





# Table of Contents

1	General Specification	3
	Typographical Conventions	3
2	Introduction	4
3	Principle of Communication / Prerequisites	5
	3.1 Activate protocol	5
	3.2 Handshake	6
4	Communication by Static Objects	7
5	Communication of Multiplex Objects	8
	5.1 Complete communication cycles for MUX Objects	9
	5.1.1 Read objects	9
	5.1.2 Write objects	9
	5.1.3 Read and write objects	9
	5.1.4 Check sum calculation MUX data (ONLY with flexotemp® PCU XXX PNIO)	10
6	Communication by Parameter Channel	11
	6.1 Types of Parameter Channels	12
	6.1.1 Type of Parameter Channel for command 1...26	12
	6.1.2 Type of Parameter Channel for command 27...28 (only PCU)	12
	6.1.3 Type of Parameter Channel for command 29...32 (only PCU)	13
	6.1.4 Type of Parameter Channel for command 33 (only ETR)	13
	6.2 Parameter Channel Settings	14
	6.2.1 Parameter Channel for System- and Channel Data	14
	6.2.2 Parameter Channel for configuration of SPL (only PCU)	15
	6.2.2.1 NodeID/ Object number	15
	6.2.2.2 Number of SPL Modules/ Object number	15
	6.2.3 Parameter Channel for configuration of Direct IO's (only PCU)	15
	6.2.3.1 Start index Direct IO's / Number of Direct IO's	15
	6.3 Parameter Channel n data words	16
	6.3.1 Parameter Channel n data words - Commands	16
7	Configuration and Online Mode with flexotempMANAGER	19
	7.1 Configuration	19
	7.1.1 Communication settings	19
	7.1.2 Define object list	23
	7.1.3 Edit object list	26
	7.1.4 Write / Read object list	27
	7.2 Online Mode	28
	7.2.1 SIO Mode	28
	7.2.2 Edit Mode	30
	7.3 Change value of a MUX object	32
8	Integration of a PCUXXXPNIO in a SIMATIC 300 PROFINET Network	34
	SIMATIC Manager step 1	34
	SIMATIC Manager step 2	35
	SIMATIC Manager step 3	35
	SIMATIC Manager step 4	37
	SIMATIC Manager step 5	38
	SIMATIC Manager step 6	38
	SIMATIC Manager step 7	39
	SIMATIC Manager step 8	39
	SIMATIC Manager step 9	40
	SIMATIC Manager step 10	41

**2** | Chapter 1  
**General Specification**

9 Appendix	42
9.1 GSD File	42
9.2 PROFINET PLC Data types	42
9.3 Version History	42

# 1 General Specification





PROFINET IO protocol is available for the following PSG controllers

## Multi loop controllers PNIO

- flexotemp® PCU XXX PNIO
- systemp® ETR XXX PNIO

## Typographical Conventions

Symbols and conventions are used in this manual for faster orientation for you.

	Caution	With this symbol, references and information are displayed which are decisive for the operation of the device. In case of non-compliance with or inaccurate compliance there can result damage to the device or injuries to persons.
	Note	The symbol refers to additional information and declarations, which serve for improved understanding.
	Example	With the symbol, a function is explained by means of an example.
	Reference	With this symbol, information in another document is referred to.
n.a.		Not applicable, not existing

## 2 Introduction

The PNIO controller works as PROFINET IO Device. It supports 1024 (PCU) | 340 (ETR) Byte I/O respectively. The transmission is done by twisted pair wire in full duplex mode with 100 MBit/s. The IP address setting, as usual for PROFINET IO, is done by the configuration of the PROFINET IO Controller (see chapter 8 Integration of a PCUXXXPNIO in a SIMATIC 300 PROFINET Network) and later transferred to the module while start-up of IO Controller.

The available LED's on the module signalize the current PROFINET IO status and possible error messages.

The PROFINET component is based on the PROFINET IO software technology by Siemens. The dual port RAM based hard- and software interface of PROFINET interfaces is standardized (IEC 61158 and IEC 61784).

The PROFINET IO devices have different characteristics per device type and manufacturer. To provide a rather simple configuration for the user, GSD's (General Station Description) are used, including GSDML based descriptions of characteristics of IO Devices for communication parameters, quantity, type, configuration data, parameter and diagnostics information of modules.

All modern configuration tools make it possible to read the GSD's in the configuration and thereby integrate the modules into the PROFINET IO system quickly and easily.

### 3 Principle of Communication / Prerequisites

Integration of the PNIO controller into the control system by means of the GSD file. Content of GSD files see chapter 9.1 GSD File.

The user can define for the PNIO controller, the I/O data areas and their data content that is what channel or system parameters, Direct IO (only PCU) read or written.

This **configuration of the PNIO controller** is done by the project setup and configuration tool

- flexotempMANAGER (see chapter 7 Configuration and Online Mode with flexotempMANAGER)

The channel and/or system parameters communicated, are named communication objects or abbreviated objects. They are classified in three areas, that are communicated in different ways:

- Static Objects (see chapter 4 Communication by Static Objects)
- Multiplex Objects (see chapter 5 Communication of Multiplex Objects)
- Parameter Channel (see chapter 6 Communication by Parameter Channel)

All objects are presented as words (16 bits) independent of the storage type in the PNIO controller on the control system side.

Writing new values the PNIO controller checks Min /Max limits of the particular parameters.

#### 3.1 Activate protocol

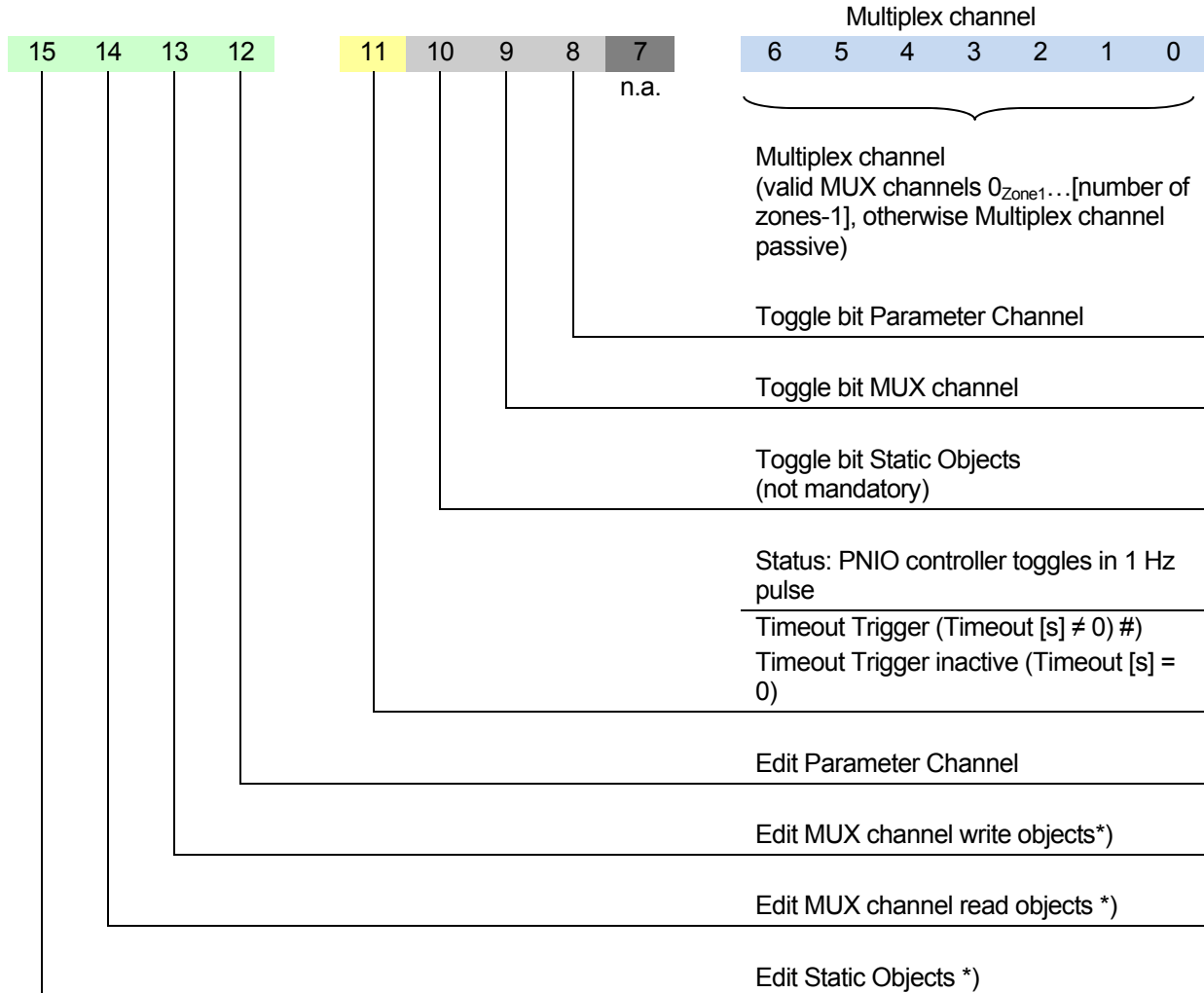
The PROFINET IO protocol is available, once the master component with the symbol "PNIO" is selected, designed and configured.

### 3.2 Handshake

The last word of the In-/Output buffer respectively is used as **Handshake** and therefore is not available for the communication of objects.

In essence, it serves to ensure **data consistency** on the control system.

It is built up as follows:



\*) 1 = Enable = ✓; NOTE: No processing for bits not set (12...15).

The handshake bits are set by the controller in the output buffer. The PNIO controller reflects the state of the bit after completion of processing in the input buffer of the controller. The handshake bit must always be toggled by the controller only if all of its data contents are written to the output buffer.

The further meaning of the hand shake is in chapters

- 4 Communication by Static Objects
- 5 Communication of Multiplex Objects
- 6 Communication by Parameter Channel

explained.

#) The timeout monitoring on PROFINETIO takes effect when the bus connection is interrupted. To detect a PLC STOP and to switch off heating outputs, bit 11 must be toggled within the timeout time when the timeout monitoring (timeout ≠ 0) is active. To do this, the PLC program must be adapted to customer requirements.

With inactive timeout monitoring (timeout = 0), bit 11 serves as a 1 Hz life signal.



## 4 Communication by Static Objects

### Characteristic of Static Objects

- Static objects are linked directly and unambiguously with a parameter on PNIO controller side. Available are system- or channel data / -parameters and Direct IO's (only PCU)
- A static channel data object is directly linked with the zone
- Static Objects have word format

As with Static Objects a unique assignment of a parameter with an in-/output word is always guaranteed, no handshake to ensure data consistency is required.

**Read:** To read Static Objects such as the setpoint value of channel 1, the input buffer has to be configured appropriately with the aid of flexotempMANAGER. The controller can read the appropriate value of there at any time.

**Write:** To write Static Objects, the output buffer has to be configured appropriately with the aid of flexotempMANAGER. Then, the value to be written must be written only on the control side.



In the PNIO controller only changed values are applied. After each writing of a new value, the autosave mechanism of PNIO controller is started. That means, when there is no new writing of Static Objects in the next 10 minutes, after autosave time of 10 minutes elapsed all previously modified values are automatically taken into the EEPROM.

## 5 Communication of Multiplex Objects

In order to communicate with a minimum of in-/output memory requirement in the control a maximum number of channels of the PNIO controller, it is possible to communicate with all of the channel parameters by means of the Multiplex Objects.

Characteristic of Multiplex Objects

- Multiplex Objects are channel parameters of the PNIO controller
- Multiplex Objects are not statically connected with the controller channel
- Multiplex Objects have word format

In the PNIO controllers offsets can be defined, where other MUX channels are accessible. Doing this multiple channels / zones can be processed per communication cycle per channel parameters.

In order to ensure consistent data on the controller side, here's a handshake necessary. The handshake is done via the handshake described in chapter 3.2.

The following norms are valid

- Valid values for MUX channel

PNIO controller:  $0_{Zone1} \dots [number\ of\ zones-1]$

- In order for a treatment of the MUX objects takes place, the handshake bit for writing and for reading the MUX objects must be set to "1" (bit 13 and bit 14 in the handshake).
- The processing of the MUX objects always takes place when the handshake bit 9 (toggle bit MUX channel) changes / toggles. The handshake bit is then reflected in the response (in the input buffer). Thus, the user can recognize that a processing has occurred.



After the change of the handshake bit, the processing of the MUX objects takes place. Therefore, it is essential, first to make all the necessary data entries in the buffer and then to change the handshake bit at last.

## 5.1 Complete communication cycles for MUX Objects

### 5.1.1 Read objects

1. Enter the desired channel number in the **MUX Channel** (Bit 0...6) of the handshake word of the output buffer.
2. Make sure, that in the handshake word of the output buffer **Handshake Bit 14** (MUX Channel anal edit read objects) is set to 1.
3. Toggle **Handshake Bit 9** (Toggle bit MUX Channel) in the handshake word of the output buffer.
4. Check, that in the handshake word of the input buffer **Handshake Bit 9** (Toggle bit MUX Channel) is reflected. If so, analyze output buffer.

