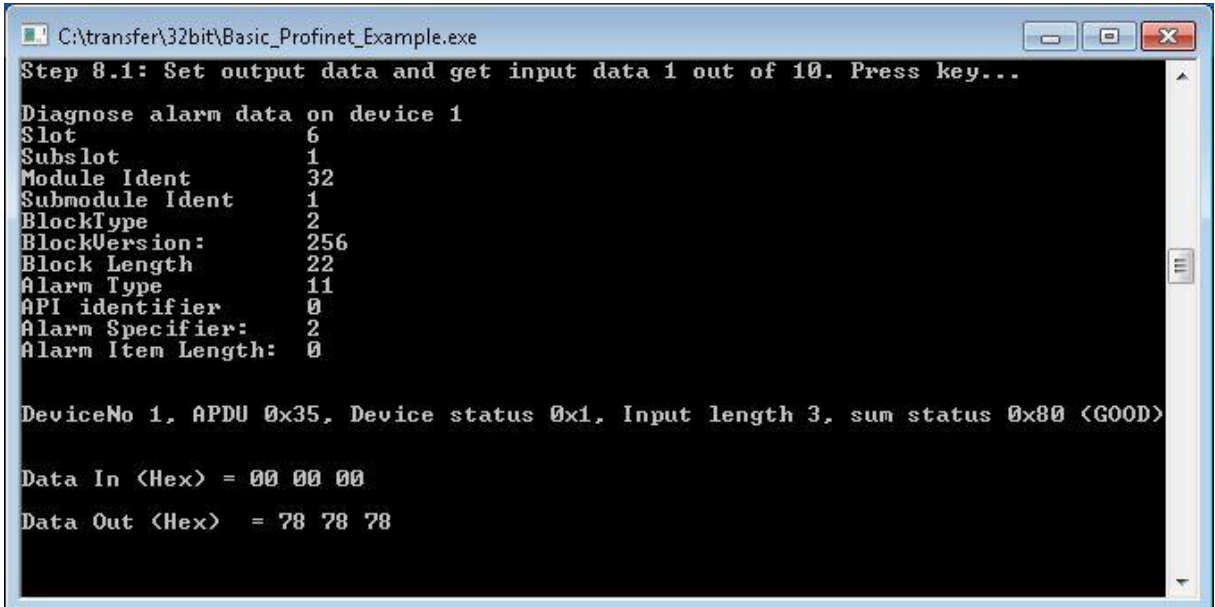
Step 7: Get APDU status of device 1. Press key...
0x21
APDU Status OK: 0x35

Step 8.1: Set output data and get input data 1 out of 10. Press key...

```

Abbildung 8: Beispielprogramm PN IO Controller - Cyclic Transfer

- Step 7: *Get APDU status*: Anzeige des APDU Status vom ausgewählten PN IO Device.



```

C:\transfer\32bit\Basic_Profinet_Example.exe
Step 8.1: Set output data and get input data 1 out of 10. Press key...

Diagnose alarm data on device 1
Slot                6
Subslot             1
Module Ident        32
Submodule Ident     1
BlockType            2
BlockVersion:       256
Block Length        22
Alarm Type           11
API identifier       0
Alarm Specifier:    2
Alarm Item Length:  0

DeviceNo 1, APDU 0x35, Device status 0x1, Input length 3, sum status 0x80 <GOOD>

Data In <Hex> = 00 00 00
Data Out <Hex> = 78 78 78

```

Abbildung 9: Beispielprogramm PN IO Controller - Cyclic Transfer

- Step 8.1 *Set output data and get input data*: Setzen der Output-Daten, Lesen der Input-Daten
- gleichzeitige Anzeige von ev. anstehenden Diagnose-Alarmen
- Anzeige des APDU- und Device-Status, des I/O-Daten - Status (sum status)

- Anzeige der Input- und Output-Daten.
- Mit beliebigem Tastendruck werden die Output-Daten inkrementiert und gesendet sowie die Input-Daten gelesen.

```

C:\transfer\32bit\Basic_Profinet_Example.exe
Step 9: Read / Write Record Data. Press key...
Read Record Data on device 1
SlotNumber      = 1
SubslotNumber   = 1
Index           = 0x2010
Slot            = 1
DataLength      = 256
ReturnValues
  Read with error
  ErrorCode      = 0xDE
  ErrorDecode    = 0x80
  ErrorCode1     = 0xB0
  ErrorCode2     = 0x0
  AddData1      = 0x0
  AddData2      = 0x0
  DataLength    = 256

Write Record Data on device 1
SlotNumber      = 1
SubslotNumber   = 1
Index           = 0x2011
Slot            = 1
DataLength      = 4
Data (Hex)      = AF FE 11 22

ReturnValues
  Write successful
  ErrorCode      = 0x0
  ErrorDecode    = 0x0
  ErrorCode1     = 0x0
  ErrorCode2     = 0x0
  AddData1      = 0x0
  AddData2      = 0x0

