User Guide

Option: Profibus (F3)

COMPAX - Software version >V3.0 and higher
Profibus - Software version >V1.3 and higher

October 98

HAUSER
We automate motion

CERTIFIED DIN EN ISO 9001 QUALITY SYSTEM
Reg. Nr. 36 38

Parker Hannifin GmbH
EMD HAUSER
Robert-Bosch-Str. 22
D-77656 Offenburg, Germany
Phone: +49 (0)781 509-0
Fax: +49 (0)781 509-176
http://www.parker-emd.com

Parker Hannifin plc
EMD Digiplan
21 Balena Close
Poole, Dorset BH17 7DX UK
Phone: +44 (0)1202 69-9000
Fax: +44 (0)1202 69-5750
http://www.parker-emd.com

Contents

Device Classification:

This documentation is valid for:

- COMPAX 25XXS with F3 option
- COMPAX 45XXS with F3 option
- COMPAX 85XXS with F3 option
- COMPAX P1XXM with F3 option
- COMPAX 02XXM with F3 option
- COMPAX 05XXM with F3 option
- COMPAX 15XXM with F3 option
- COMPAX 35XXM with F3 option

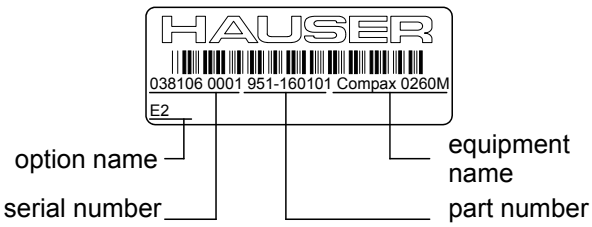
XX: any characters
F3: Profibus-Option

Key to model type

e.g..: COMPAX 0260M:
COMPAX: Name
02:Power class
60:Version e.g. "00": Standard model
M: Model M: Multi-axis model
S: Single-axis model

HAUSER product label

The product label is found on the top of the unit and contains the following information:



1. Profibus Interface Description

The Profibus interface enhances the flexibility of the Compact Servo-Controllers COMPAX-M and COMPAX-S.
You have access to all normal COMPAX functions such as:

- Changing parameters.
- Presetting commands.
- Reading and writing control inputs and outputs.
- Writing to record storage.
- Reading status.

These functions are available to you in the Profibus operating modes**FMS** and **DP** . The operating modes are set as:

- DP mode,
- FMS mode, or
- Mixed mode: FMS and DP.

In DB mode you have a cyclic channel available on which you can access current process data.

COMPAX Description

The COMPAX functions are described in the COMPAX product manual.

Profibus: General

This Profibus documentation is customized for use with COMPAX; for additional information, please refer to general Profibus literature which is available.

Profibus Software

Profibus is made up of Layers 1, 2 and 7 of the ISO/OSI Layer Model and is defined in DIN 19 245. Part 1 of this standard describes Layers 1 and 2. For Layer 7 there are two protocols, the FMS protocol, which is described in Part 2 of DIN 19 245, and the DP protocol, which was defined as Part 3 of the standard. The software implemented in the interface module (COMPAX Option F3) is based on the COMBI-Slave developed by the TMG-i-tec company and permits COMPAX operation with both FMS and DP protocol. The definition PROFIDRIVE, which was developed for speed variable drives on Profibus, is not used with COMPAX, since it was designed chiefly for frequency converters and is therefore not appropriate for the considerably wider range of functions offered by a servo controller.
Nevertheless, an attempt has been made to adapt the mechanisms described in this profile for communication on the DP channel and to create analogies.

SINEC L2

The Profibus interface allows the COMPAX to run as a DP slave on the SINEC L2-DP bus.

Abbreviations

AK: Command resp. reply identifier (range 0...15)
BDA: Command data (5th to 10th octet)
BKD: Command identifier data
BKE: Command identifier (1st and 2nd octet)
BNU: Command number (range 0...2048)
BPO: Command process data object.
IND: Subindex (3rd octet), Frame-No. (4th octet)
PAD: Process output data
PED: Process input data
PZD: Process data
SPM: Toggle bit for spontaneous message processing
STW: Control word
ZSW: Status word

Syntax

0x45: The preceding characters "**0x**" mean that "45" is represented in hex format.
Octet: An octet is 8 bits; it corresponds to one byte.

Profibus ID (ID number)

COMPAX with Profibus has ID no..**:**
0xEE95 ≡ 61 077.

Previous ID No. (for Profibus software <V1.20):
0xCCCC ≡ 52 428.

Type files / Device Master file

By installing the ServoManager, the Profibus type files as well as the device Master file are copied to your PC. These files contain the COMPAX Profibus data.

Type files

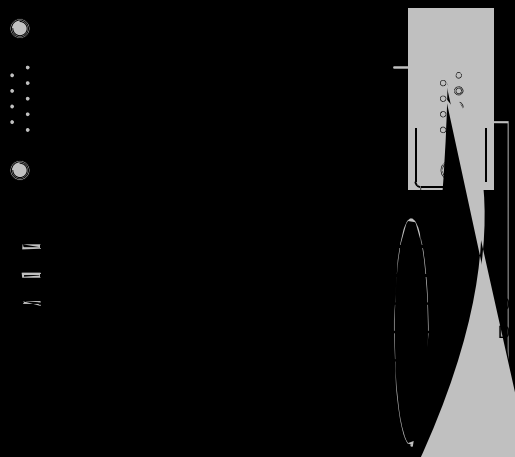
- German: ...SRVBOX\DATA\CPX300TD.200
- International: ...SRVBOX\DATA\CPX300SX.200

Device Master file

...SRVBOX\DATA\CPX00300td.GSD

Which file you need depends on the Master and its Profibus software.

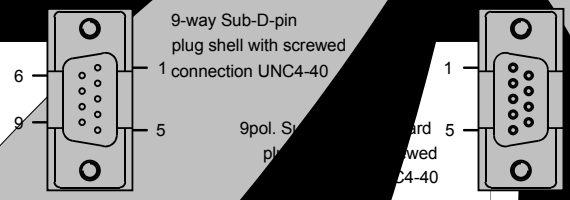
2.

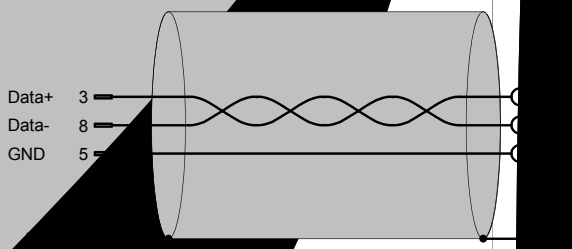


the Profibus.

the Master

Netzmo AX-S (IN)

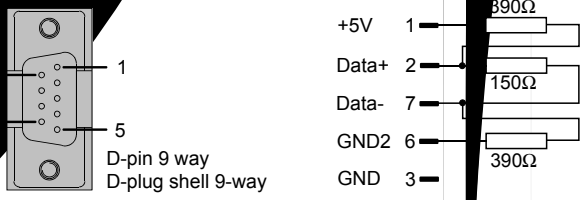




mm^2 + screen

ntains a terminatio lug

/01



2.3 Device Settings

Profibus settings are made using COMPAX parameters; these may be set from the front panel (see next page).

2.3.1 Device Address

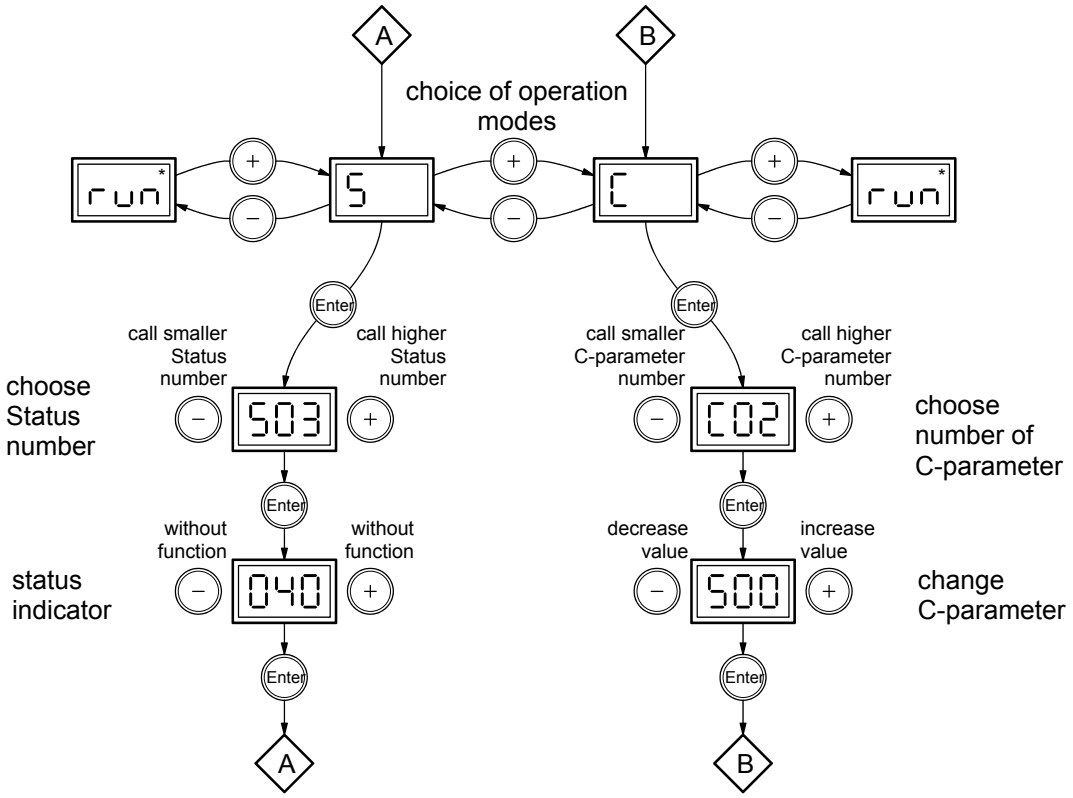
Settable on every COMPAX using parameter P194; value range: 0-126.
99:Standard setting; set an address between 1 and 126 before start-up.
Maximum number of participants without repeater: 32

2.3.2 Baud Rate

9 600 ... 1 500 000 baud.
The baud rate is automatically set!
For FMS max. 500K baud.

2.4 Bus Settings on Front Panel

The bus protocol (COMPAX parameters P194 and P196) can be set from the front panel of the COMPAX.
Procedure:



Meaning:

C-Parameter	Meaning	Range	COMPAX-Parameter	Active on
C01	**Device address**	0...126	P194	**Power on**
C02	**Baud rate**	Automatically set!		
C03	**Bus protocol**	0...255	P196	**Power on**
C04 - C11	**reserved**			

The bus parameters are loaded by cycling Power off / Power on!

3. Profibus Configuration

3.1 COMPAX Settings

P196 defines the transmission protocol of a COMPAX slave on the Profibus:

Function	Setting	Significance
Command Process Data Object Type (BPO) (For description, see pages starting at9)	BPO-Type 1 BPO-Type 2 BPO-Type 3 BPO-Type 4	0 1 2 3
Input/Output Configuration (for DP mode only) (For description see page 8)	I/O together I/O separate	0 4
DP Configuration BPO-type division (For description see page 8)	3 division (BKE+IND/BDA/PZD) 2 division (BKD/PZD)	0 8
Profibus protocol	FMS mode only FMS mode only Mixed mode (DP and FMS mode) DP mode only	0 32 64 96

Setting for DP mode only

Setting for DP mode only

For **I/O together** (e.g., for a PLC Master), the Command Process Data Objects (BPO) are in the same input and output addresses (for Type 1 for example from input byte 4 to input byte 21 and output byte 4 to output byte 21).

For **I/O separate** you may set the input and output addresses separately. (For Type 1 for example from input byte 4 to input byte 21 and output byte 40 to output byte 57)

⇨ Transmission protocol settings for input and output values must always be the same!

Setting for DP mode only

Setting for DP and/or FMS mode

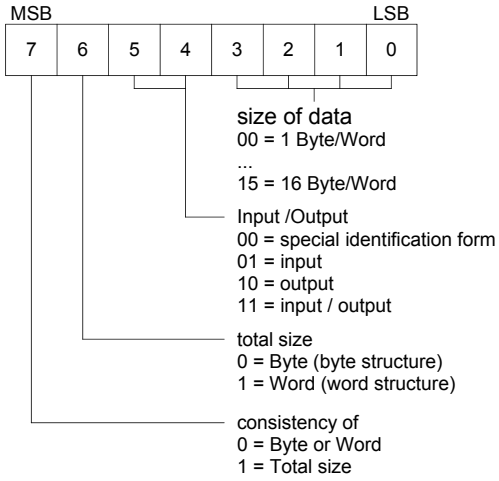
⇨ The desired setting is made by entering the sum of the significance into COMPAX parameter P196.

⇨ COMPAX units connected on the Profibus may be configured each differently!

3.2 Master Settings (DP Mode)

The Profibus Master is configured according to the running COMPAX, i.e., the configuration of each COMPAX connected on the bus must be conveyed to the Master.
This is the purpose of the identification byte. The identification byte is configured separately on the Master for each part of the process data (2 or 3 division).
Depending on the Profibus Master software in use, the identification bytes must be directly entered or may be menu selected in plain text (here the Master uses the supplied COMPAX Master filr; see page 4).

DP Configuration: Identification Byte general



COMPAX settings:

Data length:

Depending on the Profibus setting, BPO types have the following division and lengths:

Setting	Parts	No. of bytes in the BPOs			
		TYPE 1	TYPE 2	TYPE 3	TYPE 4
2 division	1. BKD	10	10	0	0
	2. PZD	8	2	8	1
3 division	1. BKE+IND	4	4	0	0
	2. BDA	6	6	0	0
	3. PZD	8	2	8	1

Input/output: **"11" input/output**

Format length: **"0": byte structure**

Consistency: **"1": over the entire length**
Type 4 = "0" (consists of only 1 byte)

This results in the following identification bytes for the COMPAX

1. Input/output byte of a BPO part are configured together with an identification byte (**I/O together**).

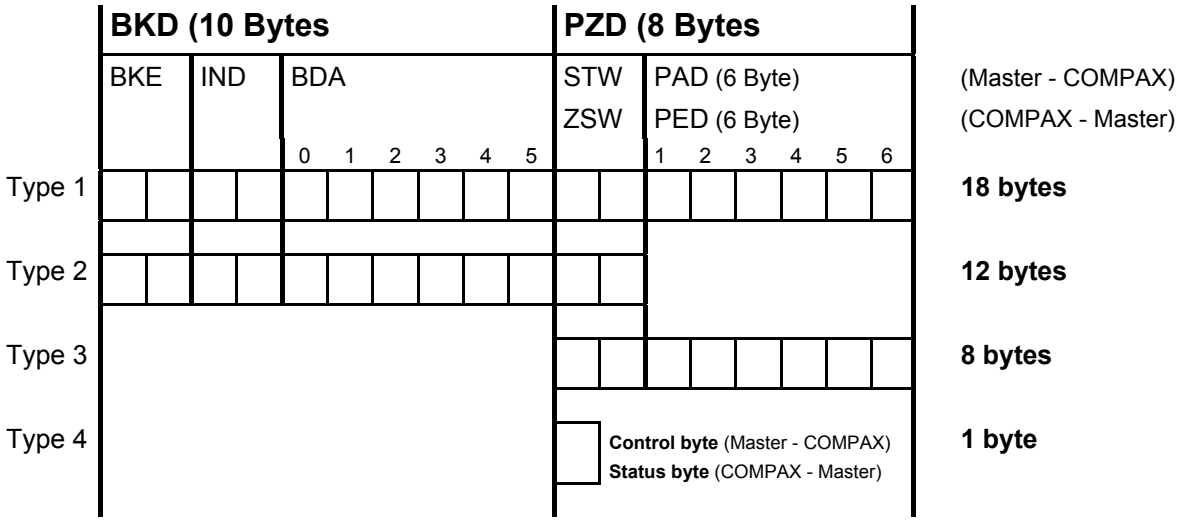
Setting	Parts	Contents			
		TYPE 1	TYPE 2	TYPE 3	TYPE 4
2 division	1. BKD	0xB9 (10111001) = 185	0xB9 (10111001) = 185	-	-
	2. PZD	0xB7 (10110111) = 183	0xB1 (10110001) = 177	0xB7 (10110111) = 183	0x30 (00110000) = 48
3 division	1. BKE+IND	0xB3 (10110011) = 179	0xB3 (10110011) = 179	-	-
	2. BDA	0xB5 (10110101) = 181	0xB5 (10110101) = 181	-	-
	3. PZD	0xB7 (10110111) = 183	0xB1 (10110001) = 177	0xB7 (10110111) = 183	0x30 (00110000) = 48

2. Input/output byte of a BPO part are configured separately with two identification bytes (**I/O separate**).

Setting	Parts		Contents			
			TYPE 1	TYPE 2	TYPE 3	TYPE 4
2 division	1. BKD	Output Input	0xA9 (10101001) = 169 0x99 (10011001) = 153	0xA9 (10101001) = 169 0x99 (10011001) = 153	-	-
	2. PZD	Output Input	0xA7 (10100111) = 167 0x97 (10010111) = 151	0xA1 (10100001) = 161 0x91 (10010001) = 145	0xA7 (10100111) = 167 0x97 (10010111) = 151	0x20 (00100000) = 32 0x10 (00010000) = 16
3 division	1. BKE+IND	Output Input	0xA3 (10100011) = 163 0x93 (10010011) = 147	0xA3 (10100011) = 163 0x93 (10010011) = 147	-	-
	2. BDA	Output Input	0xA5 (10100101) = 165 0x95 (10010101) = 149	0xA5 (10100101) = 165 0x95 (10010101) = 149	-	-
	3. PZD	Output Input	0xA7 (10100111) = 167 0x97 (10010111) = 151	0xA1 (10100001) = 161 0x91 (10010001) = 145	0xA7 (10100111) = 167 0x97 (10010111) = 151	0x20 (00100000) = 32 0x10 (00010000) = 16

4. DP Mode

The command process data object (BPO) is defined for cyclic data exchange based on the PROFIBUS profile for speed variable drives. This can be used both send both process data and commands from the Master to COMPAX and the reverse.
Four possible types of the BPO each for both directions (Master → COMPAX and COMPAX → Master) are defined (for setting, see mode types on page7):



BKD: Command identifier data
PZD: Process Data
BKE: Command ID (1st and 2nd Octet[1])
IND: Subindex (3rd octet), Frame-No. (4th octet)
BDA: Command data (5th to 10th octet)
STW: Control word
ZSW: Status word

Note!

Cyclic channel: Process data

COMPAX only generates an action when the BPO change. This means relative positioning with the same position target can only be done by handshaking. This is done by means of one bit each in the status and control words.

PAD: Process Output Data (PA data): The data which COMPAX reads from the process data channel.
PED: Process Input Data): The data which COMPAX writes to the process data channel.

BPO-write / BPO-read

Each BPO is stored in the COMPAX as BPO-Write (write-only) and BPO-Read (read-only).
The BPO type for the cyclic data transmission can be individually set in each COMPAX using a parameter (P196), i.e., the COMPAX units on a bus can be variously configured (e.g., COMPAX 1: BPO - Type 1; COMPAX 2: BPO - Type 3; ... .

BPO-Write causes the Master to give jobs to the slave for processing commands (BKD), as well as process data (control word (STW) and set points (PAD)).

When reading by means of BPO-Read, the Master gets process data (status word (ZSW) and actual values (PED)) as well as replies from the COMPAX for processing commands (BKD).

The **BPO-Type 4** is implemented for transmission of the special control/status byte.

⇨ Only one type of BPO-Read and BPO-Write is permitted in a COMPAX!
In other words, both directions Master → COMPAX and COMPAX → Master always work with the same BPO type.