### 5.1.2 Write objects

1. Enter **all** parameter data into output buffer.
2. Enter the desired channel number in the **MUX Channel** (Bit 0...6) of the handshake word of the output buffer.
3. Make sure, that in the handshake word of the output buffer **Handshake Bit 13** (MUX Channel anal edit write objects) is set to 1.
4. Toggle **Handshake Bit 9** (Toggle bit MUX Channel) in the handshake word of the output buffer.
5. Check, that in the handshake word of the input buffer **Handshake Bit 9** (Toggle bit MUX Channel) is reflected. If so, processing is ready.

### 5.1.3 Read and write objects

1. Enter **all** parameter data into output buffer.
2. Enter the desired channel number in the **MUX Channel** (Bit 0...6) of the handshake word of the output buffer.
3. Make sure, that in the handshake word of the output buffer  
**Handshake Bit 13** (MUX Channel edit write objects)  
**Handshake Bit 14** (MUX Channel edit read objects)  
 is set to 1.
4. Toggle **Handshake Bit 9** (Toggle bit MUX Channel) in the handshake word of the output buffer.
5. Check, that in the handshake word of the input buffer **Handshake Bit 9** (Toggle bit MUX Channel) is reflected. If so, analyze input buffer and writing of objects ready.

### 5.1.4 Check sum calculation MUX data (ONLY with flexotemp® PCU XXX PNIO)

(Available from flexotempMANAGER version 1.3.9; from firmware version PCU XXX PNIO 1517A)

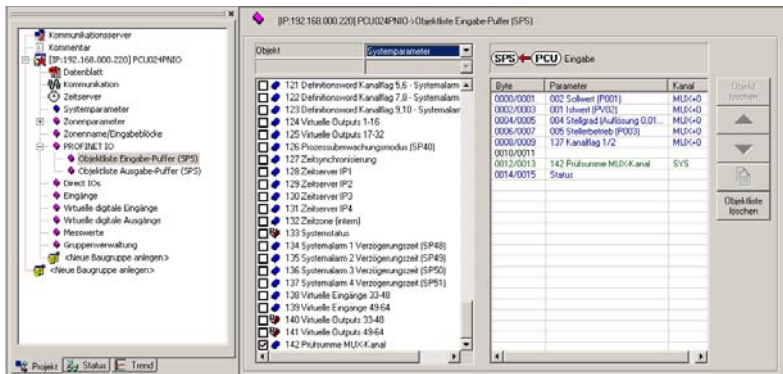
An additional system object was introduced. This contains the checksum for the MUX data of the PCU XXX PNIO. As long as the MUX object for the checksum is not inserted in the PNIO buffer, the PCU XXX PNIO operates as before.

#### Read

If the checksum is inserted into the receive buffer (PLC → PCU output), the PCU XXX PNIO checks the check sum as well as the toggle bit. If this is invalid, the MUX data are not processed.

#### Write

If the checksum is inserted into the send buffer (PCU ← PLC input), the PCU XXX PNIO calculates the checksum and enters this into the corresponding word in the buffer. The PLC must then check this accordingly.



The check sum can be calculated as follows:

```
// 1. Step set CheckSum to 0
unsigned int uiCheckSum;
uiCheckSum = 0;

// 2. Step add up all MUX objects (X=number of MUX objects)
for(cnt=0;cnt<X;cnt++)
uiCheckSum += "Data word corresponding MUX object"

// 3. Step add Handshake (only for the MUX relevant bits)
uiCheckSum += (Handshake & 0x62FF);
```

## 6 Communication by Parameter Channel

The Parameter Channel is used for fast data transfer of large amounts of data. It is suitable for example for backup, for working with recipes or Smart Power Limitation abbreviation SPL Configuration (only PCU).

The following are available

PNIO controller	Parameter Channel n data words	flexotempMANAGER
-----------------	--------------------------------	------------------

As for MUX Objects, a handshake is also necessary for Parameter Channels (see chapter 3.2 Handshake) to guarantee the data consistency on controller side.

After all the words of the Parameter Channel has been described with the appropriate values, the handshake bit (bit 8 in the handshake) is toggled. Then the PNIO controller executes the command (handshake bit 12 must be set to "1").

One can only work with one Parameter Channel type.

## 6.1 Types of Parameter Channels

Dependent on the commands there are different types (configuration) of Parameter Channels.

### 6.1.1 Type of Parameter Channel for command 1...26

Parameter channel for command 1...26 looks like follows:

Word	n data words
1	Command
2	Channel/Offset (see chapter 6.2.1)
3	Number of channels/Number of objects
4	Data word 0
5	Data word 1
6	Data word 2
7	Data word 3
8	Data word 4
9	Data word 5
10	Data word 6
...	...
n	Data word n-4

### 6.1.2 Type of Parameter Channel for command 27...28 (only PCU)

Parameter channel for command 27...28 looks like follows:

Word	n data words
1	Command
2	NodeID (see chapter 6.2.2.1 *)
3	SPL Module(s) (see chapter 6.2.2) *)
4	Current limit 1 NodeID *
5	Current limit 2 NodeID *
6	Current limit 3 NodeID *
7	SPL Status for output 1...16 Node ID *
8	SPL Status for output 17...32 Node ID *
9	Data word 5
10	Data word 6
...	...
n	Data word n-4

\*) only visible, when NodeID ≠ 0 and SPL Module ≠ 0

### 6.1.3 Type of Parameter Channel for command 29...32 (only PCU)

Parameter channel for command **29...32** looks like follows:

Word	n data words
1	Command
2	Start index Direct IO's / Number of Direct IO's (see chapter 6.2.2.1)
3	<No function>
4	Data word 0
5	Data word 1
6	Data word 2
7	Data word 3
8	Data word 4
9	Data word 5
10	Data word 6
...	...
n	Data word n-4

### 6.1.4 Type of Parameter Channel for command 33 (only ETR)

Parameter channel for command **33** looks like follows:

Word	n data words
1	Command
2	Channel (see chapter 6.2.1) x
3	Number of channels m
4	Setpoint x + 0
5	Adaptation byte x + 0
6	Setpoint x + 1
7	Adaptation byte x + 1
8	Setpoint x + 2
9	Adaptation byte x + 2
...	...
n-1	Setpoint x + (m-1)
n	Adaptation byte x + (m-1)

Allocation adaptation byte

0x01	Zone becomes active
0x02	Manual Adaptation
0x04	Automatic cooling adaptation
0x08	Channel wise current transfer

The setpoint for the specified zone is written and the specified function in the adaptation byte is executed for the zone, e.g. adaptation byte = 0x01 „Zone becomes active“.

## 6.2 Parameter Channel Settings

### 6.2.1 Parameter Channel for System- and Channel Data

#### Channel/ Offset/ Object number

The available object numbers (display decimal/ hexadecimal) communicated by PROFINET IO are stored in a word and consist of

- Channel (superior byte) and
- Offset (inferior byte)

Specification of object number

Channel		Offset	
00	For system parameter	00...FF	(Offset System parameter *)
01...80	For Zone 1...128 (PCU)	00...FF	(Offset zone parameter *)
01...20	For Zone 1...32 (ETR)	00...FF	(Offset zone parameter *)

\*) See object list of PNIO controllers

For channel and offset values a list of values exist in flexotempMANAGER (see chapter 7.2 Online Mode) to choose from. To do this, go into edit mode and select from the 2nd word of the Parameter Channel "Channel / Offset" field buffer of the output buffer with the secondary mouse button.

#### Channel

- 000 System
- 001 <user defined zone name>
- 002 <user defined zone name>
- etc. up to
- n <user defined zone name>  
(n depends on maximal number of zones of configured PNIO controller)

#### Offset

The offsets are different depending on the configured PNIO controllers. The offsets and their sequential order can be read in the document Object list ...PROFINETIO... .



## 6.2.2 Parameter Channel for configuration of SPL (only PCU)

Only for flexotemp® PCU XXX PNIO available and valid for the following flexotemp®-components:

- flexotemp® CANCT - Current Transducer Interface
- flexotemp® BACI - Bus Actuator Interface, Current Input
- flexotemp® DIO16CI - Digital In-/Output Interface, Current Input

### 6.2.2.1 NodeID/ Object number

The available object numbers (display decimal/ hexadecimal) communicated by PROFINET IO are stored in a word and consist of

- NodeID

#### NodeID

Indicates the first component of the PNIO controller with the appropriate CAN bus node ID, which can be seen from the configuration. If more than one component is addressed, this must be based on the specified CAN bus node ID in consecutive order.

### 6.2.2.2 Number of SPL Modules/ Object number

The available object numbers (display decimal/ hexadecimal) communicated by PROFINET IO are stored in a word and consist of

- Number of SPL modules

#### Number of SPL modules

Specifies the number of how many modules with SPL functionality should be handled, from NodeID (see 6.2.2.1) continuously.

## 6.2.3 Parameter Channel for configuration of Direct IO's (only PCU)

Only available with flexotemp® PCU XXX PNIO.

### 6.2.3.1 Start index Direct IO's / Number of Direct IO's

The available object numbers (display decimal/ hexadecimal) communicated by PROFINET IO are stored in a word and consist of

- Start index (superior byte) and
- number of (inferior byte)

#### Start index

Indicates the first Direct IO of the PNIO controller with the appropriate CAN bus node ID, which can be seen from the configuration of Direct IO's. If more than one Direct IO is addressed, this must be based on the specified CAN bus node ID in consecutive order.

#### Number

Specifies the number of how many Direct IO's should be handled, from start index continuously.

### 6.3 Parameter Channel n data words











Upon creation of a Parameter Channel, the operator is asked about the number of data words, the Parameter Channel should consist of. The available buffer size is calculated by the configured in-/output buffers (PLC) (see settings 7.1.1) and the possible number of data words in the list of value to be selected.

From a given object reference (see chapter 6.2) starting, the operator can set the number of channels and the number of objects.






The parameters are read/written as far as data words are available.

#### 6.3.1 Parameter Channel n data words - Commands







The available commands can be chosen from a list of values in flexotempMANAGER (see chapter 7.2 Online Mode). To do this, go into edit mode and select from the 1st word of the Parameter Channel "Command" field buffer of the output buffer with the secondary mouse button.

Command		
1	Read 1 object of channel 1	<p><b>Read 1...</b>, reads object, that is defined by object number.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value → setpoint value of zone 1 is read.</p> <p><b>Read 2...</b> and so on, reads the next (2.) object number etc. based on the defined object number.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value → setpoint value of zone 1 is read, degree of operation of zone 1 is read, etc.</p>
2	Read 2 objects of 1 channel	
3	Read 3 objects of 1 channel	
4	Read 4 objects of 1 channel	
5	Read 5 objects of 1 channel	
6	Read 6 objects of 1 channel	
7	Read 7 objects of 1 channel	
8	Read 8 objects of 1 channel	
	Settings of number of channels/number of objects are not considered.	
9	Read 1 object of 8 channels	<p>The defined object number and 7 more object numbers are read.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value → setpoint value of zone 1 is read, setpoint value of zone 2 is read, etc. setpoint value of zone 8 is read.</p>
	Settings of number of channels/number of objects are not considered.	
16	Write 1 object to 1 channel	<p><b>Write 1...</b>, writes object, that is defined by object number.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value → setpoint value of zone 1 is written by value in data word 1.</p> <p><b>Write 2...</b> and so on, writes the next (2.) object number etc. based on the defined object number.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value → setpoint value of zone 1 is written by value in data word 1 setpoint value of zone 1 is written by value in data word 2, etc.</p>
17	Write 2 objects to 1 channel	
18	Write 3 objects to 1 channel	
19	Write 4 objects to 1 channel	
20	Write 5 objects to 1 channel	
21	Write 6 objects to 1 channel	
22	Write 7 objects to 1 channel	
23	Write 8 objects to 1 channel	
	Settings of number of channels/number of objects are not considered.	
24	Write 1 object to 8 channels	<p>The defined object number and 7 more object numbers are written.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value → setpoint value of zone 1 is written by value in data word 1 setpoint value of zone 2 is written by value in data word 2, etc. setpoint value of zone 8 is written by value in data word 8.</p>
	Settings of number of channels/number of objects are not considered.	

**Command**

25	Read n objects	<p>The defined object number is read for the number of channels (see number of channels) and number of objects (see number of objects) set.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value , number of channels 3, number of objects 2 → setpoint value of zone 1 is read, degree of operation of zone 1 is read, setpoint value of zone 2 is read, degree of operation of zone 2 is read, setpoint value of zone 3 is read, degree of operation of zone 3 is read.</p>	
26	Write n objects	<p>The defined object number is written for the number of channels (see number of channels) and number of objects (see number of objects) set.</p> <p> Object number ≡ Channel: 001 Zone 1, Offset: 002 setpoint value , number of channels 3, number of objects 2 → setpoint value of zone 1 is written by data word 1, degree of operation of zone 1 is written by data word 2, setpoint value of zone 2 is written by data word 3, degree of operation of zone 2 is written by data word 4, setpoint value of zone 3 is written by data word 5, degree of operation of zone 3 is written by data word 6.</p>	
ONLY PCU	27	Read current limit and status SPL of y SPL modules from NodeID x	<p>For the defined NodeID the current limits 1...3 and the status are read for the specified SPL Modules (see number of SPL Modules).</p> <p> NodeID: 002 NodeID 2, Number of SPL Modules: 002 2 SPL Modules → starting from NodeID 2 the current limit 1...3 and the status are read into the intended data words for 2 SPL Modules.</p>
ONLY PCU	28	Stipulate current limit and de-/activate SPL of y SPL modules from NodeID x	<p>For the defined NodeID the current limits 1...3 are set for the specified SPL Modules (see number of SPL Modules) and SPL can be de-/activated.</p> <p> NodeID: 002 NodeID 2, Number of SPL Modules: 002 2 SPL Modules → starting from NodeID 2 the current limit 1...3 are written by the data words for 2 SPL Modules.</p>
ONLY Y		Information, how to use the current limits, see description of Smart Power Limitation SPL.	