Step 10: Stop the Profinet. Press key...

```

Abbildung 10: Beispielprogramm PN IO Controller - Acyclic Transfer

- Step 9 *Read / Write Record Data*: Ausführung eines ReadRec- und WriteRec Service
  - Die spezifischen Parameter für ReadRec und WriteRec sind hart codiert und können im mitgelieferten Source Code geändert werden
  - Der Service wird je 1-mal ausgeführt
- Step 10 Stop the PROFINET: Stoppen des PROFINET controllers
  - LED Grün = An, LED Gelb = Aus, LED Rot = Aus

## 3 Betrieb als PROFINET IO-Device

Zum Betrieb der Baugruppe als PROFINET IO Device muss **keine** Konfiguration mit Configurator III auf die Baugruppe geladen werden.

### 3.1 Beschreibung der LED's

 Grüne LED:

An: Firmware geladen und gestartet

Aus: Firmware nicht geladen

 Gelbe LED:

An: PROFINET IO gestartet

Aus: PROFINET IO gestoppt

 Rote LED:

An: Keine oder fehlerhafte Verbindung mit einem PN IO Controller

Aus: Kein Fehler im PROFINET IO Netzwerk

## 3.2 PROFINET IO Device C und C++ Beispiel

Dieses Beispiel zeigt exemplarisch und auf einfache Weise wie das Interface der Baugruppe zu nutzen ist.

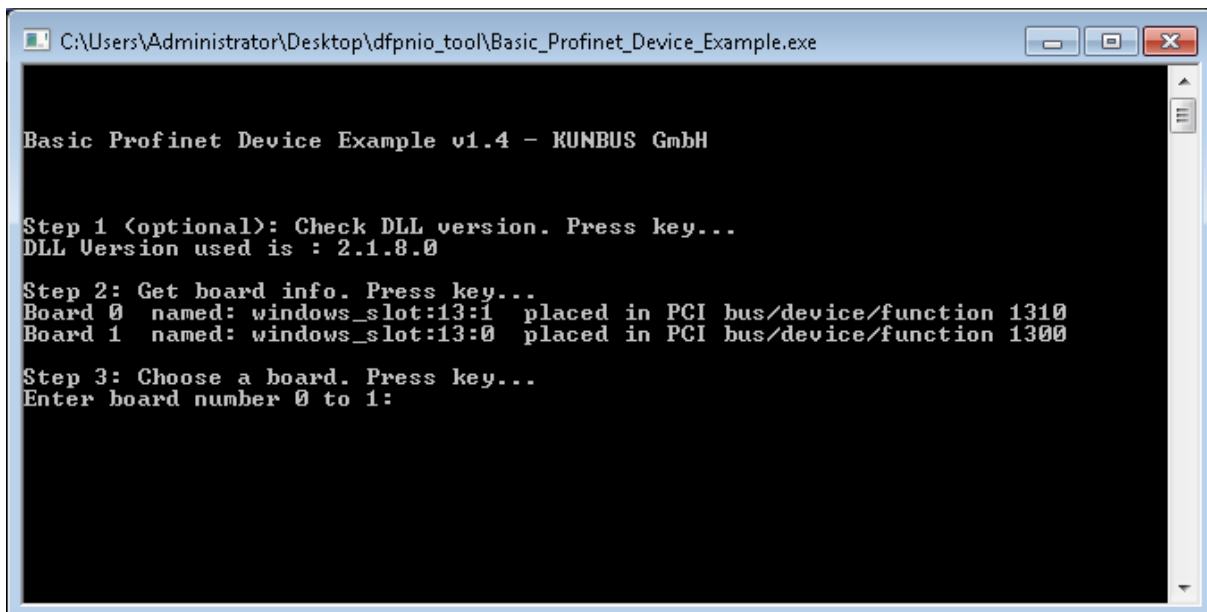
Das Beispielprogramm befindet sich im Verzeichnis

"C:\Programme\KUNBUS GmbH\Profinet IO Device\Basic\_Profinet\_Device\_Example".

oder

"C:\Programme (x86)\KUNBUS GmbH\Profinet IO Device\Basic\_Profinet\_Device\_Example".

und kann ab Visual C++ 2008 direkt editiert und übersetzt werden. Es handelt sich um eine 32-Bit Konsolapplikation, welche die einzelnen notwendigen Schritte nacheinander exemplarisch abarbeitet. Weitere Informationen über das Programm finden Sie im Sourcecode selbst. Der C-Code ist im File "Basic\_Profinet\_Example.cpp" enthalten. Die bereits mitgelieferte "Basic\_Profinet\_Example.exe" kann direkt verwendet werden:



```
C:\Users\Administrator\Desktop\dfpnio_tool\Basic_Profinet_Device_Example.exe

Basic Profinet Device Example v1.4 - KUNBUS GmbH

Step 1 (optional): Check DLL version. Press key...
DLL Version used is : 2.1.8.0

Step 2: Get board info. Press key...
Board 0 named: windows_slot:13:1 placed in PCI bus/device/function 1310
Board 1 named: windows_slot:13:0 placed in PCI bus/device/function 1300

Step 3: Choose a board. Press key...
Enter board number 0 to 1:
```

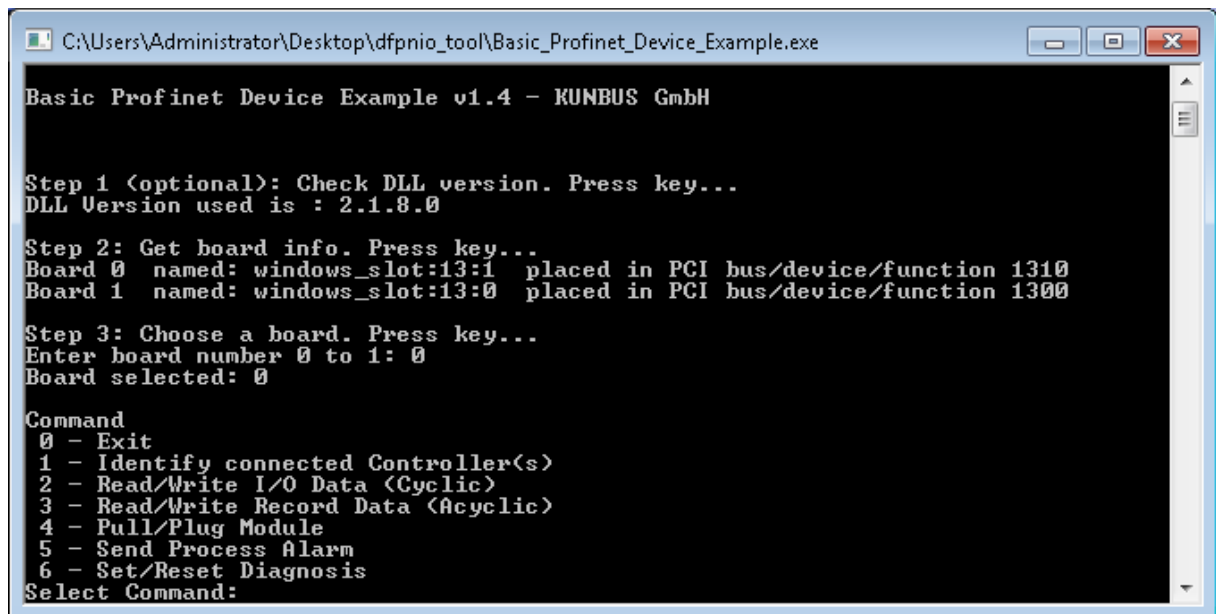
Abbildung 11: Beispielprogramm PN IO Device - Initialisierung

Das Programm läuft Schritt für Schritt ab. Mit beliebigem Tastendruck wird weitersprungen. Folgende Schritte sind verfügbar:

- Step 1: *Check DLL version*: Initialisierung der DFX-DLL und Generierung eines File Handles für den Baugruppenzugriff
- Step 2: *Get board info*: Lesen der Installierten Baugruppenkonfiguration
- Step 3: *Choose a board*: Auswahl einer installierten Baugruppe



- LED Grün = An, LED Gelb = Aus, LED Rot = Aus



```
C:\Users\Administrator\Desktop\dfpnio_tool\Basic_Profinet_Device_Example.exe

Basic Profinet Device Example v1.4 - KUNBUS GmbH

Step 1 (optional): Check DLL version. Press key...
DLL Version used is : 2.1.8.0

Step 2: Get board info. Press key...
Board 0 named: windows_slot:13:1 placed in PCI bus/device/function 1310
Board 1 named: windows_slot:13:0 placed in PCI bus/device/function 1300

Step 3: Choose a board. Press key...
Enter board number 0 to 1: 0
Board selected: 0

