ibaLink-SM-64-SD16

Engineering Guide

Manual

Issue 2.3

Measurement and Automation Systems

Manufacturer

iba AG

Koenigswarterstr. 44

90762 Fuerth

Germany

Contacts

Main office	+49 911 97282-0
Fax	+49 911 97282-33
Support	+49 911 97282-14
Engineering	+49 911 97282-13

E-Mail: iba@iba-ag.com

Web: www.iba-ag.com

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, deviations cannot be excluded completely so that the full compliance is not guaranteed. However, the information in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.

The current version is available for download on our web site http://www.iba-ag.com.

Protection note

Windows® is a label and registered trademark of the Microsoft Corporation. Other product and company names mentioned in this manual can be labels or registered trademarks of the corresponding owners.

Version	Date	Revision	Chapter	Page	Author	SW-Version
2.3	07/2012	Note added (Used under CFC)	2		st	

Table of Contents

1 About this manual

This manual describes in detail the required software configuration steps in the SIMADYN D system and in the iba systems in order to use the ibaLink-SM-64-SD16 card.

1.1 Target group

This manual addresses in particular the qualified professionals who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded to as professional if he/she is capable of assessing safety and recognizing possible consequences and risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

1.2 Basic knowledge

The following knowledge is required for the operation of the software ibaLink-SM-64-SD16.

❑ Basic knowledge of Windows operating system

❑ Basic knowledge of operating web browsers

❑ Basic knowledge of SIMADYN D, STRUC3, STRUC4

❑ Basic knowledge of ibaPDA-V6 or ibaLogic respectively

1.3 Notations

In this manual the following notations are used:

Action	Notation
Menu command	Menu „Logic diagram“
Call of menu command	„Step 1 – Step 2 – Step 3 – Step x” Example: Select menu „Logic diagram – Add – New logic diagram ”
Keys	<Key name> Example: <Alt>; <F1>
Press keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Button name> Example: <OK>; <Cancel>
File names, Paths	„File name“, „Path” Example: „Test.doc“

1.4 Symbols used

If safety instructions or other information are used in this manual, they mean:

⚠ DANGER

The non-observance of this safety information may result in an imminent risk of death or severe injury:

- By an electric shock!
- Due to the improper handling of iba software products which are coupled to input and output procedures with control function!

⚠ WARNING

The non-observance of this safety information may result in a potential risk of death or severe injury!

⚠ CAUTION

The non-observance of this safety information may result in a potential risk of injury or material damage!

Note

A note indicates special requirements or actions to be observed.

Important information

Information that a special indication has to be observed, e.g. exceptions from the general rule.

Tip

Tip or example which serves as helpful information or a trick to facilitate the work.

Other documentation

Reference to supplementary documentation or further literature.

2 General

This document describes the communication between SIMADYN D P16 systems under STRUC V2.3 until V3.x and under STRUC V4.1 until V4.2.x with the iba systems (ibaPDA, ibaScope, ibaLogic, iba periphery system) via optical fibers.

In the following, the versions are simply named as STRUC V3 and STRUC V4.

The communication partner at the end of the iba systems is the ibaFOB-4i or ibaFOB-io board or another iba component which uses the iba standard FOB telegram format.

The communication partner at the SIMADYN D end is the ibaLink-SM-64-SD16 board.

The board has 2 different SIMADYN D drivers integrated that can be selected via the switch for selecting operating modes:

Driver 1: CS3 for STRUC version 3

This variant acts like a CS3 MMC communication board.

Driver 2: CS12 for STRUC version 4

This variant acts like a CS12 interface module.

2 different drivers had to be developed because the first variant does not work under STRUC 4-in contrast to the SIMADYN D documentation and despite the possibility to program it.

Furthermore, the second variant does not work under STRUC version 3, because the emulated CS12 board is not known under STRUC version 3.

Chapter 2 describes the procedures for configuring the connection under STRUC version 3 and STRUC version 4. The differences between STRUC V3 and STRUC V4 are clearly pointed out.

Note STRUC configuration

The manual describes the integration of ibaLink-SM-64-SD16 into existing SIMADYN D system as an amendment to the STRUC manuals.

The STRUC engineering and configuration rules must be generally adhered to.

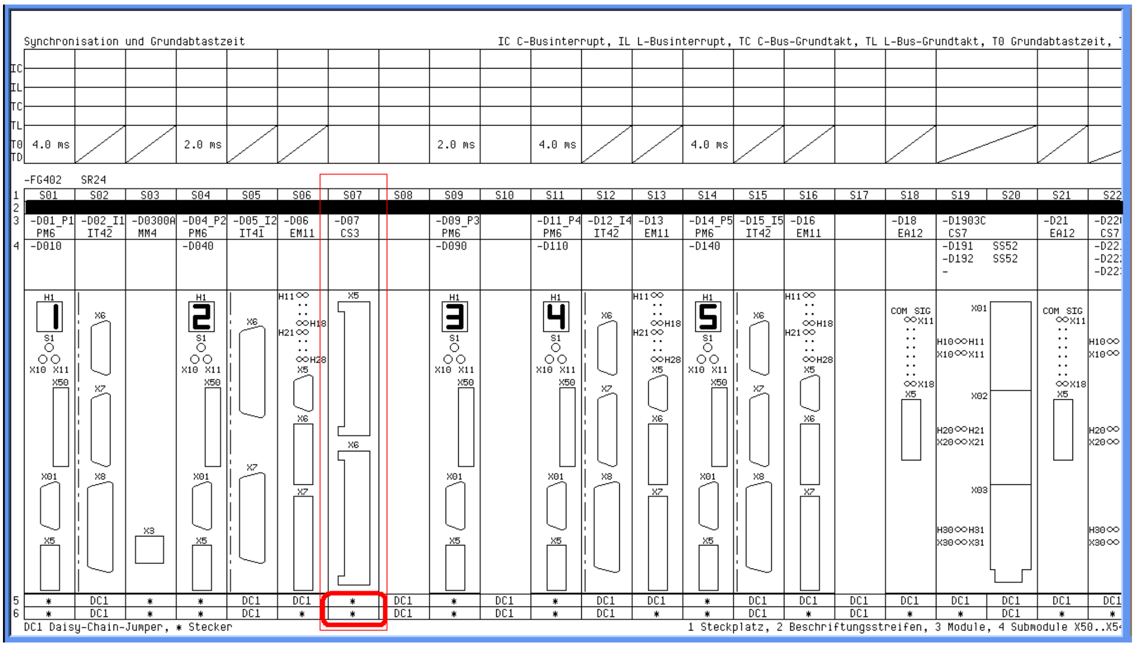
Important note

The board cannot be used under CFC, since only the expanded address range (128 kByte) is used here. (The connector EMA = N) cannot be configured.)

Note

The daisy chain jumper must be set on the board, see below.

The C3 module, which is outlined in red in the picture below, is configured for ibaLink-SM-64-SD16. In the STRUC master program the daisy chain jumper are displayed as “not set”. Contrary to this description, the daisy chain jumper on the board must be set.



Synchronisation und Grundabtastzeit

IC C-Businterrupt, IL L-Businterrupt, TC C-Bus-Grundtakt, TL L-Bus-Grundtakt, T0 Grundabtastzeit,

DC1 Daisy-Chain-Jumper, * Stecker

1 Steckplatz, 2 Beschriftungsstreifen, 3 Module, 4 Submodule X50..X5

2.1 Properties of ibaLink-SM-64-SD16 under STRUC V3

The ibaLink-SM-64-SD16 board acts in the SIMADYN D P16 system under STRUC V2.3 / V3.xx like a genuine MMC CS3 interface module.

- ❑ The board can only be used in sub-racks with C-bus connection (i. e. SR12 or SR24, or SR1 and SR5, respectively)
- ❑ The board can only communicate with P16 processor modules that have the C-bus connection, i.e. not with the technology processors (e. g. PG16, PG26, PS16, PT20/G/M).
- ❑ The board features a dual-port RAM (DPR) with a memory capacity of 16 KB.
- ❑ The MMC "COMNET" protocol is used as the communication protocol. All data is exchanged, including initialization tasks and monitoring functions, according to this protocol.
- ❑ The data communication in SIMADYN D is based on standard send/receive and telegram blocks with the @MMC initialize block.
- ❑ SIMADYN D and the iba system connected can be switched on and off fully independent of each other. SIMADYN D can also run without the iba system connected and vice versa.
- ❑ The execution cycle in the ibaLink-SM-64-SD16 is 1 ms.

2.2 Properties of ibaLink-SM-64-SD16 under STRUC V4

The ibaLink-SM-64-SD16 board acts in the SIMADYN D P16 or P32 system under STRUC V4.xx like a rack link CS12 interface module.

- ❑ The board can only be used in sub-racks with C-bus connection (i. e. SR12 or SR24).
- ❑ The board can communicate with P16 processor modules that have the C-bus connection, i. e. not with the technology processors, such as PG16, PG26, PS16, PT20/G/M.
- ❑ The board can also communicate with P32 processor modules. The P32 processors are all equipped with a C-bus connection.
- ❑ The board features a dual-port RAM (DPR) with a memory capacity of 32 KB.
- ❑ The CIP3 protocol is used as the communication protocol. All data is exchanged, including initialization tasks and monitoring functions, according to this protocol.
- ❑ The data communication in SIMADYN D is based on standard send/receive and telegram function blocks with the @CS1 initialize function block.
- ❑ SIMADYN D and the iba system connected can be switched on and off fully independent of each other. SIMADYN D can also run without the iba system connected and vice versa.
- ❑ The execution cycle in the ibaLink-SM-64-SD16 is 1 ms.

P16 / P32 note

The board can, in principle be used under STRUC V4 even by P32 processors and in P16/P32 hybrid mode. The configuration procedure on P32- is similar to that on P16 processors. Differences and deviation are explicitly pointed out.

2.3 General configuration information

- ❑ There is a maximum of 2 send channels and two receive channels. Each channel corresponds to one "module" in ibaPDA and ibaLogic-V3, respectively.
- ❑ The telegrams are configured in STRUC V3 in channel mode 0 "Handshake/no overwrite" and under STRUC V4 in channel mode 1 "Refresh/overwrite".
- ❑ Both "direct" and "indirect" communication is possible.
- ❑ Every channel has exactly 32 analog values and 32 binary values. Every channel is represented by a telegram.
- ❑ Data can be configured in the "N2" and in the "N4" format, however, not in a mixed form within one telegram.

P32 note

There is no direct / indirect communication, but only a so-called "virtual" communication which approximately corresponds to indirect communication with P16.

The "NF" data format is also possible.

3 Engineering and configuring in SIMADYN D

3.1 Engineering and configuring in general

3.1.1 Communication types

SIMADYN D offers 2 communication modes:

❑ Indirect communication
The data of different function packages can be combined in one telegram.
The receive function block is processed at the beginning of the calculation cycle, and must be placed in the special @RECEIVE function package. The send function block is processed at the end of the cycle, and must be placed in the special @SEND function package.

❑ Direct communication
The data to be transmitted is directly connected to the send or receive function block, respectively. This function block can be placed in any function package, and is processed during the processing sequence in the cycle.

P32 note-P32 features Virtual communication

The name of a "virtual communication reference" (corresponding to the telegram name with P16 indirect) is stated at receive or send function block, respectively. This channel can be referenced by all the function packages of the processor, so that all the function packages have access to the telegram data.

3.1.2 Data formats

3.1.2.1 Analog values

The analog values to be transmitted can be specified in the 16-bit format (N2, I2, O2, V2, etc.), or in the 32-bit format (N4, I4, O4, V4, etc.), however, may not be mixed within the same telegram. Data is generally transmitted in the internal presentation (raw value).

Example:

The N2 value 50.0 % is transmitted in its internal presentation, i. e. as the integer value 8192 or as the real value 8192.0. An N4 value of 10.0 % is transmitted as the integer value 107374182 or as the real value 107374182.4.

Scaling to physical values or percentage values can be carried out at the ibaPDA or ibaLogic end.

Important note

In the case of transmission in real mode, the analog values are converted to the IEEE float format. The precision of the N4 data is thereby reduced to 24 bits!

In the following chapters, all formats based on 16 bits are referred to as N2 format, whilst all the formats based on 32 bits are referred to as N4 format.

P32 note

The "NF" data format is also possible with P32. This format corresponds to the IEEE float format, so that conversion is not necessary.

3.1.2.2 Binary values:

The 32 binary values are transmitted as two N2 values. For this purpose, groups of 16 bits must be combined using a "BSC" function block, or one N2 value must be converted to 16 binary values using an "SBC" function block, respectively.

In N4-format telegrams, too, the binary values are transmitted as 2 N2 values.

3.1.3 Telegram format

The telegram format is strictly defined as follows: Exactly 32 analog values (as 32 x N2 or 32 x N4) and 32 digital values (as 2 x N2 or 1 x N4) must be defined in this order.

N4-format telegrams can be configured with indirect communication only because direct telegram blocks are not available for this format.

N2-format telegrams can be configured with direct or indirect communication.

3.1.4 Channel mode (MOD connector)

❑ STRUC V3

The ibaLink-SM-64-SD16 can process the channels in channel mode 0 (HANDSHAKE or "no overwrite") only.

This is due to the fact that the data length in channel mode 1 (REFRESH or "overwrite") is limited to 16 bytes because no coordination mechanisms are available.

