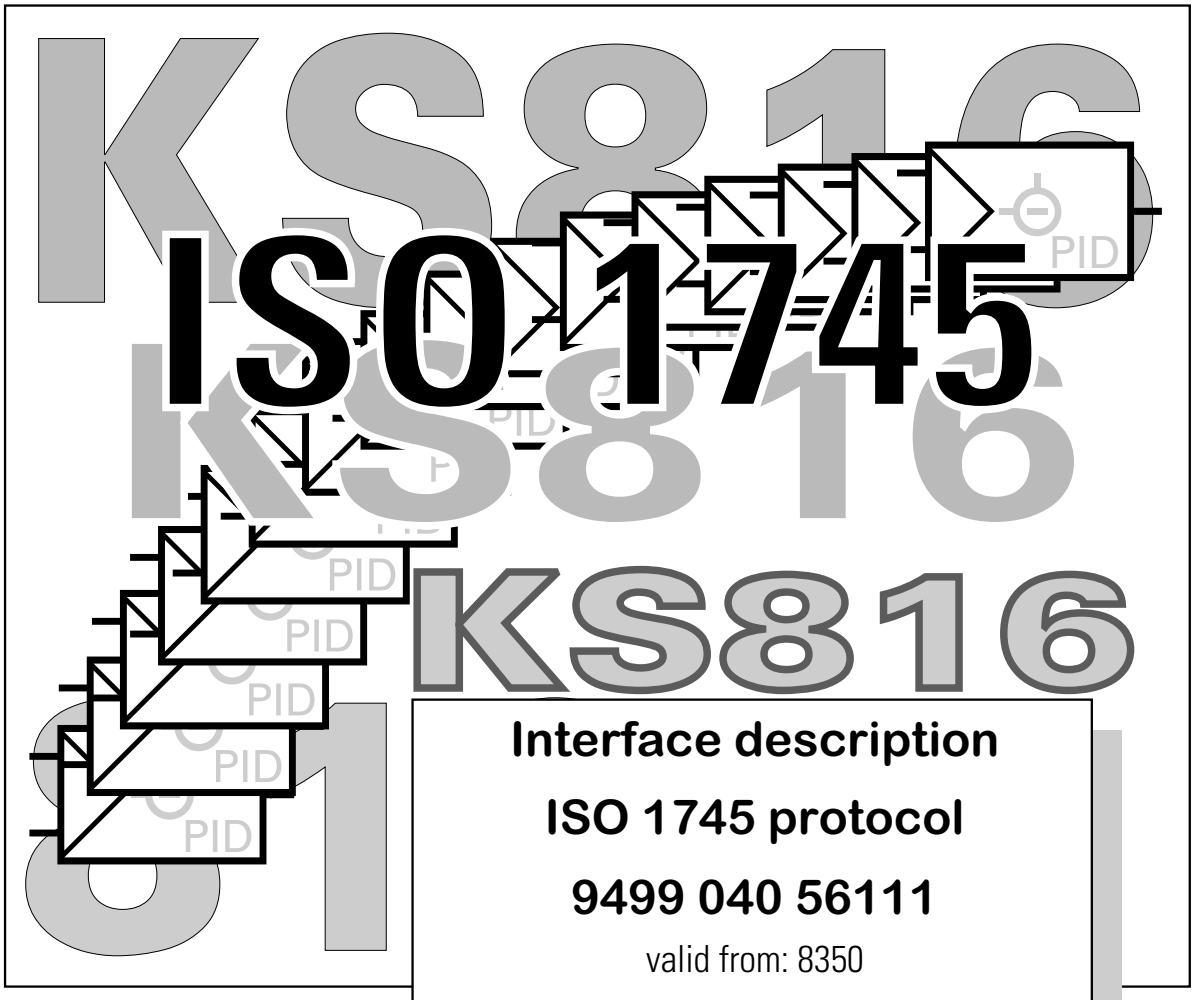
PMA Prozeß- und Maschinen-Automation GmbH

KS 816
Multi-transmitter
Multi-temperature controller



Interface description

ISO 1745 protocol

9499 040 56111

valid from: 8350

A publication of PMA Prozeß- und Maschinen Automation
P.O. Box 310229
D.34058 Kassel
Germany

Inhalt

1 Hints on operation

Multi-temperature controller version KS816-RS is provided with a serial, bussable RS485 interface which can be used for transmission of process, parameter and configuration data. Connection is via (a) 9-pole sub-D socket(s)(connector(s)). The serial communication interface permits connections to supervisory PLCs, visualization tools, etc.
An RS485/422 hardware interface is realized. The protocol available on this hardware is:

- the PCI protocol, which is based on an ISO 1745 frame,

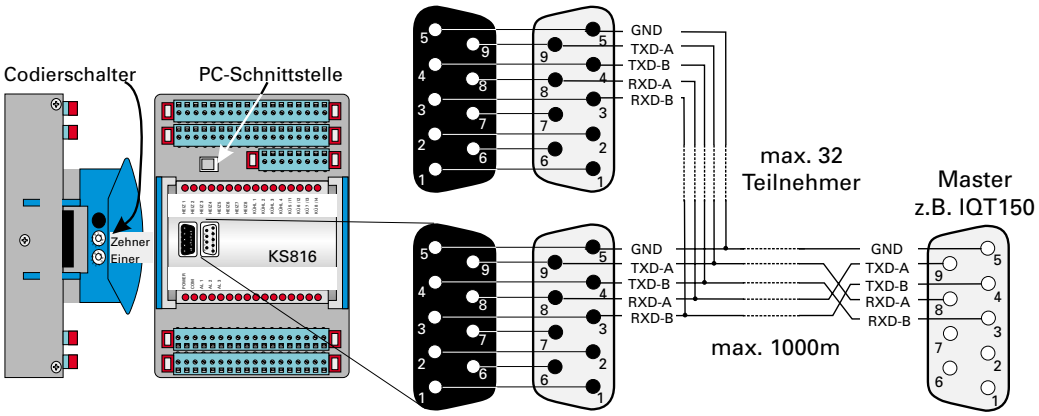
Communication is according to the master/slave principle. KS816 is always slave. The software of the serial interface is implemented as standard in the firmware.

Another standard interface is the PC interface. This interface is used for connecting an engineering tool, which runs on an external PC.

1.1 *Connecting the interface*

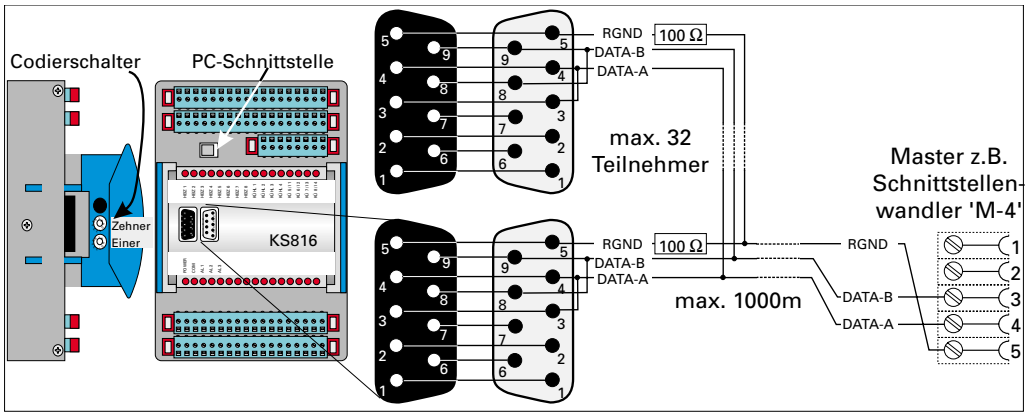
Version KS816-RS offers an RS485 or RS422 interface. ‘RS422’ as available in this product means an RS485 4-wire interface. A driver for reception and a driver for sending are available.

Fig.: 1 Connection examples RS422 interface



On the 2-wire RS 485 interface, reception and transmission lines must be galvanically connected by the user.

Fig.: 2 Connection examples RS485 interface



If an RGND connection is required with an RS485 adjustment, a 100 Ohm resistor must be mounted across terminal 5 (RGND) and terminal 5 of the interface converter by the user.
The outputs are galvanically isolated.

The interface mode is half-duplex.

Installing appropriate cables must be done by the user, whereby the general cable specifications according to EIA RS485 must be taken into account.

2 Interface protocol

2.1 *Protocol layer 1*

Bus connection is physical:

- via the PC interface as a TTL signal (COM 1)
- via an RS485/422 connection (COM 2) with version KS816-RS.

2.1.1 Data format

The following transmission format, fixed, must be used:

- 1 start bit,
- 7 bits ASCII value or 7 bits binary
- 1 parity bit (EVEN)
- 1 stop bit

LSB is transmitted first, MSB is parity bit.

2.1.2 Baud rate

The Baud rate for the serial interface is adjustable. The following Baud rates are available:

- 2400 Baud
- 4800 Baud
- 9600 Baud
- 19200 Baud

2.1.3 Parity

Parity detection is fixed to EVEN.

2.1.4 Addressing

KS 816 can be operated together with KS40, KS50, KS90, KS92, KS94 , KS98, KS4580, DIGITAL 280/380 and PRO 96 and the ICS 90 and ITS 90 systems at the same bus. Decisive for instrument selection is the address (2 bytes).