Command
0 - Exit
1 - Identify connected Controller(s)
2 - Read/Write I/O Data (Cyclic)
3 - Read/Write Record Data (Acyclic)
4 - Pull/Plug Module
5 - Send Process Alarm
6 - Set/Reset Diagnosis
Select Command:
```

Abbildung 12: Beispielprogramm PN IO Device – Kommandoübersicht

- Command 0: *Exit*: Beispielprogramm wird beendet
- Command 1: *Identify connected Controllers*: Anzeige aller mit dem PN IO Devices verbundenen PN IO Controller
- Command 2: *Read/Write I/O Data (Cyclic)*: Lesen/Schreiben von zyklischen Prozessdaten
- Command 3: *Read/Write Record Data (Acyclic)*: Lesen/Schreiben von azyklischen Record-Daten
- Command 4: *Pull/Plug Module*: Pull/Plug von Modulen, erzeugt Pull/Plug-Alarm am PN IO Controller)
- Command 5: *Send Process Alarm*: Senden eines Prozess Alarms an den PN IO Controller
- Command 6: *Set/Reset diagnosis*: Diagnosemeldung setzen/rücksetzen

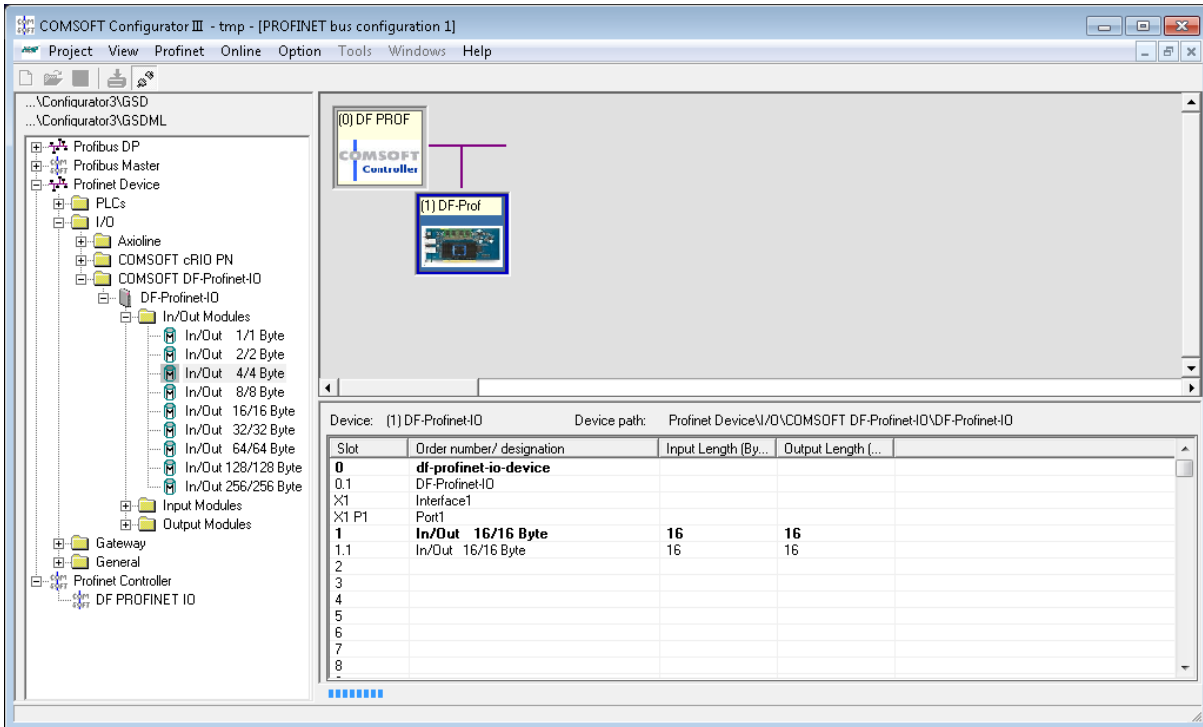


Abbildung 13: Zustand des PN IO Device am angeschlossenen PN IO Controller vor der Aktivierung

Das PN IO Device ist zu diesem Zeitpunkt noch nicht aktiviert. Eine weitere als PN IO Controller konfigurierte DF PROFINET IO Baugruppe zeigt im On Line Mode des Configurator III Tools das konfigurierte PN IO Device als fehlerhaft an (blauer Rahmen).

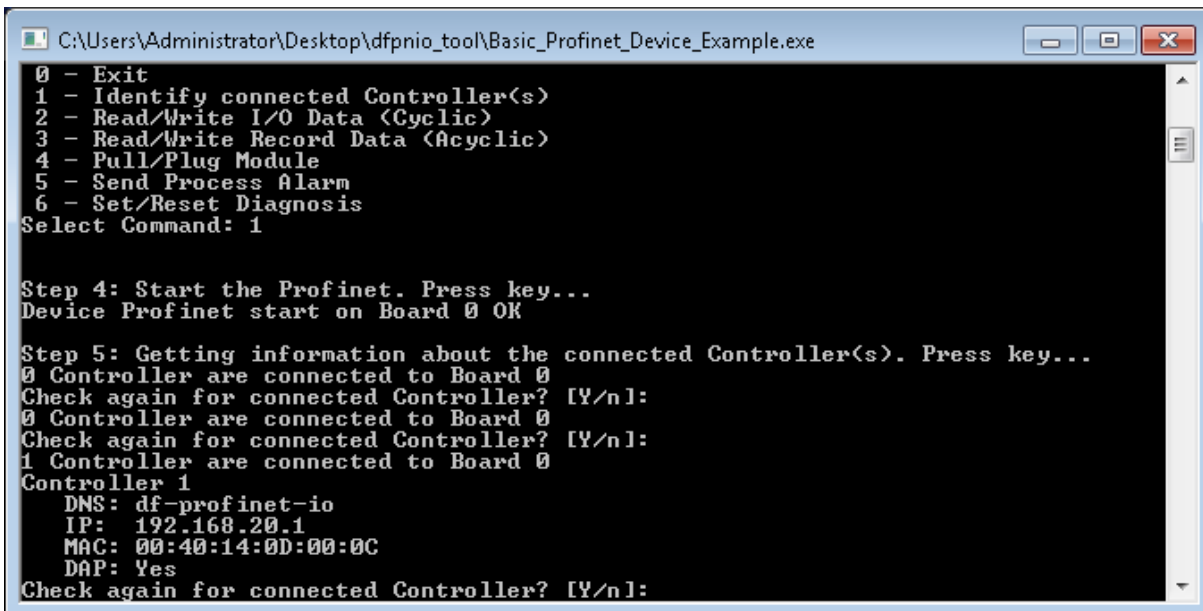