❑ STRUC V4

The ibaLink-SM-64-SD16 can process the channels in channel mode 1 (REFRESH or "overwrite") only. Channel mode "R" is specified in the case of P32.

3.1.5 Channel name (AT/AR connector)

The channel names are defined as follows:

'M0PDADAT'	send channel, module 0 (from SIMADYN D to ibaPDA or iba-Logic, respectively)
'M1PDADAT'	send channel, module 1 (from SIMADYN D to ibaPDA or iba-Logic, respectively)
'PDAM0DAT'	receive channel, module 0 (from ibaLogic to SIMADYN D)
'PDAM1DAT'	receive channel, module 1 (from ibaLogic to SIMADYN D)

3.1.6 Transport system (CTS connector)

❑ STRUC V3
The transport system (TS) of the MMC type is configured in the master program. All the function blocks involved in the communication process refer to the TS via the TS name to be stated at the CTS connector. Furthermore, the TS are also assigned to the connector of the ibaLink-SM-64 board (connector X5 at the CS3 block).

➚ Refer to chapter 3.2

❑ STRUC V4
An explicit transport system does not exist in the case of STRUC version 4. The module name defined in the master program must be stated at the CTS connectors of the send/receive function blocks and of the initialize function block.

➚ Refer to chapter 3.3.

3.1.7 Telegram length (LT connector)

The telegram length is determined by the board, depending on the SIMADYN D data format:

- ❑ N2 format: telegram length = 68 bytes (32 analog values x 2 bytes + 4 bytes binary values) with the direct send/receive function block: LT = 34 (number of words!) with the indirect telegram block: LT = 68 (number of bytes!)
- ❑ N4 format: telegram length = 132 bytes (32 analog values x 4 bytes + 4 bytes binary values)
  LT = 132 (number of bytes, indirect communication only)

P32 note

The telegram length is not explicitly stated at the CTV/CRV send/receive blocks, but is rather determined by the number and format of the data referenced.

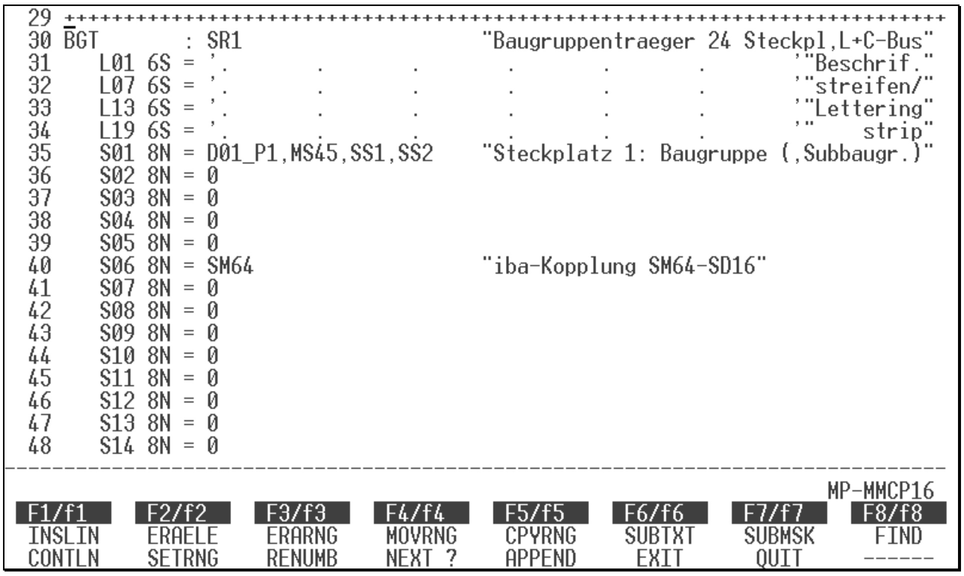
3.1.8 Initialize function block

- ❑ STRUC V3
  The initialize "@MMC" function block of the MMC communication is used to initialize the ibaLink-SM-64-SD16 board. The transport system name of the communication is stated at the CTS connector.
- ❑ STRUC V4
  The initialize "@CS1" function block of the interface module is used to initialize the ibaLink-SM-64-SD16 board. The board name of the communication is stated at the CTS connector.

3.2 Configuring the ibaLink-SM-64-SD16 board under STRUC V3

3.2.1 Configuring in the master program (MP)

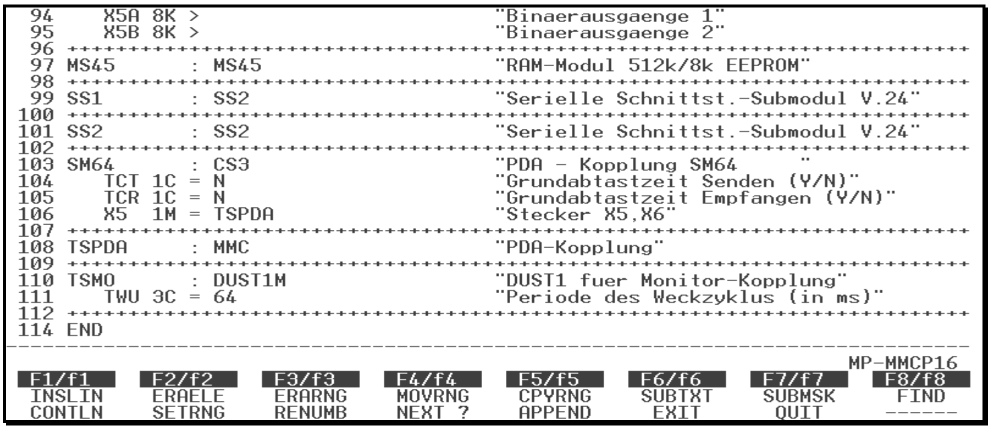
The board is configured for 1 slot in an SD sub rack with L and C bus connection.



```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
BGT         : SR1                      "Baugruppentraeger 24 Steckpl,L+C-Bus"
   L01 6S = '.     .       .       .       .       .        '"Beschrif."
   L07 6S = '.     .       .       .       .       .        '"streifen/"
   L13 6S = '.     .       .       .       .       .        '"Lettering"
   L19 6S = '.     .       .       .       .       .        '"    strip"
   S01 8N = D01_P1,MS45,SS1,SS2    "Steckplatz 1: Baugruppe (,Subbaugr.)"
   S02 8N = 0
   S03 8N = 0
   S04 8N = 0
   S05 8N = 0
   S06 8N = SM64                   "iba-Kopplung SM64-SD16"
   S07 8N = 0
   S08 8N = 0
   S09 8N = 0
   S10 8N = 0
   S11 8N = 0
   S12 8N = 0
   S13 8N = 0
   S14 8N = 0
------------------------------------------------------------------------------
```

Figure 1: Example: "SM64" board in slot 6

The ibaLink-SM-64-SD16 board is defined as type "CS3" (MMC communication). The transport system name, such as "TSPDA", must be stated at connector X5, and "N" must be assigned to connectors TCT and TCR.



```
   X5A 8K >                        "Binaerausgaenge 1"
   X5B 8K >                        "Binaerausgaenge 2"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MS45        : MS45                 "RAM-Modul 512k/8k EEPROM"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
SS1         : SS2                  "Serielle Schnittst.-Submodul V.24"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
SS2         : SS2                  "Serielle Schnittst.-Submodul V.24"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
SM64        : CS3                  "PDA - Kopplung SM64      "
   TCT 1C = N                      "Grundabtastzeit Senden (Y/N)"
   TCR 1C = N                      "Grundabtastzeit Empfangen (Y/N)"
   X5  1M = TSPDA                  "Stecker X5,X6"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TSPDA       : MMC                  "PDA-Kopplung"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TSMO        : DUST1M               "DUST1 fuer Monitor-Kopplung"
   TWU 3C = 64                     "Periode des Weckzyklus (in ms)"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
END
---------------------------------------------------------------------------
```

Figure 2: Example: SM64 board of the CS3 type

: TSPDA transport system of the MMC type.

3.2.1.1 Configuring the transport system (TS)

The transport system handles the exchange of telegrams between SD and the iba systems. Any user-defined name (maximum of 6 characters according to SD guidelines) is valid, for example, "TSPDA".

The transport system is of the MMC type.

The TS name be stated at the CTS connectors of all function blocks which refer to this communication, as well as at the connector of the SM64/CS3 board:

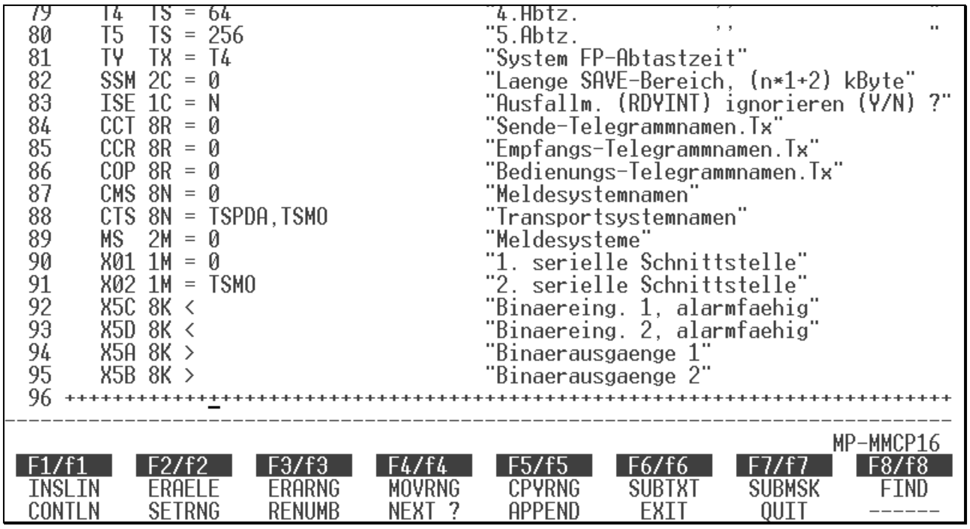
X5 at the SM64 board / type CS3 in the master program.

CTS At the processor module of the processor in which the data communication takes place.

CTS At the initialize block of the @MMC type in the special FP @SEND .

CTS At the send function block in the case of direct communication.

CTS At the telegram block in the case of indirect communication.

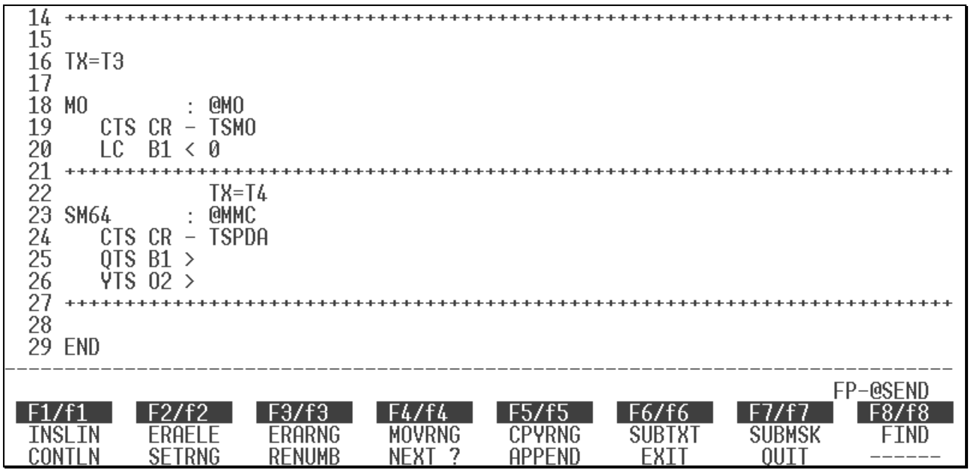


```
T4  TS = 64                   "4.Abtz.          ''          "
T5  TS = 256                  "5.Abtz.          ''          "
TY  TX = T4                   "System FP-Abtastzeit"
SSM 2C = 0                    "Laenge SAVE-Bereich, (n*1+2) kByte"
ISE 1C = N                    "Ausfallm. (RDYINT) ignorieren (Y/N) ?"
CCT 8R = 0                    "Sende-Telegrammnamen.Tx"
CCR 8R = 0                    "Empfangs-Telegrammnamen.Tx"
COP 8R = 0                    "Bedienungs-Telegrammnamen.Tx"
CMS 8N = 0                    "Meldesystemnamen"
CTS 8N = TSPDA,TSMO           "Transportsystemnamen"
MS  2M = 0                    "Meldesysteme"
X01 1M = 0                    "1. serielle Schnittstelle"
X02 1M = TSMO                 "2. serielle Schnittstelle"
X5C 8K <                      "Binaereing. 1, alarmfaehig"
X5D 8K <                      "Binaereing. 2, alarmfaehig"
X5A 8K >                      "Binaerausgaenge 1"
X5B 8K >                      "Binaerausgaenge 2"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----------------------------------------------------------------------------
                                                                    MP-MMCP16
```

Figure 3: Example: transport system name "TSPDA" at connector CTS

3.2.2 Configuring the initialize block (IB)

The initialize block for the ibaLink-SM-64-io of the @MMC type must be configured in the special FP @SEND. The transport system name must be stated at the CTS connector.