## COMMAND

ONLY PCU	29	Write Direct IO [1...255]	From start index on the number of Direct IOs set is written to Direct IOs.  Start index Direct IO's: 001, number of Direct IO's: 1 → starting from start index Direct IO's 1, 1 Direct IO is written by data word 1.
ONLY PCU	30	Read Direct IO [1...255]	From start index on the number of Direct IO's set is read from Direct IO's.  Start index Direct IO's: 001, number of Direct IO's: 1 → starting from start index Direct IO's 1, 1 Direct IO is read.
ONLY PCU	31	Write Direct IO [256...300]	From start index on the number of Direct IO's set is written to Direct IO's.  Start index Direct IO's: 256, number of Direct IO's: 1 → starting from start index Direct IO's 256, 1 Direct IO is written by data word 1.
ONLY PCU	32	Read Direct IO [256...300]	From start index on the number of Direct IO's set is read from Direct IO's.  Start index Direct IO's: 256, number of Direct IO's: 1 → starting from start index Direct IO's 256, 1 Direct IO is read.
ONLY		Information on use of Direct IO's are in the operating instructions <b>Project Setup - and Configuration Tool flexotempMANAGER Operation.</b>	
ONLY ETR	33	Write setpoint value with adaptation control	The setpoint and the adaptation byte are written and/or executed from the channel to number of channels specified.  Channel: 003 ≡ Zone 3 ≡ x, Number of channels: 2 → for Zone 3 the setpoint value of setpoint x + 0 is written, Adaptation byte x + 0 is executed, for Zone 4 the setpoint value of setpoint x + 1 is written, Adaptation byte x + 1 is executed.

Is Parameter Channel n data words not supported for the PNIO controller, an error message occurs in flexotempMANAGER and the writing for the object list is prevented.

## 7 Configuration and Online Mode with flexotempMANAGER

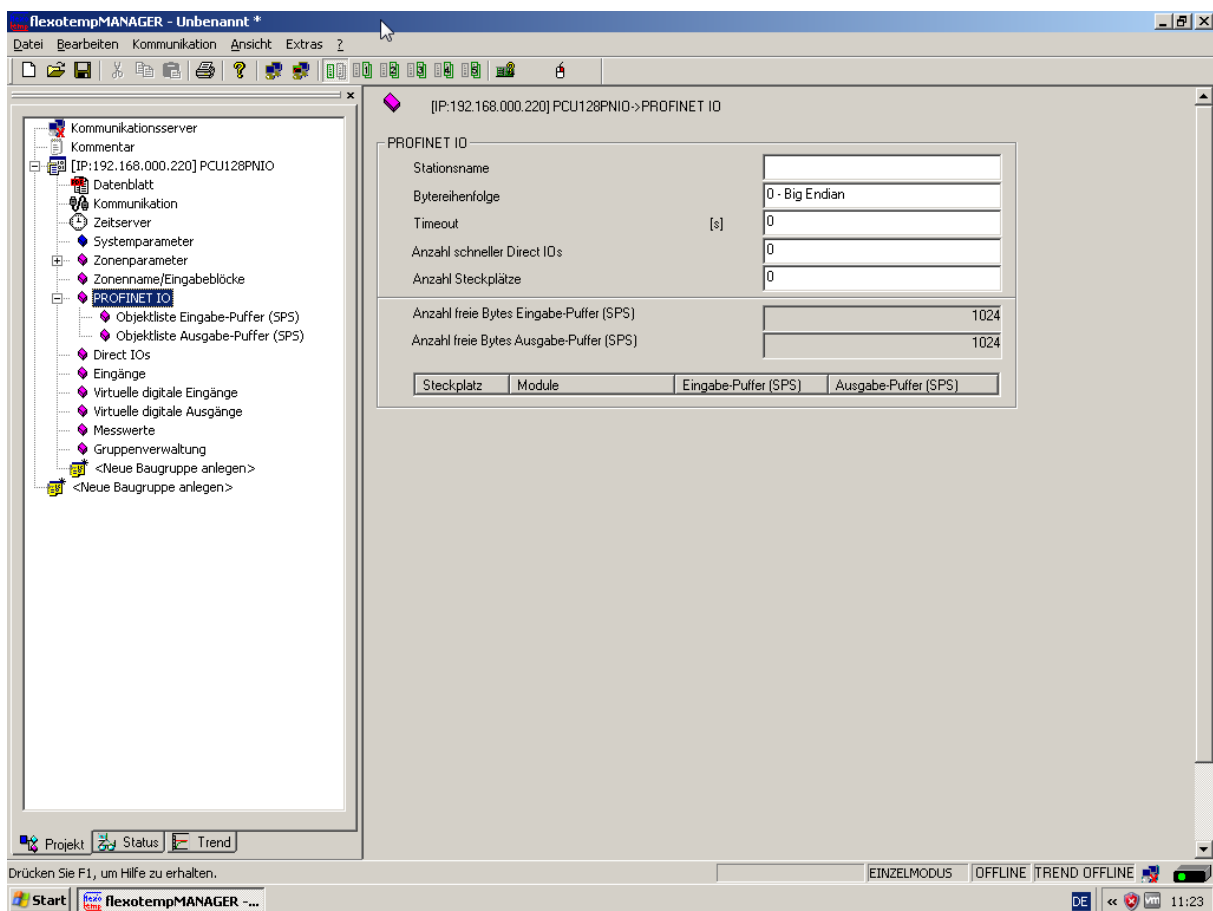
### 7.1 Configuration

Before PROFINET IO is configured for control, PNIO must be configured on controller side.

Therefore change to the project side in flexotempMANAGER and select the menu item <PROFINET IO> beneath the PNIO controller. There are

- Communication settings
- Object list for input-/output buffer (PLC)

specified.



Example PCU128PNIO

#### 7.1.1 Communication settings

##### Settings

Station name

To address an I/O device, it must be named.

Station name [1...240 characters] here must match the device name in SIMATIC Manager. Assignment of device name in SIMATIC Manager see chapter 8 Step 7 and/or chapter 8 Step 9.



Be sure to note that the name does NOT contain any capital letters!

byte sequence

Choice between

0 – Big Endian (byte with the most significant bit is stored first)

1 – Little Endian (byte with the least significant bit is stored first)



Be sure to note that the configuration of the in-/output buffer (sequence, slots, and type of modules) must later match exactly the configuration of the PNIO controller module in the SIMATIC Manager (see chapter 8).

Timeout

All outputs are set to passive upon expiration of the time in seconds set here.  
If set to 0, the function is deactivated.

Number of fast Direct IOs (only PCU)

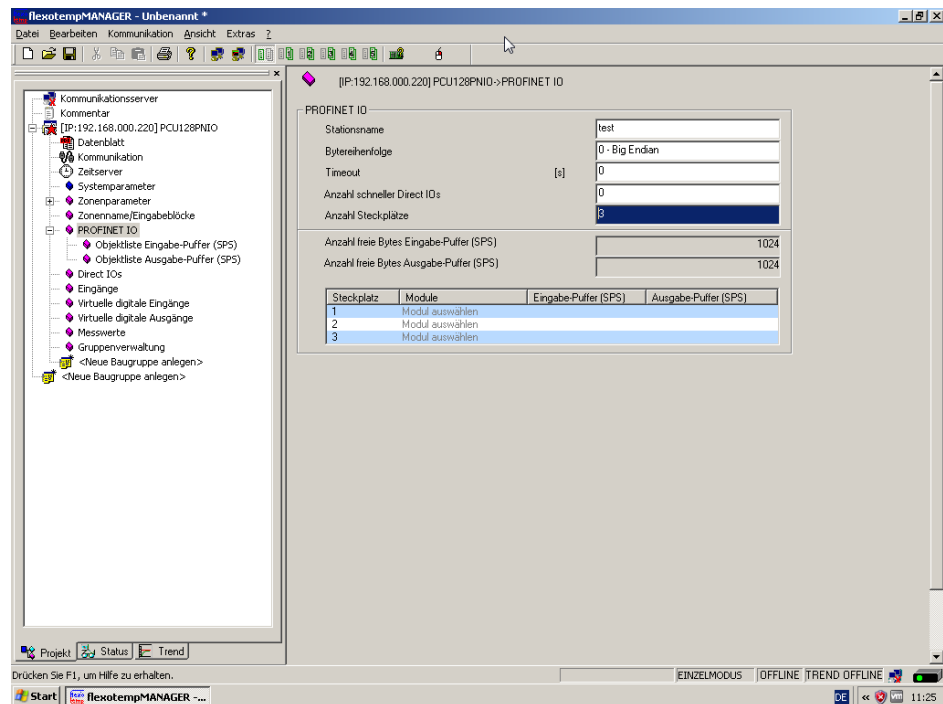
The number of the here specified Direct IO's [0 ... 20] will be handled preferentially, i.e. in each processing cycle.

Number of slots

The number of slots set here ([0 ... 32] (PCU) | [0 ... 20] (ETR), 0 = no slot occupied) must match with those in the control.



If, for example, here 3 is selected, then 3 slots are shown, which can then be assigned to in-/output modules.



Example PCU128PNIO

Number of free bytes input buffer (PLC)

Shows the number of free bytes for the current input buffer.

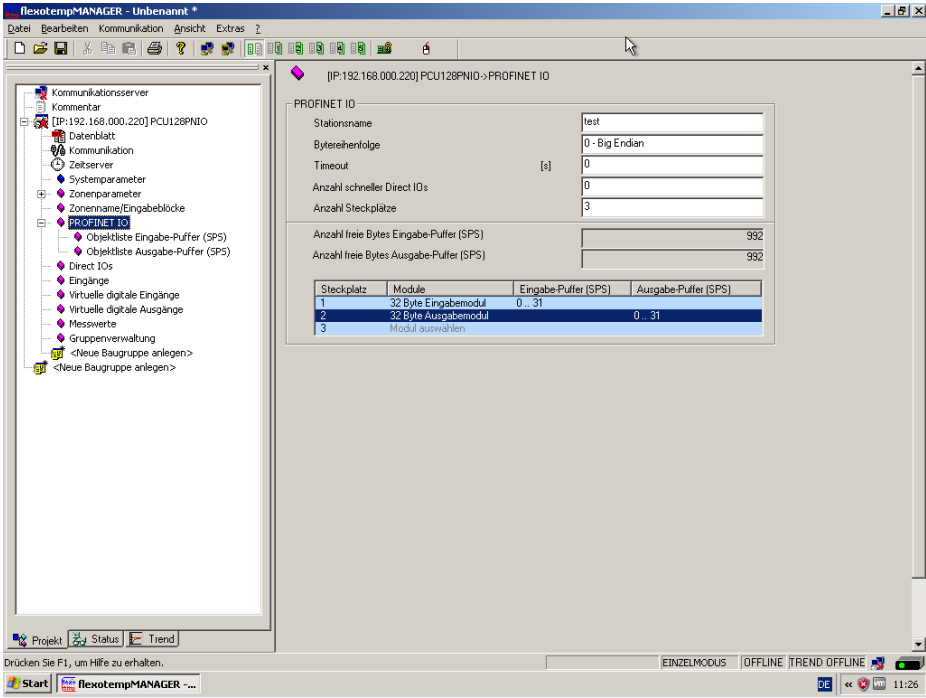
Number of free bytes output buffer (PLC)

Shows the number of free bytes for the current output buffer.

Slot If slots are occupied for in-/output modules (double click on <Select module>), words are reserved accordingly in the in-/output buffer (PLC).  
Input modules 16 Byte, 32 Byte, 64 Byte, 128 Byte,  
output modules 16 Byte, 32 Byte, 64 Byte, 128 Byte  
are available.