Abbildung 14: Beispielprogramm PN IO Device – Start des Profinet

- Step 4: *Start the Profinet*: Das PN IO Device wird aktiviert und nimmt den Datenverkehr mit den angeschlossenen PN IO Controllern auf.

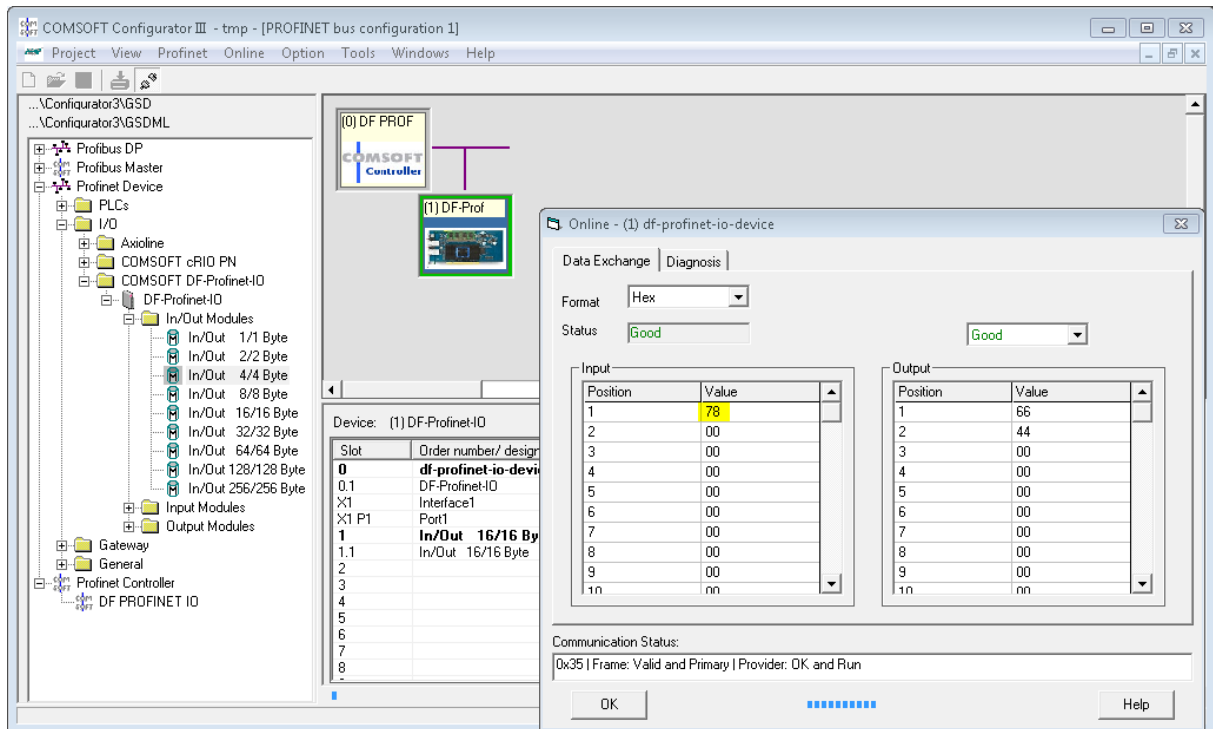


Abbildung 15: Zustand des PN IO Device am angeschlossenen PN IO Controller nach der Aktivierung

Nach Aktivierung des PN IO Devices zeigt die als PN IO Controller konfigurierte DF PROFINET IO Baugruppe im On Line Mode eine funktionierendes PN IO Device an (grüner Rahmen). Der Austausch von zyklischen I/O - Daten ist jetzt möglich.

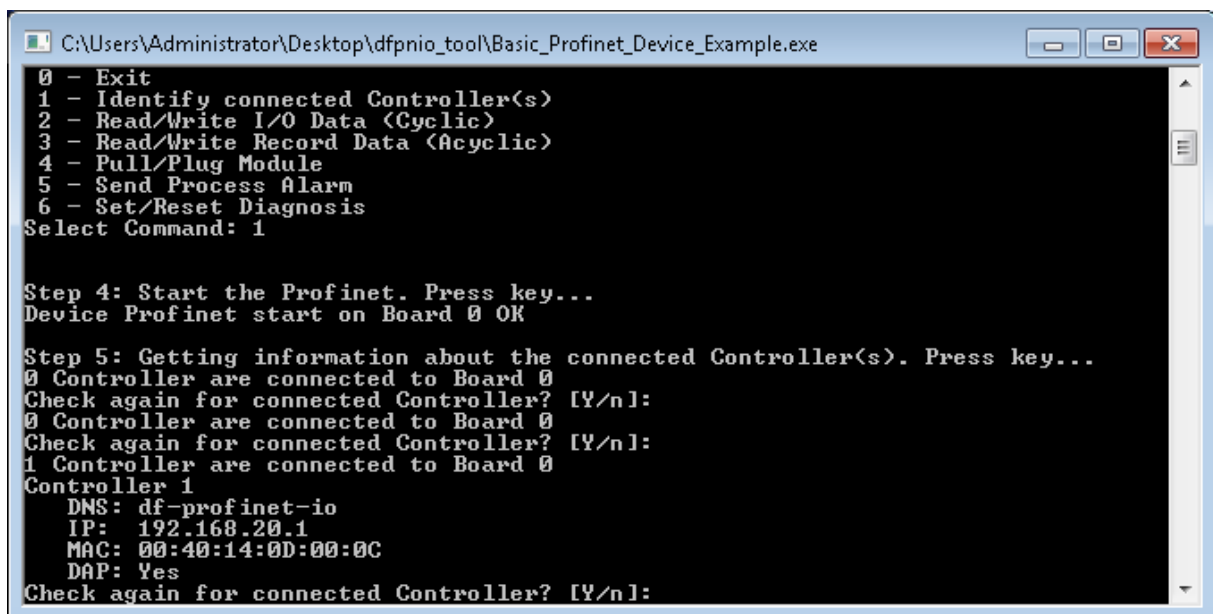


Abbildung 16: Beispielprogramm PN IO Device – Identifizierung der verbundenen PN IO Controller

- Step 5: *Identify connected Controllers*: Anzeige der aktiven mit dem PN IO Device verbundenen PN IO Controller. Step 5 kann beliebig oft wiederholt werden.

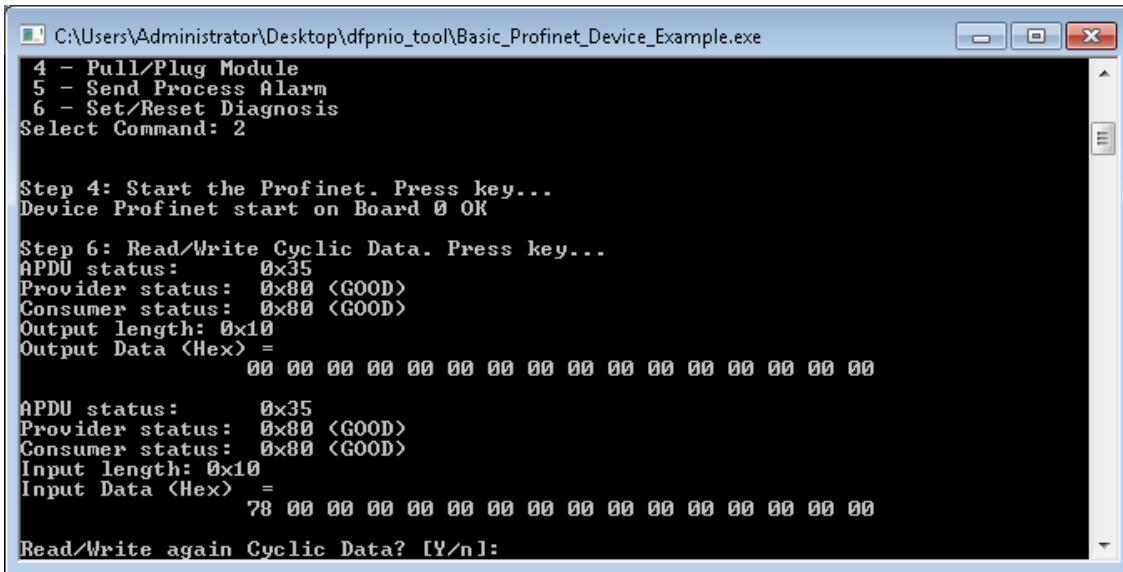


Abbildung 17: Beispielprogramm PN IO Device - Start PN IO Device und Lesen/Schreiben der zyklischen Prozessdaten

- Step 6: *Read/Write Cyclic Data*: Lesen und Schreiben von zyklischen I/O-Daten. Jeweils zu den Input- und Output-Daten werden der Status und die Längen angezeigt.

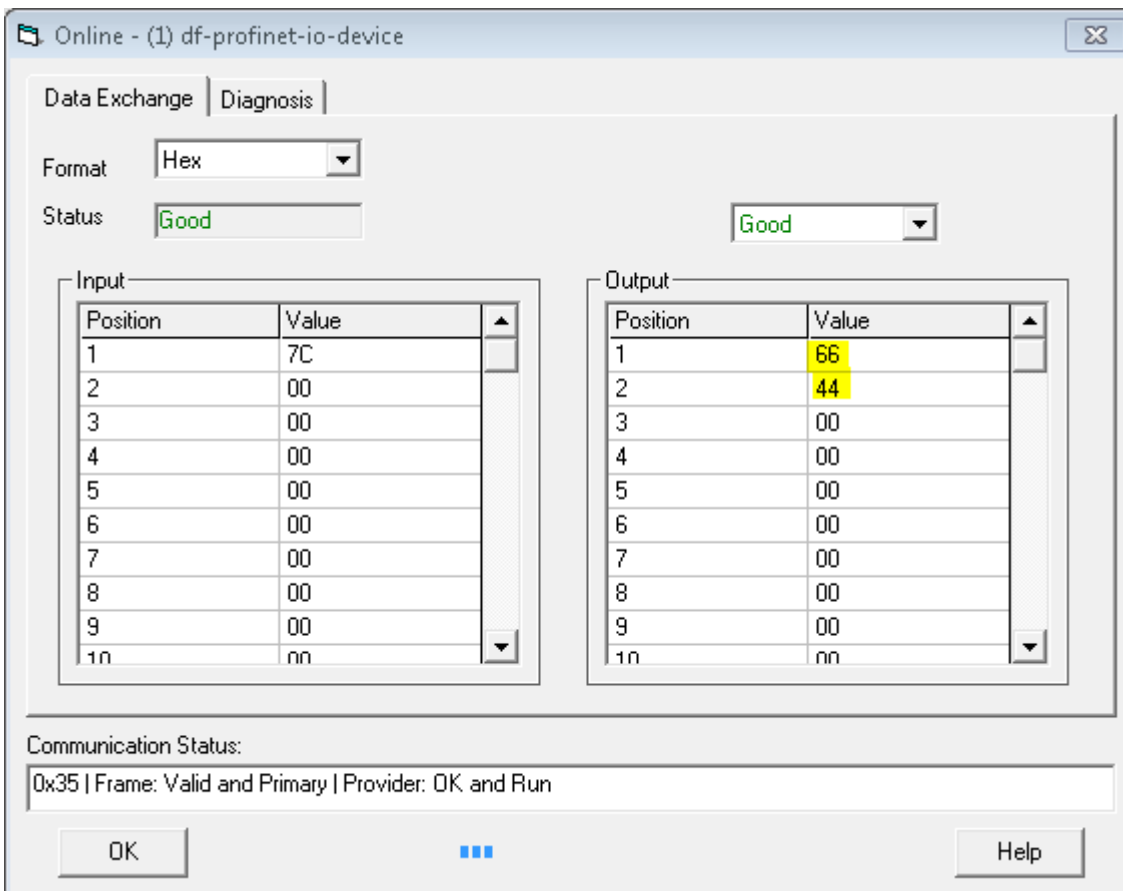


Abbildung 18: Geänderte Output-Daten am PROFINET IO Controller