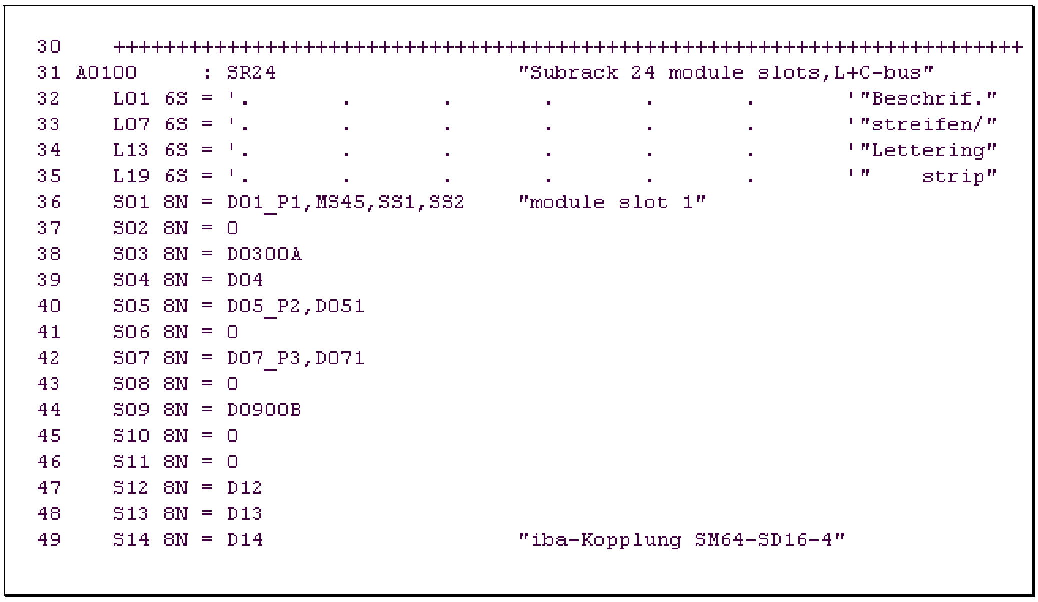
```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

TX=T3

MO          : @MO
   CTS CR - TSMO
   LC  B1 < 0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
              TX=T4
SM64        : @MMC
   CTS CR - TSPDA
   QTS B1 >
   YTS O2 >
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

END
```

Figure 4: Example: @MMC initialize block

3.3 Configuring the ibaLink-SM-64-SD16 board under STRUC V4

3.3.1 Configuring in the master program (MP)

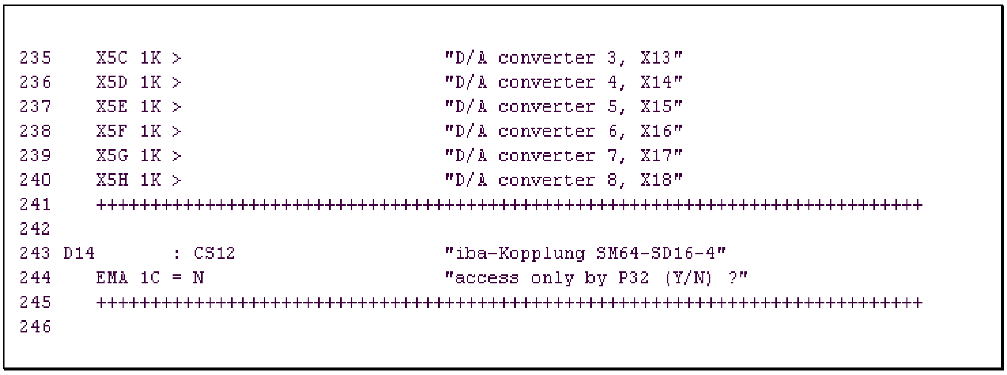
The ibaLink-SM-64 board is configured as type CS12 for one slot in an SD sub rack with L and C bus connection.



```
   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
A0100     : SR24                   "Subrack 24 module slots,L+C-bus"
   L01 6S = '.       .       .       .       .       .       '"Beschrif."
   L07 6S = '.       .       .       .       .       .       '"streifen/"
   L13 6S = '.       .       .       .       .       .       '"Lettering"
   L19 6S = '.       .       .       .       .       .       '"    strip"
   S01 8N = D01_P1,MS45,SS1,SS2    "module slot 1"
   S02 8N = 0
   S03 8N = D0300A
   S04 8N = D04
   S05 8N = D05_P2,D051
   S06 8N = 0
   S07 8N = D07_P3,D071
   S08 8N = 0
   S09 8N = D0900B
   S10 8N = 0
   S11 8N = 0
   S12 8N = D12
   S13 8N = D13
   S14 8N = D14                    "iba-Kopplung SM64-SD16-4"
```

Figure 5: Example: Board name "D14" in slot 14

"N" must be stated as a mandatory entry at the EMA connector.



```
   X5C 1K >                        "D/A converter 3, X13"
   X5D 1K >                        "D/A converter 4, X14"
   X5E 1K >                        "D/A converter 5, X15"
   X5F 1K >                        "D/A converter 6, X16"
   X5G 1K >                        "D/A converter 7, X17"
   X5H 1K >                        "D/A converter 8, X18"
   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

D14       : CS12                   "iba-Kopplung SM64-SD16-4"
   EMA 1C = N                      "access only by P32 (Y/N) ?"
   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

```

Figure 6: Example: SM64 board of the CS12 type

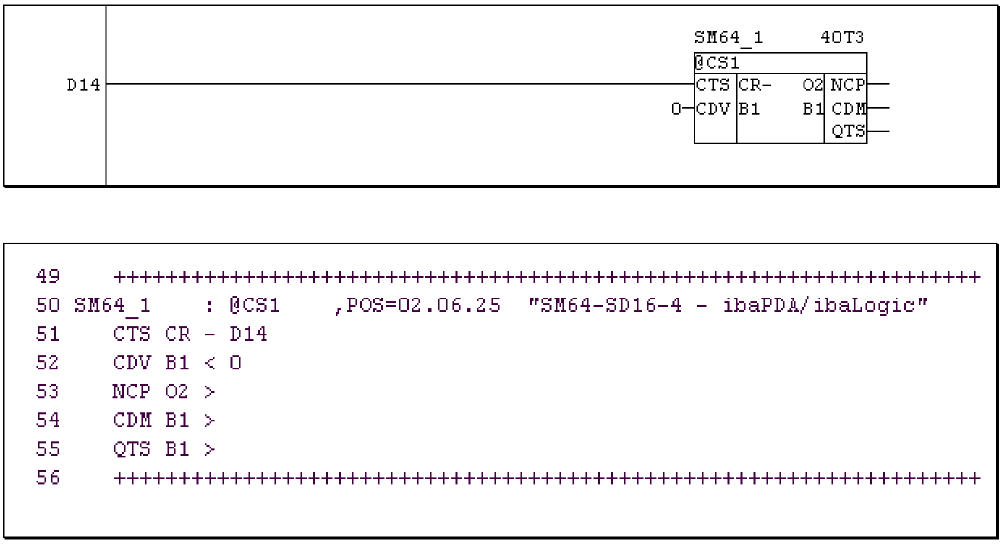
P32 note

The EMA connector must be set = N even in a pure P32 system.

This connector switches the address range of the CS12 from P32 (Y = 128 KB) to P16 (N = 32 KB). The ibaLink-SM-64-SD16 has a physical DPR capacity of just 32 KB.

3.3.2 Configuring the initialize block (IB)

The initialize block for the ibaLink-SM-64-SD16 of the @CS1 type must be configured in the special FP @SEND. The transport system name of the ibaLink-SM-64-SD16 must be stated at the CTS connector.



```
    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
SM64_1    : @CS1     ,POS=02.06.25  "SM64-SD16-4 - ibaPDA/ibaLogic"
    CTS CR - D14
    CDV B1 < 0
    NCP O2 >
    CDM B1 >
    QTS B1 >
    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 7: Example: @CS1 initialize block

3.4 Configuring a direct send channel

The direct send function block can be configured in all "standard" function packages.

Note

Direct communication is only possible for channels with N2 format. A module to which such a large number of values can be connected does not exist for the N4 format.

P32 note

Direct communication is not possible with P32.

The connectors must be connected as follows at the direct send function block (CTD501 type):

CTS with STRUC V3: transport system name as defined in the MP, for example:"TSPDA"
with STRUC V4: module name as defined in the MP, for example: "D14"

AT channel name "M0PDADAT" or "M1PDADAT" for module 0 and module 1, respectively

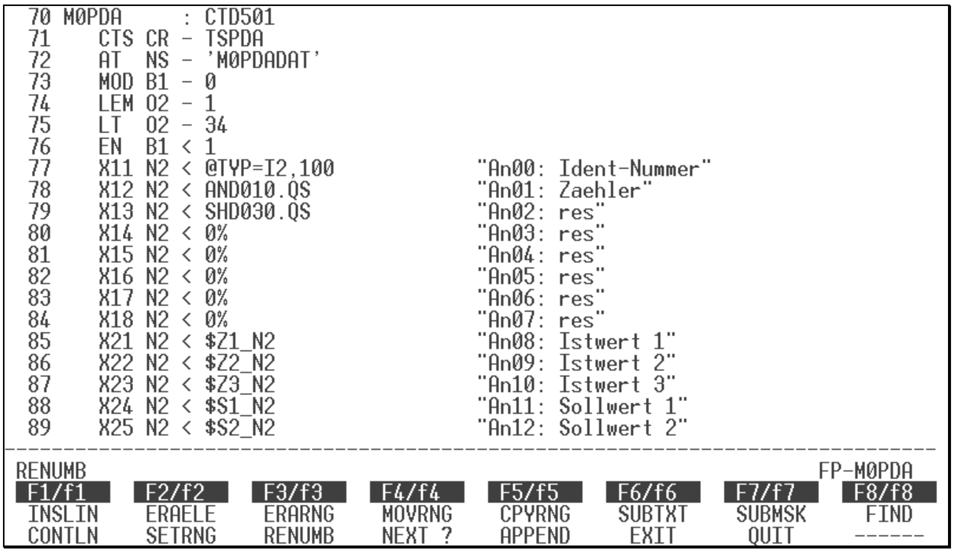
MOD with STRUC V3: channel mode = 0 (no overwrite, handshake)
with STRUC V4: channel mode = 1 (overwrite, refresh)

LEM error message limit = 1

LT telegram length = 34 (number of words)

X11...X48 analog values 0...31, user data in the N2 format

X51...X52 binary values 0...31, previously combined to 2 x 16 bits by the BSC block.