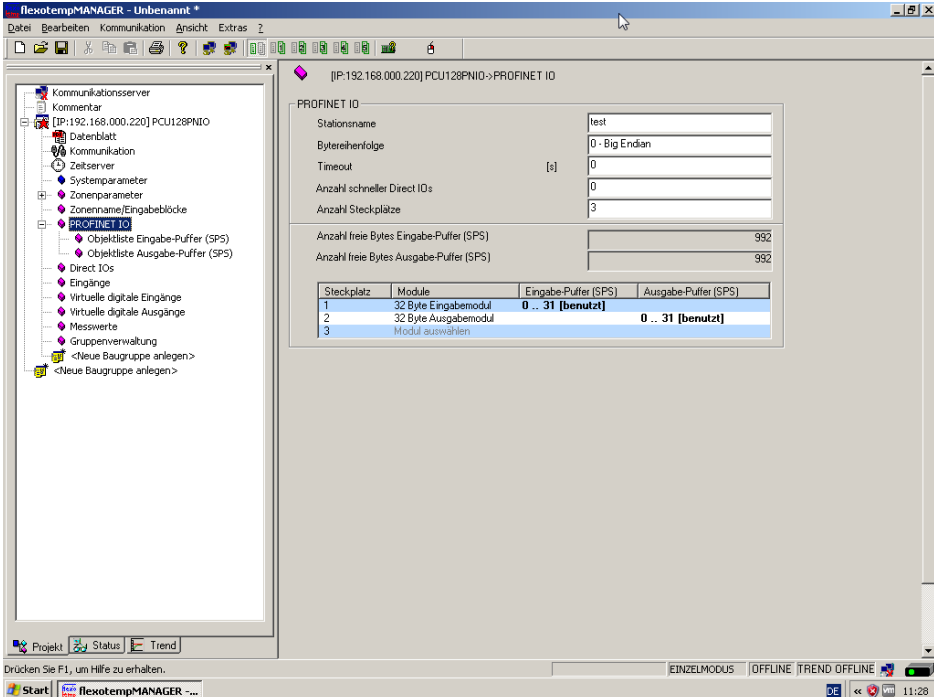


According to the still free number of bytes in the in-/output buffer only the modules are still available, not exceeding the number of bytes free.  
 I.e. if all bytes of the output buffer are full, only input modules are available for the free slots.



Example PCU128PNIO

If the object lists in-/output buffer are occupied, this will displayed under <Project> <PROFINET IO> as <(used)>.



Example PCU128PNIO



A module can now only be deleted or converted to a module of the same type but different size if the remaining/modified modules provide enough space to

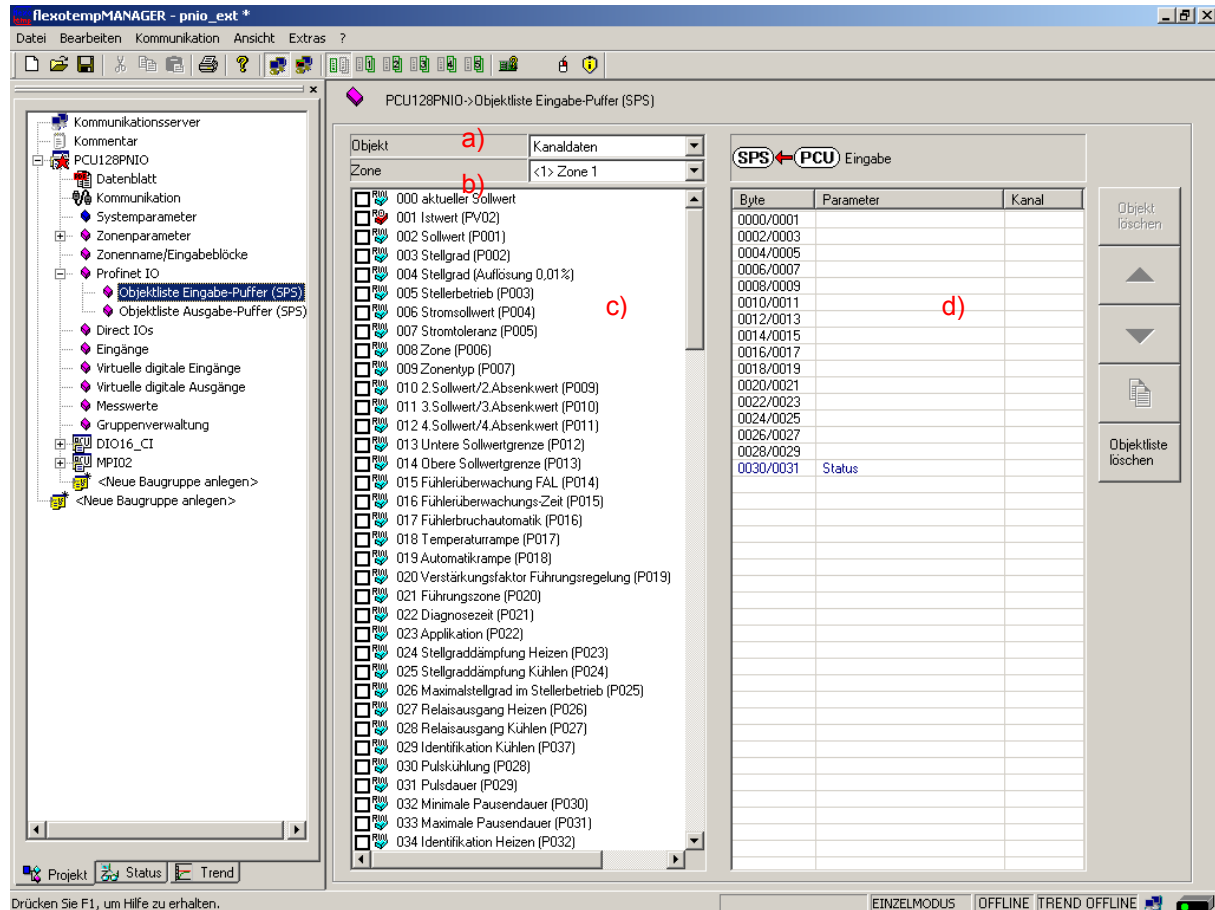
accommodate fully the configured object list.

The position of Static Objects, and MUX objects is maintained, the Parameter Channel migrates to the end of the in-/output buffer before the handshake word.



## 7.1.2 Define object list

By selecting the menu item <Object list input buffer (PLC)> or <Object list output buffer (PLC)> the list type is defined. The procedure is illustrated by the input object list buffer (PLC). The procedure for creating the object list output buffer (PLC) is identical.



Example PCU128PNIO

### Define object list

In the left display window for <Object list input\_buffer (PLC)> in a list of values a) all <Object> available for **PROFINET IO Objects** are shown

- Channel data (color see Parameter d): black
- Multiplexed Data [MUX] (color see Parameter d): blue
- System parameter [SYS] (color see Parameter d): green
- Parameter Channel [PAR] (color see Parameter d): red
- Direct IOs (only PCU) (color see Parameter d): orange)

Select in the display field b) for

- Channel data one Zone/Channel
- Multiplexed Data one MUX Channel

something.

In the display window c) all available parameters for PNIO controller, dependent on selection <Object>, are listed.

By clicking the check box of a parameter in c) the objects are moved to the right display window d).



Example PCU128PNIO

Create object list in-/output buffer (PLC) for one 32 Byte in-/output module each

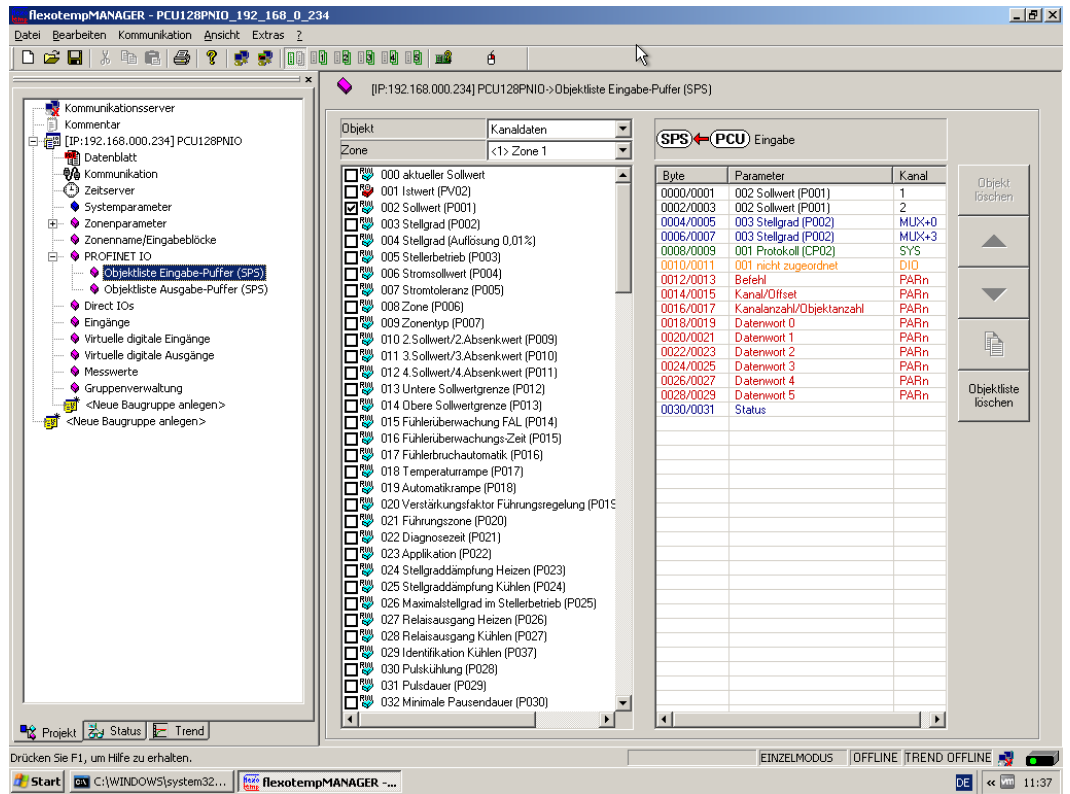
Change in flexotempMANAGER to project side  
Select menu item <PROFINET IO> beneath of PNIO controller  
Selection of menu item <Object list input buffer (PLC)>

a)	b)	b)	c) & d)
<Object>			<input type="checkbox"/> & Parameter
Channel data	<Zone>	<1> Zone 1	<input checked="" type="checkbox"/> 002 Setpoint value (P001)
Channel data	<Zone>	<2> Zone 2	<input checked="" type="checkbox"/> 002 Setpoint value (P001)
Multiplexed data	<MUX Channel>	<0> MUX Channel: +0	<input checked="" type="checkbox"/> 003 Degree of operation (P002)
Multiplexed data	<MUX Channel>	<3> MUX Channel: +3	<input checked="" type="checkbox"/> 003 Degree of operation (P002)
System parameters			<input checked="" type="checkbox"/> 001 Protocol (CP02)
Direct IO's			<input checked="" type="checkbox"/> 001 Node ID: 1...
Parameter Channel			<input checked="" type="checkbox"/> Parameter Channel n (6) Data words

Change in flexotempMANAGER to project side  
Select menu item <PROFINET IO> beneath of PNIO controller  
Selection of menu item <Object list output buffer (PLC)>

a)	b)	b)	c) & d)
<Object>			<input type="checkbox"/> & Parameter
Channel data	<Zone>	<1> Zone 1	<input checked="" type="checkbox"/> 002 Setpoint value (P001)

The following object list input buffer (PLC) is created.



Example PCU128PNIO



Creating of

- Parameter Channel
- Multiplexed Data

Note that in each case in the structure of the parameter channel in-/output buffer or handshake is applied. That it is checked prior to the application that in-/output modules are available and it is determined how large the Parameter Channel must be taking into account the size of in-/output buffer.

The following object list output buffer (PLC) is created.

The screenshot shows the 'flextempMANAGER' interface for a PCU128PNIO device. The main window displays a tree view on the left with 'Objektliste Ausgabe-Puffer (SP5)' selected. The central pane shows a list of objects for 'Zone 1', including parameters like '000 aktueller Sollwert', '002 Sollwert (P001)', and '003 Stellgrad (F002)'. The right pane shows the configuration for the 'Objektliste Ausgabe-Puffer (SP5)' with a table of data points.