BKD Function

The following tasks are processed using the BKD mechanism:

- Operating and monitoring the COMPAX: Master → COMPAX
- Sending and acknowledging spontaneous messages: COMPAX → Master → COMPAX

In the BKD mechanism the Master formulates a command; COMPAX processes the command and formulates the reply . A command as well as a reply can consist of multiple BPO reads or writes (so-called frames). The command or reply length is located in the IND (see page 15).

[1] An octet corresponds to one byte.

4.1 Structure of the output BPO of Type 1 (Master → COMPAX)

BKD

octet

BKE 1 2

Command identifier:

octet 1								octet 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				SPM	BNU										

AK: Command recognition: Type of master command. see page 12.
SPM: Spontaneous message bit: see page 14.
BNU: Command number: see page 15.

IND 3 4

octet 3	octet 4
Subindex, when using FMS objects over DP (see page 47)	Number of additional BPO's which are part of the command.

BDA 5 6 7 8 9 10

Command data (BDA)
The command data contain the argument for a command or the coding of a command.
Examples:
POSA: For the POSA command, the BDA's contain the position in the COMPAX number format.
Parameter: The command data of the master command contain the new contents of the parameter. The format in the BDA's is indicated for the respective command.
Coding for VP: BDA0 = 0x56; BDA1 = 0x50

The contents and formats of the commands are described starting on page 23.

PZD

STW 11 12

Control

octet 11								octet 12							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

-	7
-	6
New set-point	5
Target position relative (1) / absolute (0)	4
Teach real null	3
AG guiding	2
Hand-	1
Hand+	0

0	Idle / brake closed
1	Idle / brake open
2	Start
3	Find real null
4	Find machine null
5	Break
6	Stop
7	Acknowledge

PAD 13 14 15 16 17 18

Cyclical process output data: Master → COMPAX
The 6 bytes of the PAD can be configured with the following FMS objects:

Object name	Description	Index		COMPAX Param.	Byte	see
		dec	hex	P139 ... P142	Number	page
STEUERBYTE	Control byte	4812	0x12CC	1231872	1	53
CONTROL	Control commands	4816	0x12D0	1232896	1	55
POSITION_TARGET	Target position default	4822	0x12D6	1234432	4	74
SPEED	Traverse speed	4823	0x12D7	1234688	2	76
OVERRIDE	Reduce traverse speed	4826	0x12DA	1235456	1	77
OUTPUT_WORD	16 Dig. Set/reset outputs	4832	0x12E0	1236992	2	82
START_N	Execute program record N	4844	0x12EC	1240064	1	84
START_N_GO	Program start beginning at record N	4845	0x12ED	1240320	1	84

For more, see page 18.

4.2 Structure of the input BPO of Type 1 (COMPAX → Master)

BKD

octet		
BKE	1	**Command identifier:**
	2	
IND	3	
	4	
BDA	5	**Command data (BDA)**
	6	
	7	
	8	
	9	
	10	

octet 1								octet 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				SPM	BNU										

AK: Reply identifier: type of slave command. see page 12.
SPM: Spontaneous message bit: see page 14.
BNU: Command number: see page 15.

octet 3	octet 4
Subindex, when using FMS objects over DP (see page 47)	Number of additional BPO's which belong to the reply.

Command data (BDA)
The command data contain the argument for a reply or the coding of a command.
Examples:
POSA: The BDA's as reply to the POSA command contain the position in COMPAX number format.
Parameter, Status: The command data of the COMPAX reply to a parameter or status request contain the respective contents of the parameter or status. The format in the BDA's is indicated for the respective command.
Coding for VP: BDA0 = 0x56; BDA1 = 0x50 (reply)

The command contents and formats are described beginning on page 23.

PZD

ZSW	11	**Status word:**
	12	
PED	13	Cyclical Process Input Data: COMPAX → Master
	14	The 6 bytes of the PED can be configured with the following FMS objects:
	15	
	16	
	17	
	18	

octet 11								octet 12							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Motor stalled	7
Lag error	6
Set point acknowledgement	5
-	4
-	3
-	2
AG guiding	1
Program. target position reached	0

0	-
1	-
2	Ready for Start
3	Fault
4	MN was reached
5	-
6	Idle after stop
7	Warning

Object name	Description	Index		COMPAX Parameter	Byte	see
		dec	hex	P135 ... P138	Number	page
POS_ACTUAL	Actual position value	4801	0x12C1	1229056	4	75
S3	Contour error	4802	0x12C2	1229312	2	76
S4	Current traverse speed	4803	0x12C3	1229568	2	77
S5	Current motor torque	4805	0x12C5	1230080	2	69
STATUSBYTE	Status byte	4813	0x12CD	1275445	1	53
INPUT_WORD	Log. state of the 16 dig. inputs	4829	0x12DD	1236224	2	81
OUTPUT_WORD	Log. state of the 16 dig. outputs	4832	0x12E0	1236992	2	81

For more, see page 21.

4.3 BKE: Command identifier

BKE structure:

octet 1								octet 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				SPM	BNU										

4.3.1 AK: Command/ reply Processing

The command identifier AK defines the type of command to be sent, or confirms in the reply that a command was executed.

Command (AK)	Command Master → COMPAX	reply (AK)	Reply COMPAX → Master
0	no command	**0**	no reply
1	Request COMPAX parameter	**1**	Send COMPAX parameter
2	Change COMPAX parameter	**1**	COMPAX parameter changed
3	Request COMPAX record	**2**	Send COMPAX record
4	Change COMPAX record	**3**	COMPAX record changed
5	Request COMPAX status	**4**	Send COMPAX status
6	Initiate COMPAX command	**5**	COMPAX command executed
7		**7**	Command not executable (with error no.)
8	Initiate special COMPAX command	**8**	COMPAX special command executed
9	Request COMPAX variable	**9**	Send COMPAX variable
10	Change COMPAX variable	**9**	COMPAX variable changed
		10	Spontaneous message (error/event)
11	Request COMPAX curve memory value or COMPAX text memory	**11**	Send COMPAX curve memory value or COMPAX test memory
12	Change COMPAX curve memory value or COMPAX text memory	**12**	COMPAX curve memory value or COMPAX text memory changed
13		**13**	
14	Change COMPAX objects	**14**	COMPAX objects changed
15	Request COMPAX objects	**15**	Send COMPAX objects

The Master sends a command to a COMPAX using BPO-Write.

- The Master repeats this command until with BPO-Read an reply comes back from the COMPAX. This procedure ensures the transmission of the commands / replies on the user level. There is always a command in process.
- COMPAX holds the reply until the Master formulates a new command. In case of replies which contain status words, COMPAX always replies to a repetition with the current value.

Command not executable

If COMPAX is unable to carry out a command, COMPAX replies with "command not executable"; the corresponding error number (either interface error or COMPAX error) is held in BDA0.

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	Octet 3	Octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
7	-	--	0x00	0x00	F.-Nr.	--	--	--	--	--
Octet 1: 112 (+8)		Octet 2: 0								

Interface Errors

Interface errors have the following meaning:

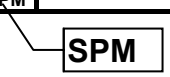
No.:	Meaning	Nr.:	Meaning
80	Command ID error.	83	Incorrect Subindex.
81	Frame number error.	84	Object cannot be read.
82	Incorrect command number.	85	Object cannot be written.

⇨ The specific COMPAX error messages are described in the COMPAX Product Handbook.

4.3.2 SPM: Spontaneous Message Processing

BKE structure:

octet 1								octet 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				SPM	BNU										



A changed SPM (0→1 or from 1←0) designates a spontaneous message.
COMPAX parameter P193 is used to set the mode by which a spontaneous message in the COMPAX is sent to the Master, for:

- "Error" (BNU=1, BDA0=error-no.),
- "Programmed target position not reached" (BNU=2) or
- "Comparator point reached" (BNU=3, BDA0=Comp. no.).

Each of these spontaneous messages can be switched on or off individually using P193.

Function:

- Normal reply processing is interrupted by the COMPAX.
- Instead, the BKD reply contains the identifier "spontaneous message" with the corresponding indication (BNU, BDA0, see below) Simultaneously, COMPAX changes the spontaneous message toggle bit.
- The spontaneous message is sent until the Master has acknowledged the message by changing the spontaneous message toggle bit.
- Then COMPAX continues with the interrupted reply processing or sends the next spontaneous message.

The spontaneous messages can be switch individually using P193:

Spontaneous messages	Significance
automatic error message	1
automatic "position reached" message	2
automatic reporting of the comparator switchpoints	4

⇨ The desired setting is made by entering the sum of the significance into COMPAX parameter P193.

Spontaneous Messages

Error

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
10	0/1	1	0x00	0x00	F.-Nr.	--	--	--	--	--
Octet 1: 160 (+8)		Octet 2: 1								

Programmed target position reached

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
10	0/1	2	0x00	0x00	--	--	--	--	--	--
Octet 1: 160 (+8)		Octet 2: 2								

Comparator point reached

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
10	0/1	3	0x00	0x00	C.-No.	--	--	--	--	--
Octet 1: 160 (+8)		Octet 2: 3								

4.4 BNU: Command Number

BKE structure:

octet 1								octet 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				SPM	BNU										

The contents of the BNU for each command and each reply is described in chapter "4.13 BKD: Error Coding of the (BPO-Type 1 and 2)" beginning on page 24.

Basic structure of the BNU for commands from Master → COMPAX:

Bit 0 ... 7: (Octet 2): For the corresponding commands, contains the parameter, status, variable or record number and, for direct COMPAX commands, a command coding (resp. a part of the command coding).

Bit 8 ... 10 (part of Octet 1): Contains: Number of relevant data in the BDA + 1.

Basic structure of the BNU for replies from COMPAX → Master:

The BNU of the reply corresponds to the BNU in the respective command.
Exception: When requesting records the length of the command and its command code is contained in the BNU.

4.5 IND:

IND structure:

octet 3	octet 4
Subindex, when using FMS objects over DP (see page 47)	The number of BPO's which belong to the command or reply.

A job can consist of one, two, three or four BPO-Writes (Write Frames);
an reply of one, two or three BPO-Reads (Read Frames).
Frame no. (4th octet = 2nd Octet of IND) contains:

For command: How many write frames the master is still sending for a command.

For the reply: How many read frames are still be be read for the issued command or the selection of which read frame is requested by the master (when requesting record data).

Example: Write Frames

Frame-Nr. 4. octet	Command: Master → COMPAX Reply: COMPAX → Master	Request next reply: Master → COMPAX
= 3	fourth from last frame	
= 2	third from last frame	last frame
= 1	second from last frame	second frame
= 0	last frame	first frame

4.6 BDA: Command Data

The following data may be in the BDA's:

- Value of a parameter or of a variable (in DSP format).
- Contents of a status (differing formats).
- Record contents (Formats corresponding to the command code table)
- Command code of a direct COMPAX command.

The formats of the BDA's for each command and each reply are described in Chapter "4.13 BKD: Error Coding of the (BPO-Type 1 and 2)" beginning on page 24.

4.6.1 DSP Number Format

All COMPAX parameters are transmitted in DSP number format.
The number in DSP format is represented as a fractional number. 24 bits are reserved as a whole-number component and 24 bits as decimal places:

2^{23} ..... 2^{2} 2^{1} 2^{0}	2^{-1} 2^{-2} 2^{-3} ..... 2^{-24}
3 bytes whole-number	3 bytes after decimal place

Negative numbers are represented as 2's complement.
Forming a 2's complement:

- Determine the bit combination of the positive number value.
- Negate the binary value.
- Add 1.

Format Conversion

You can generate this format from any number with decimal places as follows:
Example: number = 450.5

1. Multiply number by 2^{24} .
   $450.5*2^{24}$ = 7 558 135 808.

Convert 2. 7 558 135 808 into a hex number (or first into an integer) =>.0x00 01 C2 80 00 00 ≡ whole number, decimal≡ MSB,.... LSB, MSB,.... LSB.

3. These bytes must now be entered in the given sequence in the commands. The sequence of the bytes is reversed. Do not alter the order of the bits.

This conversion also applies to negative numbers.

Examples of the number format of "xx xx xx xx xx xx"

Number	MSB					LSB
10	00	00	0A	00	00	00
360	00	01	68	00	00	00
450.5	00	01	C2	80	00	00
-1	FF	FF	FF	00	00	00
	whole numbers		decimal places			

The result, for example, for **360.0** is:

"00 00 00 68 01 00"

as BDA entries.

4.7 CONTROL WORD (BPO-Type 1...3)

Activating COMPAX control commands.
You can only control the COMPAX using the control word if the corresponding bits are enabled with the FMS object INPUT-MASK or the COMPAX parameter P221.
Enabling the control word functions using COMPAX parameters with: P221 = 63.
Now the COMPAX inputs E1 ... E6 are no longer reserved for fixed functions, but are freely accessible.

Data Description

Data byte [bit]	Significance	Corresponding input logic states	Function enable using "INPUT_MASK" (not relevant when P221=63)
1 [7] MSB	-		
1 [6]	-		
1 [5]	New set point		
1 [4]	Position target relative (=1) or absolute (=0)		
1 [3]	Teach Real Null	E1 and E4 ="1"	Data byte 2[0]="1" and 2[3]="1"
1 [2]	AG guiding		
1 [1]	Hand-	E3="1"	Data byte 2[2]="1"
1 [0]	Hand+	E2="1"	Data byte 2[1]="1"
2 [7]	Quit	E4="1"	Data byte 2[3]="1"
2 [6]	Stop	E6="1"	Data byte 2[5]="1"
2 [5]	Break	E1 and E6 ="1"	Data byte 2[0]="1" and 2[5]="1"
2 [4]	Find machine home	E1 and E2 ="1"	Data byte 2[0]="1" and 2[1]="1"
2 [3]	Find Real Null	E1 and E3 ="1"	Data byte 2[0]="1" and 2[2]="1"
2 [2]	Start	E5="1"	Data byte 2[1]="1"
2 [1]	Idle and brake closed open		
2 [0] LSB	Idle and brake closed		

⇨ By partially switching input functions to the STEUERWORT, the multi-function of E1 permits a function limitation: Example: If a function with E1 occupies the control word (e.g. teach real null), then additional E1 functions (such as the "QUIT" function) are ignored by the inputs.

Therefore: If you need all the input functions, the function must be completely reassigned, either to the inputs (P221 = 0) or to the control word (P221 = 63).

Command Recognition

The control word is sent cyclically on the bus from the Profibus Master.
Note! When a PLC is the Master, the control word may not be present for too short a time.
The PLC and Profibus cycle are asynchronous. If the control word is output for just one PLC cycle (scan), data may be lost.
Rectify the problem with:

- A control word which is available for a sufficiently long time

or

- by reading back the FMS object "STEUERWORT" (over FMS or DP with the "Request object" command; see page 47): If the change is in this object, then the command was recognized.

COMPAX - I/O - Functions using the Control Word (Data bits 1[1], 1[0], 2[7], ... 2[2])

Direct switching of the I/O functions by removing a function and simultaneously setting another function is not recognized by the COMPAX; **Exception:** STOP and BREAK (these are always recognized immediately).
Therefore proceed as follows:

- Remove the previous functions by sending a "null telegram" (allow status to remain until it has been recognized by the COMPAX).
- When the COMPAX is ready (Data bit 1[2]="0"), set a new function.

Example: Switch from Hand+ to Hand-

- Reset Hand+: data bit 1[0] = "0"
- Wait until the COMPAX set Data bit 1[2]="0" or with handshake (see below).
- Set Hand-: data bit 1[1] = "1"

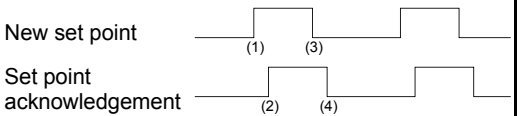
New Set Point

Handshake for transferring PAD target position values

You can place the "target position" object on the cyclic process output data channel of the DP mode. Then you can cyclically specify new set points. Note that in DP mode, the data can only be newly handed over if the BPO has changed. For relative positioning this has as a consequence that identical target positions coming right after each other are not accepted. In this case, a handshake must be implemented for transferring the positions. This is done using the following bits:

- Control word byte 1 bit 5 "new set point" and
- Status word byte 1 bit 5 "set point acknowledgement"

Function:



Transition	Meaning	Condition
1	New set-point	Set-point acknowledgement ="0" Set-point can be sent
2	Set point acknowledgement	Set-point acknowledgement ="1" Set-point recognized New set-point ="0"
3	New set point	Set-point acknowledgement ="0" New set-point can be sent
4	Set point acknowledgement	

To ensure reliable establishment of a handshake using the FMS object "P__ENABLE" bit 7, the automatic transfer of a changed LAGE_ZIEL can be turned off (see page 90).

4.8 STATUSWORT

The status word displays information concerning the status of the device, as well as messages.

Data Description

Data byte [bit]		Significance	Data byte [bit]		Significance
1 [7]	="1"	Motor locked up	2 [7]	="1"	Warning or stop indicator (signal has the reverse meaning as A2 for the COMPAX)
1 [6]	="1"	Contouring Error	2 [6]	="1"	Standstill after Stop
1 [5]	="1"	Set Point Acknowledgement (Profibus)	2 [5]	="1"	-
1 [4]	="1"	Set Point Not Acknowledgement (COMPAX)	2 [4]	="1"	Machine was homed
1 [3]	="1"	-	2 [3]	="1"	Fault (signal has the reverse meaning as A1 for the COMPAX)
1 [2]	="1"	I/O - Function active*	2 [2]	="1"	Ready for Start
1 [1]	="1"	AG Guiding	2 [1]	="1"	-
1 [0]	="1"	Programmed Set Point Reached	2 [0]	="1"	-

* active I/O-function: A COMPAX control signal becomes activated through the „STEUERWORT“ or „CPX STW“. When an I/O-function is activated no further I/O-functions are recognized from the COMPAX except STOP and BREAK. Therefore, send the next I/O-functions only when Data bit 1[2]=„0“
I/O-functions are all the control functions which can normally become activated with E1...E6.

4.9 PAD-control

The following COMPAX communications objects (FMS) can be written cyclically using the Process Output Data of BPO-Type 1 / Type 3.

Object Name	Description	Index		COMPAX Parameter[2]	Byte	see
		dec	hex	P139 ... P142	Number	page
STEUERBYTE	Control byte	4812	0x12CC	1231872	1	53
CONTROL	Control commands	4816	0x12D0	1232896	1	55
LAGE_ZIEL[3]	Target position default	4822	0x12D6	1234432	4	73
SPEED	Traverse speed	4823	0x12D7	1234688	2	76
OVERRIDE	Reduce traverse speed	4826	0x12DA	1235456	1	77
OUTPUT_WORD	16 Dig. Set/reset outputs	4832	0x12E0	1236992	2	82
START_N	Execute program record N	4844	0x12EC	1240064	1	84
START_N_GO	Program start beginning at record N	4845	0x12ED	1240320	1	84

Since the PAD channel has a length of 6 bytes, it is not possible to have simultaneous access to all the objects described here. This means you need to make an appropriate selection.

Setting the PAD:

- using the object "Process Output Data Description" (PA_SELECT; see page 89),

or

- using the COMPAX parameters P139, P140, P141, P142 (corresponds to the object PAD_INI; see page 92).

You may place each of the named objects on the PAD channel according to its required bytes.

Set the corresponding COMPAX parameter to the value given for the respective object (see table above).

FMS - Object:	Length in byte	Possible contents in the PAD channel					
		PAD1	PAD2	PAD3	PAD4	PAD5	PAD6
		P139	P140	P141		P142	
STEUERBYTE	1	▬	▬	▬	▬	▬	▬
CONTROL	1	▬	▬	▬	▬	▬	▬
OVERRIDE	1	▬	▬	▬	▬	▬	▬
START_N	1	▬	▬	▬	▬	▬	▬
START_N_GO	1	▬	▬	▬	▬	▬	▬
SPEED	2	▬	▬	▬		▬	▬
OUTPUT_WORD	2	▬	▬	▬		▬	▬
LAGE_ZIEL	4	▬	▬	▬	▬	▬	▬

⇨ Be sure that there is no double addressing in the PAD channel.
Double addressing occurs, for example, if the LAGE_ZIEL is in PAD1 - PAD4, and P141 is used to address PAD3 again.
The proper action in this case would be: LAGE_ZIEL in PAD1 - PAD4 using P139 = 1234432 and P140 = P141 = 0!