```

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
APDU status:      0x35
Provider status:  0x80 (GOOD)
Consumer status:  0x80 (GOOD)
Input length:    0x10
Input Data (Hex) =
                7B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Read/Write again Cyclic Data? [Y/n]:
APDU status:      0x35
Provider status:  0x80 (GOOD)
Consumer status:  0x80 (GOOD)
Output length:   0x10
Output Data (Hex) =
                66 44 00 00 00 00 00 00 00 00 00 00 00 00 00 00

APDU status:      0x35
Provider status:  0x80 (GOOD)
Consumer status:  0x80 (GOOD)
Input length:    0x10
Input Data (Hex) =
                7C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Read/Write again Cyclic Data? [Y/n]:

```

Abbildung 19: Beispielprogramm PN IO Device - Wiederholtes Lesen/Schreiben der zyklischen Prozessdaten und Anzeige der am PROFINET IO Controller geänderten Output-Daten

```

Read/Write again Cyclic Data? [Y/n]: n
Step 12: Stop the Profinet. Press key...
Device Profinet stop on Board 0 OK

Command
0 - Exit
1 - Identify connected Controller(s)
2 - Read/Write I/O Data (Cyclic)
3 - Read/Write Record Data (Acyclic)
4 - Pull/Plug Module
5 - Send Process Alarm
6 - Set/Reset Diagnosis
Select Command: 4