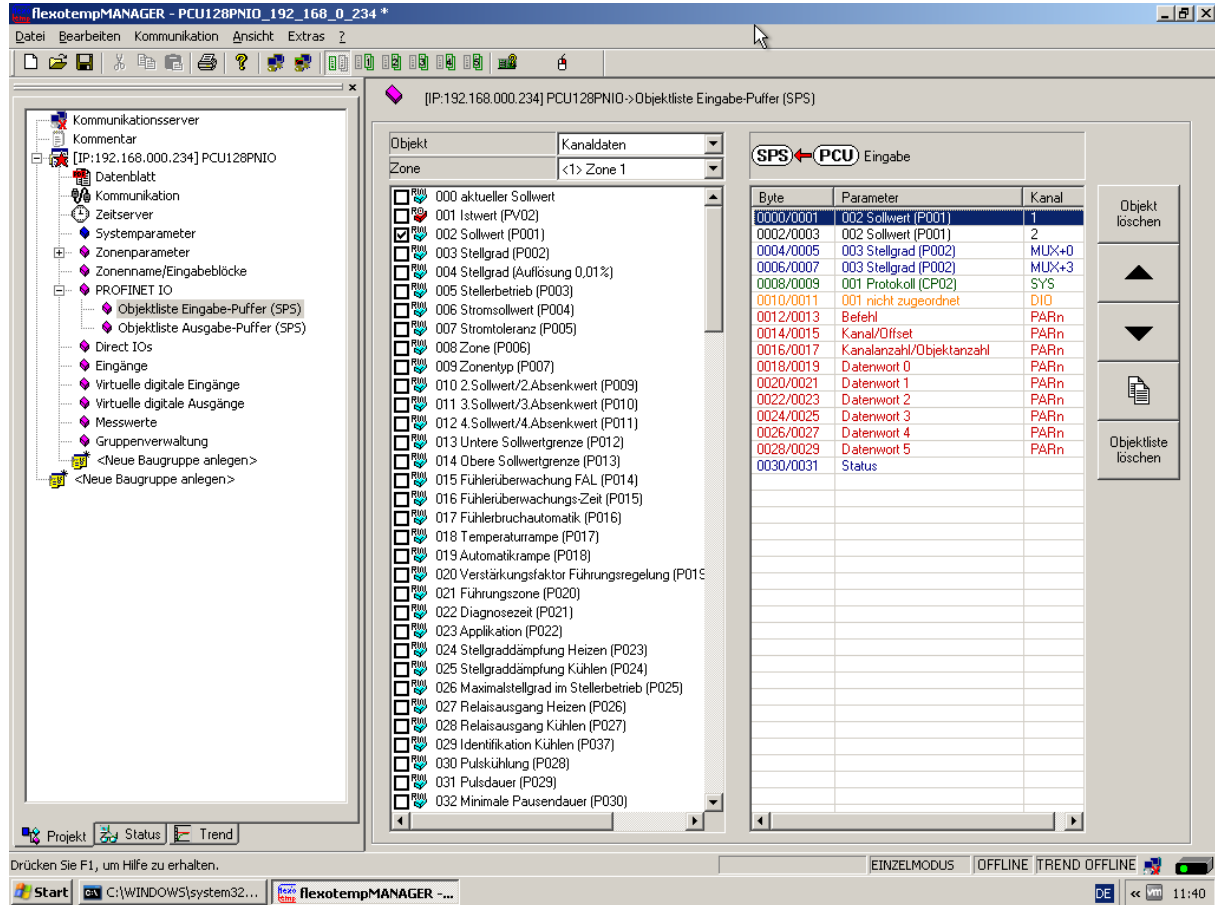
Byte	Parameter	Kanal
000/0001	002 Sollwert (P001)	1
002/0003		
0004/0005		
0006/0007		
0008/0009		
0010/0011		
0012/0013	Befehl	PAFn
0014/0015	Kanal/Offset	PAFn
0016/0017	Kanalanzahl/Objektanzahl	PAFn
0018/0019	Datenwort 0	PAFn
0020/0021	Datenwort 1	PAFn
0022/0023	Datenwort 2	PAFn
0024/0025	Datenwort 3	PAFn
0026/0027	Datenwort 4	PAFn
0028/0029	Datenwort 5	PAFn
0030/0031	Handshake	

Example PCU128PNIO

### 7.1.3 Edit object list

In an already created object list an element of the buffer is selected.

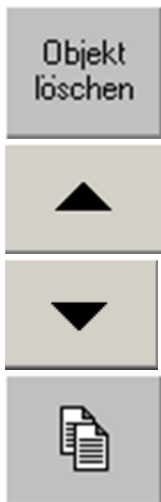
Beside the right display window, the keys for processing the object list are activated.



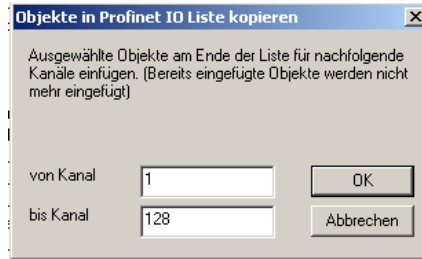
Example PCU128PNIO

#### Further editing of object lists

By the keys



- <Delete object>  
(selected object is deleted from the preset object list)
- <Arrow key up / down>,  
(selected object is moved up / down in the preset object list)
- <Copy objects in PROFINET IO list>  
(ONLY for Channel Data!)  
To get a list of similar objects quickly, that only differs by channel number, use the key <Copy objects in PROFINET IO list>. The objects are copied in the preset object list.



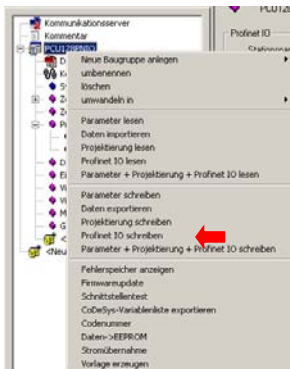
Example PCU128PNIO



- <Delete object list>  
 (all objects are deleted from the displayed list after confirmation of the query  
 <Delete object list really? >  
 with <OK>)

the object lists can be edited.

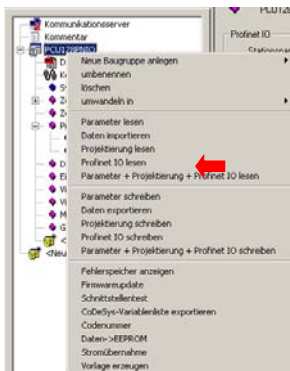
### 7.1.4 Write / Read object list



Example PCU128PNIO

#### Write object list

To activate newly created object lists for in-/output buffers (PLC) in flexotempMANAGER, they must be written into the PNIO controller. For this purpose, the context menu of PNIO controller is selected by the secondary mouse button and <Write PROFINET IO> is chosen. The object list is written directly to the PNIO controller.



Example PCU128PNIO

#### Read object list

By menu item <Read PROFINET IO> (display context menu of PNIO controller by secondary mouse button) the object lists in the PNIO controller for in-/output buffers (PLC) are read into flexotempMANAGER. The object list is directly read into flexotempMANAGER.

**!** Caution! Previously created lists in flexotempMANAGER that have not yet been written to the PNIO controller will be overwritten by this operation

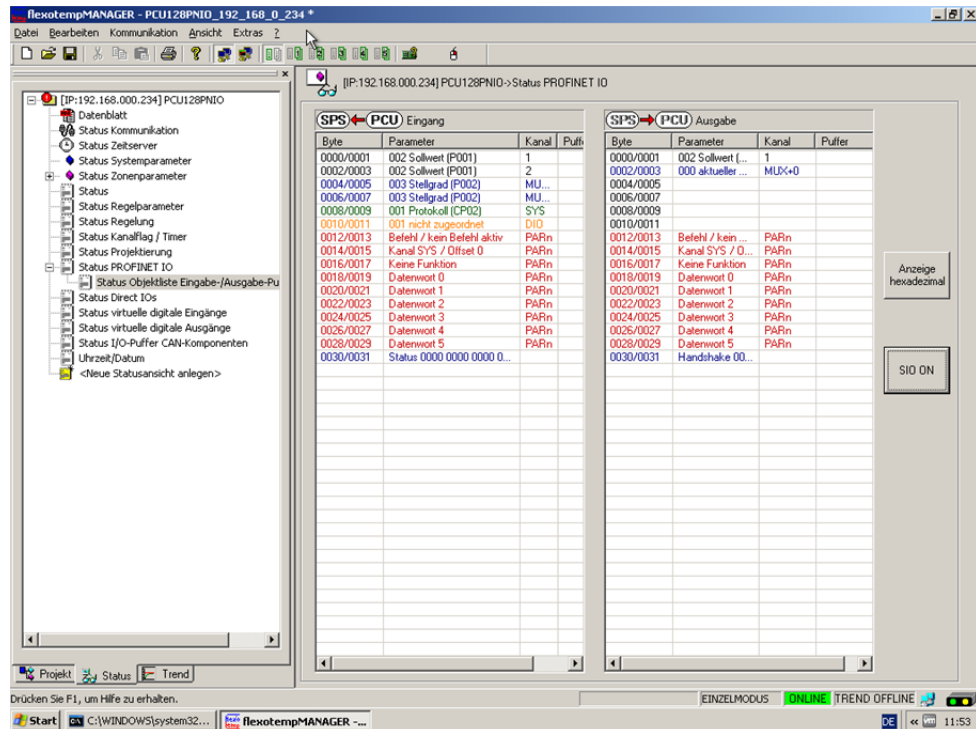


If you switch between menu items PROFINET IO of the project back and forth and you are ONLINE, a query, whether the respective PROFINET IO information should be read, occurs.

## 7.2 Online Mode

To test the function PROFINET IO without control or to debug the buffer sent by the controller, you can switch to the view <Project> <Status> in flexotempMANAGER. When switching to the menu item <Status PROFINET IO> the current object lists are read from the PNIO controller.

Then, the object lists of the input and the output buffer (PLC) are presented.



Example PCU128PNIO

### 7.2.1 SIO Mode

Change into SIO mode ON, the in-/output buffers (PLC) can be handled by flexotempMANAGER.

SIO ON

Is the SIO mode turned off, the buffers are only read and displayed in online mode. The buffers are described in this mode only by the controller.

The key indicates that the SIO mode can be turned **ON** in this mode.

SIO OFF

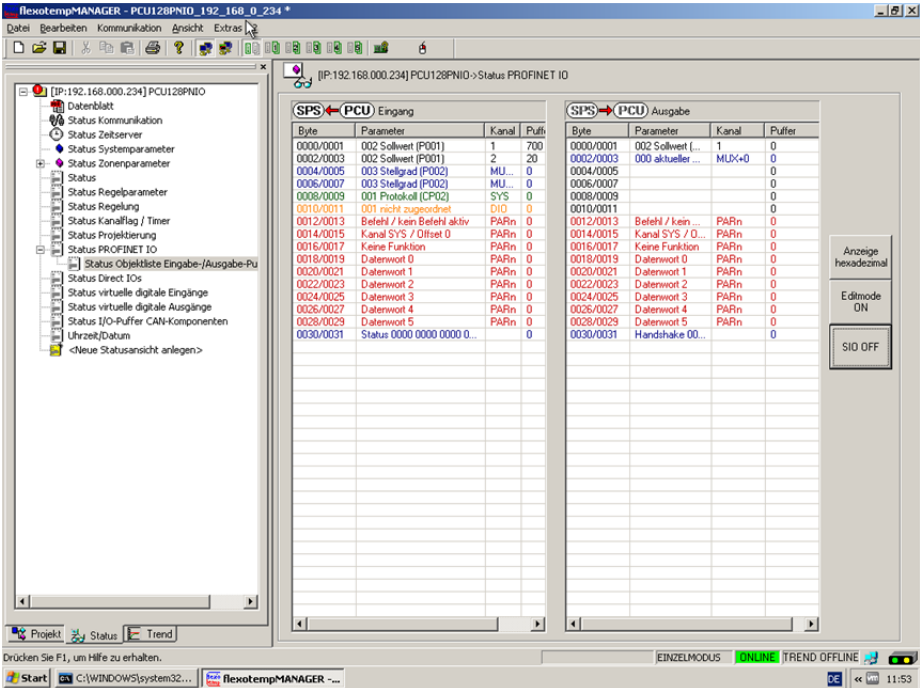
Is the SIO mode switched on, then the read and write buffers that are sent from the controller via PROFINET IO to the PNIO controller, are no longer evaluated. The buffers are now operated via flexotempMANAGER!

The key indicates that the SIO mode can be turned **OFF** in this mode.

**SIO Mode ON**

Anzeige  
 dezimal

Display  
 of buffer content  
 decimal

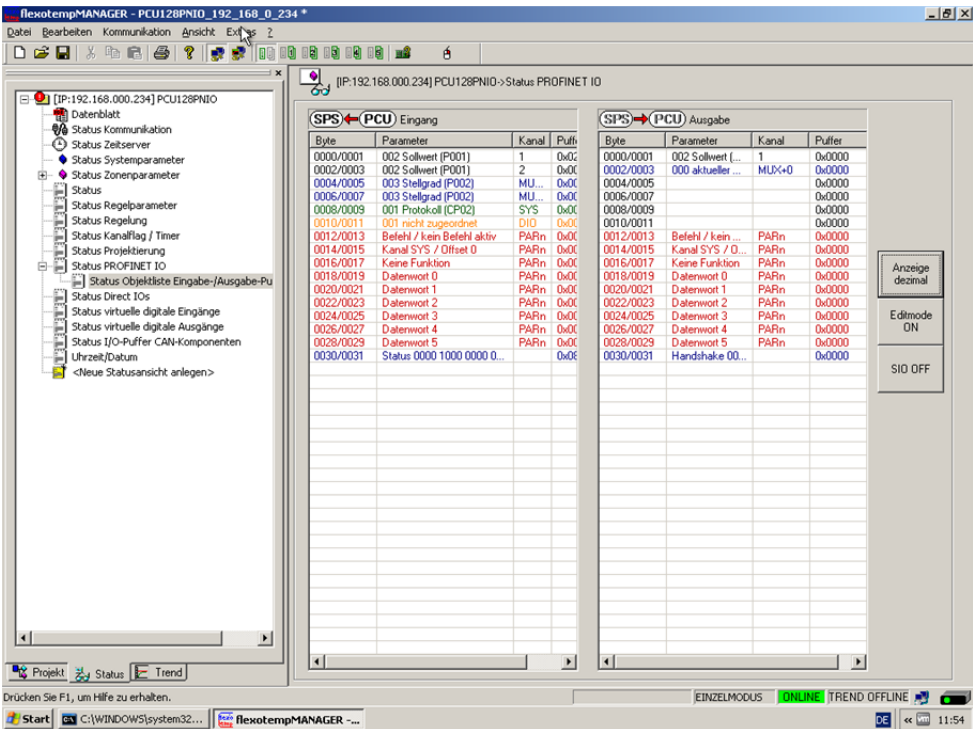


Example PCU128PNIO

**SIO Mode ON**

Anzeige  
 hexadezimal

Display  
 of buffer content  
 hexadecimal



Example PCU128PNIO

## 7.2.2 Edit Mode

The Edit Mode is only available in **SIO Mode ON**.