⇨ The channels are freely addressable using PA_SELECT. Shown are the possibilities using parameters.

2 Index * 256 + Subindex

3 The target position is taken over in the normal setting only if the value has changed (cf. page 18).

Disabling / enabling PAD's

The PAD's can be individually disabled and enabled using the object "Enable Process Output Data" (PA_ENABLE see page 90). This means an object is only written with the value from the PAD channel if the corresponding PAD's are also enabled.

After "Power on" the PAD's are enabled, i.e., normally no setting needs to be made here.

After power on, the COMPAX parameters P139, P140, P141 and P142 initialize the objects PA_INI and PA_SELECT and thereby the PAD channel.
The PAD_INI object can be used to read and write these parameters.

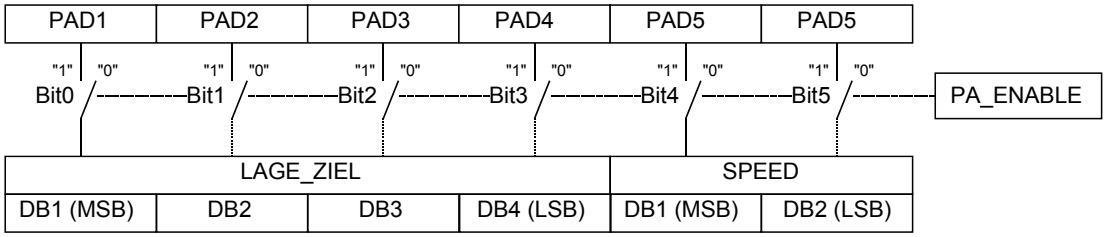
Notes

Note the following when configuring the PAD:

- After Power On, the PAD's are enabled if a valid configuration for the PAD is entered in the COMPAX parameters P139 ... P142.
- Note the length (number of bytes) of an object. An object can be represented on the PADs if the corresponding number of PAD bytes are free, i.e. not occupied by any other objects.
- Using the null object (Index and Subindex = 0) or by setting the corresponding COMPAX parameter to "0", an object can again be removed (deleted) from the PAD channel.

Example allocations for the PAD's

The LAGE_ZIEL and SPEED objects are represented on the PAD's.



FMS - Object "PA_SELECT"

Using the FMS object "PA_SELECT, the PAD assignment can be changed during operation (for details see page 90):

Subindex	Meaning	Value	
		dec	hex
1	PAD length (not variable)	6	0x06
2	Index of object which occupies PAD1	4822	0x12D6
3	Subindex of object which occupies PAD1	0	0x00
4	Index of object which occupies PAD2	0	0x0000
5	Subindex of object which occupies PAD2	0	0x00
6	Index of object which occupies PAD3	0	0x0000
7	Subindex of object which occupies PAD3	0	0x00
8	Index of object which occupies PAD4	0	0x0000
9	Subindex of object which occupies PAD4	0	0x00
10	Index of object which occupies PAD5	4823	0x12D7
11	Subindex of object which occupies PAD5	0	0x00
12	Index of object which occupies PAD6	0	0x0000
13	Subindex of object which occupies PAD6	0	0x00

After changing the PAD assignment using the FMS object "PA_SELECT", PA_ENABLE is set to "0" in order to avoid an undefined state. After a PAD change the PAD's must be manually enabled again using the PA_ENABLE object.

So that this setting of the PAD channel is already available upon power-up, the corresponding COMPAX parameters (P139 ... P142) resp. PAD_INI are to be assigned as follows:

Subindex (Parameter)	Meaning	Value	
		dec	hex
1 (P139)	Index and subindex of object which occupies PAD1	1234432	0x12D600
2 (P140)	Index and subindex of object which occupies PAD2	0	0x000000
3 (P141)	Index and subindex of object which occupies PAD3	0	0x000000
4 (P142)	Index and subindex of object which occupies PAD5	1234688	0x12D700

4.10 PED-Control

Using the Process Input Data of BPO-Type 1 / Type 3 it is possible to cyclically read the following COMPAX communication objects (FMS).

Object name	Description	Index		COMPAX Parameter[4]	Byte	see
		dec	hex	P135 ... P138	Anzahl	page
POS_ACTUAL	Actual position value	4801	0x12C1	1229056	4	75
S3	Contour error	4802	0x12C2	1229312	2	76
S4	Current traverse speed	4803	0x12C3	1229568	2	77
S5	Current motor torque	4805	0x12C5	1230080	2	69
STATUSBYTE	Status byte	4813	0x12CD	1275445	1	53
INPUT_WORD	Log. state of the 16 dig. inputs	4829	0x12DD	1236224	2	81
OUTPUT_WORD	Log. state of the 16 dig. outputs	4832	0x12E0	1236992	2	81

Since the PED channel has a length of 6 bytes, it is not possible to simulataneously read all the objects described here. This means you need to make an appropriate selection.

Setting the PED

- using the object "Process Input Data Description" (PE_SELECT; see page 88),

or

- using COMPAX parameters P135, P136, P137, P138 (corresponds to the object PED_INI; see page 91).

You may place each of the named objects on the PAD channel corresponding to its required bytes.

Set the corresponding COMPAX parameter to the value given for the respective object (see table above).

FMS - Object:	Length in byte	Possible assignment in the PED channel					
		PED1	PED2	PED3	PED4	PED5	PED6
		P135	P136	P137		P138	
STEUERBYTE	1						
S3	2						
S4	2						
S5	2						
INPUT_WORD	2						
OUTPUT_WORD	2						
LAGE_IST	4						

Be sure that there is no double addressing in the PED channel.
Double addressing occurs, for example, if the LAGE_IST is in PED1 - PED4, and P137 is used to address PED3 again.
The correct action in this case would be: LAGE_IST to PED1 - PED4 using P135 = 1229056 and P136 = P137 = 0!

The COMPAX parameters P135, P136, P137 and P138 initiate the object PI_SELECT and thereby the PED channel after the COMPAX is turned on.
These parameters can be read and written to using the PED_INI object.

The channels can be freely assigned using PE_SELECT. Shown are the possibilities using parameters.

[4] Index * 256 + Subindex

Notes

Note the following when configuring the PED:

- Note the length (number of bytes) of an object. An object can be represented on the PED's only if the corresponding number of PED bytes is available, i.e. are not occupied by any other objects.
- An object can be removed (deleted) again from the PED channel either by using the null object (Index and Subindex = 0) or by setting the corresponding COMPAX parameter to "0".

Example for configuring the PED's

Represent the object INPUT_WORD, S3 and S4 on the PEDs.

PED1	PED2	PED3	PED4	PED5	PED6
INPUT_WORD		S3		S4	
DB1 (MSB)	DB2 (LSB)	DB1 (MSB)	DB2 (LSB)	DB1 (MSB)	DB2 (LSB)

Configure PE_SELECT as follows:

The PAD assignment can be changed during operation by using the "PI_SELECT" FMS object 88):

Subindex	Meaning	Value	
		dec	hex
1	PED length (not variable)	6	0x06
2	Index of object which occupies PED1	4829	0x12DD
3	Subindex of object which occupies PED1	0	0x00
4	Index of object which occupies PED2	0	0x0000
5	Subindex of object which occupies PED2	0	0x00
6	Index of object which occupies PED3	4802	0x12C2
7	Subindex of object which occupies PED3	0	0x00
8	Index of object which occupies PED4	0	0x0000
9	Subindex of object which occupies PED4	0	0x00
10	Index of object which occupies PED5	4803	0x12C3
11	Subindex of object which occupies PED5	0	0x00
12	Index of object which occupies PED6	0	0x0000
13	Subindex of object which occupies PED6	0	0x00

To ensure that this setting of the PED channel is present upon Power On, the corresponding COMPAX parameters (P135 ... P138) or PED_INI must be assigned as follows:

Subindex	Meaning	Value	
(Parameter)		dec	hex
1 (P135)	Index and subindex of object which occupies PED1	1236224	0x12DD00
2 (P136)	Index and subindex of object which occupies PED2	0	0x000000
3 (P137)	Index and subindex of object which occupies PED3	1229312	0x12C200
4 (P138)	Index and subindex of object which occupies PED5	1229568	0x12C300

4.11 STEUERBYTE (BPO-Type 4)

The STEUERBYTE is used for program control and contains the following functions :

- Enables program start beginning at record 1 - 15.
  The record pointer is set to the corresponding program record.
- The program can be started, stopped, and continued.
- An error acknowledgement is possible.

⇨ Record select ="0000" and "Start" homes the machine.

Data Description

Data byte [Bit]	Significance	Meaning
[7]	Acknowledge	Error acknowledge with a positive edge
[6]	Stop with ramp in P10 without idling	Triggered by a positive edge
[5]	Continue ("1") / New start ("0") for start	Continue: continue program. New start: Program start at selected record.
[4]	Start / Stop	Start ⎍ / Stop ⎍ Start after condition defined in bit 5
[3]	Record select (2^3)	**Note!** Record select ="0000" and "new start" causes the machine to home itself.
[2]	Record select (2^2)	
[1]	Record select (2^1)	
[0]	Record select (2^0)	

4.12 STATUSBYTE (BPO-Typ 4)

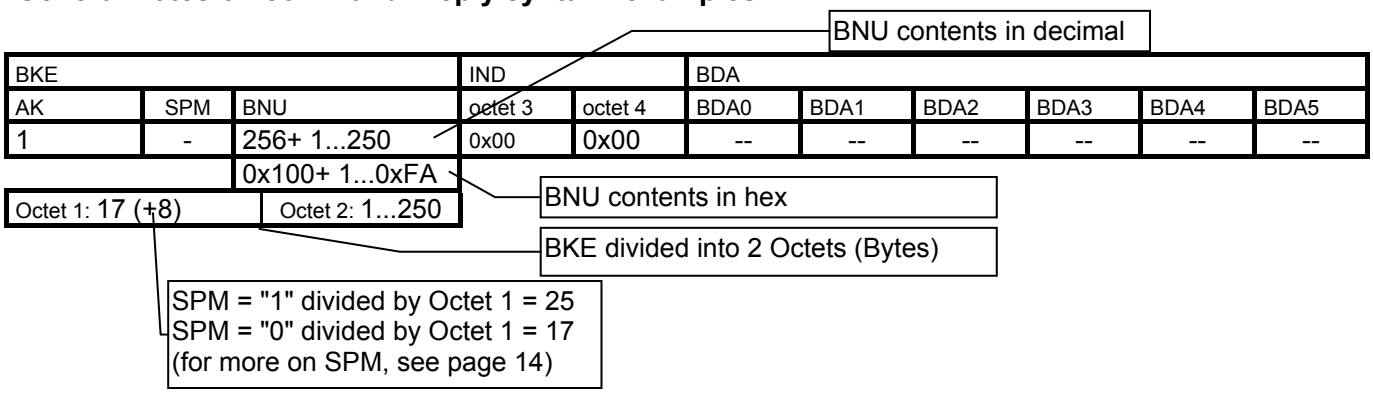
The status byte shows information about the status of the device as well as messages.

Data Description

Data byte [bit]		Significance
[7]	="1"	Machine was homed
[6]	="1"	Idle after stop
[5]	="1"	Programmed set point in the positioning window
[4]	="1"	Motor stalled
[3]	="1"	Contour error
[2]	="1"	Ready for Start
[1]	="1"	Warning or standstill indicator (Signal has the opposite meaning of A2 in the COMPAX)
[0]	="1"	Fault(Signal has the reverse meaning as A1 for the COMPAX)

4.13 BKD: Error Coding of the (BPO-Type 1 and 2)

General notes on command / reply syntax - examples



BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
1	-	256+ 1...250	0x00	0x00	--	--	--	--	--	--
		0x100+ 1...0xFA								
Octet 1: 17 (+8)		Octet 2: 1...250								

4.13.1 Requesting/changing COMPAX Parameters

Requesting COMPAX parameters

The parameter number is transferred with the command number.
The BNU is comprised of the offset 256 + parameter number.

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
1	-	256+ 1...250	0x00	0x00	--	--	--	--	--	--
		0x100+ 1...0xFA								
Octet 1: 17 (+8)		Octet 2: 1...250								

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
1	-	256+ 1...250	0x00	0x00	LSB		MSB	LSB		MSB
		0x100+ 1...0FA								
Octet 1: 17 (+8)		Octet 2: 1...250								

Change COMPAX-Parameter

As long as the corresponding password is enabled, parameters can be changed over the bus (password enable using command GOTO 302; see page 32).
The parameter number is transferred with the command number.
The BNU is comprised of the offset 1792 + parameter number..

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
2	-	1792+ 1...250	0x00	0x00	LSB		MSB	LSB		MSB
		0x700+ 1...0xFA								
Octet 1: 39 (+8)		Octet 2: 1...250								

Reply COMPAX→ Master

If the parameter was able to be overwritten, the COMPAX replies with the received command and AK=1.

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
1	-	1792+ 1...250	0x00	0x00	LSB		MSB	LSB		MSB
		0x700+ 1...0xFA								

Octet 1: 23 (+8)	Octet 2: 1...250

In case of error, COMPAX sends the error reply.

4.13.2 Request COMPAX status S1 - S110

The status number is transferred in the command number.
The BNU is comprised of the offset 256 + status number.

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
5	-	256+ 1...110	0x00	0x00	--	--	--	--	--	--
		0x100+ 1...0x6E								
Octet 1: 81 (+8)		Octet 2: 1...110								

The **status values 1 - 16, 31, 33 - 35, 37 - 39 and greater than 40** are sent in DSP data format.

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
4	-	256 + status no.	0x00	0x00	LSB		MSB	LSB		MSB
		0x100+ status no.								
Octet 1: 65 (+8)		Octet 2: 1...110								

In **Status 18** is a 6-level fault history. The value "99" indicates acknowledgement of the previous error.

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	274	0x00	0x00	current fault					
		0x112								
Octet 1: 65 (+8)		Octet 2: 18								

The**status 16, 17, 19-26, and 30** are byte values and are located in BDA0; the least significant bit always corresponds to bit 1 (see COMPAX Product Manual).

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	256 + status no.	0x00	0x00	octet 1					
		0x100+ status no.								
Octet 1: 65 (+8)		Octet 2: 1...110								

In **Status 32** is information pertaining to COMPAX software.

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	288	0x00	0x00	Day	Month	Year	Version		
		0x120								
Octet 1: 65 (+8)		Octet 2: 32								

In **Status 36** is information pertaining to Profibus software.

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	292	0x00	0x00	Day	Month	Year	Version		ID
		0x124								
Octet 1: 65 (+8)		Octet 2: 36								

4.13.3 COMPAX Commands

The reply to transmitted commands uses the reply ID AK=5; in all other respects it corresponds to the sent command.

VALID PARAMETER

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x56	0x50	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

VALID CONFIGURATION

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x56	0x43	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

VALID FIXPOINT

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	--	769	0x00	0x00	0x56	0x46	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

START

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x53	0x54	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

STOP

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x53	0x50	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

⇨ The reply COMPAX → Master corresponds to the sent command, but with AK=5.

STOP with ramp in P10

with idle: Command Master → COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x45	0x53	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

without idle: Command Master → COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x45	0x54	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

QUIT

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x51	0x54	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

TEACH ZERO

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x54	0x5A	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

BREAK

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x42	0x4B	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								

POSA HOME

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	769	0x00	0x00	0x50	0x48	--	--	--	--
		0x301								
Octet 1: 99 (+8)		Octet 2: 1								


The reply COMPAX → Master corresponds to the sent command, but with AK=5.

POSA

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1857	0x00	0x00	LSB		MSB	LSB		MSB
		0x741								
Octet 1: 103 (+8)		Octet 2: 65								

POSR

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1874	0x00	0x00	LSB		MSB	LSB		MSB
		0x752								
Octet 1: 103 (+8)		Octet 2: 82								

SPEED

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1875	0x00	0x00	LSB		MSB	LSB		MSB
		0x753								
Octet 1: 103 (+8)		Octet 2: 83								

POSRSPEED

Command Master→ COMPAX

1. POSR

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1874	0x00	0x01	LSB		MSB	LSB		MSB
		0x752								
Octet 1: 103 (+8)		Octet 2: 82								

2. SPEED

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1875	0x00	0x00	LSB		MSB	LSB		MSB
		0x753								
Octet 1: 103 (+8)		Octet 2: 83								

⇨ The reply COMPAX → Master corresponds to the sent command, but with AK=5.

POSR SPEED ACCEL

Command Master→ COMPAX

1. POSR

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1874	0x00	0x02	LSB		MSB	LSB		MSB
		0x752								
Octet 1: 103 (+8)		Octet 2: 82								

2. SPEED

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1875	0x00	0x01	LSB		MSB	LSB		MSB
		0x753								
Octet 1: 103 (+8)		Octet 2: 83								

3. ACCEL

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	844	0x00	0x00	MSB	LSB	--	--	--	--
		0x34C								
Octet 1: 99 (+8)		Octet 2: 76								

POSR0SPEED

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1923	0x00	0x00	LSB		MSB	LSB		MSB
		0x783								
Octet 1: 103 (+8)		Octet 2: 131								

WAITPOSA

1. WAIT: Command Master→ COMPAX **Special command for COMPAX XX50**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	855	0x00	0x01	0x00	0x00				
		0x357								
Octet 1: 99 (+8)		Octet 2: 87								

2. POSA: **Command Master→ COMPAX**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1857	0x00	0x00	LSB		MSB	LSB		MSB
		0x741								
Octet 1: 103 (+8)		Octet 2: 65								

⇨ The reply COMPAX → Master corresponds to the sent command, but with AK=5.

WAITPOSR

1. WAIT: Command Master→ COMPAX **Special command for COMPAX XX50**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	855	0x00	0x01	0x00	0x00				
		0x357								
Octet 1: 99 (+8)		Octet 2: 87								

2. POSR: **Command Master→ COMPAX**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1874	0x00	0x00	LSB		MSB	LSB		MSB
		0x752								
Octet 1: 103 (+8)		Octet 2: 82								

POSROUTPUT

Command Master→ COMPAX

1. POSR

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1874	0x00	0x01	LSB		MSB	LSB		MSB
		0x752								
Octet 1: 103 (+8)		Octet 2: 82								

2. OUTPUT x = y

x = 1 - 16 (output no.)
y = 0x30 for set output
y = 0x31 for reset output

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	1103	0x00	0x00	0x00	x	y	--	--	--
		0x44F								
Octet 1: 100 (+8)		Octet 2: 79								

ACCEL x

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	844	0x00	0x00	x MSB	x LSB	--	--	--	--
		0x34C								
Octet 1: 99 (+8)		Octet 2: 76								

⇨ The reply COMPAX → Master corresponds to the sent command, but with AK=5.

ACCEL- x

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	836	0x00	0x00	x MSB	x LSB	--	--	--	--
		0x344								
Octet 1: 99 (+8)		Octet 2: 68								

GOTO n

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	839	0x00	0x00	n MSB	n LSB	--	--	--	--
		0x347								
Octet 1: 99 (+8)		Octet 2: 71								

Password Enable

Parameters except for P40 ... P49 are password protected. There is no additional password protection.

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	839	0x00	0x00	0	302	--	--	--	--
		0x347								
Octet 1: 99 (+8)		Octet 2: 71								

START n

Run indicated record.

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	1281	0x00	0x00	0x53	0x4E	n MSB	n LSB	--	--
		0x501								
Octet 1: 101 (+8)		Octet 2: 1								

START n GO

Start program at indicated record.