Step 4: Start the Profinet. Press key...
Device Profinet start on Board 0 OK

Step 9: Pull/Plug Modules. Press key...
Pull module 1. Press key...
    successful.
Plug module 1. Press key...
    successful.
Pull/Plug Modules again? [Y/n]:

```

Abbildung 20: Beispielprogramm PN IO Device - Command 4- Pull/Plug Module

- Step 4: *Start the Profinet*: Das PN IO Device wird aktiviert und nimmt den Datenverkehr mit den angeschlossenen PN IO Controllern auf
- Step 9: *Pull Module / Plug Module* – Senden von Pull / Plug – Alarmen an den PROFINET IO Controller

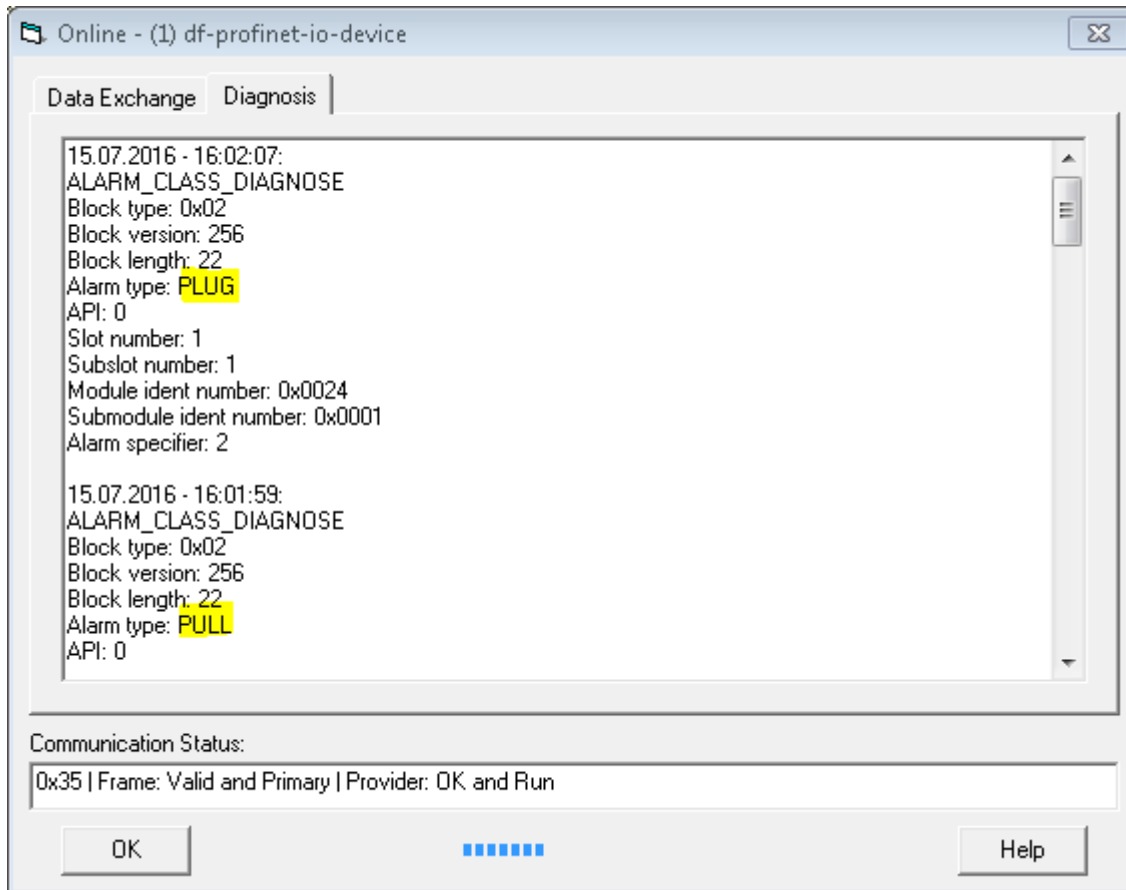


Abbildung 21: Anzeige der vom PROFINET IO Device gesendeten Pull / Plug – Alarme am PROFINET IO Controller

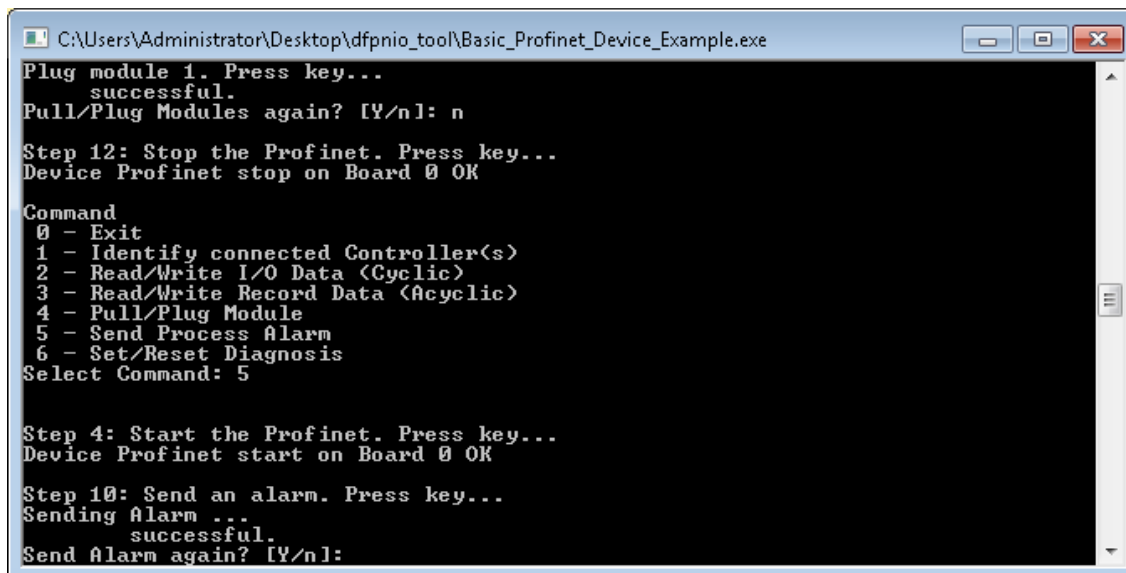


Abbildung 22: Beispielprogramm PN IO Device - Command 5- Senden eines Process-Alarmes

- Step 4: *Start the Profinet*: Das PN IO Device wird aktiviert und nimmt den Datenverkehr mit den angeschlossenen PN IO Controllern auf

- Step 10: *Send an alarm* – Senden eines Process-Alarms an den PROFINET IO Controller

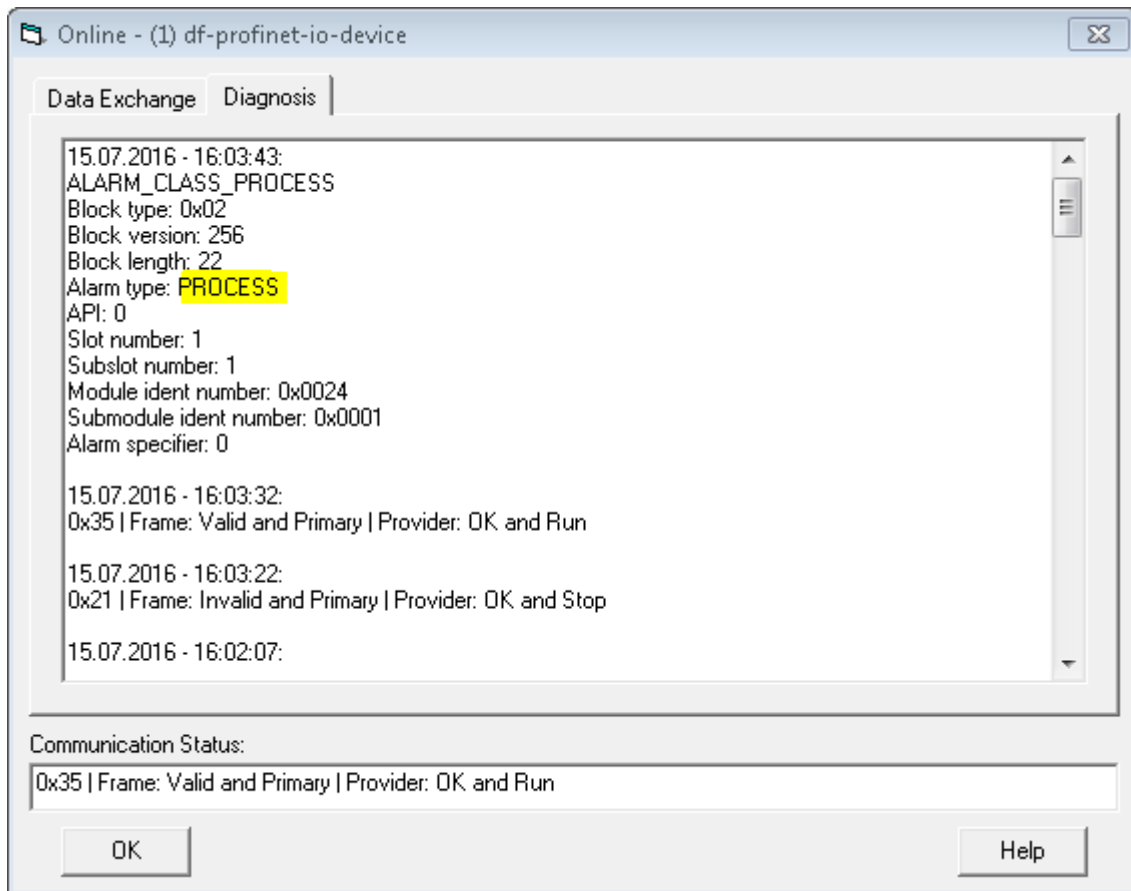


Abbildung 23: Anzeige des vom PROFINET IO Device gesendeten Process – Alarms am PROFINET IO Controller

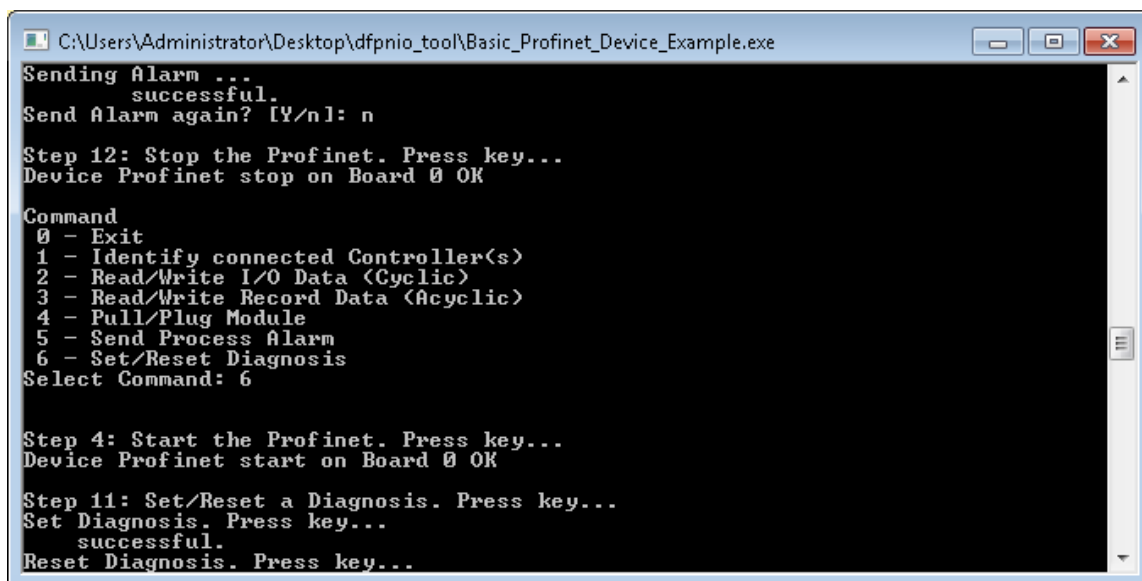