```
M0PDA      : CTD501
   CTS CR - TSPDA
   AT  NS - 'M0PDADAT'
   MOD B1 - 0
   LEM O2 - 1
   LT  O2 - 34
   EN  B1 < 1
   X11 N2 < @TYP=I2,100          "An00: Ident-Nummer"
   X12 N2 < AND010.QS            "An01: Zaehler"
   X13 N2 < SHD030.QS            "An02: res"
   X14 N2 < 0%                   "An03: res"
   X15 N2 < 0%                   "An04: res"
   X16 N2 < 0%                   "An05: res"
   X17 N2 < 0%                   "An06: res"
   X18 N2 < 0%                   "An07: res"
   X21 N2 < $Z1_N2               "An08: Istwert 1"
   X22 N2 < $Z2_N2               "An09: Istwert 2"
   X23 N2 < $Z3_N2               "An10: Istwert 3"
   X24 N2 < $S1_N2               "An11: Sollwert 1"
   X25 N2 < $S2_N2               "An12: Sollwert 2"
-------------------------------------------------------------------------------
RENUMB                                                                 FP-M0PDA
 F1/f1    F2/f2    F3/f3    F4/f4    F5/f5    F6/f6    F7/f7    F8/f8
 INSLIN   ERAELE   ERARNG   MOVRNG   CPYRNG   SUBTXT   SUBMSK    FIND
 CONTLN   SETRNG   RENUMB   NEXT ?   APPEND    EXIT     QUIT    ------
```

Figure 8: Example: send function block, direct (STRUC V3)

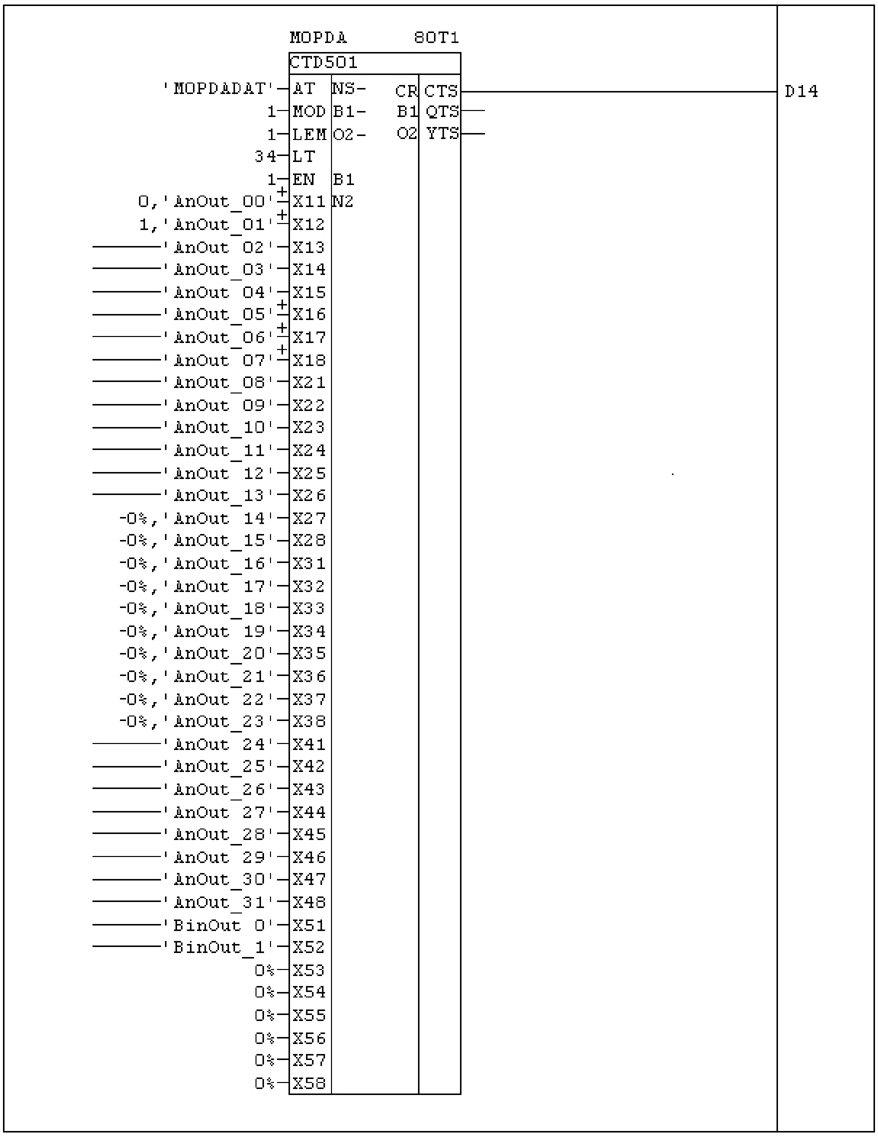
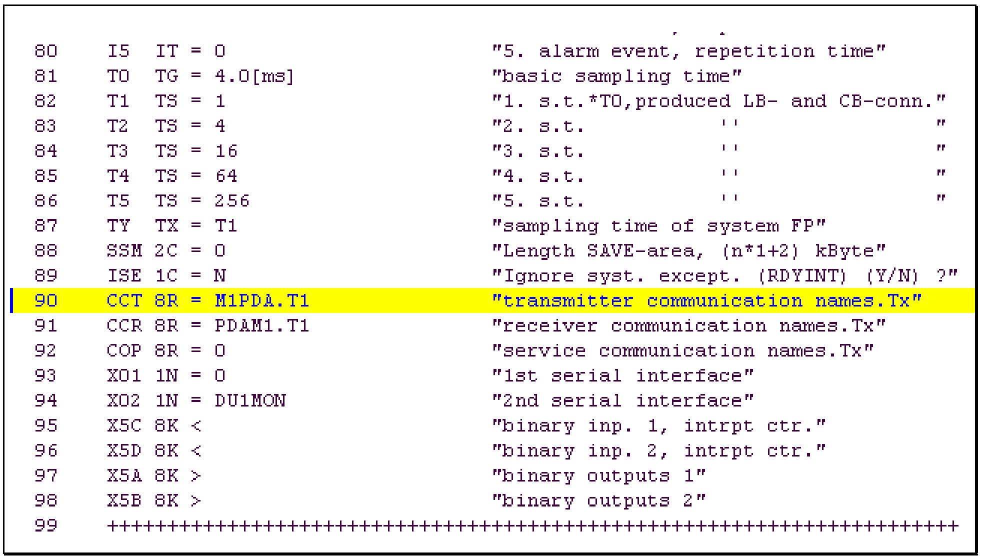


Figure 9: Example: send function block, direct (STRUC V4)

3.5 Configuring an indirect send channel

3.5.1 Configuring in the master program

In addition to the configuration of the board and of the transport system, the send telegram name and the time slice must be specified in the master program at the CCT connector of the processor module.



```
I5  IT = 0                    "5. alarm event, repetition time"
TO  TG = 4.0[ms]              "basic sampling time"
T1  TS = 1                    "1. s.t.*TO,produced LB- and CB-conn."
T2  TS = 4                    "2. s.t.          ''              "
T3  TS = 16                   "3. s.t.          ''              "
T4  TS = 64                   "4. s.t.          ''              "
T5  TS = 256                  "5. s.t.          ''              "
TY  TX = T1                   "sampling time of system FP"
SSM 2C = 0                    "Length SAVE-area, (n*1+2) kByte"
ISE 1C = N                    "Ignore syst. except. (RDYINT) (Y/N) ?"
CCT 8R = M1PDA.T1             "transmitter communication names.Tx"
CCR 8R = PDAM1.T1             "receiver communication names.Tx"
COP 8R = 0                    "service communication names.Tx"
X01 1N = 0                    "1st serial interface"
X02 1N = DU1MON               "2nd serial interface"
X5C 8K <                      "binary inp. 1, intrpt ctr."
X5D 8K <                      "binary inp. 2, intrpt ctr."
X5A 8K >                      "binary outputs 1"
X5B 8K >                      "binary outputs 2"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 10: Example: "M1PDA" send telegram and "T1" time slice

3.5.2 Configuring the indirect send function blocks

The indirect send function blocks do not handle the communication. Instead, they only store the data received in a telegram buffer.

The following must be noted when configuring the indirect send blocks in the function packages:

❑ The N2 and N4 data formats may not be mixed. All the values must be either in the N2 format or in the N4 format.

Important note

Exception:
The digital values are transmitted as two N2 variables.

❑ 3 types (1, 4 or 8 values) exist for each format. These types can be used in any manner (even in different function packages). The designer is, however, responsible for the assignment of addresses within the telegrams. The system is unable to check gaps and/or overlaps.

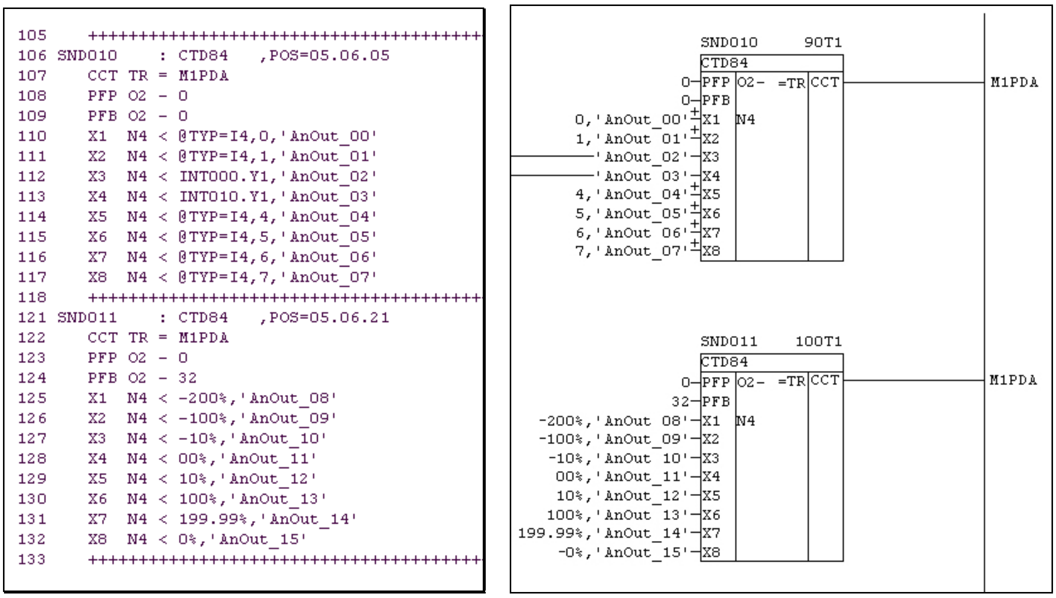
❑ The telegram format must be adhered to, no matter whether all the values are needed or not.

❑ The telegram length must be 68 bytes for N2 telegrams and 132 bytes for N4 telegrams.

Example: N4 send telegram

❑ 1st SND010 send block of the CTD84 type (send 8 x N4) with connectors

CCT Telegram names as defined in the master program
PFP, PFB Offset of the data in the telegram
X1 – X8 Analog values 0-7 in the N4 format



```
    ++++++++++++++++++++++++++++++++++++++++
SND010     : CTD84    ,POS=05.06.05
    CCT TR = M1PDA
    PFP O2 - 0
    PFB O2 - 0
    X1  N4 < @TYP=I4,0,'AnOut_00'
    X2  N4 < @TYP=I4,1,'AnOut_01'
    X3  N4 < INT000.Y1,'AnOut_02'
    X4  N4 < INT010.Y1,'AnOut_03'
    X5  N4 < @TYP=I4,4,'AnOut_04'
    X6  N4 < @TYP=I4,5,'AnOut_05'
    X7  N4 < @TYP=I4,6,'AnOut_06'
    X8  N4 < @TYP=I4,7,'AnOut_07'
    ++++++++++++++++++++++++++++++++++++++++
SND011     : CTD84    ,POS=05.06.21
    CCT TR = M1PDA
    PFP O2 - 0
    PFB O2 - 32
    X1  N4 < -200%,'AnOut_08'
    X2  N4 < -100%,'AnOut_09'
    X3  N4 < -10%,'AnOut_10'
    X4  N4 < 00%,'AnOut_11'
    X5  N4 < 10%,'AnOut_12'
    X6  N4 < 100%,'AnOut_13'
    X7  N4 < 199.99%,'AnOut_14'
    X8  N4 < 0%,'AnOut_15'
    ++++++++++++++++++++++++++++++++++++++++
```

Figure 11: Example: send blocks, indirect

❑ 2nd send block to 4th send block (SND011 to SND013) of the CTD84 type, connectors same as with SND010, the offset only is passed on:

SND011.PFB 32
SND011.X1-X8 analog values 8-15

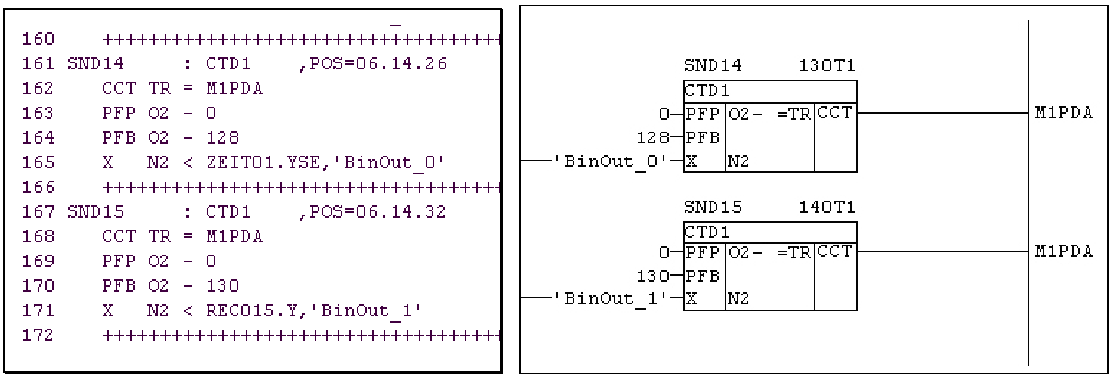
SND012.PFB 64
SND012.X1-X8 analog values 16-23

SND013.PFB 96
SND013.X1-X8 analog values 24-31

❑ 5th send block and 6th send block (SND015 to SND016) of the CT1 type (send 1 x N2), connectors same as with SND010, the offset only is passed on:

SND015.PFB 128
SND015.X an N2 value which represents bits 0-15.

SND016.PFB 130
SND016.X an N2 value which represents bits 16-31.



```
    ++++++++++++++++++++++++++++++++++++
SND14      : CTD1    ,POS=06.14.26
    CCT TR = M1PDA
    PFP O2 - 0
    PFB O2 - 128
    X   N2 < ZEIT01.YSE,'BinOut_0'
    ++++++++++++++++++++++++++++++++++++
SND15      : CTD1    ,POS=06.14.32
    CCT TR = M1PDA
    PFP O2 - 0
    PFB O2 - 130
    X   N2 < REC015.Y,'BinOut_1'
    ++++++++++++++++++++++++++++++++++++
```

Figure 12: Example: send blocks, binary values, indirect

3.5.3 Configuring the telegram block

The telegram block (@CTT type) must be placed in the special @SEND function package. This block sends the telegram buffer, which was filled with data by the indirect send blocks, to the specified interface. For this purpose, the data relevant for communication must be stated at the connectors:

CCT Telegram name corresponding to the specification in the master program and at the send blocks, for example, "M1PDA"

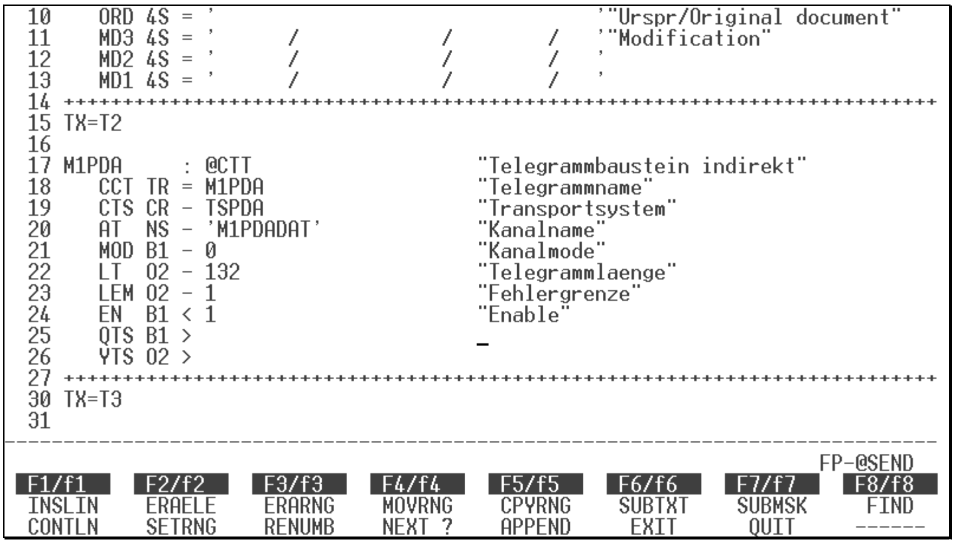
CTS With STRUC V3: transport system name as defined in the MP, for example "TSPDA
With STRUC V4: board name as defined in the MP, for example: "D14"