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	1281	0x00	0x00	0x53	0x47	n MSB	n LSB	--	--
		0x501								
Octet 1: 101 (+8)		Octet 2: 1								

⇨ The reply COMPAX → Master corresponds to the sent command, but with AK=5.

TEACH n

Command Master ® COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	1281	0x00	0x00	0x54	0x4E	n MSB	n LSB	--	--
		0x501								
Octet 1: 101 (+8)		Octet 2: 1								

OUTPUT Ax = y

x = (1) 7 - 16 (output no.)
y = 0x31 for set output
y = 0x30 for reset output

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	1103	0x00	0x00	0x00	x	y	--	--	--
		0x44F								
Octet 1: 100 (+8)		Octet 2: 79								

OUTPUT A0 = y

y = 0x30 for "drive is under torque with opened brake"
y = 0x31 for "drive dead with closed brake"
y = 0x32 for "drive dead with opened brake"

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	1103	0x00	0x00	0x00	0x00	y	--	--	--
		0x44F								
Octet 1: 100 (+8)		Octet 2: 79								

SETC x

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	835	0x00	0x00	x MSB	x LSB	--	--	--	--
		0x343								
Octet 1: 99 (+8)		Octet 2: 67								

SETM

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1869	0x00	0x00	LSB		MSB	LSB		MSB
		0x74D								
Octet 1: 103 (+8)		Octet 2: 77								

⇨ The reply COMPAX → Master corresponds to the sent command, but with AK=5.

SETS

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1873	0x00	0x00	LSB		MSB	LSB		MSB
		0x751								
Octet 1: 103 (+8)		Octet 2: 81								

POSR CAM

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
6	-	1874	0x00	0x00	FF	FF	FF	FF	FF	FF
		0x752								
Octet 1: 103 (+8)		Octet 2: 82								

LOOP x

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
6	-	843	0x00	0x00	x MSB	x LSB	--	--	--	--
		0x34B								
Octet 1: 99 (+8)		Octet 2: 75								

COMPAX - Request Curve Memory

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					memory location number					
11	-	1091	0x00	0x00	MSB		LSB	-	-	-
		0x443								
Octet 1: 180 (+8)		Octet 2: 67								

Reply COMPAX → Master **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					memory location number			memory contents		
11	-	1091	0x00	0x00	MSB		LSB	MSB		LSB
		0x443								
Octet 1: 180 (+8)		Octet 2: 67								

⇨ The reply COMPAX → Master corresponds to the sent command, but with AK=5.

COMPAX - Change Curve Memory

Command Master→ COMPAX **Special command for COMPAX XX70**

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					memory location number			memory contents		
12	-	1859	0x00	0x00	MSB		LSB	MSB		LSB
		0x743								
Octet 1: 199 (+8)		Octet 2: 67								

The reply COMPAX → Master corresponds to the sent command, but with AK=5.

4.13.4 Request/Change COMPAX Variable

The variable number is transferred in the command number.
All COMPAX variables are transmitted in DSP number format.
The COMPAX variable with Index 0 (V0) is used to initialize all variable with the same value, i.e., if variable 0 is changed for example to value 10, all 39 variables are changed to this value.

Request COMPAX variable

The BNU is comprised of offset 256 + variable number.

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
9	-	256+ 0...39	0x00	0x00	--	--	--	--	--	--
		0x100+ 0...0x27								
Octet 1: 145 (+8)		Octet 2: 0...39								

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
9	-	256+ 0...39	0x00	0x00	LSB		MSB	LSB		MSB
		0x100+ 0...0x27								
Octet 1: 145 (+8)		Octet 2: 0...39								

Change COMPAX Variable

The BNU is comprised of offset 1792 + variable number.
The **reply** uses reply ID **AK=9**; in other respects it corresponds to the sent command.

Master → COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
10	-	1792+ 0...39	0x00	0x00	LSB		MSB	LSB		MSB
		0x700+ 0...0x27								
Octet 1: 167 (+8)		Octet 2: 0...39								

4.13.5 COMPAX - Special Commands

OUTPUT WORD

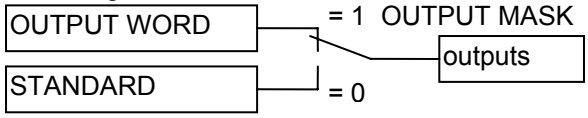
BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
8	-	1103	0x00	0x00	0xFF	MSB	LSB	--	--	--
		0x44F								

Octet 1: 132 (+8)	Octet 2: 79

Set logic state of the 16 digital outputs.
A fixed status information is assigned to some outputs; the free outputs are accessed through the command "OUTPUT Ax=y".

"OUTPUT WORD“ can be used to write the outputs only if they have been enabled using the "OUTPUT_MASK“ (FMS) object or the COMPAX parameters P223 (where P223=255).
The output thereby loses any status information which was assigned to it.

Note that access to the outputs is only possible selectively: either using "OUTPUT WORD", or "OUTPUT Ax=y" resp. the fixed status informations (standard).



Output	Standard configuration	Writeable as free outputs after being enabled by
1	no fault	Data byte 2 [0]
2	no warning	Data byte 2 [1]
3	Machine was homed	Data byte 2 [2]
4	Ready for Start	Data byte 2 [3]
5	Programmed target position reached	Data byte 2 [4]
6	Idle after stop	Data byte 2 [5]
7 ... 15	freely assignable in standard model	Data byte 2 [6]/[7]...Data byte 1 [0]...[6]
16		Data byte 1 [7]

This object can be used to set and/or reset the outputs.

The fixed assignments of outputs A9...A16 in other COMPAX versions (COMPAX XX30, ...) cannot be placed on the OUTPUT_WORD object.

4.13.6 Request/Change COMPAX Record

This requires one or more BPO-Write / BPO-Read depending on record contents.

Request COMPAX Record

The record number is sent in the command number (BNU).
The BNU is comprised of offset 256 + record number.
Depending on the record contents, COMPAX uses the frame number (IND Octet 4) to tell the master how many BPO reads still have to be executed in order to read out the entire record contents.

1. Record contents requires 1 BPO read

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
3	-	256 + Record-No.	0x00	0x00	--	--	--	--	--	--
		0x100+ record no.								

Octet 1: 49 (+8)	Octet 2: record no.

The reply consists of:

- the reply identifier AK=2,
- the command number (BNU) of the command which is in the requested record memory, and
- its associated command data.

The command number (BNU) consists of:

- the length of the command (no. of bytes) and
- the COMPAX command code (BC).
  - BNU (bits 10,9,8) = no. of bytes, and
  - BNU (bits 7..0) = command code.

BKE of the reply:

octet 1								octet 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				**SPM**	**BNU**										
2				**-**	**Length of relevant BDA data + 1**			**Command code (SD1)**							

Reply COMPAX → Master

BKE				IND		BDA					
AK	SPM	BNU		octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
2	-	Length	BC	0x00	0x00	command data					

The command data and the command code of a command are listet in the COMPAX record memory - command code table starting on page 42!

2. Record contents requires 2 or more BPO reads

The reply consists of:

- the reply identifier AK=2
- number of BPO reads (frame no.) required to receive the complete record contents.
- number of relevant record data in this BPO read (bits 10, 9 and 8 of the BNU).
- the**re**cord**d**ata SD1 to max. SD7 (SD1 = bits 7..0 of the BNU; SD2 ... SD7 = BDA0 ... BDA5).

The sequentially read relevant data are assembled in order.
The record contents can be interpreted with the help of the command code table provided.

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
3	-	256 + Record-No.	0x00	0x00	--	--	--	--	--	--
		0x100+ record no.								
Octet 1: 49 (+8)		Octet 2: Satz-Nr.								

Reply COMPAX → Master

BKE				IND		BDA					
AK	SPM	BNU		octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
		Number rec. data	rec.data SD1		Number BPO-rd	record data SD2	SD3	SD4	SD5	SD6	SD7
2	-			0x00							

Example: In record N20 is "IF P40 >= V10 GOTO 50"

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
3	-	276	0x00	0x00	--	--	--	--	--	--
		0x100+ 0x14								
Octet 1: 49 (+8)		Octet 2: 20								

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
2	-	1866	0x00	0x02	0x50	0x00	0x28	0x00	0x00	0x00
		0 x 7 4 A								
Octet 1: 39 (+8)		Octet 2: 74								

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
2	-	1792	0x00	0x01	0xBE	0x56	0x00	0x0A	0x00	0x00
		0 x 7 0 0								
Octet 1: 39 (+8)		Octet 2: 0								

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
2	-	1280	0x00	0x00	0x00	0x47	0x00	0x32	-	-
		0 x 5 0 0								
Octet 1: 37 (+8)		Octet 2: 0								

"7": length of relevant BDA data + 1

Still 2 BPO - reads

0xBE stands for comparison operator ">="

0x50 stands for the "Parameter" operand

0x28 stands for Parameter P40 (Parameter-No. Low Byte

"4A" stands for "IF"

0x56 stands for the "Variable" operand

0x0A stands for the variable no. 10

"5": length of relevant BDA data + 1

0x47 stands for GOTO

0x32 stands for 50 (GOTO **50**)

Data from the command code table (see starting page 42)

IF <Operand1> <Comparison Operator> <Operand2> **GOTO** n	19	7	0x4A	O1Type	O1D1	O1D2	O1D3	O1D4	O1D5
		7	O1D6	Vglop	O2Type	O2D1	O2D2	O2D3	O2D4

	5	O2D5	O2D6	0x47	n MSB	n LSB	--	--

Change COMPAX Record

- The record number is transferred with the command number. The BNU is comprised of the offset 512 + record number.
- Depending on the record contents, COMPAX uses the frame number (IND Octet 4) to tell the master how many BPO writes still have to be executed in order to write the entire record contents.

Reserved for future applications

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	512+ record-No.	0x00	Number of BPO-wr	0x00	--	--	--	--	--
		0x200+ record no.								
Octet 1: 68 (+8)		Octet 2: record no.								

Command Master → COMPAX (BPO - Write 1)

BKE				IND		BDA					
AK	SPM	BNU		octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
		Number	record data		Number	record data					
4	-	record data	SD1 (BC)	0x00	BPO-wr - 1	SD2	SD3	SD4	SD5	SD6	SD7

Command Master → COMPAX (BPO - Write 2)

BKE				IND		BDA					
AK	SPM	BNU		octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
		Number	record data		Number	record data					
4	-	record data	SD1 (BC)	0x00	BPO-wr - 2	SD2	SD3	SD4	SD5	SD6	SD7

...

The command data for a command are listed in the COMPAX record memory - command code table starting on page 42!

COMPAX - Reply

- The reply to the sent commands has the reply identifier AK=3; otherwise it corresponds to the sent process data.

Examples

Following are several examples for writing to the record memory.

1. POSR

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	512+ record-No.	0x00	0x01	0x00	--	--	--	--	--
		0x200+ record no.								
Octet 1: 66 (+8)		Octet 2: record no.								

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
4	-	1874	0x00	0x00	LSB		MSB	LSB		MSB
		0x752								
Octet 1: 71 (+8)		Octet 2: 82								

2. POSRSPEED

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	512+ record no.	0x00	0x02	0x00	--	--	--	--	--
		0x200+ record no.								
Octet 1: 66 (+8)		Octet 2: record no.								

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
4	-	1874	0x00	0x01	LSB		MSB	LSB		MSB
		0x752								
Octet 1: 71 (+8)		Octet 2: 82								

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
					decimal places			whole numbers		
4	-	1875	0x00	0x00	LSB		MSB	LSB		MSB
		0x753								
Octet 1: 71 (+8)		Octet 2: 83								

3. IF Ex = y GOTO n

y = 0x30 Input is logic 0; y = 0x31 Input is logic 1

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	512+ record no.	0x00	0x01	0x00	--	--	--	--	--
		0x200+ record no.								
Octet 1: 66 (+8)		Octet 2: record no.								

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
4	-	1865	0x00	0x00	0x00	x	y	0x47	n MSB	n LSB
		0x749								
Octet 1: 71 (+8)		Octet 2: 73								

COMPAX Command Codes

Definition of the command code: (A command code consists of 1 byte).

Sorted according to command code

Code	Command
0x20	Empty instruction (No Operation)
0x01	**VALIDP / C / F**
0x41	**POSA** Value / **POSA HOME**
0x42	**GOSUB**
0x43	**SETC** n
0x45	**END**
0x47	**GOTO**
0x49	**IF E**x**=**y ... / **IF ERROR ... / IF STOP ...**
0x4A	**IF <**Operand1**> <**Comparison Operator**> <**Operand2**> ...**
0x4B	**LOOP** n
0x4C	**ACCEL** Value
0x4D	**SETM** Value
0x4F	**OUTPUT A**x**=**y
0x50	**P**n**=. . .**
0x51	**SETS** Value
0x52	**POSR** Value / **POSR CAM**
0x53	**SPEED** Value / **SPEED SYNC**
0x54	**REPEAT** Value
0x55	**RETURN**
0x56	**V**n**=. . .**
0x57	**WAIT** Value / **WAIT START**
0x61	**POSA** Parameter
0x6B	**LOOP** Parameter
0x6C	**ACCEL** Parameter
0x6D	**SETM** Parameter
0x71	**SETS** Parameter
0x72	**POSR** Parameter
0x73	**SPEED** Parameter
0x74	**REPEAT** Parameter
0x77	**WAIT** Parameter
0xC1	**POSA** Variable
0xCB	**LOOP** Variable
0xCC	**ACCEL** Variable
0xCD	**SETM** Variable
0xD1	**SETS** Variable
0xD2	**POSR** Variable
0xD3	**SPEED** Variable
0x4D	**REPEAT** Variable
0xD7	**WAIT** Variable

Sorted by command

Code	Command
0x6C	**ACCEL** Parameter
0xCC	**ACCEL** Variable
0x4C	**ACCEL** Value
0x45	**END**
0x42	**GOSUB**
0x47	**GOTO**
0x4A	**IF <**Operand1**> <**Comparison Operator**> <**Operand2**> ...**
0x49	**IF E**x**=**y ... / **IF ERROR ... / IF STOP ...**
0x20	Empty instruction (No Operation)
0x4B	**LOOP** n
0x6B	**LOOP** Parameter
0xCB	**LOOP** Variable
0x4F	**OUTPUT A**x**=**y
0x50	**P**n**=. . .**
0x61	**POSA** Parameter
0xC1	**POSA** Variable
0x41	**POSA** Value / **POSA HOME**
0x72	**POSR** Parameter
0xD2	**POSR** Variable
0x52	**POSR** Value / **POSR CAM**
0x74	**REPEAT** Parameter
0x4D	**REPEAT** Variable
0x54	**REPEAT** Value
0x55	**RETURN**
0x43	**SETC** n
0x6D	**SETM** Parameter
0xCD	**SETM** Variable
0x4D	**SETM** Value
0x71	**SETS** Parameter
0xD1	**SETS** Variable
0x51	**SETS** Value
0x73	**SPEED** Parameter
0xD3	**SPEED** Variable
0x53	**SPEED** Value / **SPEED SYNC**
0x01	**VALIDP / C / F**
0x56	**V**n**=. . .**
0x77	**WAIT** Parameter
0xD7	**WAIT** Variable
0x57	**WAIT** Value / **WAIT START**

Definition of Operands

An operand consists of 7 bytes; 1 byte for the type indicator and 6 data bytes.

Operand	Type	D1	D2	D3	D4	D5	D6
Parameter	0x50	No.H	No.L	0x00	0x00	0x00	0x00
Status	0x53	No.H	No.L	0x00	0x00	0x00	0x00
Variable	0x56	No.H	No.L	0x00	0x00	0x00	0x00
Constants	0x20	NL	NM	NH	VL	VM	VH

Definition of Comparison Operators

A comparison operator consists of 1 byte.

Comparison operator	Symbols	Code
Equal	=	0x3D
Less than	<	0x3C
Greater than	>	0x3E
Equal to/less than	<=	0xBC
Equal to/greater than	>=	0xBE
Does not equal	<>	0xBB

Definition of Arithmetical Operators

An arithmetic operator consists of 1 byte.

Arithmetic Operator	Symbols	Code
Addition	+	0xB2
Subtraction	-	0x2D
Multiplication	*	0x2A
Division	/	0x2F
Whole number division	\	0x5C
Modulo calculation	%	0x25

Use of this coding results in the following record memory - command code table. All of the commands are listed individually here!

COMPAX Record Memory - Command Code Table

Command	GL	DL	SD1	SD2	SD3	SD4	SD5	SD6	SD7
ACCEL Parameter	3	3	0x6C	No.H	No.L	--	--	--	--
ACCEL Variable	3	3	0xCC	No.H	No.L	--	--	--	--
ACCEL value	3	3	0x4C	MSB	LSB	--	--	--	--
END	2	2	0x45	0x00	--	--	--	--	--
GOSUB EXT	3	3	0x42	0x00	0x00	--	--	--	--
GOSUB Value	3	3	0x42	MSB	LSB	--	--	--	--
GOTO EXT	3	3	0x47	0x00	0x00	--	--	--	--
GOTO Value	3	3	0x47	MSB	LSB	--	--	--	--
IF ERROR GOSUB n	7	7	0x49	0x00	0xFF	0x31	0x42	n MSB	n LSB
IF ERROR GOTO n	7	7	0x49	0x00	0xFF	0x31	0x47	n MSB	n LSB
IF Ex=y **GOSUB** n	7	7	0x49	x MSB	x LSB	y*	0x42	n MSB	n LSB
IF Ex=yy **GOSUB** n	8	7	0x49	x MSB	x LSB	y1*	y2*	0x42	n MSB
		1	n LSB	--	--	--	--	--	--
IF Ex=yyy **GOSUB** n	9	7	0x49	x MSB	x LSB	y1*	y2*	y3*	0x42
		2	n MSB	n LSB	--	--	--	--	--
IF Ex= . . .									
IF Ex=yyyyyyyy **GOSUB** n	14	7	0x49	x MSB	x LSB	y1*	y2*	y3*	y4*
		7	y5*	y6*	y7*	y8*	0x42	n MSB	n LSB
IF Ex=y **GOTO** n	7	7	0x49	x MSB	x LSB	y*	0x47	n MSB	n LSB
IF Ex=yy **GOTO** n	8	7	0x49	x MSB	x LSB	y1*	y2*	0x47	n MSB
		1	n LSB	--	--	--	--	--	--
IF Ex=yyy **GOTO** n	9	7	0x49	x MSB	x LSB	y1*	y2*	y3*	0x47
		2	n MSB	n LSB		--	--	--	--
IF Ex= . . .									
IF Ex=yyyyyyyy **GOTO** n	14	7	0x49	x MSB	x LSB	y1*	y2*	y3*	y4*
		7	y5*	y6*	y7*	y8*	0x47	n MSB	n LSB
IF <Operand1**> <**Comparison Operator**> <**Operand2**> GOTO** n	19	7	0x4A	O1Typ	O1D1	O1D2	O1D3	O1D4	O1D5
		7	O1D6	Vglop	O2Typ	O2D1	O2D2	O2D3	O2D4
		5	O2D5	O2D6	0x47	n MSB	n LSB	--	--
IF <Operand1**> <**Comparison Operator**> <**Operand2**> GOSUB** n	19	7	0x4A	O1Typ	O1D1	O1D2	O1D3	O1D4	O1D5
		7	O1D6	Vglop	O2Typ	O2D1	O2D2	O2D3	O2D4
		5	O2D5	O2D6	0x42	n MSB	n LSB	--	--
IF STOP GOSUB n	7	7	0x49	0x00	0xFE	0x31	0x42	n MSB	n LSB
IF STOP GOTO n	7	7	0x49	0x00	0xFE	0x31	0x47	n MSB	n LSB
LOOP n	3	3	0x4B	n MSB	n LSB	--	--	--	--
LOOP Parameter	3	3	0x6B	No.H	No.L	--	--	--	--
LOOP Variable	3	3	0xCB	No.H	No.L	--	--	--	--
OUTPUT Ax=y	4	4	0x4F	x MSB	x LSB	y*	--	--	--
OUTPUT Ax=yy	5	5	0x4F	x MSB	x LSB	y1*	y2*	--	--
OUTPUT Ax=. . .									
OUTPUT Ax=yyyyyyyy	11	7	0x4F	x MSB	x LSB	y1*	y2*	y3*	y4*
		4	y5*	y6*	y7*	y8*	--	--	--
OUTPUT A0=y	4	4	0x4F	0x00	0x00	y	--	--	--
POSA HOME	7	7	0x41	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
POSA Parameter	7	7	0x61	No.H	No.L	0x00	0x00	0x00	0x00
POSA Variable	7	7	0xC1	No.H	No.L	0x00	0x00	0x00	0x00
POSA Value	7	7	0x41	NL	NM	NH	VL	VM	VH
POSR CAM	7	7	0x52	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
POSR Parameter	7	7	0x72	No.H	No.L	0x00	0x00	0x00	0x00
POSR Variable	7	7	0xD2	No.H	No.L	0x00	0x00	0x00	0x00
POSR Value	7	7	0x52	NL	NM	NH	VL	VM	VH
REPEAT Parameter	3	3	0x74	No.H	No.L	--	--	--	--
REPEAT Variable	3	3	0xD4	No.H	No.L	--	--	--	--
REPEAT Value	3	3	0x54	MSB	LSB	--	--	--	--
RETURN	2	2	0x55	0x00	--	--	--	--	--
SETC n	3	3	0x43	n MSB	n LSB	--	--	--	--