Editmode  
ON

In order to test the function of the handshake, it is possible to switch to **Edit mode**. In Edit mode, no more communication takes place. The user can edit the values of the output buffer.

Press key **Edit mode ON**.

Byte	Parameter	Kanal	Puffer
0000/0001	002 Sollwert (P001)	0	700
0002/0003	002 Sollwert (P001)	1	1000
0004/0005	003 Stellgrad (P002)	MU...	0
0006/0007	003 Stellgrad (P002)	MU...	0
0008/0009	001 Protokoll (CP02)	SYS	0
0010/0011	001 nicht zugeordnet	DIO	21449
0012/0013	Befehl / kein Befehl aktiv	PARn	0
0014/0015	Kanal SYS / Offset 0	PARn	0
0016/0017	keine Funktion	PARn	0
0018/0019	Datenwort 0	PARn	0
0020/0021	Datenwort 1	PARn	0
0022/0023	Datenwort 2	PARn	0
0024/0025	Datenwort 3	PARn	0
0026/0027	Datenwort 4	PARn	0
0028/0029	Datenwort 5	PARn	0
0030/0031	Status 0000 0000 0000 0000		0

Byte	Parameter	Kanal	Puffer
0000/0001	002 Sollwert (P001)	0	0
0002/0003			
0004/0005			0
0006/0007			0
0008/0009			0
0010/0011			0
0012/0013	Befehl / kein Befehl aktiv	PARn	0
0014/0015	Kanal SYS / Offset 0	PARn	0
0016/0017	keine Funktion	PARn	0
0018/0019	Datenwort 0	PARn	0
0020/0021	Datenwort 1	PARn	0
0022/0023	Datenwort 2	PARn	0
0024/0025	Datenwort 3	PARn	0
0026/0027	Datenwort 4	PARn	0
0028/0029	Datenwort 5	PARn	0
0030/0031	Handshake 0000 0000 00...		0

Example PCU128PNIO



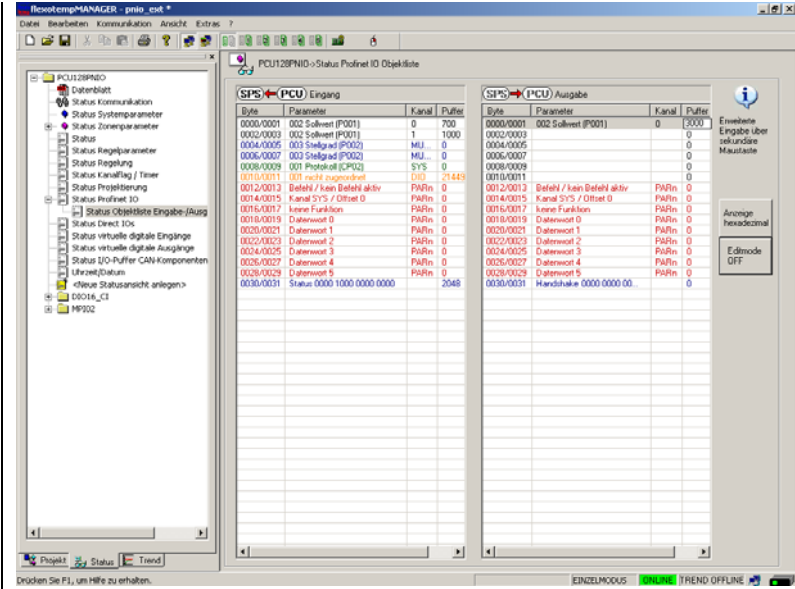
Example PCU128PNIO

For the first setpoint value object channel 1 in the object list output buffer, another value (example: 300) should be specified.

Select the field buffer behind the object on the right side and activate the entering of values by secondary mouse button. The value 3000 is entered.

Take care of the data format at specification of values of parameters.





Example PCU128PNIO

<RETURN> confirms the entered value.

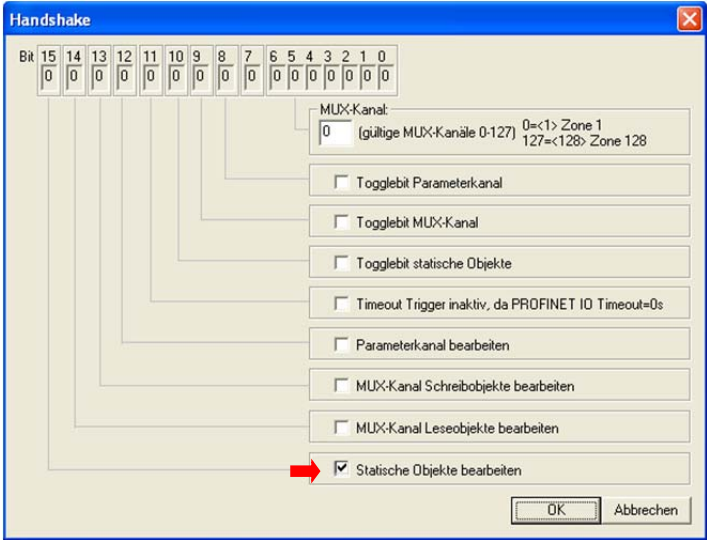
Handshake

The bits in the handshake (see chapter 3.2) should be set accordingly before writing the object list.

The object "handshake" (right window) must be select and the secondary mouse button must be pressed. A dialogue box appears.

In the dialog box

- Set <Edit static objects> (✓)



Example PCU128PNIO

Exit dialog box by <OK>.

According to the settings you will see a value other than 0 in the buffer for handshake.

**Editmode  
 OFF**

When you exit the **Edit mode**, the specified are written to the PNIO controller and trigger the appropriate actions.

The appearing box <Write object list output buffer?> has to be acknowledge accordingly.

After an error-free processing the content is copied into the input buffer.

### 7.3 Change value of a MUX object

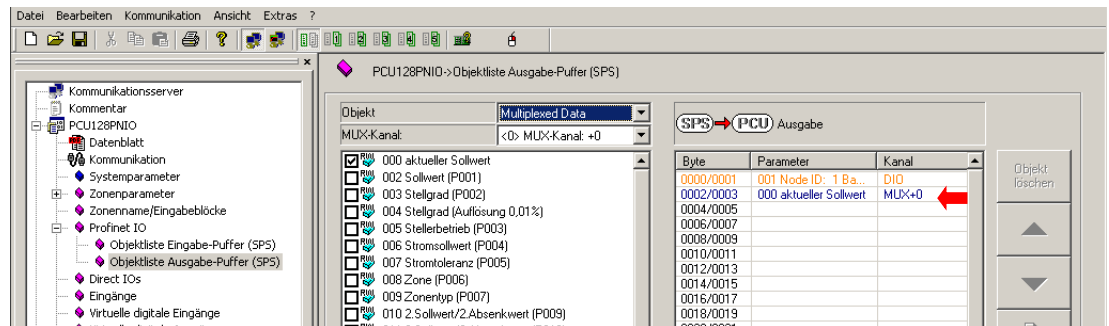


Example  
PCU128PNIO

Change value of a MUX object

To change the value of a MUX object, assign the MUX object "Setpoint value" MUX+0 to the object list of the output buffer (see chapter 7.1.3 Edit object list).

a)	b)	b)	c) & d)
<Object>			<input type="checkbox"/> & Parameter
Multiplexed data	<MUX Channel>	<0> MUX Channel: +0	<input checked="" type="checkbox"/> 000 Current setpoint value



Example PCU128PNIO

#### Write object list

See chapter 7.1.4 Write / Read object list

#### Change to Online mode

See chapter 7.2 Online Mode

#### Change to SIO mode

See chapter 7.2.1 SIO Mode

#### Change to Edit mode

See chapter 7.2.2 Edit Mode



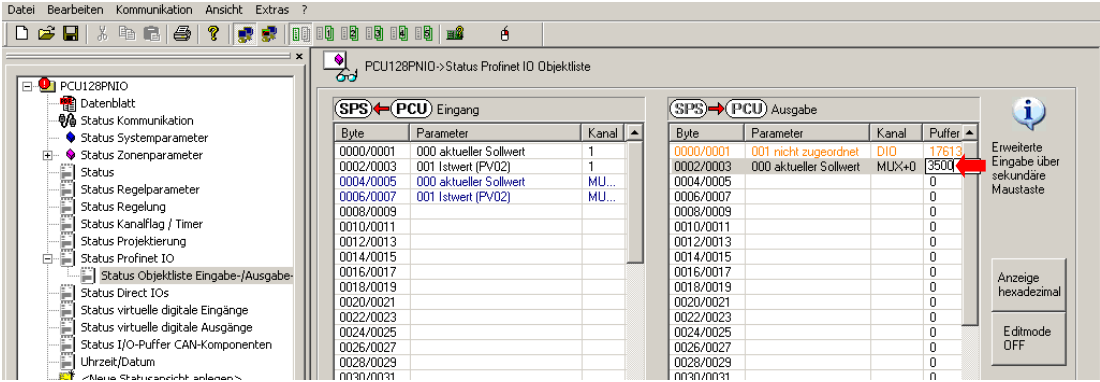
Example  
PCU128PNIO

For the MUX object in the object list of output buffer „Current setpoint value“, another value (example: 350) should be specified for MUX Channel 0.

Select the field buffer behind the object on the right side and activate the entering of values by secondary mouse button. The value 3500 is entered and confirmed by <RETURN>.



Take care of the data format at specification of values of parameters.



Example PCU128PNIO

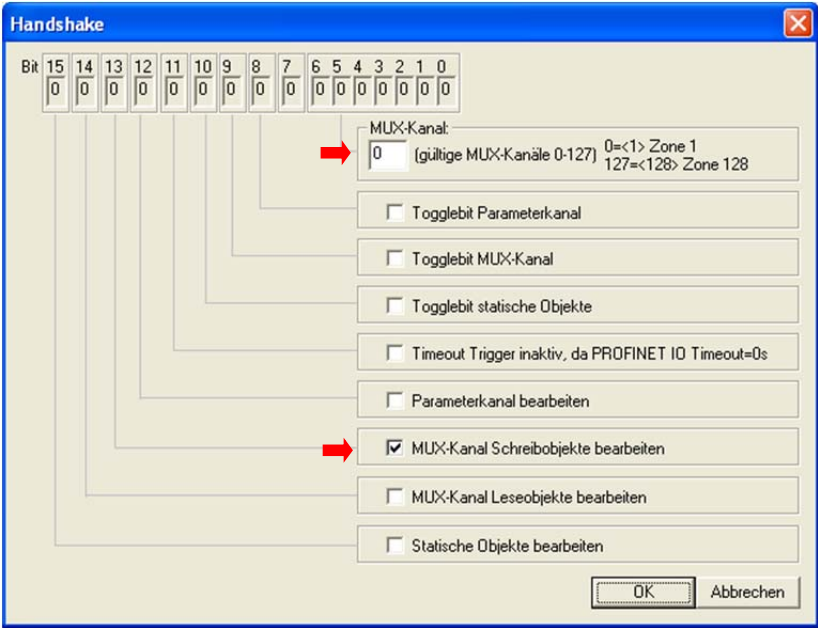
<RETURN> confirms the entered value.

**Handshake**

Bits in the handshake (see chapter 3.2) should be set accordingly before writing the object list. The object "handshake" must be select and the secondary mouse button must be pressed. A dialogue box appears.

In the dialog box

- Set <MUX Channel edit write objects> (✓)



Example PCU128PNIO

Exit dialog box by <OK>.

According to the settings you will see a value other than 0 in the buffer for handshake.

Editmode  
 OFF

When you exit the **Edit mode**, the specified values are written to the PNIO controller and trigger the appropriate actions. The appearing box <Write object list output buffer?> has to be acknowledge accordingly. After an error-free processing the content is copied into the input buffer.