AT Channel name M0PDADAT or M1PDADAT for send module 0 or module 1, respectively

MOD With STRUC V3: channel mode = 0 (handshake, no overwrite)
With STRUC V4: channel mode = 1 (refresh, overwrite)

LT Telegram length: 68 bytes with N2 telegram, 132 bytes with N4 telegram

LEM Error message limit



```
   ORD 4S = '                               '"Urspr/Original document"
   MD3 4S = '      /            /        /   '"Modification"
   MD2 4S = '      /            /        /   '
   MD1 4S = '      /            /        /   '
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TX=T2

M1PDA      : @CTT                    "Telegrammbaustein indirekt"
   CCT TR = M1PDA                    "Telegrammname"
   CTS CR - TSPDA                    "Transportsystem"
   AT  NS - 'M1PDADAT'               "Kanalname"
   MOD B1 - 0                        "Kanalmode"
   LT  O2 - 132                      "Telegrammlaenge"
   LEM O2 - 1                        "Fehlergrenze"
   EN  B1 < 1                        "Enable"
   QTS B1 >
   YTS O2 >
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TX=T3
```

Figure 13: Example: telegram block: send, indirect (STRUC V3)

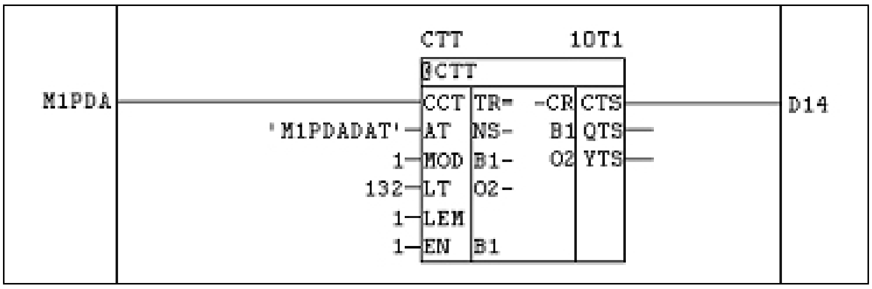


Figure 14: Example: telegram block: send, indirect (STRUC V4)

3.6 Configuring a direct receive channel

The direct receive function block can be configured in all "standard" function packages.

Tip

Direct communication is only possible for channels with N2 format. A module to which such a large number of values can be connected does not exist for the N4 format.

The connectors must be connected as follows at the direct receive block (CRD501 type):

CTS With STRUC V3: transport system name as defined in the MP, for example: "TSPDA"
With STRUC V4: board name as defined in the MP, for example: "D14"

AR Channel name "PDAM0DAT" or "PDAM1DAT" for module 0 and module 1, respectively

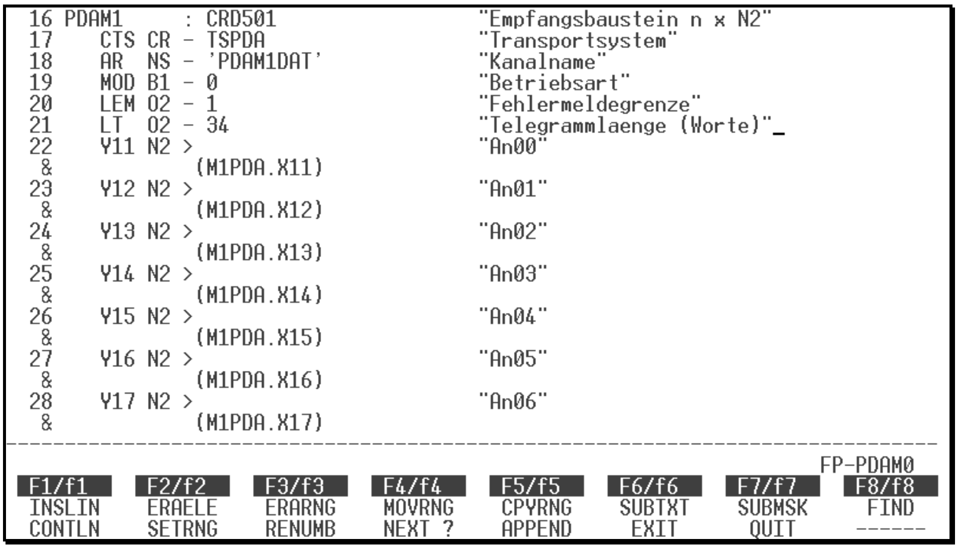
MOD With STRUC V3: channel mode = 0 (no overwrite, handshake)
with STRUC V4: channel mode = 1 (overwrite, refresh)

LEM Error message limit = 1

LT Telegram length = 34 (words

Y11...Y48 Analog values 0...31, user data in the N2 format

Y51...Y52 Binary values 0...31, combined to 2 x 16 Bits



```
 PDAM1      : CRD501                "Empfangsbaustein n x N2"
   CTS CR - TSPDA                   "Transportsystem"
   AR  NS - 'PDAM1DAT'              "Kanalname"
   MOD B1 - 0                       "Betriebsart"
   LEM 02 - 1                       "Fehlermeldegrenze"
   LT  02 - 34                      "Telegrammlaenge (Worte)"_
   Y11 N2 >                         "An00"
&          (M1PDA.X11)
   Y12 N2 >                         "An01"
&          (M1PDA.X12)
   Y13 N2 >                         "An02"
&          (M1PDA.X13)
   Y14 N2 >                         "An03"
&          (M1PDA.X14)
   Y15 N2 >                         "An04"
&          (M1PDA.X15)
   Y16 N2 >                         "An05"
&          (M1PDA.X16)
   Y17 N2 >                         "An06"
&          (M1PDA.X17)
-------------------------------------------------------------------------------
                                                                      FP-PDAM0
 F1/f1    F2/f2    F3/f3    F4/f4    F5/f5    F6/f6    F7/f7    F8/f8
 INSLIN   ERAELE   ERARNG   MOVRNG   CPYRNG   SUBTXT   SUBMSK   FIND
 CONTLN   SETRNG   RENUMB   NEXT ?   APPEND   EXIT     QUIT     ------
```

Figure 15: Example: receive function block, direct (STRUC V3

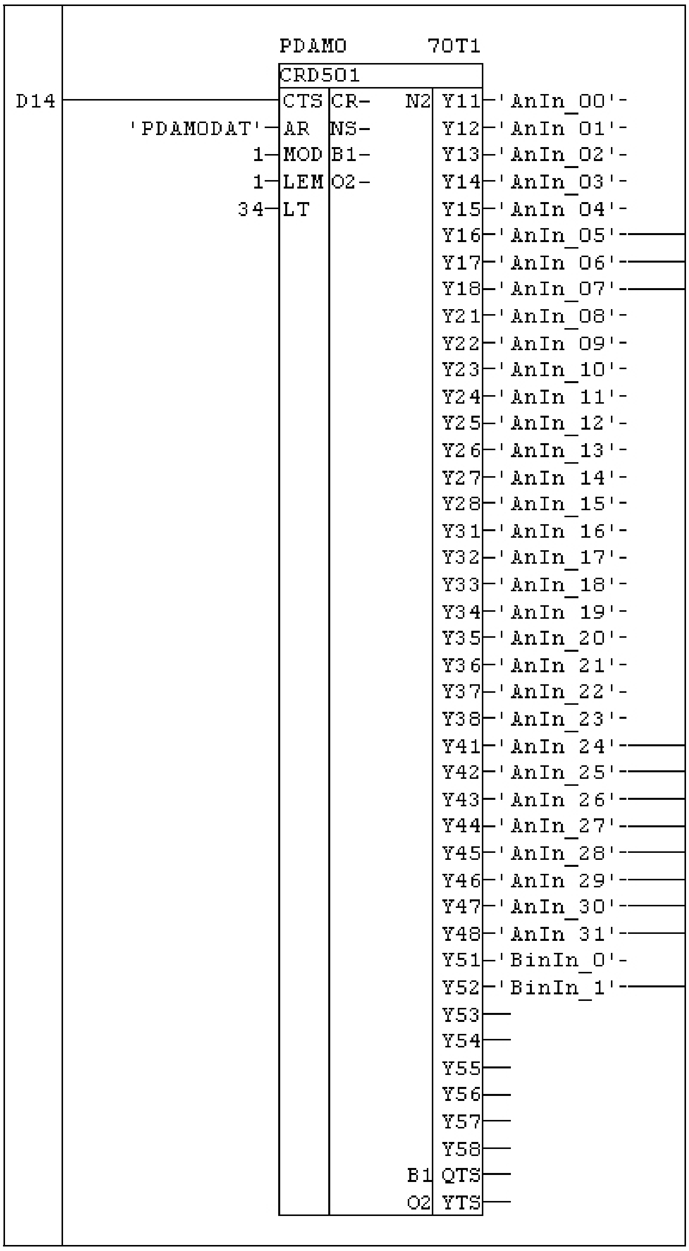
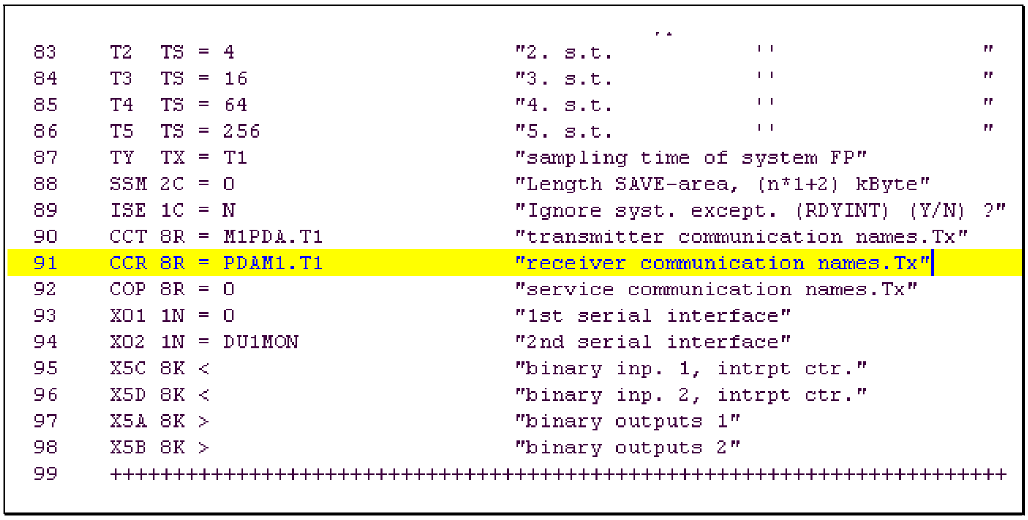


Figure 16: Example: receive function block, direct (STRUC V4)