Command	GL	DL	SD1	SD2	SD3	SD4	SD5	SD6	SD7
SETM Value	7	7	0x4D	NL	NM	NH	VL	VM	VH
SETM Parameter	7	7	0x6D	No.H	No.L	0x00	0x00	0x00	0x00
SETM Variable	7	7	0xCD	No.H	No.L	0x00	0x00	0x00	0x00
SETS Value	7	7	0x51	NL	NM	NH	VL	VM	VH
SETS Parameter	7	7	0x71	No.H	No.L	0x00	0x00	0x00	0x00
SETS Variable	7	7	0xD1	No.H	No.L	0x00	0x00	0x00	0x00
SPEED Parameter	7	7	0x73	No.H	No.L	0x00	0x00	0x00	0x00
SPEED Variable	7	7	0xD3	No.H	No.L	0x00	0x00	0x00	0x00
SPEED Value	7	7	0x53	NL	NM	NH	VL	VM	VH
SPEED SYNC	7	7	0x53	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
VALIDP	3	3	0x01	0x56	0x50	--	--	--	--
VALIDC	3	3	0x01	0x56	0x43	--	--	--	--
VALIDF	3	3	0x01	0x56	0x46	--	--	--	--
WAIT Parameter	3	3	0x77	No.H	No.L	--	--	--	--
WAIT Variable	3	3	0xD7	No.H	No.L	--	--	--	--
WAIT Value	3	3	0x57	MSB	LSB	--	--	--	--
WAIT START	3	3	0x57	0x00	0x00	--	--	--	--
POSA Value **WAIT** Value	10	7	0x41	NL	NM	NH	VL	VM	VH
		3	0x57	MSB	LSB	--	--	--	--
POSA . . . **WAIT** . . .	10	7	...	...	...	...	...	...	...
		3	...	...	...	...	...	...	...
POSA Variable **WAIT** Variable	10	7	0xC1	No.H	No.L	0x00	0x00	0x00	0x00
		3	0xD7	No.H	No.L	--	--	--	--
POSR Value **OUTPUT A**x**=**y	11	7	0x52	NL	NM	NH	VL	VM	VH
		4	0x4F	x MSB	x LSB	y*	--	--	--
POSR Parameter **OUTPUT A**x**=**y	11	7	0x72	No.H	No.L	0x00	0x00	0x00	0x00
		4	0x4F	x MSB	x LSB	y	--	--	--
POSR Variable **OUTPUT A**x**=**y	11	7	0xD2	No.H	No.L	0x00	0x00	0x00	0x00
		4	0x4F	x MSB	x LSB	y*	--	--	--
POSR Value **SPEED** Value	14	7	0x52	NL	NM	NH	VL	VM	VH
		7	0x53	NL	NM	NH	VL	VM	VH
POSR . . . **SPEED** . . .	14	7	...	...	...	...	...	...	...
		7	...	...	...	...	...	...	...
POSR Variable **SPEED** Variable	14	7	0xD2	No.H	No.L	0x00	0x00	0x00	0x00
		7	0xD3	No.H	No.L	0x00	0x00	0x00	0x00
POSR Value **SPEED** Value **ACCEL** Value	17	7	0x52	NL	NM	NH	VL	VM	VH
		7	0x53	NL	NM	NH	VL	VM	VH
		3	0x4C	MSB	LSB	--	--	--	--
POSR ... **SPEED** ... **ACCEL** ...	14	7	...	...	...	...	...	...	...
		7	...	...	...	...	...	...	...
		3	...	...	...	...	...	...	...
POSR Variable **SPEED** Variable **ACCEL** Variable	14	7	0xD2	No.H	No.L	0x00	0x00	0x00	0x00
		7	0xD3	No.H	No.L	0x00	0x00	0x00	0x00
		3	0xCC	No.H	No.L	--	--	--	--
POSR Value **WAIT** Value	10	7	0x52	NL	NM	NH	VL	VM	VH
		3	0x57	MSB	LSB	--	--	--	--
POSR . . . **WAIT** . . .	10	7	...	...	...	...	...	...	...
		3	...	...	...	...	...	...	...
POSR Variable **WAIT** Variable	10	7	0xD2	No.H	No.L	0x00	0x00	0x00	0x00
		3	0xD7	No.H	No.L	--	--	--	--
SPEED Value **WAIT** Value	10	7	0x53	NL	NM	NH	VL	VM	VH
		3	0x57	MSB	LSB	--	--	--	--
SPEED . . . **WAIT** . . .	10	7	...	...	...	...	...	...	...
		3	...	...	...	...	...	...	...
SPEED Variable **WAIT** Variable	10	7	0xD3	No.H	No.L	0x00	0x00	0x00	0x00
		3	0xD7	No.H	No.L	--	--	--	--
WAIT POSA Value	10	7	0x57	0x00	0x00	0x41	NL	NM	NH
		3	VL	VM	VH	--	--	--	--
WAIT POSR Value	10	7	0x57	0x00	0x00	0x52	NL	NM	NH
		3	VL	VM	VH	--	--	--	--

Command	GL	DL	SD1	SD2	SD3	SD4	SD5	SD6	SD7
Pn=**<**Operand1**> [<**Arithmetic Operator**> <**Operand2**>]**	18	7	0x50	n MSB	n LSB	O1Typ	O1D1	O1D2	O1D3
		7	O1D4	O1D5	O1D6	AriOp	O2Typ	O2D1	O2D2
		4	O2D3	O2D4	O2D5	O2D6	--	--	--
Vn=**<**Operand1**> [<**Arithmetic Operator**> <**Operand2**>]**	18	7	0x56	n MSB	n LSB	O1Typ	O1D1	O1D2	O1D3
		7	O1D4	O1D5	O1D6	AriOp	O2Typ	O2D1	O2D2
		4	O2D3	O2D4	O2D5	O2D6	--	--	--

Key:

GL: Total length (number of bytes for the complete command)
DL: Data length (number of relevant bytes (record data) in this BPO read/write
SD1...SD7: Record data (SD1 of the 1st BPO corresponds to the command code)
Nr.H: High Byte of the parameter/variable number
Nr.L: Low Byte of the parameter/variable number
MSB: High Byte of an integer value
LSB: Low Byte of an integer value
NL: Low Byte of the decimal place of a value in DSP number format
NM: Mid Byte of the decimal place of a value in DSP number format
NH: High Byte of the decimal place of a value in DSP number format
VL: Low Byte of the decimal place of a value in DSP number format
VM: Mid Byte of the decimal place of a value in DSP number format
VH: High Byte of the decimal place of a value in DSP number format
O1Typ: Type indicator of the 1st operand
O1D1...O1D6: Data of the 1st operand
O2Typ: Type indicator of the 2nd operand
O2D1...O2D6: Data of the 2nd operand
Vglop: Comparison operator
AriOp: Arithmetic operator
***y (y1, y2, ...)** y=0x30 for high (y="1"); y=0x31 for low (y="0")

4.13.7 Request/Change COMPAX Objects

Any COMPAX object in the FMS object directory whose length is not more than 6 bytes can be requested or changed, as long as there are access rights for the object.
The object number (Index) is transferred with the command number.
The BNU is comprised of the Object Index - 4000.
If necessary the subindex must be entered in octet 3.

COMPAX - FMS - Request Object

Command Master→ COMPAX

Subindex

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
15	-	Index - 4000	0x00	0x00	--	--	--	--	--	--

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
15	-	Index - 4000	0x00	0x00	Object data					

Change COMPAX - FMS Object

Command Master→ COMPAX

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
14	-	Index - 4000	0x00	0x00	Object data					

Reply COMPAX → Master

If the object was able to be overwritten, COMPAX replies with the received command.
In case of error, COMPAX sends the error reply.

Important FMS objects for DP mode

The following FMS objects are needed for working with the process data (cyclic channel) in DP mode:

- To allocate the process output data (PAD): PA_SELECT, PA_ENABLE, PAD_INI.
- To allocate the process input data (PED): PE_SELECT, PED_INI.
- To enable the STEUERWORT: INPUT_MASK
- To enable the command OUTPUT WORD: OUTPUT_MASK

Also important are the objects which you can place on the process output or the process input data channel. Format and units for these objects can be found under the corresponding object description.

Command	Symbol	Service	Index	Subind.	Byte	PD	see page
Objects for allocating process input and process output data							
Select PE data	PE_SELECT	rd/wr	4849	0 ... 13	19	-	88
Select PA data	PA_SELECT	rd/wr	4850	0 ... 13	19	-	89
Enable PA data	PA_ENABLE	rd/wr	4851	0	1	-	90
Initialization of the PE data write	PED_INI	rd/wr	4852	0 ... 4	3	-	91
Initialization of the PA data write	PAD_INI	rd/wr	4853	0 ... 4	3	-	92
Objects for enabling the "CONTROL WORD" and "OUTPUT WORD"							
Mask inputs	INPUT_MASK	rd/wr	4830	0	2	-	81
Mask outputs	OUTPUT_MASK	rd/wr	4833	0	2	-	83

Objects which you can place on the process output or process input data channel							
Position actual value	LAGE_IST	rd	4801	0	4	I	75
Contour error	S3	rd	4802	0	2	I	75
Current traverse speed	S4	rd	4803	0	2	I	77
Current motor torque	S5	rd	4805	0	2	I	69
Control byte	STEUERBYTE	rd/wr	4812	0	1	O	55
Status byte	STATUSBYTE	rd	4813	0	1	I	53
Control commands	CONTROL	wr	4816	0	1	O	55
Define target position	LAGE_ZIEL	rd/wr	4822	0	4	O	73
Traverse speed	SPEED	rd/wr	4823	0	2	O	76
Reduce traverse speed	OVERRIDE	rd/wr	4826	0	1	O	77
Logic state of the 16 digital inputs	INPUT_WORD	rd	4829	0	2	I	81
Logic state of the 16 digital outputs	OUTPUT_WORD	rd/wr	4832	0	2	I/O	82
Run program record n	START_N	wr	4844	0	1	O	84
Start program at record N	START_N_GO	wr	4845	0	1	O	84
COMPAX - control word	CPX_STW	rd/wr	4876	0	2	A	55
COMPAX - status word	CPX_ZSW	rd	4877	0	2	E	56

All objects listed in the object directory are accessible in this mode through the READ/WRITE services. Definitive for the allocation of the object directory are those device functions of the connected Profibus participant which are to be implemented.

4.13.8 Command cannot be executed

Reply COMPAX → Master

BKE			IND		BDA					
AK	SPM	BNU	octet 3	octet 4	BDA0	BDA1	BDA2	BDA3	BDA4	BDA5
7	-	--	0x00	0x00	F.-Nr.	--	--	--	--	--
Octet 1: 112 (+8)		Octet 2: 0								

5. FMS Object Directory

5.1 Communications Objects: Overview sorted by symbol

Command	Symbol	Service	Index	Subind.	Byte	PD	see page
Acceleration/deceleration time	ACCEL	rd/wr	4828	0 ... 2	2	-	80
BPO-Read Type 1	BPO_R_T1	rd	4847	0	18	-	62
BPO-Write Type 1	BPO_W_T1	wr	4848	0	18	-	68
Commands for COMPAX XX70	CAM_CMD	wr	4872	0	5	-	86
Read and write the curve memory.	CAM_MEM	rd/wr	4874	0	3	-	87
Set and read curve memory pointer.	CAM_MEM_P	rd/wr	4873	0	2	-	86
Command input in ASCII-Format	COMMAND	rd/wr	4817	0	20	-	57
Control commands	CONTROL	wr	4816	0	1	O	55
COMPAX - control word	CPX_STW	rd/wr	4876	0	2	A	55
COMPAX - status word	CPX_ZSW	rd	4877	0	2	E	56
Traverse speed-Actual value	GESCHW_IST	rd	4804	0	6	-	79
Set and read record pointer	GOTO	rd/wr	4843	0	1	-	84
Disable/enable automatic incrementing of PZ, VZ or NZ when accessing objects PX_INC, VX_INC or NX_INC.	INC_DISABLE	rd/wr	4864	0	1	-	67
Mask inputs	INPUT_MASK	rd/wr	4830	0	2	-	81
Logic state of the 16 digital inputs	INPUT_WORD	rd	4829	0	2	I	81
Actual position value	LAGE_IST	rd	4801	0	4	I	75
Target position default	LAGE_ZIEL	rd/wr	4822	0	4	O	73
Write or read out COMPAX record memory in binary format.	NX_INC	rd/wr	4863	0	20	-	85
Set or read COMPAX record memory pointer.	NZ	rd/wr	4862	0	1	-	85
Set or reset O digital output	OUTPUT	wr	4831	1 ... 16	1	-	82
Mask outputs	OUTPUT_MASK	rd/wr	4833	0	2	-	83
Logic state of the 16 digital outputs	OUTPUT_WORD	rd/wr	4832	0	2	I/O	82
Reduce traverse speed	OVERRIDE	rd/wr	4826	0	1	O	77
Parameter 1-30 read/write	P1_P30	rd/wr	4835	1 ... 30	6	-	58
Parameter 121-150 read/write	P121_P150	rd/wr	4839	1 ... 30	6	-	60
Parameter 151-180 read/write	P151_P180	rd/wr	4840	1 ... 30	6	-	61
Parameter 181-200 read/write	P181_P200	rd/wr	4841	1 ... 20	6	-	61
Parameter 201-250 read/write	P201_P250	rd/wr	4842	1 ... 50	1	-	62
Parameter 31-60 read/write	P31_P60	rd/wr	4836	1 ... 30	6	-	58
COMPAX-Change or read parameter 35	P35	rd/wr	4865	0	4	-	58
COMPAX-Change or read parameter 36	P36	rd/wr	4866	0	4	-	58
Parameter 61-90 read/write	P61_P90	rd/wr	4837	1 ... 30	6	-	59
Parameter 91-120 read/write	P91_P120	rd/wr	4838	1 ... 30	6	-	60
Enable PA data	PA_ENABLE	rd/wr	4851	0	1	-	90
Select PA data	PA_SELECT	rd/wr	4850	0 ... 13	19	-	89
Initialize writing of PA data	PAD_INI	rd/wr	4853	0 ... 4	3	-	92
Select PE data	PE_SELECT	rd/wr	4849	0 ... 13	19	-	88
Initialize writing of PE data	PED_INI	rd/wr	4852	0 ... 4	3	-	91
Absolute positioning	POSA	wr	4818	0	6	-	72
Relative positioning	POSR	wr	4819	0	6	-	72
Change traverse speed	POSR0SPEED	wr	4825	0	6	-	76
Comparator function	POSROUTPUT	wr	4834	0	8	-	83
Speed step profile	POSRXSPEEDY	wr	4827	0	12	-	77
Speed step profile	PRXSDYALZ	wr	4875	0	14	-	78
Read or change COMPAX parameter	PX_I32_INC	rd/wr	4858	0	4	-	64
Read or change COMPAX parameter	PX_INC	rd/wr	4857	0	6	-	63
Read or set COMPAX parameter pointer.	PZ	rd/wr	4856	0	1	-	63
Actual position, target position and absolute encoder	S1_S2_S12	rd	4800	0 ... 3	6	-	74
COMPAX run hours	S10	rd	4809	0	6	-	70
Loop counter of a running REPEAT loop	S11	rd	4810	0	2	-	70
Lag error	S3	rd	4802	0	2	I	75
Error message	S30	rd	4811	0 ... 2	1	-	71

Command	Symbol	Service	Index	Subind.	Byte	PD	see page
Current traverse speed	S4	rd	4803	0	2	I	77
Current motor torque	S5	rd	4805	0	2	I	69
Temperature of the power final stage	S6	rd	4806	0	2	-	69
Control voltage and intermediate circuit voltage	S7_S8	rd	4807	0 ... 2	2	-	69
No. of axis motion cycles	S9	rd	4808	0	6	-	70
Traverse speed	SPEED	rd/wr	4823	0	2	O	76
Execute program record N	START_N	wr	4844	0	1	O	84
Program start beginning at record N	START_N_GO	wr	4845	0	1	O	84
Status byte	STATUSBYTE	rd	4813	0	1	I	53
Status word	STATUSWORD	rd	4815	0	2	-	55
Control byte	STEUERBYTE	rd/wr	4812	0	1	O	53
Control word	STEUERWORT	rd/wr	4814	0	2	-	54
Read current position in record N	TEACH_N	wr	4846	0	1	-	85
Traverse speed	VERF_GESCHW	rd/wr	4824	0	6	-	76
Change or read COMPAX-Variable.	VX_I32_INC	rd/wr	4871	0	4	-	66
Change or read COMPAX-Variable.	VX_INC	rd/wr	4861	0	6	-	66
Read or set COMPAX variable pointer.	VZ	rd/wr	4860	0	1	-	66
Synchronization with automatic reverse travel	WAITPOSA	wr	4820	0	6	-	72
Synchronization without automatic reverse travel	WAITPOSR	wr	4821	0	6	-	73
Change COMPAX parameter.	WR_PX	wr	4854	0	7	-	62
Change COMPAX parameter.	WR_PX_I32	wr	4855	0	5	-	62
Change COMPAX variable.	WR_VX	wr	4859	0	7	-	65
Change COMPAX variable.	WR_VX_I32	wr	4870	0	5	-	65

The information in column PD is coded as follows:

I Object can be mapped on the process input data.
O Object can be mapped on the process output data.
I/O Object can be mapped on the process in- and output data.
- no representation possible on the process data channel.