## 8 Integration of a PCUXXPNIO in a SIMATIC 300 PROFINET Network

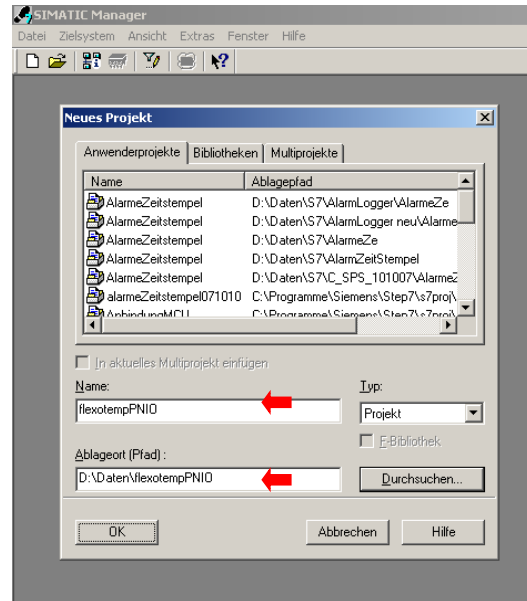
To integrate a temperature PNIO controller in a SIMATIC 300 PROFINET network, the following steps have to be executed.



The example describes the procedure using the SIMATIC Manager of Siemens.

Create empty and/or new project.

In this example the project has the name „**flexotempPNIO**“ and is stored in the folder „**D:\Daten\flexotempPNIO**“.

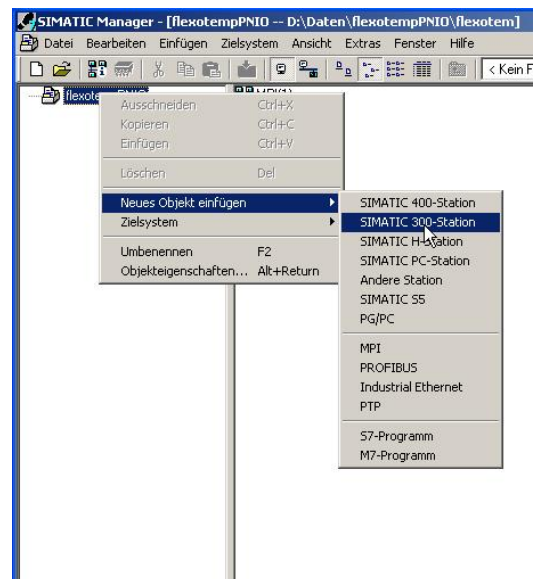


### SIMATIC Manager step 1

Before the PROFINET IO connection is established, the hardware of the PLC Controllers must be configured first.

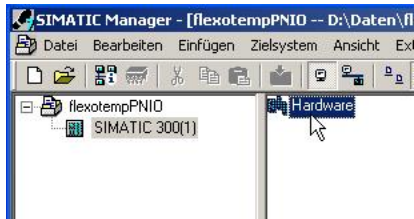
In this example a **S7315-2 CPU** is used.

Select in the dialog <Neues Objekt einfügen> of SIMATIC Managers <**SIMATIC 300-Station**>.

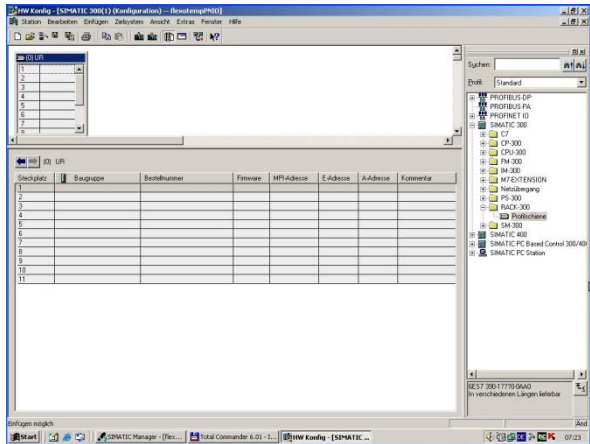


**SIMATIC Manager step 2**

Create hardware configuration  
 Click on <Hardware> to reach the hardware Manager.



Create module rack for hardware.



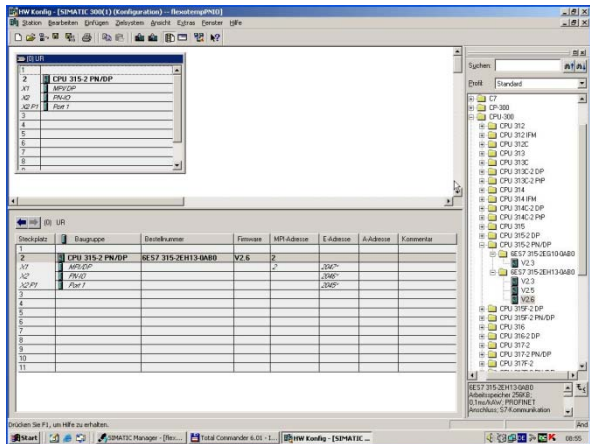
**SIMATIC Manager step 3**

Equip slots of the module rack (rail) with the required components.

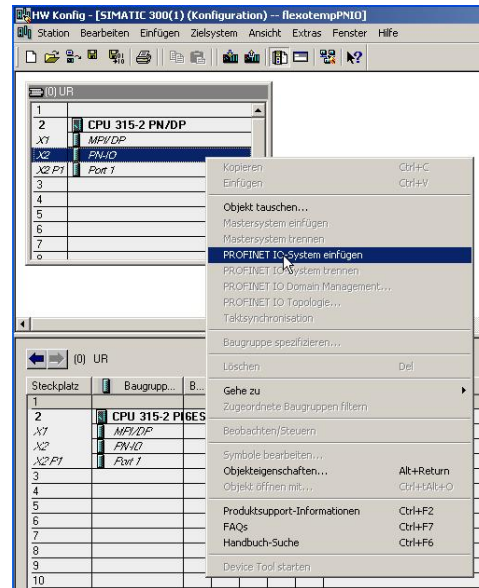
The are available in the hardware catalog of SIMATIC Manager.

In this example **CPU 315-2 PN/DP** is selected.

This CPU has a Profibus DP and a PROFINET Controller.

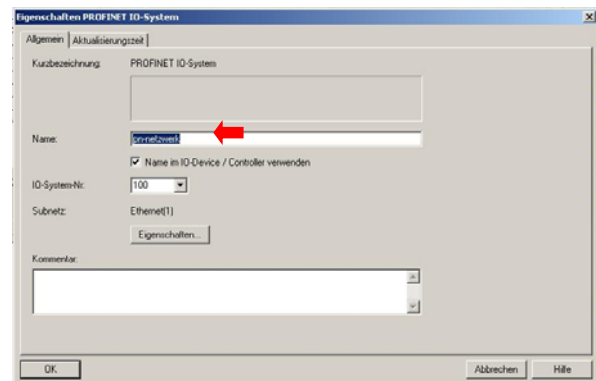


One <PROFINET IO-System einfügen>



Give the PROFINET IO System a name and set a check for <Namen im IO-Device / Controller verwenden>.

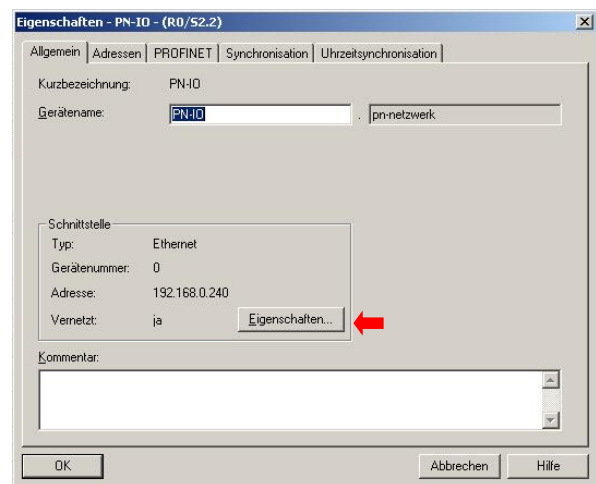
In this example the PROFINET IO system is named „pn-netzwerk“.



Click on the PROFINET IO Controller Module, to reach the property dialog of the PROFINET IO Controller.

Configure the PROFINET IO network.

Click on <Eigenschaften>, to define the properties of the interface.

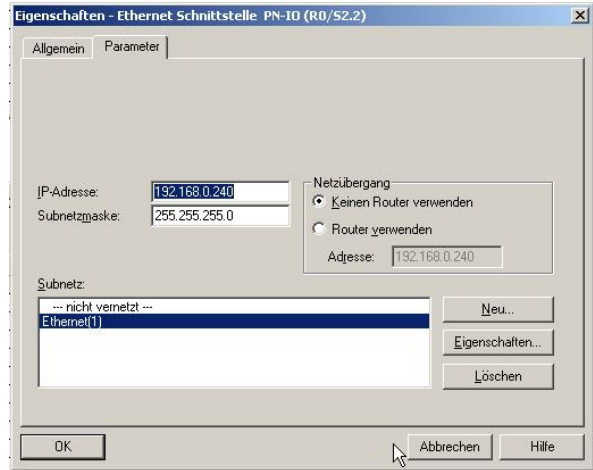


In the dialog **<Eigenschaften–Ethernet Schnittstelle>** IP address and subnet mask of PROFINET IO Controller are defined.

Select the appropriate subnet by selection menu for **<Subnetz>**.

Select desired settings and confirm by **<OK>**.

The network settings for PROFINET IO Controllers are completed.



### SIMATIC Manager step 4

After completion of the Control Setup the following steps are explained, that are necessary for the Setup of the PROFINET IO component **PCU128PNIO**.

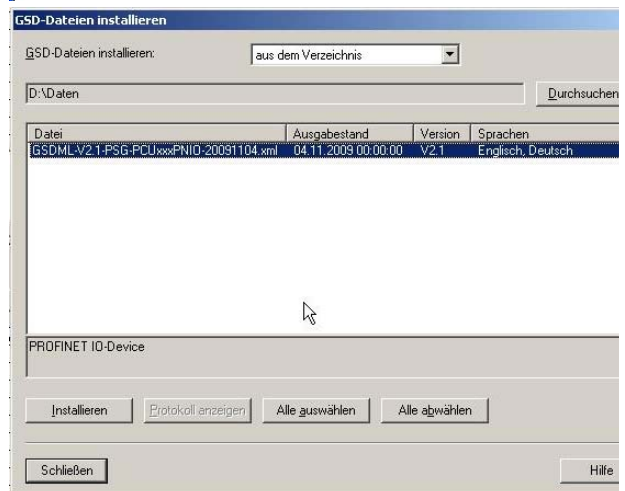
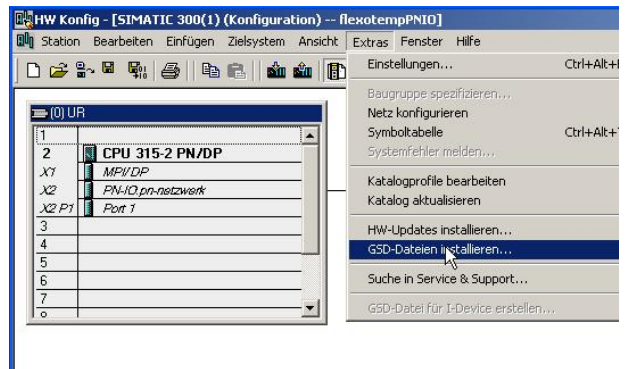
The characteristic features of the communication module **PCU128PNIO** are defined in the form of an electronic data sheet, the GSDML file. If the file does not exist, it must be installed in the SIMATIC Manager.

Select **<Extras> <GSD-Dateien installieren>**.

Select XML file



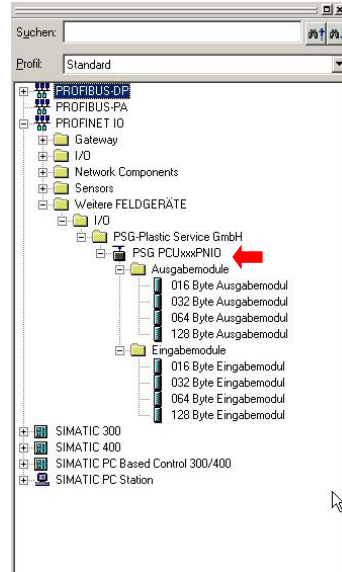
Note: Please download the latest version of the Meusburger homepage (at least 09/2011)



**SIMATIC Manager step 5**

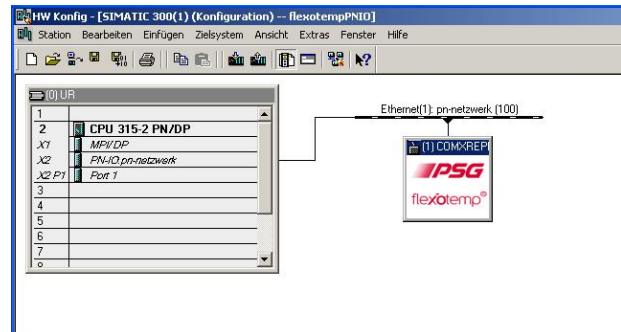
After installing the GSD file the PROFINET IO module PCU128PNIO is in the catalog of the SIMATIC Manager available. The module can be connected to the network section of the PROFINET IO network.

Drag the element „PSG PCUxxxPNIO“ with the mouse button to the network section and keep the primary mouse button pressed.