3.7 Configuring an indirect receive channel

3.7.1 Configuring in the master program

In addition to the configuration of the board and of the transport system, the receive telegram name and the time slice must be specified in the master program at the CCR connector of the processor module.



```
T2  TS = 4                         "2. s.t.            ''              "
T3  TS = 16                        "3. s.t.            ''              "
T4  TS = 64                        "4. s.t.            ''              "
T5  TS = 256                       "5. s.t.            ''              "
TY  TX = T1                        "sampling time of system FP"
SSM 2C = 0                         "Length SAVE-area, (n*1+2) kByte"
ISE 1C = N                         "Ignore syst. except. (RDYINT) (Y/N) ?"
CCT 8R = M1PDA.T1                  "transmitter communication names.Tx"
CCR 8R = PDAM1.T1                  "receiver communication names.Tx"
COP 8R = 0                         "service communication names.Tx"
X01 1N = 0                         "1st serial interface"
X02 1N = DU1MON                    "2nd serial interface"
X5C 8K <                           "binary inp. 1, intrpt ctr."
X5D 8K <                           "binary inp. 2, intrpt ctr."
X5A 8K >                           "binary outputs 1"
X5B 8K >                           "binary outputs 2"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 17: Example: "PDAM1" receive telegram and "T1" time slice

3.7.2 Configuring the indirect receive function blocks

The indirect receive blocks do not handle the communication. Instead, they only retrieve the data received from the telegram buffer.

The following must be noted when configuring the indirect receive blocks in the function packages:

❑ The N2 and N4 data formats may not be mixed. All the values must be either in the N2 format or in the N4 format.

Important note

Exception:

The digital values are transmitted as two N2 variables.

❑ 3 types (1, 4 or 8 values) exist for each format. These types can be used in any manner (even in different function packages). The designer is, however, responsible for the assignment of addresses within the telegrams. The system is unable to check gaps and/or overlaps.

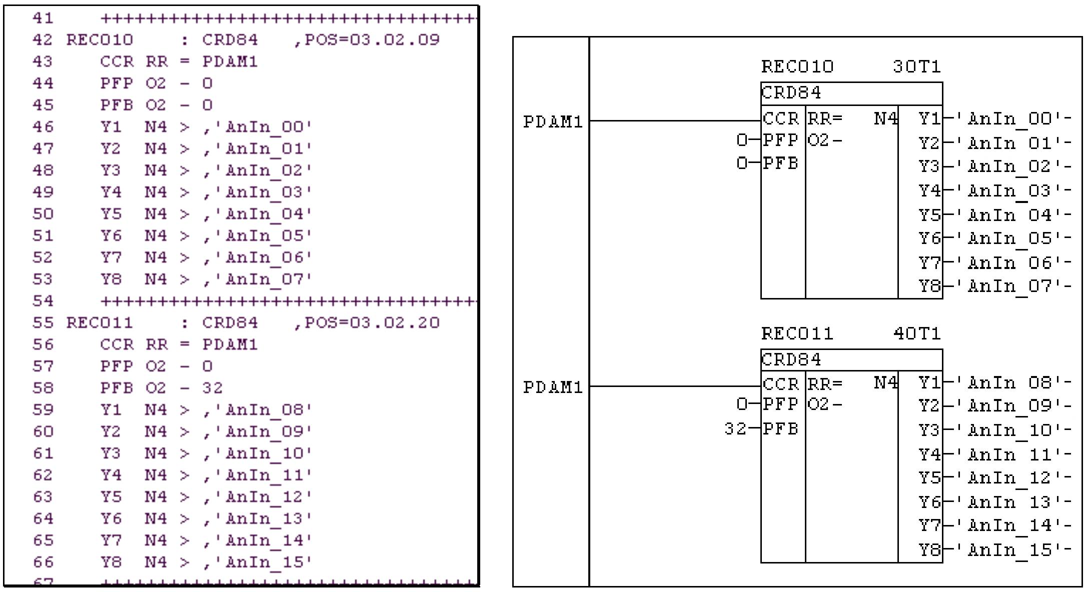
❑ The telegram format must be adhered to, no matter whether all the values are needed or not.

❑ The telegram length must be 68 bytes for N2 telegrams and 132 bytes for N4 telegrams.

Example: N4 receive telegram

❑ 1st REC010 receive block of the CRD84 type (receive 8 x N4) with connectors CCR Telegram names as defined in the master program

PFP, PFB	Offset of the data in the telegram
Y1 – Y8	analog values 0-7 in the N4 format



```
    ++++++++++++++++++++++++++++++++++
REC010     : CRD84   ,POS=03.02.09
    CCR RR = PDAM1
    PFP O2 - 0
    PFB O2 - 0
    Y1  N4 > ,'AnIn_00'
    Y2  N4 > ,'AnIn_01'
    Y3  N4 > ,'AnIn_02'
    Y4  N4 > ,'AnIn_03'
    Y5  N4 > ,'AnIn_04'
    Y6  N4 > ,'AnIn_05'
    Y7  N4 > ,'AnIn_06'
    Y8  N4 > ,'AnIn_07'
    ++++++++++++++++++++++++++++++++++
REC011     : CRD84   ,POS=03.02.20
    CCR RR = PDAM1
    PFP O2 - 0
    PFB O2 - 32
    Y1  N4 > ,'AnIn_08'
    Y2  N4 > ,'AnIn_09'
    Y3  N4 > ,'AnIn_10'
    Y4  N4 > ,'AnIn_11'
    Y5  N4 > ,'AnIn_12'
    Y6  N4 > ,'AnIn_13'
    Y7  N4 > ,'AnIn_14'
    Y8  N4 > ,'AnIn_15'
```

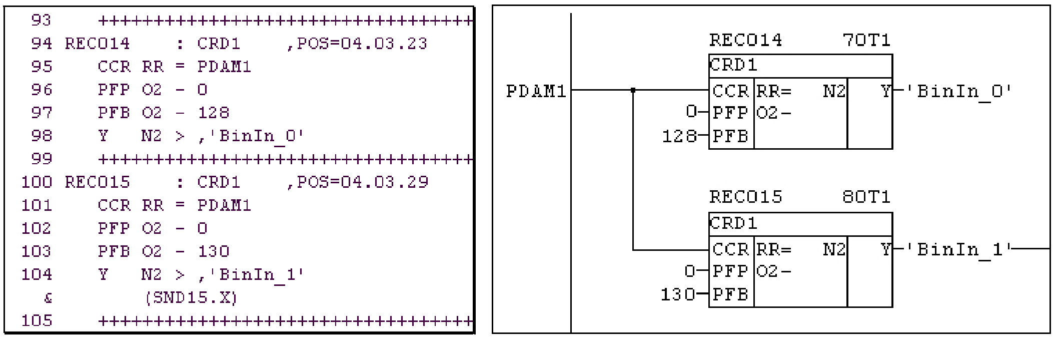
Figure 18: Example: receive function block, indirect

❑ 2nd receive block to 4th receive block (REC011 to REC013) of the CRD84 type, connectors same as with REC010, the offset only is passed on:

REC011.PFB 32	
REC011.Y1-Y8	analog values 8-15
REC 012.PFB 64	
REC 012.Y1-Y8	analog values 16-23
REC 013.PFB 96	
REC 013.Y1-Y8	analog values 24-31

❑ 5th receive block and 6th receive block (REC014 to REC015) of the CRD1 type (receive 1 x N2), connectors same as with REC010, the offset only is passed on:

REC014.PFB 128	
REC014.Y	an N2 value which represents bits 0-15
REC015.PFB 130	
REC015.Y	an N2 value which represents bits 16-31.



```
   +++++++++++++++++++++++++++++++++++
RECO14     : CRD1    ,POS=04.03.23
   CCR RR = PDAM1
   PFP O2 - 0
   PFB O2 - 128
   Y   N2 > ,'BinIn_0'
   +++++++++++++++++++++++++++++++++++
RECO15     : CRD1    ,POS=04.03.29
   CCR RR = PDAM1
   PFP O2 - 0
   PFB O2 - 130
   Y   N2 > ,'BinIn_1'
 &        (SND15.X)
   +++++++++++++++++++++++++++++++++++
```

Figure 19: Example: receive function blocks, binary values, indirect

3.7.3 Configuring the telegram block

The telegram block (@CRT type) must be placed in the special @RECEIVE function package. This block receives the data from the specified interface and stores them in the telegram buffer. For this purpose, the data relevant for communication must be stated at the connectors:

CCR Telegram name corresponding to the specification in the master program and at the send function blocks, for example, "PDAM1"

CTS With STRUC V3: transport system name as defined in the MP, for example: "TSPDA"

With STRUC V4: board name as defined in the MP, for example: "D14"

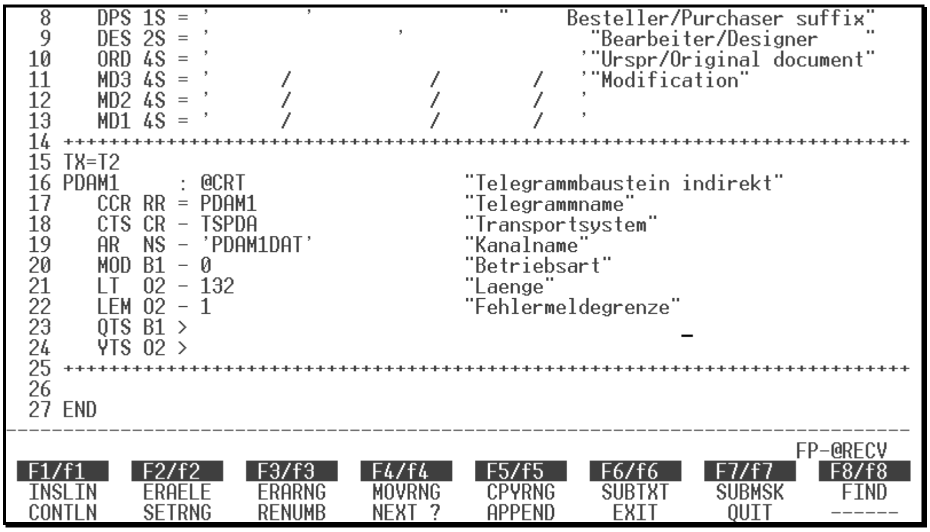
AT Channel name PDAM0DAT or PDAM1DAT for send module 0 or module 1, respectively

MOD With STRUC V3: channel mode = 0 (handshake, no overwrite)

With STRUC V4: channel mode = 1 (refresh, overwrite)

LT Telegram length in bytes, = 68 bytes with N2 telegram, = 132 bytes with N4 telegram

LEM Error message limit



```
   DPS 1S = '         '           "    Besteller/Purchaser suffix"
   DES 2S = '                '           "Bearbeiter/Designer    "
   ORD 4S = '                           '"Urspr/Original document"
   MD3 4S = '     /          /      /   '"Modification"
   MD2 4S = '     /          /      /   '
   MD1 4S = '     /          /      /   '
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TX=T2
PDAM1      : @CRT                 "Telegrammbaustein indirekt"
   CCR RR = PDAM1                 "Telegrammname"
   CTS CR - TSPDA                 "Transportsystem"
   AR  NS - 'PDAM1DAT'            "Kanalname"
   MOD B1 - 0                     "Betriebsart"
   LT  O2 - 132                   "Laenge"
   LEM O2 - 1                     "Fehlermeldegrenze"
   QTS B1 >
   YTS O2 >
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