5.2 Communikations objectes: Overview sorted by Index

Command	Symbol	Service	Index	Subind.	Byte	PD	see page
Actual position, target position and absolute encoder	S1_S2_S12	rd	4800	0 ... 3	6	-	74
Actual position value	LAGE_IST	rd	4801	0	4	I	75
Lag error	S3	rd	4802	0	2	I	75
Current traverse speed	S4	rd	4803	0	2	I	77
Traverse speed-Actual value	GESCHW_IST	rd	4804	0	6	-	79
Current motor torque	S5	rd	4805	0	2	I	69
Temperature of the power final stage	S6	rd	4806	0	2	-	69
Control voltage and intermediate circuit voltage	S7_S8	rd	4807	0 ... 2	2	-	69
No. of axis motion cycles	S9	rd	4808	0	6	-	70
COMPAX run hours	S10	rd	4809	0	6	-	70
Loop counter of a running REPEAT loop	S11	rd	4810	0	2	-	70
Error message	S30	rd	4811	0 ... 2	1	-	71
Control byte	STEUERBYTE	rd/wr	4812	0	1	O	53
Status byte	STATUSBYTE	rd	4813	0	1	I	53
Control word	STEUERWORT	rd/wr	4814	0	2	-	54
Status word	STATUSWORD	rd	4815	0	2	-	55
Control commands	CONTROL	wr	4816	0	1	O	55
Command input in ASCII-Format	COMMAND	rd/wr	4817	0	20	-	57
Absolute positioning	POSA	wr	4818	0	6	-	72
Relative positioning	POSR	wr	4819	0	6	-	72
Synchronization with automatic reverse travel	WAITPOSA	wr	4820	0	6	-	72
Synchronization without automatic reverse travel	WAITPOSR	wr	4821	0	6	-	73
Target position default	LAGE_ZIEL	rd/wr	4822	0	4	O	73
Traverse speed	SPEED	rd/wr	4823	0	2	O	76
Traverse speed	VERF_GESCHW	rd/wr	4824	0	6	-	76
Change traverse speed	POSR0SPEED	wr	4825	0	6	-	76
Reduce traverse speed	OVERRIDE	rd/wr	4826	0	1	O	77
Speed step profile	POSRXSPEEDY	wr	4827	0	12	-	77
Acceleration/deceleration time	ACCEL	rd/wr	4828	0 ... 2	2	-	80
Logic state of the 16 digital inputs	INPUT_WORD	rd	4829	0	2	I	81
Mask inputs	INPUT_MASK	rd/wr	4830	0	2	-	81
Set or reset O digital output	OUTPUT	wr	4831	1 ... 16	1	-	82
Logic state of the 16 digital outputs	OUTPUT_WORD	rd/wr	4832	0	2	I/O	82
Mask outputs	OUTPUT_MASK	rd/wr	4833	0	2	-	83
Comparator function	POSROUTPUT	wr	4834	0	8	-	83
Parameter 1-30 read/write	P1_P30	rd/wr	4835	1 ... 30	6	-	58
Parameter 31-60 read/write	P31_P60	rd/wr	4836	1 ... 30	6	-	58
Parameter 61-90 read/write	P61_P90	rd/wr	4837	1 ... 30	6	-	59
Parameter 91-120 read/write	P91_P120	rd/wr	4838	1 ... 30	6	-	60
Parameter 121-150 read/write	P121_P150	rd/wr	4839	1 ... 30	6	-	60
Parameter 151-180 read/write	P151_P180	rd/wr	4840	1 ... 30	6	-	61
Parameter 181-200 read/write	P181_P200	rd/wr	4841	1 ... 20	6	-	61
Parameter 201-250 read/write	P201_P250	rd/wr	4842	1 ... 50	1	-	62
Set and read record pointer	GOTO	rd/wr	4843	0	1	-	84
Execute program record N	START_N	wr	4844	0	1	O	84
Program start beginning at record N	START_N_GO	wr	4845	0	1	O	84
Read current position in record N	TEACH_N	wr	4846	0	1	-	85
BPO-Read Type 1	BPO_R_T1	rd	4847	0	18	-	62
BPO-Write Type 1	BPO_W_T1	wr	4848	0	18	-	68
Select PE data	PE_SELECT	rd/wr	4849	0 ... 13	19	-	88
Select PA data	PA_SELECT	rd/wr	4850	0 ... 13	19	-	89
Enable PA data	PA_ENABLE	rd/wr	4851	0	1	-	90
Initialize writing of PE data	PED_INI	rd/wr	4852	0 ... 4	3	-	91
Initialize writing of PA data	PAD_INI	rd/wr	4853	0 ... 4	3	-	92
Change COMPAX parameter.	WR_PX	wr	4854	0	7	-	62
Change COMPAX parameter.	WR_PX_I32	wr	4855	0	5	-	62
Read or set COMPAX parameter pointer.	PZ	rd/wr	4856	0	1	-	63
Read or change COMPAX parameter	PX_INC	rd/wr	4857	0	6	-	63
Read or change COMPAX parameter	PX_I32_INC	rd/wr	4858	0	4	-	64

Command	Symbol	Service	Index	Subind.	Byte	PD	see page
Change COMPAX variable.	WR_VX	wr	4859	0	7	-	65
Read or set COMPAX variable pointer.	VZ	rd/wr	4860	0	1	-	66
Change or read COMPAX-Variable.	VX_INC	rd/wr	4861	0	6	-	66
Set or read COMPAX record memory pointer.	NZ	rd/wr	4862	0	1	-	85
Write or read out COMPAX record memory in binary format.	NX_INC	rd/wr	4863	0	20	-	85
Disable/enable automatic incrementing of PZ, VZ or NZ when accessing objects PX_INC, VX_INC or NX_INC.	INC_DISABLE	rd/wr	4864	0	1	-	67
COMPAX-Change or read parameter 35	P35	rd/wr	4865	0	4	-	58
COMPAX-Change or read parameter 36	P36	rd/wr	4866	0	4	-	58
Change COMPAX variable.	WR_VX_I32	wr	4870	0	5	-	65
Change or read COMPAX-Variable.	VX_I32_INC	rd/wr	4871	0	4	-	66
Commands for COMPAX XX70	CAM_CMD	wr	4872	0	5	-	86
Set and read curve memory pointer.	CAM_MEM_P	rd/wr	4873	0	2	-	86
Read and write the curve memory.	CAM_MEM	rd/wr	4874	0	3	-	87
Speed step profile	PRXSDYALZ	wr	4875	0	14	-	78
COMPAX - control word	CPX_STW	rd/wr	4876	0	2	A	55
COMPAX - status word	CPX_ZSW	rd	4877	0	2	E	56

The information in column PD means the following:
I Object can be mapped on the process input data.
O Object can be mapped on the process output data.
I/O Object can be mapped on the process in- and output data.
- no representation possible on the process data channel.

5.3 Control

5.3.1 STEUERBYTE

Permits program start from records 1 - 15.
The record pointer is set to the corresponding program record.
The program can be started, stopped, and continued.

⇨ Record select ="0000" and "Start" homes the machine.

An error acknowledgement is possible.

Object Description

Index	4812				
Symbol	STEUERBYTE	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	read all/write all	**PD Map**	PAD

Data Description

Data byte [Bi]	Significance	Meaning
[7]	Acknowledge	Error acknowledgement with positive edge
[6]	Stop with ramp in P10 without de-energization	with a positive edge
[5]	Continue ("1") / New start ("0") for start	Continue: continue program. New start: Program start at selected record.
[4]	Start / Stop	Start ⎍ / Stop ⎍ Start after condition defined in bit 5
[3]	Record select (2^3)	**Note!**
[2]	Record select (2^2)	Record select ="0000" and "new start" causes
[1]	Record select (2^1)	the machine to home itself.
[0]	Record select (2^0)	

5.3.2 STATUSBYTE

The status byte shows information about the status of the device as well as messages.

Object Description

Index	4813				
Symbol	STATUSBYTE	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	PED

Data Description

Data byte [Bi]	Significance
[7]	Machine was homed
[6]	Idle after stop
[5]	Programmed target position reached
[4]	reserved
[3]	reserved
[2]	Ready for start (see below)
[1]	Warning
[0]	Fault

Meaning of "ready for start"

- "Ready for START" is used for program control.
- A4 is set,
  - if the program is at a WAIT START instruction and waiting for the START signal,
  - after an interruption with STOP or BREAK and these signals are no longer active,
  - after a corrected error condition and
  - after Power On.
  - at program end with the END command.
- A4 has no significance for direct command statement.

5.3.3 CONTROL WORD

Activates device control commands.
COMPAX can only be controlled with the control word if the corresponding bits for it are enabled by the INPUT-MASK object.

Object Description

Index	4814				
Symbol	Control word	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	PAD

Data Description

Data byte [Bit]	Significance	Corresponding input logic states	Function enable using "INPUT_MASK" (Data byte [Bit])
1 [7] MSB	-		
1 [6]	-		
1 [5]	New set point		
1 [4]	Position target relative or absolute (see page 73)		
1 [3]	Teach real null	E1 and E4 ="1"	Data byte 2[0]="1" and 2[3]="1"
1 [2]	AG guidance		
1 [1]	Hand-	E3="1"	Data byte 2[2]="1"
1 [0]	Hand+	E2="1"	Data byte 2[1]="1"
2 [7]	Acknowledge	E4="1"	Data byte 2[3]="1"
2 [6]	Stop	E6="1"	Data byte 2[5]="1"
2 [5]	Break	E1 and E6 ="1"	Data byte 2[0]="1" and 2[5]="1"
2 [4]	Find machine null	E1 and E2 ="1"	Data byte 2[0]="1" and 2[1]="1"
2 [3]	Find real null	E1 and E3 ="1"	Data byte 2[0]="1" and 2[2]="1"
2 [2]	Start	E5="1"	Data byte 2[1]="1"
2 [1]	Idle and brake open		
2 [0] LSB	Idle and brake closed		

⇨ By partially switching input functions to the STEUERWORT, the multi-function of E1 permits a function limitation: Example: If a function with E1 occupies the control word (e.g. teach real null), then additional E1 functions (such as the "QUIT" function) are ignored by the inputs.

Therefore: If you require all the input functions, the assignments must be completely changed over; either to the inputs or to the control word.

COMPAX - I/O - Functions using the Control Word (Data bits 1[1], 1[0], 2[7], ... 2[2])

Direct switching of the I/O functions by removing a function and simultaneously setting another function is not recognized by the COMPAX; **Exception:** STOP and BREAK (these are always recognized immediately). Therefore proceed as follows::

- Remove the previous functions (allow status to remain until it has been recognized by the COMPAX).
- Then set a new function.

Example: Switch from Hand+ to Hand-

- Reset Hand+: Data bit 1[0] = "0"
- Wait until the COMPAX has recognized it via the Profibus (or handshake via FMS - Object Control Word).
- Set Hand-: Data bit 1[1] = "1"

5.3.4 STATUSWORT

The status word shows information about the device status as well as messages.

Object Description

Index	4815				
Symbol	Status word	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	PED

Data Description

Data byte [bit]	Significance at ="1"	Data byte [bit]	Significance at ="1"
1 [7]	Motor stalled	2 [7]	Warning or stop indicator
1 [6]	Lag error	2 [6]	Idle after stop
1 [5]	Set Point Acknowledgement (Profibus)	2 [5]	-
1 [4]	Set Point Not Acknowledgement (COMPAX)	2 [4]	Machine was homed
1 [3]	-	2 [3]	Fault
1 [2]	I/O Function active*	2 [2]	Ready for Start
1 [1]	AG guided	2 [1]	-
1 [0]	Programmed target position reached	2 [0]	-

* active I/O-function: A COMPAX control signal becomes activated through the „STEUERWORT“ or „CPX STW“. When an I/O-function is activated no further I/O-functions are recognized from the COMPAX except STOP and BREAK. Therefore, send the next I/O-functions only when Data bit 1[2]=„0“
I/O-functions are all the control functions which can normally become activated with E1...E6.

5.3.5 CPX_STW

Activates device control commands and set/Reset the virtuell Inputs I17...I32
COMPAX can only be controlled with the CPX_STW if the corresponding bits for it are enabled by the bits in P221.

Object Description

Index	4876

Symbol	CPX_STW	Length	2	Access groups	0
Object-code	Simple-Var.			Password	0
Data type	Octet-String	Access rights	read/write all	PD Map	PAD

Data Description

Data byte [Bit]	Significance	Data byte [Bit]	Significance
1 [7..0]	I32...I25	2 [7..0]	I24...I17

Data byte [Bit]	Function without shift	Function with shift	Enable
1 [7..0]	-	-	P222/Bit 7 ... 0
2 [7..6]	-	-	P221/Bit 7 ... 6
2 [5]	STOP	BREAK	P221/Bit 5
2 [4]	START	-	P221/Bit 4
2 [3]	QUIT	Teach real zero	P221/Bit 3
2 [2]	Hand-	Go to real zero	P221/Bit 2
2 [1]	Hand+	machine home	P221/Bit 1
2 [0]	SHIFT		P221/Bit 0

By partially switching input functions to the STEUERWORT, the multi-function of E1 permits a function limitation: Example: If a function with E1 occupies the control word (e.g. teach real null), then additional E1 functions (such as the "QUIT" function) are ignored by the inputs.

Therefore: If you require all the input functions, the assignments must be completely changed over; either to the inputs or to the control word.

5.3.6 CPX_ZSW

The status word shows information about the device status as well as messages.
From COMPAX Software Version 3.64 or higher the status information S16 and S17 onto CPX-ZSW can be placed with parameter P203 Bit 0=1.

Object Description

Index	4877				
Symbol	CPX_ZSW	Length	2	Access groups	0
Object-code	Simple-Var.			Password	0
Data type	Octet-String	Access rights	read all	PD Map	PED

Data Description (P203 bit 0 = 0)

Data byte [Bit]	Significance	Data byte [Bit]	Significance
1 [7]	Status O16	2 [7]	Motor stalled
1 [6]	Status O15	2 [6]	Lag error
1 [5]	Status O14	2 [5]	Idle after stop
1 [4]	Status O13	2 [4]	target position reached
1 [3]	Status O12	2 [3]	Ready for Start
1 [2]	Status O11	2 [2]	Machine was homed
1 [1]	Status O10	2 [1]	no warning
1 [0]	Status O9	2 [0]	no error

Data Description (P203 Bit 0 = 1)

Data byte [Bit]	Significance	Data byte [Bit]	Significance
1 [7]	frei	**2 [7] 2 [6]** 0 0	**OUTPUT A0 = x** after OUTPUT A0 = 0
1 [6]	RUN (“0“= off or currentless with error)	0 1 1 0	after OUTPUT A0 = 1 after OUTPUT A0 = 2
1 [5]	-	2 [5]	Idle after stop
1 [4]	Stop with Input I6	2 [4]	target position reached
1 [3]	programm runs	2 [3]	Ready for Start
1 [2]	command aktive	2 [2]	Machine was homed
1 [1]	Service Password aktive	2 [1]	no warning
1 [0]	Passwort 302 aktive	2 [0]	no error

5.3.5 CONTROL

Control commands. COMPAX commands which need no additional values.
The desired values are activated by transferring the correspondig command number (1 ... 16).

Object Description

Index	4816				
Symbol	CONTROL	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	write all	**PD Map**	PAD

Data Description

Command no..	Function	Command no..	Function
1	Go to machine home	11	E-stop without idle
2	Start program	12 ... 15	no function
3	Stop program/positioning	16	Drive under torque with opened brake
4	Break off program/positioning	17	Drive dead with closed brake
5	Acknowledge error	18	Drive dead with opened brake
6	Accept current position as real null	19	Traverse speed from external encoder
7	Declare valid	20	Program jump via external inputs
8	Declare configuration valid	21	Deactivate password protection (GOTO 302)
9	Declare curve valid	22	Activate password protectin (GOTO 270)
10	E-stop with idle	23	Deactivate password protection (GOTO 620)

5.3.6 COMMAND

All COMPAX commands which exist for the RS232 interface can be transferred in plain text (as ASCII-string) using this object.

Object Description

Index	4817				
Symbol	COMMAND	**Length**	20	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Visible-String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Coding	ASCII	**Value range**	0x20 ... 0x7F

Data byte	Significance	Data byte	Significance
1	1. command string character	20	20. command string character

5.4 Edit parameter

5.4.1 P1_P30

Input or read out COMPAX parameter 1 to 30.
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex).

Object Description

Index	4835				
Symbol	P1_P30	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	30	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	P1	30	P30

5.4.2 P35

Change or read COMPAX parameter 35

Object Description

Index	4865				
Symbol	P35	**Length**	4	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer32	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	Integer32	**Resolution**	1 ⇔ 0.000001

5.4.3 P36

Change or read COMPAX parameter 36

Object Description

Index	4866				
Symbol	P36	**Length**	4	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer32	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	Integer32	**Resolution**	1 ⇔ 0.000001

5.4.4 P31_P60

Input or read out COMPAX parameters 31 to 60
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex+30).

Object Description

Index	4836				
Symbol	P31_P60	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	30	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	P31	30	P60

5.4.5 P61_P90

Input or read out COMPAX parameter 61 to 90
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex+60).

Object Description

Index	4837				
Symbol	P61_P90	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	30	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	P61	30	P90

5.4.6 P91_P120

Input or read out COMPAX parameter 91 to 120
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex+90).

Object Description

Index	4838				
Symbol	P91_P120	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	30	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	P91	30	P120

5.4.7 P121_P150

Input or read out COMPAX parameter 121 to 150
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex+120).

Object Description

Index	4839				
Symbol	P121_P150	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	30	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	P121	30	P150

5.4.8 P151_P180

Input or read out COMPAX parameter 151 to 180
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex+150).

Object Description

Index	4840				
Symbol	P151_P180	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	30	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	P151	30	P180

5.4.9 P181_P200

Input or read out COMPAX parameter 181 to 200
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex+180).

Object Description

Index	4841				
Symbol	P181_P200	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	20	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	P181	20	P200

5.4.10 P201_P250

Input or read out COMPAX parameter 201 to 250
The subindex is used to select the corresponding parameter (Parameter-No. = Subindex+200).

Object Description

Index	4842				
Symbol	P200_P250	**Length**	1	**Access groups**	0
Object code	Array	**Elements**	50	**Password**	0
Data type	Unsigned8	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Subindex	Significance	Subindex	Significance
1	P201	40	P250

5.4.11 WR_PX

Change COMPAX parameter.
Parameter no. 1 is transferred in the 1st data byte. (Parameter-N\o. = 1st data byte).
After access the parameter pointer (PZ) points to the changed parameter.

Object Description

Index	4854				
Symbol	WR_PX	**Length**	7	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data byte	Significance	Data byte	Significance
1	Parameter number	2 ... 7	Parameter value

Data byte	Significance	Data byte	Significance
2	Decimal place [LSB]	5	Whole number [LSB]
3	decimal places	6	whole numbers
4	Decimal place [MSB]	7	Whole number [MSB]

Data format	DSP	**Unit**	parameter-specific

5.4.12 WR_PX_I32

Change COMPAX parameter.
Parameter no. 1 is transferred in the 1st data byte. (Parameter-N\o. = 1st data byte).
After this object is accessed, the parameter pointer (PZ) points to the changed parameter.

Object Description

Index	4855				
Symbol	WR_PX_I32	**Length**	5	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data byte	Significance	Data byte	Significance
1	Parameter number	2 ... 5	Parameter value

Data byte	Significance	Data byte	Significance
2	MSB	5	LSB

Data format	Integer32	Unit	parameter-specific

Parameter	Resolution	Parameter	Resolution
001 .. 005	1 ⇔ 0.001	035 .. 036	1 ⇔ 0.000001
006 .. 010	1	037 .. 049	1 ⇔ 0.001
011 .. 016	1 ⇔ 0.001	050 .. 072	1
017 .. 020	1	073 .. 099	1 ⇔ 0.001
021 .. 022	1 ⇔ 0.000001	100 .. 186	1
023 .. 029	1	187 .. 196	1 ⇔ 0.001
030 .. 034	1 ⇔ 0.001	197 .. 250	1

5.4.13 PZ

Read or set COMPAX parameter pointer.
The parameter pointer is changed by the objects WR_PX, WR_PX_I32, PX_INC and PX_I32_INC.

Object Description

Index	4856				
Symbol	PZ	Length	1	Access groups	0
Object code	Simple-Var.			Password	0
Data type	Unsigned8	Access rights	read all/write all	PD Map	not possible

5.4.14 PX_INC

Change or read COMPAX parameter
The parameter number is defined by the current value of the parameter pointer (PZ).
The parameter pointer is automatically incremented after this object is accessed.
The object DISABLE_INC can be used to disable/enable automatic incrementing.