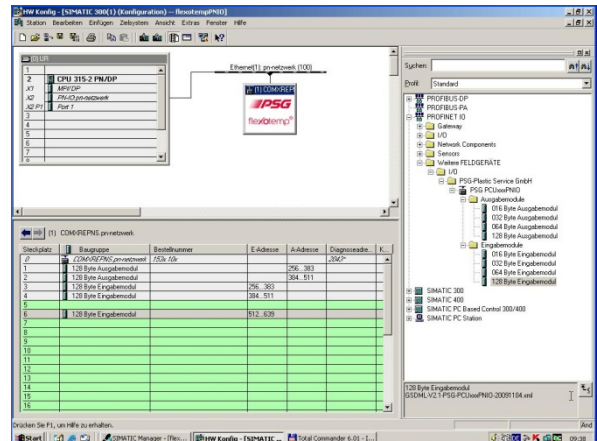


**SIMATIC Manager step 6**

The PROFINET IO Slave participant is now connected with the PROFINET IO bus section.



In the next step, the process image of the I/O data can be specified. All relevant data types of the PCUxxxPNIO are defined in the GSD file. The appropriate modules can now easily be drawn with the help of the mouse pointer on the appropriate slots. Here, keep the primary mouse button pressed.



Be sure to note that the configuration of the I/O data (sequence, slots, and type of modules) must later match exactly the configuration of the PCUxxxPNIO component in the flexotempMANAGER.

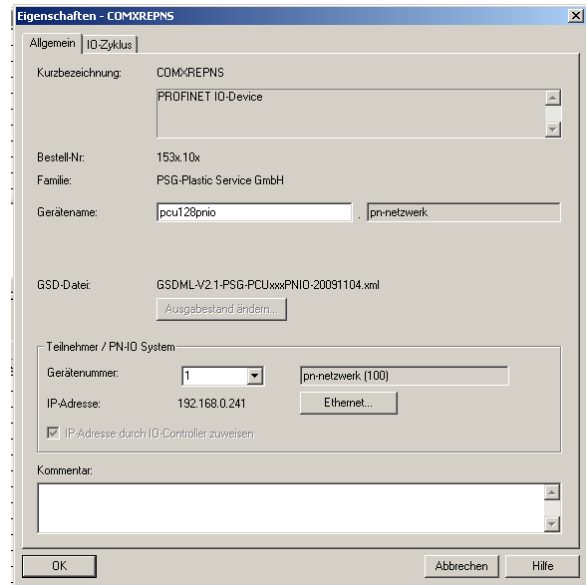


### SIMATIC Manager step 7

Open the properties dialog by clicking on the PROFINET IO slave on the bus section.

Unambiguous device name  
 see chapter 8 Step 9.

The device name must match the station name of **PCUxxxPNIO**, that is used in flexotempMANAGER (see chapter 7.1.1)



Before an I/O device can be addressed by an I/O controller, you must have a device name. In PROFINET IO, this approach has been chosen because names are easier to handle than complex IP addresses. Assigning a device name for a specific I/O device can be compared with the setting of the PROFIBUS address for a DP slave.

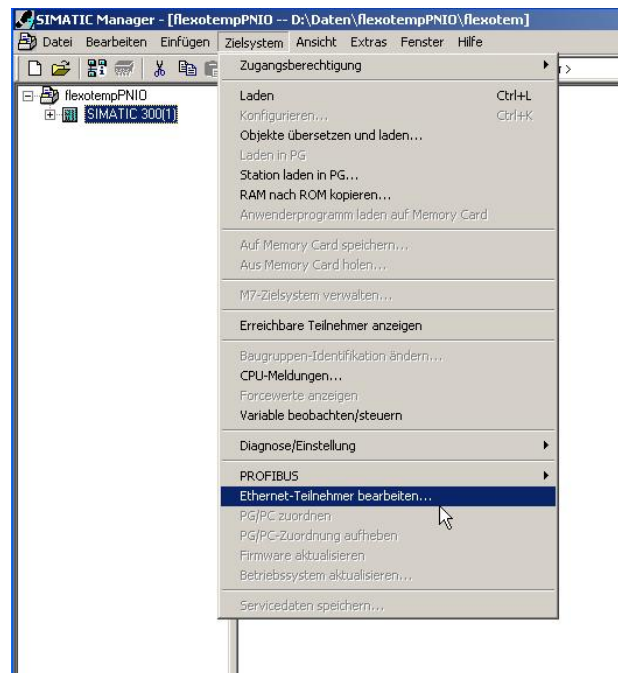
### SIMATIC Manager step 8

Exit hardware configuration and change to main window of SIMATIC Manager.

Select item **<Ethernet-Teilnehmer bearbeiten>** in menu **<Zielsystem>**.

The following dialog contains the button **<Durchsuchen>**.

After selection of the button, all online reachable participants are shown.

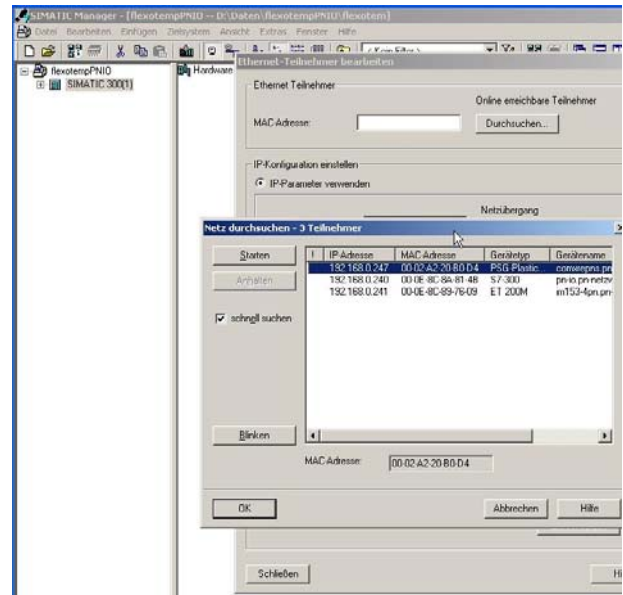


This function is only available, when as PG/PC interface an Ethernet interface was chosen in SIMATIC Manager.

## Integration of a PCUXXXPNIO in a SIMATIC 300 PROFINET Network

From the list of participants reached select the corresponding slave module and select it (a connection to the PROFINET network must exist).

Confirm with the button **<OK>**.



### SIMATIC Manager step 9

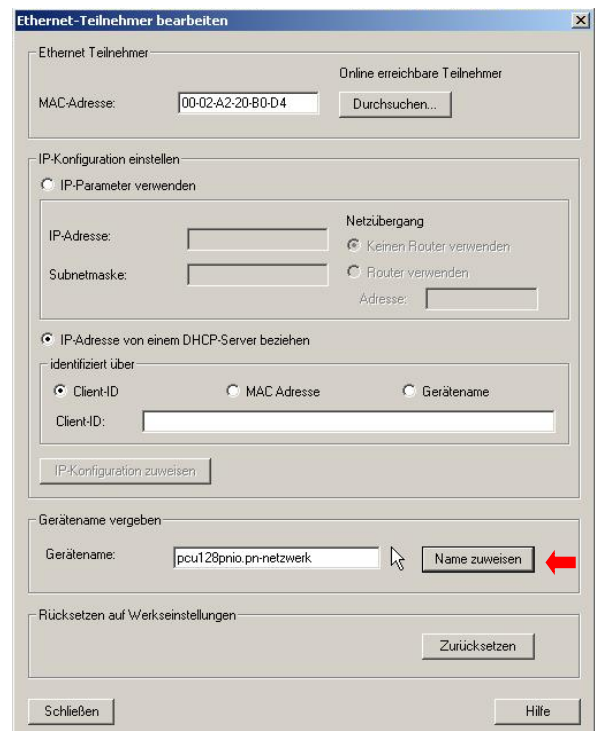
The dialog **<Ethernet-Teilnehmer bearbeiten>** contains in the window MAC address the MAC address of the **PCUxxxPNIO** component.

The device name can be changed here or extended by the network name. In this case consider the following syntax:

[device name].[network name]

Change of name assigned by pressing the **<Name zuweisen>** button. If the name is successfully assigned, this is confirmed by a corresponding dialog. Only now the participant is addressable by the PROFINET controller.

The device name SIMATIC Manager is also valid as station name flexotempMANAGER (see chapter 7.1.1).



Be sure to note that the name does NOT contain any capital letters!

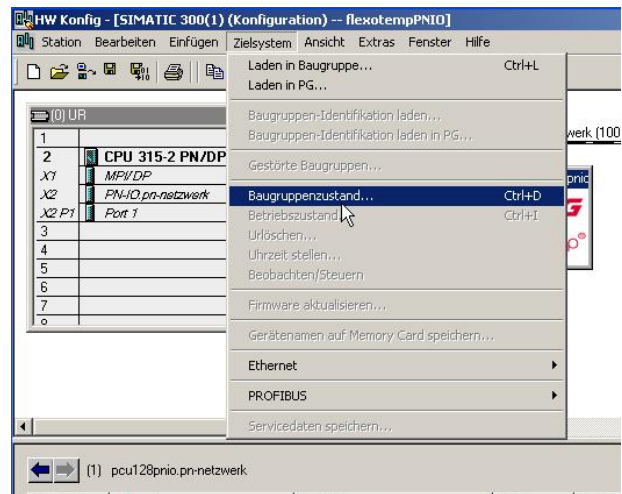
**SIMATIC Manager step 10**

Upon successful transfer

- of project into S7
- of configuration with the aid of flexotempMANAGER into **PCU128PNIO**

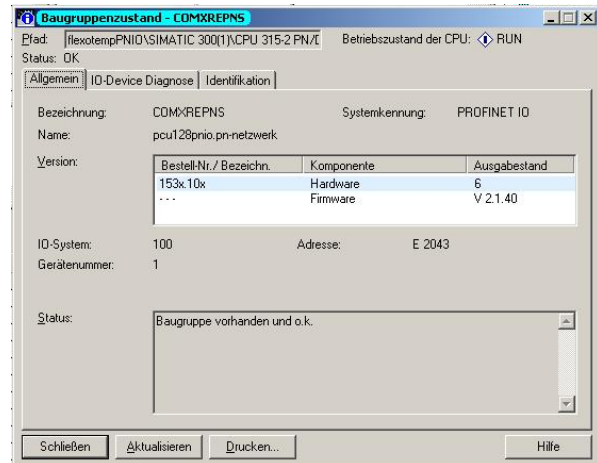
the component status can be checked in the hardware configuration in the SIMATIC Manager.

Select the appropriate component by mouse and then menu item <Zielsystem> <Baugruppenzustand>.



If everything works properly,

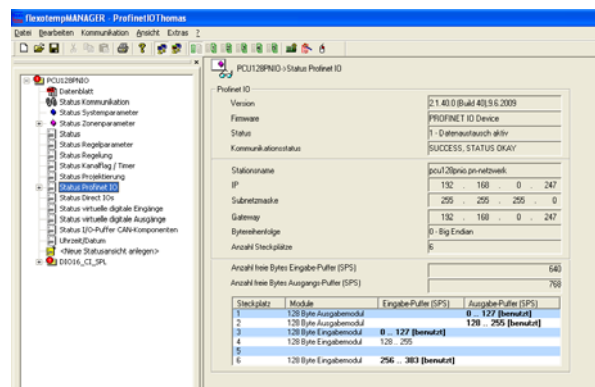
- the specified name
- the status „**Baugruppe vorhanden und o.k.**“ are shown.



If the status message "**Baugruppe projiziert, aber nicht vorhanden...**" is displayed, please check the module configuration in the S7 and the flexotempMANAGER to match. Also check the station name see flexotempMANAGER and the device name in the S7.

The current status of PROFINET IO can also be seen on the status side PROFINET IO in flexotempMANAGER:

- Version: Firmware version PCUXXXPNIO
- Firmware: Firmware type on PCUXXXPNIO
- Status: Status of data exchange
- Communication status
- Active station name
- Active IP address
- ...
- Active module configuration



## 9 Appendix

### 9.1 GSD File

See [www.meusburger.com](http://www.meusburger.com) under Downloads the relevant PNIO controller.

### 9.2 PROFINET PLC Data types

Data type	Description	Range	PLC
Boolean	8-bit Boolean	False (0x00) and True (0x01)	byte
Integer8	8-bit signed integer	-128...127	byte
Integer16	16-bit signed integer	-32,768...32767	word
Integer32	32-bit signed integer	-2147438648...2147438647	dword
Unsigned8	8-bit unsigned integer	0...255	byte
Unsigned16	16-bit unsigned integer	0...65,535	word
Unsigned32	32-bit unsigned integer	0...4294967295	dword
FloatingPoint	32-bit IEEE-754 floating point value	1.19209290e-38...3.4028235e+38	dword

### 9.3 Version History

Version	Date	Changes
1.00.05	9/13/2017	flexotempMANAGER Station name = SIMATIC Manager device name;
1.00.04	7/18/2017	Chapter Check sum calculation MUX data added; Timeout
1.00.03	7/3/2014	Parameter Channel for command 33 added (only ETR)
1.00.02	12/5/2013	SIMATC step 4 specified; ETR132PNIO added
1.00.01	3/24/2010	Project PROFINET IO Parameter Timeout amended
1.00.00	1/25/2010	First edition
0.00.01	1/18/2010	Preliminary version