END
```

Figure 20: Example: telegram block: receive, indirect (STRUC V3)

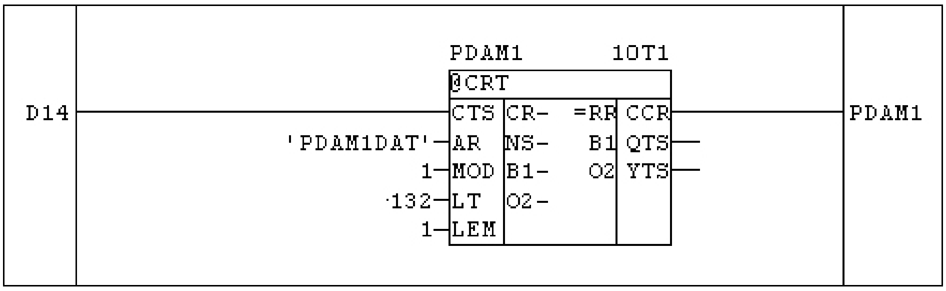


Figure 21: Example: telegram block: receive, indirect (STRUC V4)

4 Operating modes

4.1 “Mode” rotary switch

The front of the board is fitted with a rotary switch with the following positions:

- ❑ CS3 mode: Driver for STRUC V3.
  An MMC CS3 communication board is emulated.
- ❑ CS12 mode: Driver for STRUC V4.
  A CS12 fiber-optic interface module board is emulated.
- ❑ Real mode: Data is transmitted in IEEE float format.
  During conversion, the precision of the 32-bit analog values (N4, I4 formats) is reduced to 24 bits.
  This mode must be set as the default mode for the ibaPDA and ibaLogic connection. This is where the lowest loss in precision occurs if data is transmitted in N4 format.
- ❑ Integer mode: Data is transmitted as 16-bit integers.
  Here, the precision of the 32-bit analog values (N4, I4 formats) is reduced to 16 bits.
  This mode must be set as the default mode for connecting ibaScope und iba periphery devices.

Position	Mode	Notes
0	CS3-Real Mode	CS3 driver for STRUC V3, IEEE float format
1	CS3-Integer Mode	CS3 driver for STRUC V3, 16-bit integer format
2	CS3	Free for future expansion
3	CS3	Free for future expansion
4	CS12-Real Mode	CS12 driver for STRUC V4, IEEE float format
5	CS12-Integer Mode	CS12 driver for STRUC V4, 16-bit integer format
6	CS12	Free for future expansion
7	CS12	Free for future expansion
8		Free for future expansion
9	Test-Mode	Irrelevant of the SIMADYN D configuration, a test sample is sent

Table 1: Positions for rotary switch

Important note

The rotary switch should not be moved when the board is running, because if the driver is switched this can result in a disabling of the dual-port RAM and this may lead to an acknowledgement delay during backplane bus access.

4.2 LED indicators for operating states

The 3 LEDs represent the following states:

LED	State	Meaning
Green: Run	On/off	Processor is not running, board is defective or no power
	Flashing	Board has power, processor is running
Yellow: Link	Off	No telegrams active, no send or receive channels have been configured for this board at the SD end!
	Flashing	At least one send or receive channel has been configured for this board, however, no telegrams are received, i.e. it is running in unidirectional mode only (e.g. with ibaPDA) or the connector is not inserted. The send direction is OK if - at the opposite end - the yellow LED on the ibaFOB-4i or ibaFOB-io is permanently lit.
	On	At least one send or receive channel is configured and the receive direction is ok, i.e. bidirectional mode (e.g. *iba-Logic* ←→ SIMADYN D) or unidirectional mode from the iba periphery.
Red: Error	Off	No error, the LED is automatically reset when the error is gone!
	On	Error has occurred within the device

Table 2: LED indicators

5 Settings in ibaPDA-V6

5.1 Introduction

Generally, ibaPDA-V6 is a passive system like ibaPDA V5 before. But ibaPDA-V6 offers the possibility to configure a limited number of alarm output signals which can be output via the fiber optical link of an ibaFOB-io or ibaFOB-4o card and transmitted to the input of an ibaLink-SM-64-SD16 board.

If you don’t want to use the alarms just a simplex FO cable is enough for connecting the systems. If you want to use the alarms too, a second or duplex FO cable is required.

➚ For more information please refer to the ibaPDA-V6 manual or online help.

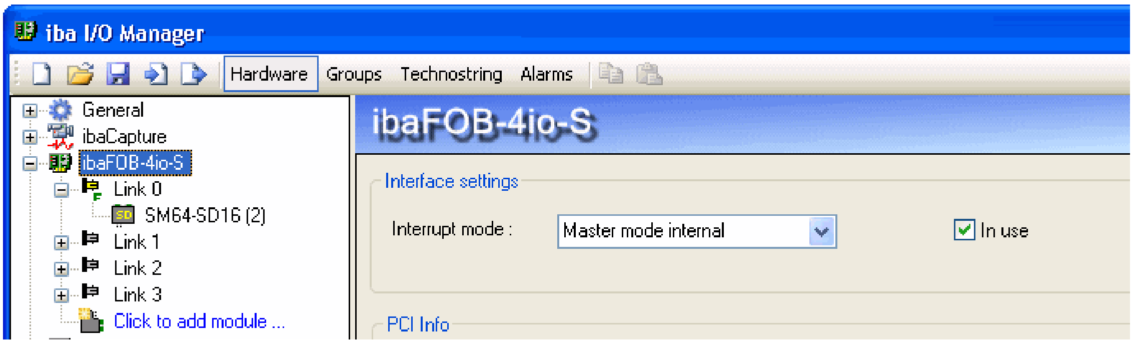
If in one-directional operation the yellow LED on the ibaLink-SM-64-SD16 board should be blinking (provided the SD engineering is correct) and the yellow LED on the ibaFOB-4i card should be on.

The following settings have to be made in ibaPDA-V6:

- Setting up the ibaFOB card (IO-manager, data interface, card level)
- Adding a SM64-SD16 module (IO-manager, data interface, link level)
- Activating and setting up the module (IO-manager, module “General” tab)
- Activating and setting up the signals (IO-manager, module “signal” tables)

5.2 Card Settings of ibaFOB-4i-S in the IO-manager

Generally, the card is detected by the system automatically. If not done yet, only the interrupt mode should be set to “Master/internal” or “Slave”. The mode “Master/external” is not permitted.



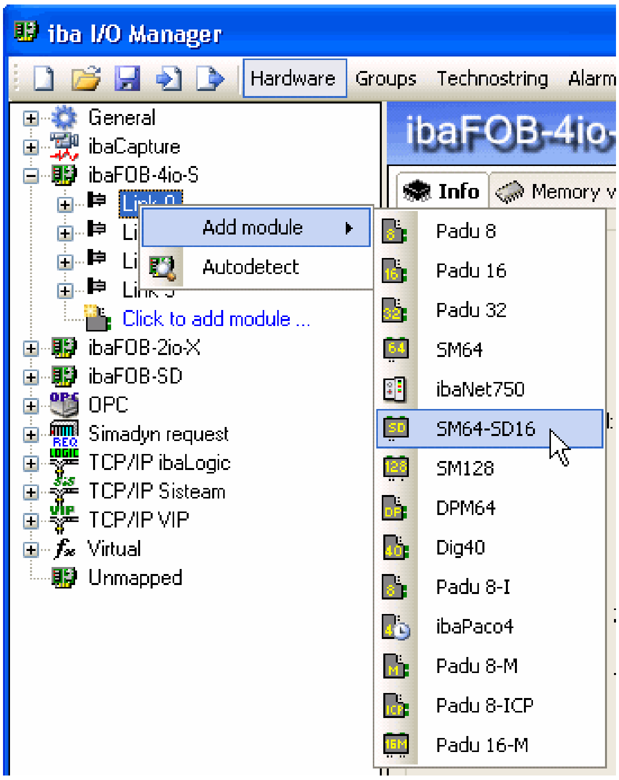
Interrupt mode : Master mode internal

☑ In use

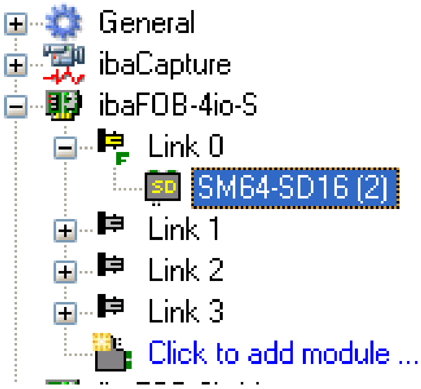
Figure 22: Interrupt setting ibaPDA-V6

5.3 Adding a SM64-SD16 module

If an active and correctly configured connection has been established to an ibaLink-SM-64-SD16 board then an appropriate module is automatically displayed at the corresponding link of the ibaFOB-4i card. If not, make a right mouse click on the link in the data interface tree which is connected to the ibaLink-SM64-SD16 board and choose “Add module” in the context menu and then select “SM64-SD16” from the module list.



After, the module is shown in the data interface tree.

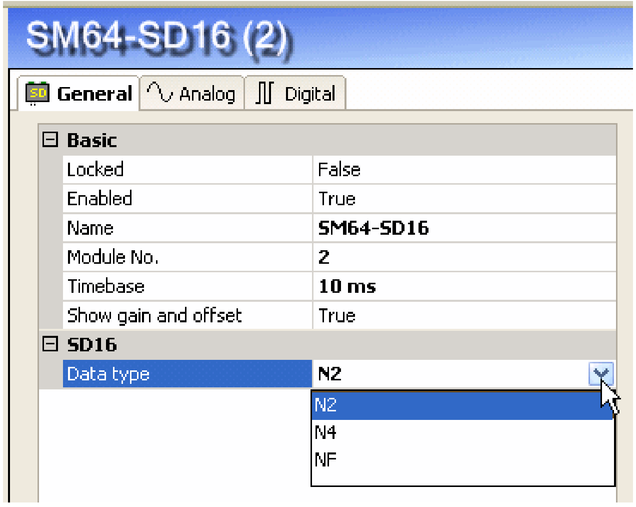


5.4 Module Settings

Mark the SM64-SD16 module in the tree and perform the basic module setup in the “General” tab.

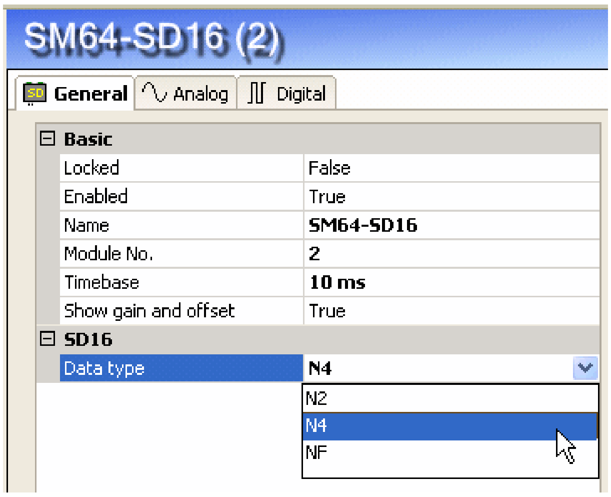
➚ See ibaPDA-V6 manual. Have a special look at the data type setting.

Setting N2: default values for gain and offset are set (± 200 %)
ibaPDA uses N2 percentage values by default for the scaling of the input values, i. e. gain = 0.00610352 and offset = 0.
This corresponds to y = (200 / 2^{15}) * x; the 16-bit integer range is scaled to ±200 %.



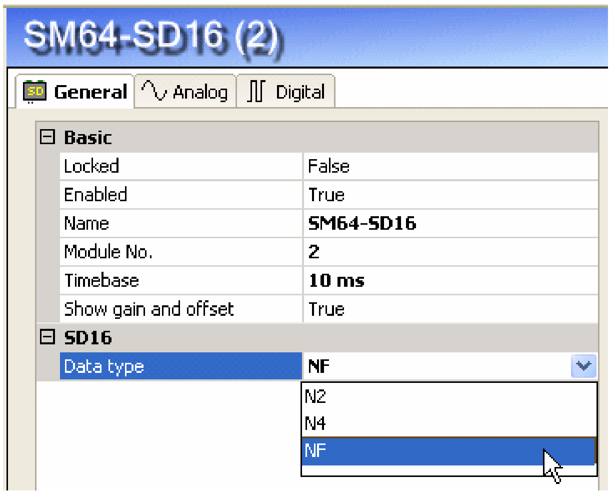
Basic	
Locked	False
Enabled	True
Name	SM64-SD16
Module No.	2
Timebase	10 ms
Show gain and offset	True
SD16	
Data type	N2

Setting N4: default values for gain and offset are set (±200 %)
ibaPDA uses N4 percentage values by default for the scaling of the input values, i. e. gain = 9.313226e-08 and offset = 0.
This corresponds to y = (200 / 2^{31}) * x; the 32-bit integer range is scaled to ±200 %.



Basic	
Locked	False
Enabled	True
Name	SM64-SD16
Module No.	2
Timebase	10 ms
Show gain and offset	True
SD16	
Data type	N4

Setting NF: default values for gain and offset are set (1)
ibaPDA does not scale the input values by default, i.e. gain = 1 and offset = 0.
The 32-bit IEEE float values are used without conversion.



SM64-SD16 (2)

General | Analog | Digital

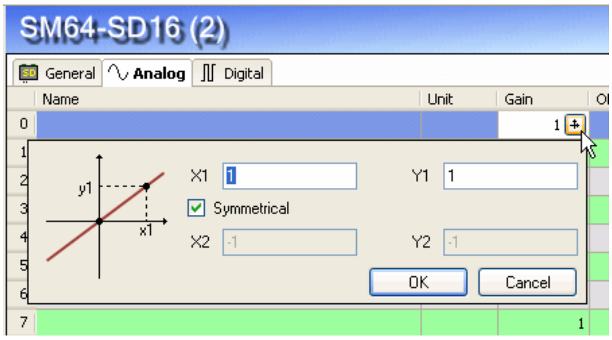
⊟ **Basic**	
Locked	False
Enabled	True
Name	**SM64-SD16**
Module No.	**2**
Timebase	**10 ms**
Show gain and offset	True
⊟ **SD16**	
Data type	**NF**

N2
N4
NF

5.5 Configuring the Signals

The signals to be measured can now be entered and activated in the signal tables (*Analog* and "Digital" tabs).

The scaling of the signals can be changed manually, if required, in order to convert the values to physical units. To do so, just click in the column "Gain" (in the row of the signal in question) and click again on the little browser button in order to open the two-point-scaling dialog.



X1 1   Y1 1
☑ Symmetrical
X2 -1   Y2 -1

Figure 23: Configuring the Signals

6 Settings in ibaLogic-V3

The ibaLogic-V3 connection requires bidirectional operation. This means that a duplex fiber-optic cable is required. Given correct connection of the cable, both the yellow LED on the ibaLink-SM-64 SD16 and the yellow LED on the ibaFOB-io counterpart board must be permanently lit (on condition that the input and output channels are correctly configured at the SD end).

The following settings are necessary in ibaLogic-V3:

- ❑ General settings ("System setup" menu; refer to the ibaLogic manual)
- ❑ FOB-io PCI settings ("PCI configuration" menu/FOB IO PCI Link settings)
- ❑ Selection of input and output resources
- ❑ Scaling of inputs and outputs

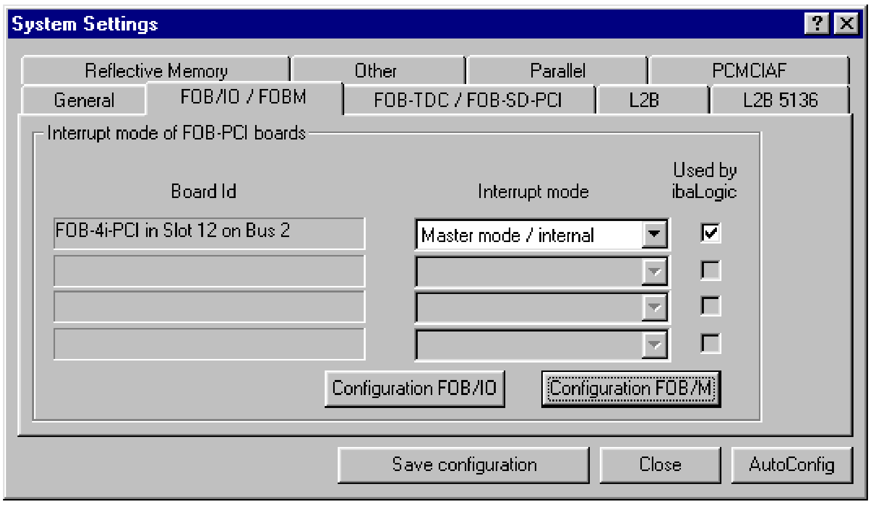
6.1 System setup

The system setup is explained in the ibaLogic-V3 manual.

Note

The FOBio, FOB-F, FOB-M and FOB4i boards are addressed by ibaLogic using the same drive; their interfaces are identical at the ibaLogic level. The different names in the masks have historical reasons, but they always refer to the same board.

Menu "File System setup"



Interrupt mode of FOB-PCI boards

Board Id	Interrupt mode	Used by ibaLogic
FOB-4i-PCI in Slot 12 on Bus 2	Master mode / internal	☑
		□
		□
		□

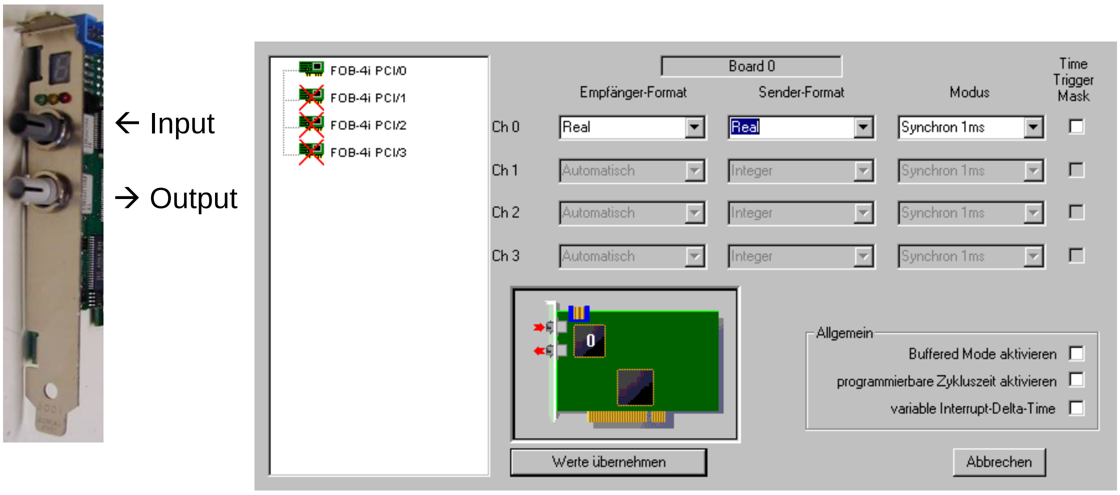
Figure 24: ibaLogic system setup

The communication partner of the ibaLink-SM-64 SD16 at the PDA end is the FOB-io PCI board. The board is configured automatically. The user only has to set the "Master internal" or "Slave" interrupt mode. The "Master external" mode is not permitted.

FOBio setup

The mode corresponding to the telegram format set on the ibaLink-SM-64-SD16 must be selected. Typically, the real mode should be set.

Menu "File-PCI configuration-FOBio PCI link setup"



	Empfänger-Format	Sender-Format	Modus	Time Trigger Mask
Ch 0	Real	Real	Synchron 1ms	□
Ch 1	Automatisch	Integer	Synchron 1ms	□
Ch 2	Automatisch	Integer	Synchron 1ms	□
Ch 3	Automatisch	Integer	Synchron 1ms	□

Allgemein
Buffered Mode aktivieren □
programmierbare Zykluszeit aktivieren □
variable Interrupt-Delta-Time □

Figure 25: FOB IO setup

6.2 Selection of input and output resources

The 2 SIMADYN D send channels (telegrams M0PADAT and M1PDADAT) are defined as 2 consecutive FOB IO modules. Irrespective of the connector to which the ibaLink-SM-64 SD16 is connected, both modules must be selected in ibaLogic-V3.

Example:

If SIMADYN D is connected to the first connector (from above) of the FOBio PCI, the SIMADYN D connection must be assigned to the 1st and 2nd FOBIO module in the input resource.

Input		
SD channel	ibaLogic analog values	ibaLogic digital values
M0PDADAT	FOB-F/FOB-IO → analog(real) → FOB-F M0 Ana.	FOB-F/FOB-IO → digital → FOB-F M0 dig.
M1PDADAT	FOB-F/FOB-IO → analog(real) → FOB-F M1 ana.	FOB-F/FOB-IO → digital → FOB-F M1 dig.

Table 3: Example for selection

The same procedure is applicable to the outputs (SIMADYN D inputs).

1st connector (from top)	→ 1st module and 2nd module
2nd connector (from top)	→ 3rd module and 4th module, etc.

Note

It goes without saying that, if the data is transmitted in integer mode, then the analog (integer) modules must be selected.

6.3 Scaling of inputs and outputs

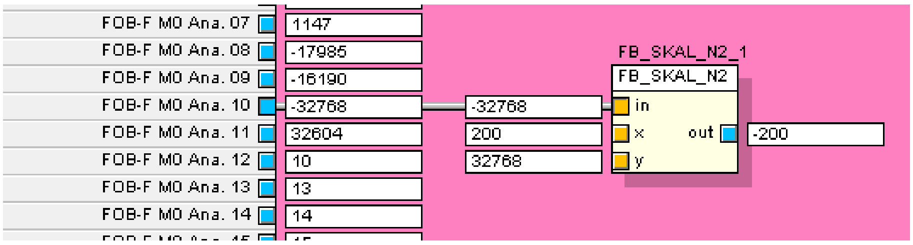
The signals are received as raw values as described in chapter 3.1.2.1.

The users themselves are responsible for scaling, using the variety of options made available by ibaLogic.

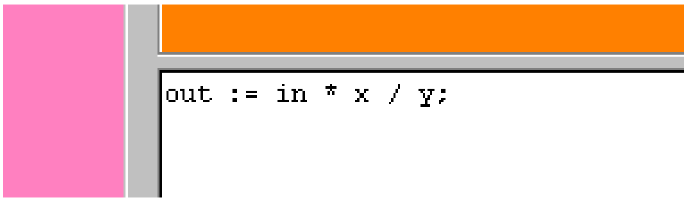
Scaling of N2 values to ±200 %: y = (200/32768) * x;

Scaling of N4 values to ±200 %: y = (200/2147483648) * x;

It is, for example, possible to generate a scaling module that scales the "raw values" to the N2 percentage range.



The module only performs a multiplication and a division.