Object Description

Index	4857				
Symbol	PX_INC	Length	6	Access groups	0
Object code	Simple-Var.			Password	0
Data type	Octet String	Access rights	read all/write all	PD Map	not possible

Data Description

Data format	DSP	Unit	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.4.15 PX_I32_INC

Change or read COMPAX parameter
The parameter number is defined by the current value of the parameter pointer (PZ).
The parameter pointer is automatically incremented after this object is accessed.
The object DISABLE_INC can be used to disable/enable automatic incrementing.

Object Description

Index	4858				
Symbol	PX_I32_INC	**Length**	4	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer32	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	Integer32	**Unit**	parameter-specific
Parameter	**Resolution**	**Parameter**	**Resolution**
001 .. 005	1 ⇔ 0.001	035 .. 036	1 ⇔ 0.000001
006 .. 010	1	037 .. 049	1 ⇔ 0.001
011 .. 016	1 ⇔ 0.001	050 .. 072	1
017 .. 020	1	073 .. 099	1 ⇔ 0.001
021 .. 022	1 ⇔ 0.000001	100 .. 186	1
023 .. 029	1	187 .. 196	1 ⇔ 0.001
030 .. 034	1 ⇔ 0.001	197 .. 250	1

5.5 Edit variables

5.5.1 WR_VX

Change COMPAX variable.
The variable number is transferred in the 1st data byte. (Variable-No. = 1st data byte).
The variable pointer (VZ) points to the changed variable after this object is accessed.

Object Description

Index	4859				
Symbol	WR_VX	**Length**	7	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data byte	Significance
1	Variable number

Data byte	Significance	Data byte	Significance
2	Decimal place [LSB]	5	Whole number [LSB]
3	decimal places	6	whole numbers
4	Decimal place [MSB]	7	Whole number [MSB]

Data format	DSP	**Unit**	variable-specific

5.5.2 WR_VX_I32

Change COMPAX variable. The variable is transferred
in the 1st data byte. (Variable-No. = 1st data byte).
The variable pointer (VZ) points to the changed variable after this object is accessed.

Object Description

Index	4870				
Symbol	WR_VX_I32	**Length**	5	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data byte	Significance	Data byte	Significance
1	Variable number	2 ... 5	Variable value

Data byte	Significance	Data byte	Significance
2	MSB	5	LSB

Data format	Integer32	**Resolution**	1 ⇔ 0.001

5.5.3 VZ

Read or set COMPAX variable pointer.
The variable pointer is changed by the objects WR_VX, WR_VX_I32, VX_INC and VX_I32_INC.

Object Description

Index	4860				
Symbol	VZ	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	read all/write all	**PD Map**	not possible

5.5.4 VX_INC

Change or read COMPAX variable.
The variable number is defined by the current value of the variable pointer (VZ).
The variable pointer is automatically incremented after this object is accessed.
The object DISABLE_INC can be used to disable/enable automatic incrementing.

Object Description

Index	4861				
Symbol	VX_INC	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	parameter-specific

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.5.5 VX_I32_INC

Change or read COMPAX variable.
The variable number is defined by the current value of the variable pointer (VZ).
The variable point is autoamatically incremented after this object is accessed.
The object DISABLE_INC can be used to disable/enable automatic incrementing.

Object Description

Index	4871				
Symbol	VX_I32_INC	**Length**	4	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer32	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	Integer32	**Resolution**	1 ⇔ 0.001

5.5.6 INC_DISABLE

Disabling/enabling the automatic incrementing of PZ, VZ or NZ when accessing objects PX_INC, VX_INC or NX_INC.

Object Description

Index	4864				
Symbol	INC_DISABLE	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Bit	Assignment
7	no function
6	no function
5	no function
4	disable/enable. auto. increm. when accessing the object CAM_MEM_P
3	disable/enable automatic incrementing of TZ when accessing the object TX_INC
2	disable/enable automatic incrementing of NZ when accessing the object NX_INC
1	disable/enable automatic incrementing of VZ when accessing the object VX_INC
0	disable/enable automatic incrementing of PZ when accessing the object PX_INC

Bit	Function	Data byte	Function
= 0 (FALSE)	enable autom. incrementing	= 1 (TRUE)	autom. incrementing disabled

5.6 Access to BPO-Typ 1

5.6.1 BPO_R_T1

Read access to BPO Type 1

Object Description

Index	4847				
Symbol	BPO_R_T1	**Length**	18	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	not possible

Data Description

Data format	BIN	**Value range**	0x00 .... 0xFF

Data byte	Significance	Data byte	Significance
1	1. Octet of the BKD	11	1. Octet of the PZD
10	10. Octet of the BKD	18	8. Octet of the PZD

5.6.2 BPO_W_T1

Write access to the BPO Type 1

Object Description

Index	4848				
Symbol	BPO_W_T1	**Length**	18	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data format	BIN	**Value range**	0x00 .... 0xFF

Data byte	Significance	Data byte	Significance
1	1. Octet of the BKD	11	1. Octet of the PZD
10	10. Octet of the BKD	18	8. Octet of the PZD

5.7 Diagnosis

5.7.1 S5

Current motor torque.
Value in % of the rated torque.

Object Description

Index	4805				
Symbol	S5	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer16	**Access rights**	read all	**PD Map**	PED

Data Description

Unit	%	**Resolution**	$1 \Leftrightarrow {}^{1}/_{64}$%; (6400 ⇔ 100%)

5.7.2 S6

Temperature of the power final stage.

Object Description

Index	4806				
Symbol	S6	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned16	**Access rights**	read all	**PD Map**	not possible

Data Description

Unit	degrees Celsius	**Resolution**	$1 \Leftrightarrow {}^{1}/_{64}$ °C

5.7.3 S7_S8

Control voltage and power or intermediate circuit voltage.

Object Description

Index	4807				
Symbol	S7_S8	**Length**	2	**Access groups**	0
Object code	Array	**Elements**	2	**Password**	0
Data type	Unsigned16	**Access rights**	read all	**PD Map**	not possible

Data Description

Unit	Volt	**Resolution**	$1 \Leftrightarrow {}^{1}/_{64}$V

Subindex	Significance	Subindex	Significance
1	Control voltage	2	Intermediate circuit voltage

5.7.4 S9

Number of axis motion cycles.

Object Description

Index	4808				
Symbol	S9	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.7.5 S10

COMPAX run hours

Object Description

Index	4809				
Symbol	S10	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	h

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.7.6 S11

Loop counter of a running REPEAT loop.

Object Description

Index	4810				
Symbol	S11	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned16	**Access rights**	read all	**PD Map**	not possible

5.7.7 S30

Error message.
This object contains the error number of the current error and the last occuring error.
If the error number of the current error = 0, there is no error.

Object Description

Index	4811				
Symbol	S30	**Length**	1	**Access groups**	0
Object code	Array	**Elements**	2	**Password**	0
Data type	Unsigned8	**Access rights**	read all	**PD Map**	not possible

Data Description

Subindex	Significance	Subindex	Significance
1	Error number of current error	2	Error number of last error

5.8 Positioning

5.8.1 POSA

Absolute positioning: Reference point is real null (RN).
Positioning is done with the acceleration time (brake time) set by ACCELL-POS (ACCEL-NEG) and the velocity set by SPEED. If these values were not set, then valid are **substitute values:** SPEED: Parameter P002; ACCEL: Parameter P006

Object Description

Index	4818				
Symbol	POSA	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	mm (or inch)

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.8.2 POSR

Relative positioning: The reference point is the current position.
Positioning is done with the acceleration time (brake time) set by ACCELL-POS (ACCEL-NEG) and the velocity set by SPEED. If these values were not set, then valid are **substitute values:** SPEED: Parameter P002; ACCEL: Parameter P006

Object Description

Index	4819				
Symbol	POSR	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	mm (or inch)

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.8.3 WAITPOSA

Synchronisation: with automatic return travel (synchro-pulse command). Starting from the rest position of the drive, a complete synchronisation move is carried out. The value for this element is the processing status (when sawing, the length of the material).
Positioning is done with the acceleration time (brake time) set by ACCELL-POS (ACCEL-NEG) and the velocity set by SPEED. If these values were not set, then valid are **substitute values:** SPEED: Parameter P002; ACCEL: Parameter P006

Object Description

Index	4820				
Symbol	WAITPOSA	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	mm (or inch)

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.8.4 WAITPOSR

Synchronization: without automatic return travel (synchro-pulse command). Starting from the rest position of the drive, a complete synchronization move is carried out. The value for this element is the processing status (when sawing, the length of the material).

Positioning is done with the acceleration time (brake time) set by ACCELL-POS (ACCEL-NEG) and the velocity set by SPEED. If these values were not set, then valid are **substitute values:** SPEED: Parameter P002; ACCEL: Parameter P006

Object Description

Index	4821				
Symbol	WAITPOSR	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	mm (or inch)

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.8.5 LAGE_ZIEL

Set Point Preset.

Absolute positioning (reference point is real null (RN)): settable with data byte 1 bit 4 in the control word)see page 54).

Positioning is done with the acceleration time (brake time) set by ACCELL-POS (ACCEL-NEG) and the velocity set by SPEED. If these values were not set, then valid are **substitute values:** SPEED: Parameter P002; ACCEL: Parameter P006

Object Description

Index	4822				
Symbol	LAGE_Ziel	**Length**	4	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer32	**Access rights**	read all/write all	**PD Map**	PAD

Data Description

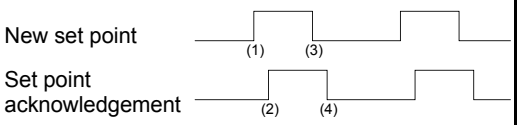
Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

Handshake for transferring PAD target position values

You can place the "target position" object on the cyclic process output data channel of the DP mode. Then you can cyclically specify new set points. Note that in DP mode, the data can only be newly handed over if the BPO has changed. For relative positioning this has as a consequence that identical target positions coming right after each other are not accepted. In this case, a handshake must be implemented for transferring the positions. This is done using the following bits:

- Control word byte 1 bit 5 "new set point" and
- Status word byte 1 bit 5 "set point acknowledgement"

Function:



Transition	Meaning	Condition
1	New set-point	Set-point acknowledgement ="0" Set-point can be sent
2	Set point acknow-ledgement	Set-point acknowledgement ="1" Set-point recognized
		New set-point ="0"
3	New set point	Set-point acknowledgement ="0" New set-point can be sent
4	Set point acknow-ledgement	

To ensure reliable establishment of a handshake using the FMS object "P__ENABLE" bit 7, the automatic transfer of a changed LAGE_ZIEL can be turned off (see page 90).

5.8.6 S1_S2_S12

Actual position: Current position referenced to real null.
Target position: Final position of the running or last carried out positioning cycle.
Absolute encoder: Position value of the absolute encoder (Option A1).

Object Description

Index	4800				
Symbol	S1_S2_S12	**Length**	6	**Access groups**	0
Object code	Array	**Elements**	3	**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	mm (or inch)

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Subindex	Significance	Subindex	Significance
1	Actual position	3	Absolute encor value
2	Target position		

5.8.7 LAGE_IST

Position actual value.
Current drive position.

Object Description

Index	4801				
Symbol	POS_ACTUAL	**Length**	4	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer32	**Access rights**	read all	**PD Map**	PED

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

5.8.8 S3

Contour Error.
Difference between set and actual position in a positioning cycle.

Object Description

Index	4802				
Symbol	S3	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer16	**Access rights**	read all	**PD Map**	PED

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ $^{1}/_{256}$ mm (or inch)

5.9 Speed

5.9.1 SPEED

Traverse speed in % of the nominal speed (nominal rpm * travel per motor revolution).
The value is valid until a new value is programmed.
The set speed can be reduced by using the OVERRIDE object.
A speed change during the positioning cycle is possible by using the POSR0SPEED object.

Object Description

Index	4823				
Symbol	SPEED	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer16	**Access rights**	read all/write all	**PD Map**	PAD

Data Description

Unit	%	**Resolution**	$1 \Leftrightarrow {}^{1}/_{64}\%$; (6400 ⇔ 100%)

5.9.2 VERF_GESCHW

Traverse speed.
Given in % of the nominal speed (nominal rpm * travel per motor revolution).
The value is valid until a new value is programmed.
The set speed can be reduced by using the OVERRIDE object.

Object Description

Index	4824				
Symbol	**VERF_GESCHW**	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	%

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.9.3 POSR0SPEED

Changing traverse speed during a positioning cycle.

Object Description

Index	4825				
Symbol	**POSR0SPEED**	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	DSP	Unit	%

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.9.4 OVERRIDE

Reduce traverse speed.
Software emulation of an external potentiometer on the override input (X11.6).

Object Description

Index	4826				
Symbol	OVERRIDE	Length	1	Access groups	0
Object code	Simple-Var.			Password	0
Data type	Unsigned8	Access rights	read all/write all	PD Map	PAD

Data Description

Unit	%	Resolution	$1 \Leftrightarrow {}^{1}/_{255}\%$; (255 ⇔ 100%)

5.9.5 POSRXSPEEDY

Speed Step Profile: Each speed (rpm) step profile can have a maximum of 8 speed steps. The position value is given as a relative measure. It is referenced to the positioning start point.

Object Description

Index	4827				
Symbol	POSRXSPEEDY	Length	12	Access groups	0
Object code	Simple-Var.			Password	0
Data type	Octet String	Access rights	write all	PD Map	not possible

Data Description

Position

Data format	DSP	Unit	mm (or inch)

Data byte	Significance	Data byte	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Speed

Data format	DSP	Unit	%

Data byte	Significance	Data byte [bit]	Significance
7	Decimal place [LSB]	10	Whole number [LSB]
8	decimal places	11	whole numbers
9	Decimal place [MSB]	12	Whole number [MSB]

5.9.6 PRXSDYALZ

Speed step profile: Each speed (rpm) step profile can have a maximum of 8 speed steps. The position value is given as a relative measure. It is referenced to the positioning start point.

Object Description

Index	4875				
Symbol	PRXSDYALZ	**Length**	14	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Position

Data format	DSP	**Unit**	mm (or inch)

Data byte	Significance	Data byte	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Speed

Data format	DSP	**Unit**	%

Data byte	Significance	Data byte [bit]	Significance
7	Decimal place [LSB]	10	Whole number [LSB]
8	decimal places	11	whole numbers
9	Decimal place [MSB]	12	Whole number [MSB]

Rampenzeit

Data format	Unsigned16	**Unit**	ms

Data byte	Significance	Data byte [bit]	Significance
13	MSB	14	LSB

5.9.7 S4

Current axis speed.
Value in % of the nominal speed (nominal rpm * travel per motor revolution).

Object Description

Index	4803				
Symbol	S4	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Integer16	**Access rights**	read all	**PD Map**	PED

Data Description

Unit	%	**Resolution**	$1 \Leftrightarrow {}^{1}/_{64}\%$; (6400 ⇔ 100%)

5.9.8 GESCHW_IST

Traverse Speed Actual Value.
Value in % of the nominal speed (nominal rpm * travel per motor revolution).

Object Description

Index	4804				
Symbol	GESCHW_IST	**Length**	6	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	%

Data byte	Significance	Data byte [bit]	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

5.10 Acceleration

5.10.1 ACCEL

Acceleration Time: Time specification for the acceleration process
Deceleration Time: Time specification for the deceleration process
The time specification for the acceleration process is by default also the time for the deceleration process unless the decel time is explicitly specified.

The time specification applies to nominal speed (100%). $ta = \frac{SPEED}{100\%} * ACCEL - POS$

Object Description

Index	4828				
Symbol	ACCEL	**Length**	2	**Access groups**	0
Object code	Array	**Elements**	2	**Password**	0
Data type	Unsigned16	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Value range	0 ... 65 000		
Unit	ms	**Resolution**	1 ⇔ 1 ms

Subindex	Significance	Subindex	Significance
1	Acceleration time	2	Deceleration time

5.11 Input/Outputs

5.11.1 INPUT_WORD

Logic state of the 16 digital inputs.
Some inputs have a fixed control function assigned, unless this was turned off with INPUT_MASK.

Input	Significance	Input	Significance
1	SHIFT	1 & 3	Find real null (RN)
2	Hand+	1 & 4	Teach real null
3	Hand-	1 & 5	reserved
4	Acknowledge	1 & 6	Break
5	Start	9 ... 16	freely assignable in standard model
6	Stop		
7 ... 8	Freely assignable in standard model		
1 & 2	Find machine home (MN)		

Object Description

Index	4829				
Symbol	INPUT_WORD	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all	**PD Map**	PED

Data Description

Data byte [Bit]	Significance	Data byte [bit]	Significance
1 [7]	Status input 16	2 [7]	Status input 8
1 [0]	Status input 9	2 [0]	Status input 1

5.11.2 INPUT_MASK

Mask Inputs.
Standard configuration is for functions assigned to the COMPAX inputs.
If the corresponding mask bit is set, access to this COMPAX input function is enabled with the STEUERWORT, while the corresponding COMPAX input loses this function and is available for other uses.
After Power On INPUT_MASK has a value of 0, i.e. all inputs have their standard functions and COMPAX cannot be controlled using the STEUERWORT.

Object Description

Index	4830				
Symbol	INPUT_MASK	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data byte [Bit]	Significance	Data byte [bit]	Significance
1 [7]	no function (="0")	2 [7]	no function (="0")
...	...	2 [6]	no function (="0")
...	...	2 [5]	Mask input 6
1 [0]	no function (="0")	2 [0]	Mask input 1

5.11.3 OUTPUT

Setting or resetting a digital output.
The corresponding output is selected using the subindex (Subindex = output no.).
Some outputs have a fixed status information assigned (see OUTPUT-WORD).
Any output which shall be affected through the Profibus must be explicitely enabled for this using the OUTPUT-MASK object (for assigning see below under OUTPUT_WORD). This causes the output to lose any assigned status information.

Object Description

Index	4831				
Symbol	OUTPUT	**Length**	1	**Access groups**	0
Object code	Array	**Elements**	16	**Password**	0
Data type	Boolean	**Access rights**	write all	**PD Map**	not possible

Data Description

Data byte	Function	Data byte	Function
= 0xFF (TRUE)	Output [Subindex] = 1	= 0x00 (FALSE)	Output [Subindex] = 0

5.11.4 OUTPUT_WORD

Logic state of the 16 digital outputs.
Some outputs are assigned a fixed status information.

Output	Assignment when reading the outputs	Writeable as free outputs after enabling with OUTPUT_MASK
1	no fault	Data byte 2 [0]
2	no warning	Data byte 2 [1]
3	Machine was homed	Data byte 2 [2]
4	Ready for Start	Data byte 2 [3]
5	Programmed target position reached	Data byte 2 [4]
6	Idle after stop	Data byte 2 [5]
7 ... 15	Freely assignable in standard model	Data byte 2 [6]/[7]...Data byte 1 [0]...[6]
16		Data byte 1 [7]

This object can be used to set and/or reset the outputs.
Each object to be activated over the Profibus must be enabled explicitely with the corresponding OUTPUT_MASK object. This causes the output to lose any assigned status information.

Object Description

Index	4832				
Symbol	**OUTPUT_WORD**	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	PED & PAD

Data Description

Data byte [Bit]	Significance	Data byte [bit]	Significance
1 [7]	Status output 16	2 [7]	Status output 8
1 [0]	Status output 9	2 [0]	Status output 1

5.11.5 OUTPUT_MASK

Mask outputs.
Any output which shall be affected through the Profibus must be explicitly enabled (masked) for this using the OUTPUT-MASK object.
This causes the output to lose any assigned status information.
After Power On, OUTPUT-MASK has a value of 0, i.e. all outputs are disabled for Profibus (not masked).