Abbildung 24: Beispielprogramm PN IO Device - Command 6- Setzen/Rücksetzen einer Diagnosemeldung

- Step 4: *Start the Profinet*: Das PN IO Device wird aktiviert und nimmt den Datenverkehr mit den angeschlossenen PN IO Controllern auf
- Step 11: *Set/Reset a Diagnosis* – Setzen/Rücksetzen einer Diagnosemeldung

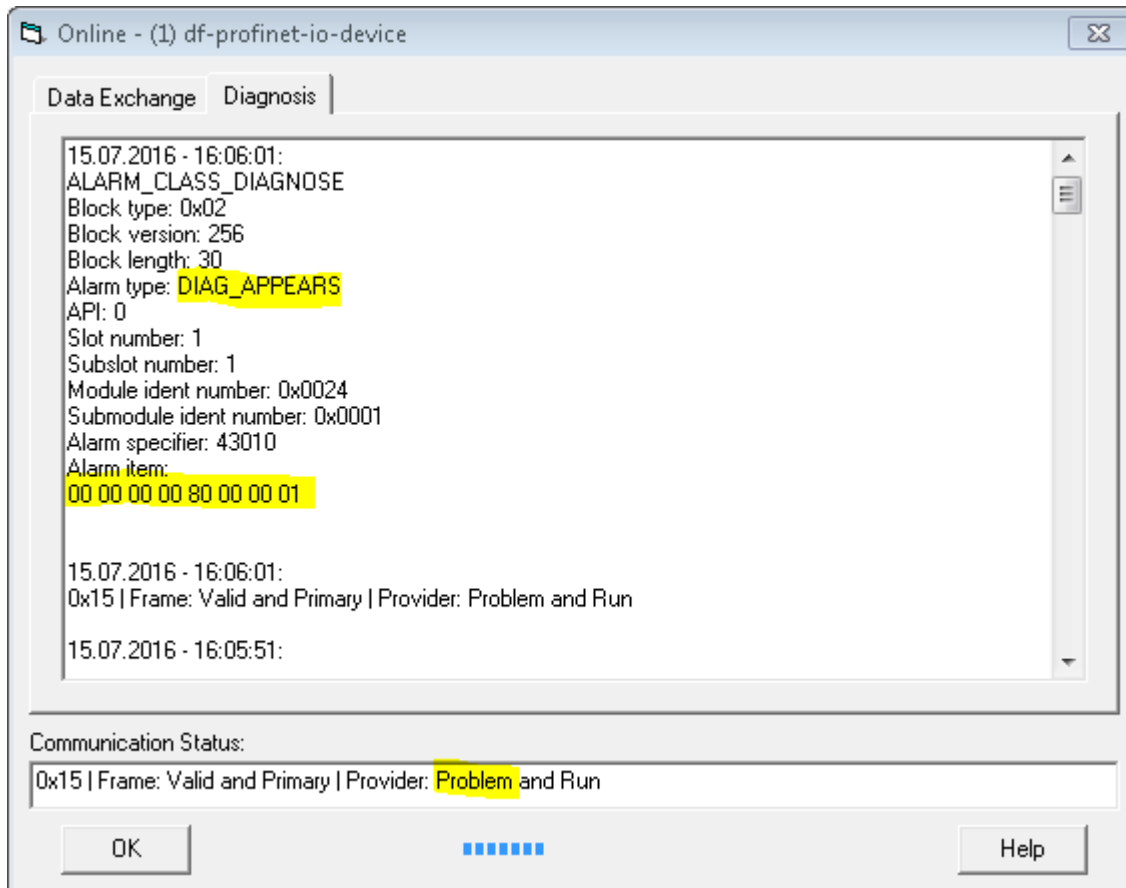


Abbildung 25: Anzeige der vom PROFINET IO Device gesendeten Diagnosemeldung am PROFINET IO Controller



## 4 Parallelbetrieb von PN IO Controller und PN IO Device

Mit der DF PROFINET IO Baugruppe können die Betriebsarten Controller und Device parallel ausgeführt werden, d.h. die Baugruppe verhält sich gleichzeitig als Controller und als Device im PN IO - Netzwerk.

Zum Test können die im Lieferumfang enthaltenen Beispielapplikationen für PN IO Controller und PN IO Device parallel gestartet werden.

### 4.1 Parallel-Betrieb PN IO Controller/Device auf 1 Ethernet Port

Bei DF PROFINET IO Baugruppen, welche nur mit 1 Ethernet Port bestückt sind, ist bei der PN IO Konfiguration folgendes zu beachten:

Bei der Projektierung des DF PROFINET IO Device über die mitgelieferte GSDML-Datei im Konfigurationstool des externen PN IO Controllers (z.B. einer SPS) ist bei der DCP-Konfiguration die identische TCP IP – Adresse zu projektieren, wie diejenige, die bei der Konfiguration des internen DF PROFINET IO Controllers über Configurator III eingestellt wurde.