```
out := in * x / y;
```

7 Settings in ibaScope

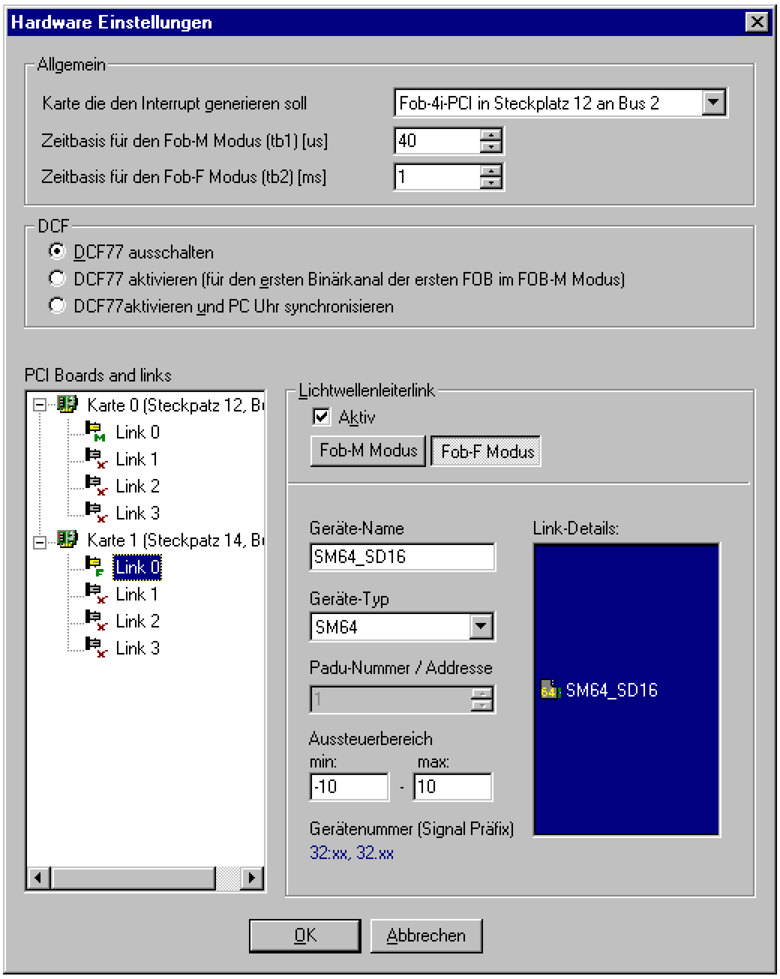
The ibaScope connection requires only one direction. This means that a simplex fiber-optic cable is required. In this unidirectional mode, the yellow LED on the ibaLink-SM-64 SD16 flashes-on condition that the SD configuration is current-and the yellow LED on the iba-FOB-4i is permanently lit.

The ibaLink-SM-64 SD16 must be operated in integer mode for ibaScope.

➚ Refer to chapter 4.1

7.1 HW configuration

The SM64 device type must be selected in Fob-F mode.



Hardware Einstellungen

Allgemein

Karte die den Interrupt generieren soll: Fob-4i-PCI in Steckplatz 12 an Bus 2

Zeitbasis für den Fob-M Modus (tb1) [us]: 40

Zeitbasis für den Fob-F Modus (tb2) [ms]: 1

DCF

● DCF77 ausschalten

○ DCF77 aktivieren (für den ersten Binärkanal der ersten FOB im FOB-M Modus)

○ DCF77aktivieren und PC Uhr synchronisieren

PCI Boards and links

- Karte 0 (Steckpatz 12, B
  - Link 0
  - Link 1
  - Link 2
  - Link 3
- Karte 1 (Steckpatz 14, B
  - Link 0
  - Link 1
  - Link 2
  - Link 3

Lichtwellenleiterlink

☑ Aktiv

Fob-M Modus | Fob-F Modus

Geräte-Name: SM64_SD16

Geräte-Typ: SM64

Padu-Nummer / Addresse: 1

Aussteuerbereich min: -10 - max: 10

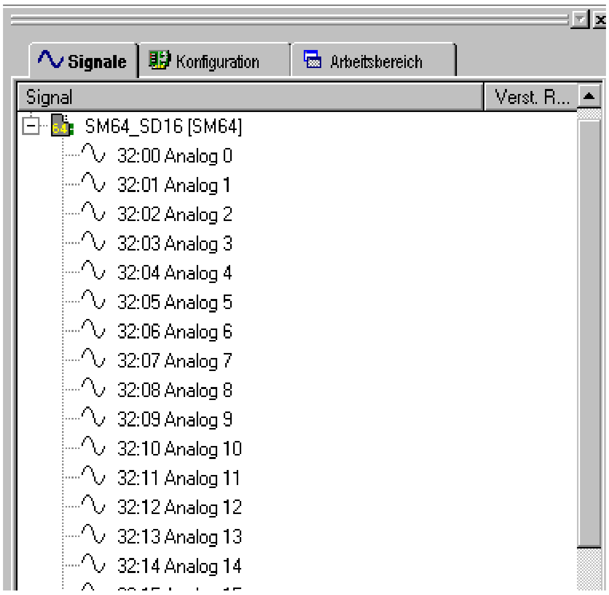
Gerätenummer (Signal Präfix): 32:xx, 32.xx

Link-Details: SM64_SD16

OK | Abbrechen

7.2 Selection of input and output resources

The 2 SIMADYN D send channels (M0PADAT and M1PDADAT telegrams) appear as 1 SM64 module with 64 analog and digital values in the signal tree.



SM64_SD16 [SM64]
32:00 Analog 0
32:01 Analog 1
32:02 Analog 2
32:03 Analog 3
32:04 Analog 4
32:05 Analog 5
32:06 Analog 6
32:07 Analog 7
32:08 Analog 8
32:09 Analog 9
32:10 Analog 10
32:11 Analog 11
32:12 Analog 12
32:13 Analog 13
32:14 Analog 14

8 Connecting other devices

8.1 iba periphery boards

The configuration at the SIMADYN D end is implemented in the manner described in chapter 3.

The following must be remembered in this context:

❑ The periphery boards solely use integer values, with the value range of –32768 to +32767 corresponding to -10 V to +10 V, for example. (In the N2 format, this integer value range corresponds to the range of 200 % to +199 %).

❑ The integer mode must be set at the rotary switch of the ibaLink-SM-64 SD16 board.

❑ Data in the 32-bit format (N4) is reduced to the 16-bit format (N2).

The following devices can be connected:

❑ Analog and digital inputs with ibaPADU-8, ibaPADU-16 and ibaPADU-32

❑ Analog and digital outputs with ibaPADU8-O

❑ Connection of ibaNet750-BM, so that all the features of the WAGO process modules (inputs, outputs, shaft-angle encoder or SSI encoder, etc.) are available.

8.2 Communication with other systems

The configuration at the SIMADYN D end is implemented in the manner described in chapter 3.

Note that the same telegram format (real or integer) must be set on the ibaLink-SM-64 SD16 board and on the counterpart board.

The following communication options exist:

❑ Communication with ibaLink-SM-64-io at SIMATIC S5 and SIMICRO MMC

❑ Communication with another ibaLink-SM-64-SD16 at SIMADYN P16 or SIMADYN P32

❑ Communication with ibaLink-SM-128V-i-2o at VME bus systems, such as ALSTOM Logidyn

❑ Communication with ibaBM-DPM-64at Profibus DP

9 Support and Contact

Support

Phone:	+49 911 97282-14
Fax:	+49 911 97282-33
E-Mail:	support@iba-ag.com

Note

If you require support, specify the serial number (iba-S/N) of the product.

Contact

Headquarters

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Phone:	+49 911 97282-0
Fax:	+49 911 97282-33
E-Mail:	iba@iba-ag.com
Contact:	Mr. Harald Opel

Regional and Worldwide

For contact data of your regional iba office or representative please refer to our web site

www.iba-ag.com.