Object Description

Index	4833				
Symbol	OUTPUT_MASK	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data byte [Bit]	Significance	Data byte [bit]	Significance
1 [7]	Mask output 16	2 [7]	Mask output 8
1 [0]	Mask output 9	2 [0]	Mask output 1

⇨ The fixed assignments of outputs A9...A16 in other COMPAX versions (COMPAX XX30, ...) cannot be placed on the OUTPUT_WORD object.

5.11.6 POSROUTPUT

Comparator function.
Setting/resetting a non-assigned output during a positioning process.
The position value is given as a relative measure. It is referenced to the positioning start point.
A maximum of 4 comparators can be set for a positioning process.

Object Description

Index	4834				
Symbol	POSROUTPUT	**Length**	8	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data format	DSP	**Unit**	mm (or inch)

Data byte	Significance	Data byte	Significance
1	Decimal place [LSB]	4	Whole number [LSB]
2	decimal places	5	whole numbers
3	Decimal place [MSB]	6	Whole number [MSB]

Data byte	Significance	Data byte	Significance
7	Output number (1 ... 16)	8	1 = set output 0 = reset output

5.12 Programming

5.12.1 GOTO

Set and read record pointer.

Object Description

Index	4843				
Symbol	GOTO	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

5.12.2 START_N

Run program record N.
Only this record is processed. The record pointer remains at this program record.

Object Description

Index	4844				
Symbol	START_N	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	write all	**PD Map**	PAD

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

5.12.3 START_N_GO

Start program at record N.
The record pointer is set to the corresponding program record and then the program is started.

Object Description

Index	4845				
Symbol	START_N_GO	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	write all	**PD Map**	PAD

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

5.12.4 TEACH_N

Take over current position in record N.
The command "POSA *current position*" is stored in record N.

Object Description

Index	4846				
Symbol	TEACH_N	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	write all	**PD Map**	not possible

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

5.12.5 NZ

Read or set COMPAX record memory pointer.
The record memory is changed using the object NX_INC.

Object Description

Index	4862				
Symbol	NZ	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned8	**Access rights**	read all/write all	**PD Map**	not possible

5.12.6 NX_INC

Read from or write to the COMPAX record memory in binary format.
The record memory number is defined by the current value of the record memory pointer (NZ).
The record memory pointer is automatically incremented after this object is accessed.
The object DISABLE_INC can be used to disable/enable automatic incrementing.

Object Description

Index	4863				
Symbol	NX_INC	**Length**	20	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data byte	Significance
1	command length (Number of relevant bytes)
2	1. character of the record contents
...	
20	last character of the record contents

Coding for the record contents can be found on page 42.

5.13 COMPAX XX70 Commands

5.13.1 CAM_CMD

Special commands for COMPAX XX70.
With the contents of the 1st data byte the corresponding command is selected (1st data byte = CAM command).

Object Description

Index	0x4872				
Symbol	CAM_CMD	**Length**	5	**Access groups**	0
Object code	Simple-Var.	**Elements**		**Password**	0
Data type	Octet String	**Access rights**	write all	**PD Map**	not possible

Data Description

Data byte	Significance	Data byte	Significance
1	CAM command select	2 ... 5	Data for CAM command

Data byte	Significance	Data byte	Significance
2	MSB	5	LSB

Data byte 1	Command	Data format	Resolution
1	SETC	Integer32	1
2	SETM	Integer32	1 ⇔ 0.001
3	SETS	Integer32	1 ⇔ 0.001
4	POSR CAM	-	-
5	LOOP	Integer32	1
6	VF	-	-

5.13.2 CAM_MEM_P

Set and read curve memory pointer.

Object Description

Index	4873				
Symbol	CAM_MEM_P	**Length**	2	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Unsigned16	**Access rights**	read/write all	**PD Map**	not possible

Data Description

Data format	binary	**Unit**	Curve memory number
Value range	1 ... 5460	**Resolution**	1

5.13.3 CAM_MEM

Reading and writing the curve memory.
The curve memory number is defined by the current value of the curve memory pointer (CAM_MEM_P).
The curve memory pointer is automatically incremented after this object is accessed.
The object DISABLE_INC can be used to disable/enable automatic incrementing.

Object Description

Index	0x4874				
Symbol	CAM_MEM	**Length**	3	**Access Groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read/write all	**PD Map**	not possible

Data Description

Data byte	1	2	3
Meaning	Record memory contents		
Assignment	MSB	...	LSB

5.14 Process Data Control

These functions can be used to define the process data (PAD and PED) for DP mode (BPO-Type 1 or 3). The process data channel of the COMPAX is 6 bytes wide and can be assigned objects which permit process data emulation.
Each byte can be read and written by COMPAX.
Through the emulation of COMPAX communication objects to the PE data, the latter are cyclically read on the process data channel. The PA data which are emulated to a COMPAX communication object cyclically describe this object.

- Process output data (PA data): The data which COMPAX reads from the process data channel.
- Process input data (PE data): The data which COMPAX writes to the process data channel.

The assignment of process data to certain communication objects is determined by the objects "PE_SELECT" and "PA_SELECT".
The two objects "PED_INI" and "PAD_INI" determine which assignment shall be valid after Power On.
The PA data can be enabled and disabled with the "PA_ENABLE" object.
After Power On the PA data are enabled!

5.14.1 PE_SELECT

Process Input Data - Description.

This parameter contains the data that define which process input data are emulated on which communication objects. Communication objects which can be emulated on PED data are designated in the respective object descriptions.

Object Description

Index	4849				
Symbol	PE_SELECT			**Access groups**	0
Object code	Record			**Password**	0
Data type	PDB structure	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Subindex	Significance	Data type	Length
1	Length of the process data channel	Unsigned 8	1
2	Index of object which occupies the 1st PE data byte	Unsigned16	2
3	Subindex of object which occupies the 1st PE data byte	Unsigned 8	1
4	Index of object which occupies the 2nd PE data byte	Unsigned16	2
5	Subindex of object which occupies the 2nd PE data byte	Unsigned 8	1
6	Index of object which occupies the 3rd PE data byte	Unsigned16	2
7	Subindex of object which occupies the 3rd PE data byte	Unsigned 8	1
8	Index of object which occupies the 4th PE data byte	Unsigned16	2
9	Subindex of object which occupies the 4th PE data byte	Unsigned 8	1
10	Index of object which occupies the 5th PE data byte	Unsigned16	2
11	Subindex of object which occupies the 5th PE data byte	Unsigned 8	1
12	Index of object which occupies the 6th PE data byte	Unsigned16	2
13	Subindex of object which occupies the 6th PE data byte	Unsigned 8	1

Note the length of the object which you intend to place on the process input data.

The "INPUT_WORD" object has a length of 2 bytes. If you use subindex 2 to place this object on the process input data, then the first two bytes are thereby occupied.

With subindex 6 you can then place an additional object on the process input data starting with the 3rd PE data byte.

Arrangement of bit sizes represented using the above example ("INPUT_WORD" with subindex 2)

Data byte 1 = 1st PE-Data byte (Bit 7 = High-Bit: input 16)

Data byte 2 = 2. PE-Data byte (Bit 7 = High-Bit)

For 2- or 4- byte values the High-Byte is to the left.

5.14.2 PA_SELECT

Writing Process Output Data (Master → COMPAX).

This object is used to define which FMS objects are placed on the Process Output Data channel PAD. Communications objects which can be mapped on process output data are marked in the respective object descriptions (under the heading "PD mapping" or are listed on page 18.
Defining the PAD using PA-SELECT is done as follows:

- Using the COMPAX parameters P139, P140, P141 and P142.
  These parameters are used to define the configuration of the PAD after "Power on".
  The parameters are automatically written to the "PAD_INI" object and to the "PA-SELECT" objectS.
  COMPAX parameters P139, P140, P141 and P142 are handed over at "Power on".
- Directly by using the FMS object "PA-SELECT" (in FMS mode or over DP with the "Change COMPAX object" function) The FMS object "PA-SELECT" allows a change in the PAD configuration during operation.

  ⇨ After changing the PAD assignment using the FMS object "PA_SELECT", PA_ENABLE is set to "0" in order to avoid an undefined state. After a PAD change the PAD's must be manually enabled again using the PA_ENABLE object.

Object Description

Index	4850				
Symbol	PA_SELECT			**Access groups**	0
Object code	Record			**Password**	0
Data type	PDB structure	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Subindex	Significance	Data type	Length
1	Length of the process data channel	Unsigned 8	1
2	Index of object which occupies the 1st PA data byte	Unsigned16	2
3	Subindex of object which occupies the 1st PA data byte	Unsigned 8	1
4	Index of object which occupies the 2nd PA data byte	Unsigned16	2
5	Subindex of object which occupies the 2nd PA data byte	Unsigned 8	1
6	Index of object which occupies the 3rd PA data byte	Unsigned16	2
7	Subindex of object which occupies the 3rd PA data byte	Unsigned 8	1
8	Index of object which occupies the 4th PA data byte	Unsigned16	2
9	Subindex of object which occupies the 4th PA data byte	Unsigned 8	1
10	Index of object which occupies the 5th PA data byte	Unsigned16	2
11	Subindex of object which occupies the 5th PA data byte	Unsigned 8	1
12	Index of object which occupies the 6th PA data byte	Unsigned16	2
13	Subindex of object which occupies the 6th PA data byte	Unsigned 8	1

⇨ Note the length of the object which you intend to place on the process output data.

The "OUTPUT_WORD" object, for example, is 2 bytes long. If you place this object on the process output data using subindex 2, then the first two bytes are thereby occupied.

Subindex 6 can be used to then place an additional object on the process output data using the 3rd Place PI data bytes; Subindex 4 must then be set to "0".

Assignment of the bit sizes represented by the above example ("OUTPUT_WORD" with subindex 2)

Data byte 1 = 1st PI-Data byte (Bit 7 = High-Bit: output 16)

Data byte 2 = 2. PI-Data byte (Bit 7 = High-Bit)

For 2- or 4- byte values the High-Byte is to the left.

5.14.3 PA_ENABLE

Enable process output data.
Each bit of this parameter is associated with a byte of the process output data channel.
Meaning: Bit = 0 the corresponding process data value is disabled
Bit = 1 the corresponding process data value is enabled
If an object takes up several bytes on the PA data channel, the logic state of the bit which is associated with the first byte of this object is the one used, and the other associated bits are not relevant.

Object Description

Index	4851				
Symbol	PA_ENABLE	**Length**	1	**Access groups**	0
Object code	Simple-Var.			**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Bit	Assignment
7	="1": Automatic accept of a changed LAGE_Ziel of the process output data is turned off.
6	="0": The control word in the process output data is turned off, i.e. the control word can be accessed via FMS. Standard setting is "1", i.e. the control word is located on the process output data.
5	6. Byte of the PA-data
4	5. Byte of the PA-data
3	4. Byte of the PA-data
2	3. Byte of the PA-data
1	2. Byte of the PA-data
0	1. Byte of the PA-data

Bit	Function	Data byte	Function
= 0 (FALSE)	Process data value disabled	= 1 (TRUE)	Process data value enabled

5.14.4 PED_INI

Initializing the Process Input Data - Description.

This object contains the data which define which process input data are emulated on which communication objects after Power On of the COMPAX.

Communication objects which can be emulated on PE data are designated in the respective object descriptions.

Object description

Index	4852				
Symbol	PED_INI	**Length**	3	**Access groups**	0
Object code	Array	**Elements**	4	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data byte	Significance	Data byte	Significance
1	Object Index (High Byte)	3	Object Subindex
2	Objekt Index (Low Byte)		

Subindex	PE-Belegung	Data byte 1	Data byte 2	Data byte 3
1 (P135)	Index and subindex of the object which occupies the 1nd PE data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex
2 (P136)	Index and subindex of the object which occupies the 2nd PE data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex
3 (P137)	Index and subindex of the object which occupies the 3rd PE data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex
4 (P138)	Index and subindex of the object which occupies the 5th PE data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex

The setting is stored in the COMPAX parameter (as indicated in the parentheses).

5.14.5 PAD_INI

Initializing the Process Output Data - Description.

This object contains the data which define which process output data are emulated on which communication objects after Power On of the COMPAX.

Communication objects which can be emulated on PA data are designated in the respective object descriptions.

Object description

Index	4853				
Symbol	PAD_INI	**Length**	3	**Access groups**	0
Object code	Array	**Elements**	4	**Password**	0
Data type	Octet String	**Access rights**	read all/write all	**PD Map**	not possible

Data Description

Data byte	Significance	Data byte	Significance
1	Object Index (High Byte)	3	Object Subindex
2	Objekt Index (Low Byte)		

Subindex	Significance	Data byte 1	Data byte 2	Data byte 3
1 (P139)	Index and subindex of object which occupies the 1st PA data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex
2 (P140)	Index and subindex of object which occupies the 2nd PA data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex
3 (P141)	Index and subindex of object which occupies the 3rd PA data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex
4 (P142)	Index and subindex of object which occupies the 5th PA data byte	Object Index (High Byte)	Object Index (Low Byte)	Object Subindex

The setting is stored in the COMPAX parameter (as indicated in the parentheses).

6. Profibus Parameters

No.	Meaning	Minimum value	Standard value	Maximum value	valid as of...	see page
P191	Bus-Time-out ="0": no reaction to a time-out ="1": Stop with E73 and shut down with activation of holding brake				VP	
P193	Spontaneous messages Significance1: autom. Error message Significance 2: autom. "position reached" - message Significance 4: autom. comparator switchpoints report				immediately	9
P194	Device address	0	99	126	Power on	6
P196	Operating mode	0	0	255	Power on	7
P135	Index and subindex of object which occupies the 1st PE data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	21
P136	Index and subindex of object which occupies the 2.st PE data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	
P137	Index and subindex of object which occupies the 3.st PE data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	
P138	Index and subindex of object which occupies the 5.st PE data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	
P139	Index and subindex of object which occupies the 1st PA data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	18
P140	Index and subindex of object which occupies the 2.st PA data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	
P141	Index and subindex of object which occupies the 3.st PA data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	
P142	Index and subindex of object which occupies the 5.st PA data byte after power on. **Value:** Index • 256 + Subindex	0	0	16777215	Power on	
P203	Place status S16 and S17 onto CPX_ZSW	Bit 0 ="0" CPX_ZSW (default) Bit 0 ="1" S16, S17 onto CPX_ZSW			immediately	56
P221	Standard functions of the digital inputs accessible from the STEUERWORT object. Physical inputs freely available. Is written by the object INPUT_MASK data byte 2.	0	0	255	immediately	81
P223	Outputs A1...A8 are accessible from object OUTPUT_WORD. Written by the object OUTPUT_MASK data byte 2.	0	0	255	immediately	81
P224	Outputs A9...A16 accessible from the object OUTPUT_WORD. Written by the object OUTPUT_MASK data byte 1.	0	0	255	immediately	81

7. Special Error Messages error messages

No.	Cause	Remedy / Causes	Acknowledge with	Drive dead
E73	Time-out error	Resend the characters	[1]	no[2]

[1] No acknowledgement necessary; the error message is cancelled after the next errorless transmission.
[2] Depends on P191.

8. Appendix

8.1 The Communications List - KBL

The KBL of the Slave follows the suggestion of the SA profile with the addition of Phys-Read and Phys-Write to acyclic connections. For FMS the SAPs 20-27 and 63 are used, so that there is no overlapping with the SAPs 55-62 and NIL (no SAP) which DP requires.
The KBL headers:
KR=0
No. of KBL entries: 9
Poll lists - SAP: 255(No); has meaning only in the Master, not relevant here
ASS\ABT-CI: 3000 (30sec); control interval for establishing the connection
Symbol length: 0; i.e. KBL entries have no symbolic names

The KBL entries:

Nr. im SA Profil	KR	Local SAP	Remote Address	Remote SAP	TYPE	LLI SAP	Attribute	maxSCC	max RCC	Nr. im SA Profil	maxRAC	Control Interval	Multiplier	Max PDU Send Lo Prio	Max PDU Send Lo Prio	Max PDU Rec Hi Prio	Max PDU Rec Lo Prio	Features Supported	
1	2	20	All	All	MSZY	0	O	0	0	0	0	3000	0	0	241	0	241	00 00 00 00 20 00	Rd. ind
2	3	21	All	All	MSZY	0	O	0	0	0	0	3000	0	0	241	0	241	00 00 00 00 10 00	Wr. ind
3	4	22	All	All	MSZY_SI	0	O	0	0	1	0	3000	0	241	241	0	241	00 00 10 00 20 00	Rd. ind Evt.req
4	5	23	All	All	MSZY_SI	0	O	0	0	1	0	3000	0	241	241	0	241	00 00 10 00 10 00	Wr. ind Evt.req
5	6	24	All	All	MSAZ_SI	0	O	0	1	1	0	0	0	241	241	0	241	00 00 10 80 33 06	(Phys)Rd (Phys)Wr GetOving Evt.teq AckEv AlterEv
6	7	25	All	All	MSAZ	0	O	0	1	0	0	0	0	0	241	0	241	00 00 00 80 33 00	(Phys)Rd (Phys)Wr GetOving
7	8	26	All	All	MSAZ	0	O	0	1	0	0	0	0	0	241	0	241	00 00 00 80 33 06	(Phys)Rd (Phys)Wr GetOving AckEv AlterEV
8	9	27	All	All	MULT	0	x	0	0	0	0	0	0	0	0	241	241	00 00 00 00 00 80	Info.ind
9	10	63	All	All	BRCT	0	x	0	0	0	0	0	0	0	0	241	241	00 00 00 00 00 80	Info.ind

The figure for maximum PDU lengths is referenced to an FMS-PDU. The maximum user data length of Layer 2 is 246 bytes according to the Standard. When using source and target SAPs as well as segment addresses, the length is reduced to 242 bytes. Assuming a 1 byte LLI header, the remaining maximum FMS-PDU length specified in the KBL is therefore 241 bytes.

The VFD pointer of the KBL entries is always FFFF FFFF h, i.e. without meaning.

8.2 Default Values for Bus Parameters

Default values for mixed networds or dedicated FMS networks: (as of: 05.17.93)

Baud rate (kbit/s)	**9.6**	**19.2**	**93.75**	**187.5**	**500**	**1500**
Default values for FMS Master						
$T_{SL}(T_{Bit})$	125	250	600	1500	3500	
$minT_{SDR}(T_{Bit})$	30	60	125	250	500	
$maxT_{SDR}(T_{Bit})$	60	120	250	500	1000	
$T_{SET}(T_{Bit})$	1	1	1	1	1	
$T_{QUI}(T_{Bit})$	0	0	0	0	0	
G	1	1	1	1	1	1
HSA	126	126	126	126	126	126
max_retry_limit	1	1	1	1	1	1
Default values for FMS Slaves						
$minT_{SDR}(T_{Bit})$	30	60	125	250	255	

Each master must be capable of a ready time of $T_{RDY} \leq 570$ µs (at 1.5 MBaud).

Default values for DP networks: (as of:17.5.1993)

Baud values (kbit/s)	**up to 187,5**	**500**	**1500**
$T_{RDY}(T_{Bit})$	<=11	<=11	<=11
$T_{SDI}(T_{Bit})$	<=80	<=180	<=280
Default values for DP Masters			
$T_{SL}(T_{Bit})$	100	200	300
$minT_{SDR}(T_{Bit})$	11	11	11
$maxT_{SDR}(T_{Bit})$	60	100	150
$T_{SET}(T_{Bit})$	1	1	1
$T_{QUI}(T_{Bit})$	0	0	0
G	100	100	100
HSA	126	126	126
max_retry_limit	1	1	1
Default values for DP Slaves			
$maxT_{SDR}(T_{Bit})$	<=60	<=100	<=150
$minT_{SDR}(T_{Bit})$	11	11	11

A DP Master must maintain the times T_{RDY} and T_{SDI} shown in the table.
A DP Slave must maintain the time max T_{SDR} shown in the table.
The $maxT_{SDR}$ as well as the T_{RDY} and T_{SDI} are characteristic properties of an implementation.

9. Index