Beispiel: Für den Controller-Betrieb der DF PROFINET IO – Baugruppe wurde in Configurator III die TCP IP – Adresse **192.168.20.37** eingestellt:

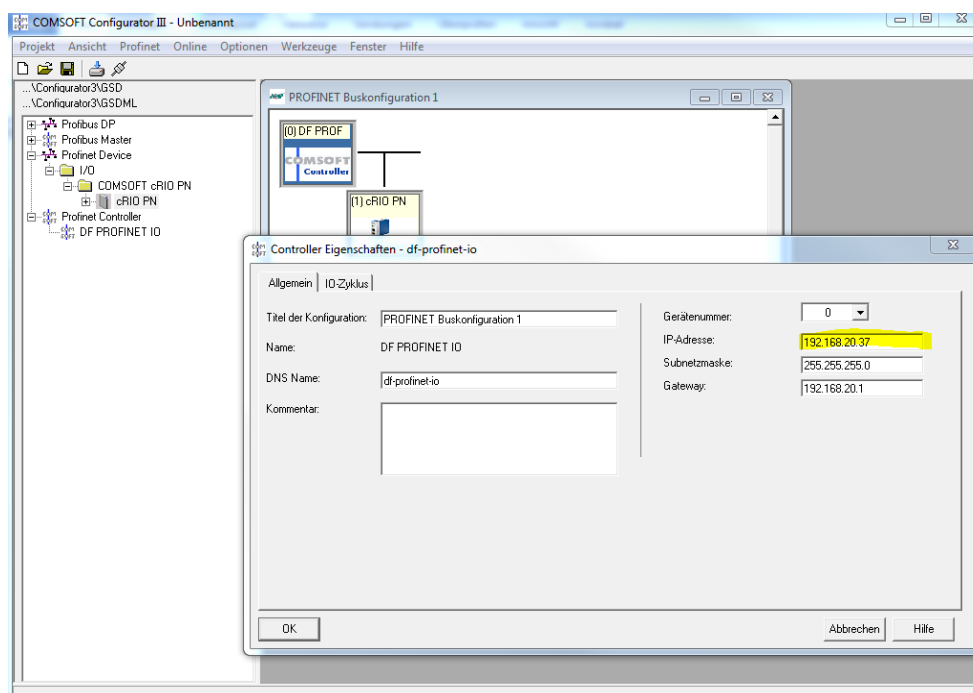


Abbildung 26: Konfigurierte TCP IP-Adresse des internen PN IO Controllers

Im Konfigurationstool des externen Controllers muss in der DCP-Konfiguration (Zuweisung eines PROFINET IO - Namens und der TCP IP – Adresse an das PN IO Device) die identische TCP IP Adresse **192.168.20.37** konfiguriert werden.

## 4.2 Beschreibung der LED's

### Grüne LED:

An: Firmware geladen und gestartet

Aus: Firmware nicht geladen

### Gelbe LED:

An: PROFINET IO gestartet

Aus: PROFINET IO gestoppt

### Rote LED:

An: Fehler im PROFINET IO Netzwerk

Aus: Kein Fehler im PROFINET IO Netzwerk

## 4.3 Parallel-Betrieb PN IO Controller/Device auf 2 Ethernet Ports

Bei DF PROFINET IO Baugruppen, welche mit 2 getrennten Ethernet Ports bestückt sind, wird die PN IO Controller Funktion auf **Ethernet Port 1**, die PN IO Device Funktion auf **Ethernet Port 2** ausgeführt. Die Ethernet Ports arbeiten unabhängig voneinander, sodass für jeden Ethernet Port unterschiedliche TCP IP – Adressen konfiguriert werden können.