The KS 816 (0...99) address is adjusted via the “KS816 Engineering Tool” (general instrument settings → communication → address).

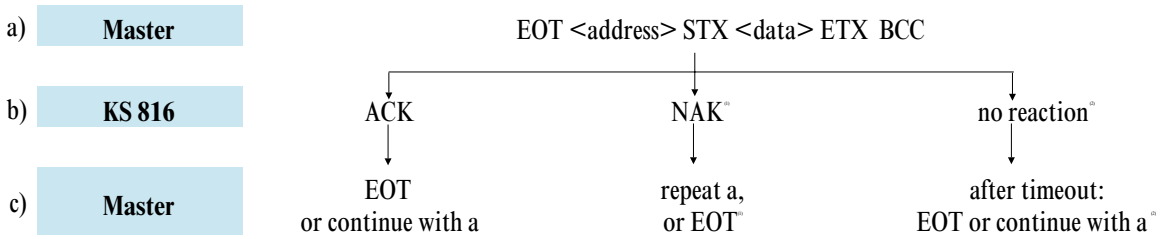
2.2 *Protocol layer 2*

A fixed master/slave principle is used, whereby KS816 is always slave. Transmission control (communication start and cancelation by EOT) is always by the master.

Two communication services are available:

- ❑ for data sending: SDA (Send Data with Acknowledge) , acknowledged by KS 816

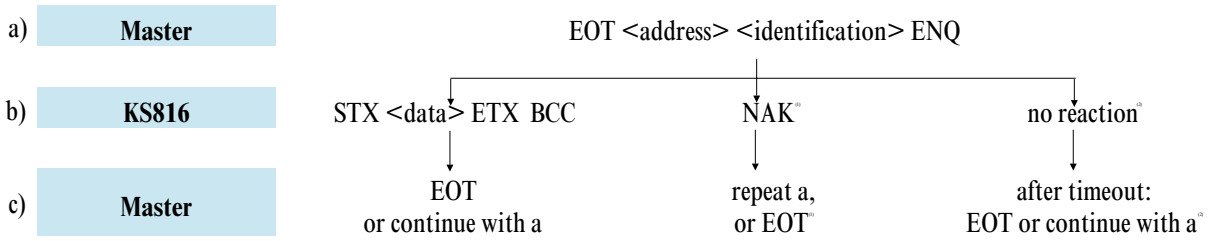
Data flow direction : master → KS 816



(1) Possible after disturbance of transmission or after sending inadmissible data.
(2) Possible after KS 800 failure, bus failare or faulty address specification.

❑ for data request: RDR (Request Data with Reply)
data request with reply in one message cycle.

Data flow direction : KS816 Master



2.2.1 Transmission control characters

The following transmission control characters are used:

Abbreviation	HEX	Description
STX	02	Start of Text - data introduction
ETX	03	End of Text - end of data
EOT	04	End of Transmission - reset the interface units or transmission cancelation
ENQ	05	Enquiry - request for reply
ACK	06	Acknowledge - confirmation
NAK	15	Not Acknowledge - no confirmation

2.2.2 Character format

Numbers and characters in the address, identification or data field are always transmitted as ASCII characters.
The following 7-bit ASCII characters with parity (EVEN) are valid

CHR	HEX	Description
,	2C	Comma as separator
=	3D	Separator between identification and value
0...9	30...39	Values for numbers and codes
B	41	Additional for codes
:...?	3A...3F	Values for floating point format (FP)
@...;	40...7F	Values for status and control bytes
...;	20...7F	Characters for text string (CHAR16)
.	2E	Decimal point

(1) Possible after disturbance of transmission or after sending inadmissible data.
(2) Possible after KS 816 failure, bus failare or faulty address specification.

3 Message structure

3.1 *Message elements*

In the following section, some expressions which shall be explained as follows are used:

Element	Description	Rem.
<addr>	Address of a participating unit, always 2 bytes long, adjustable on the instruments	**A**
<daten>	Data field composed of a) fields <identification> a. <value>, separated by character '=' b) a series of successive <value> with several block accesses	**B**
<identification>	identification field composed of a) field <code> and b) different selection criteria <selection>	**C**
<value>	Value of a datum, which is addressed with a key.	
<code>	Addressing key of a datum, 2-digit, decimal number range, first digit also 'B'.	**D**
<selection>	further addressing field for selection of <function block no.> a. <function no.>	**E**
<BCC>	Block Check Count. All characters between STX (exclusive) and ETX (inclusive) are connected bytewisely by an EXOR function and output as 1 byte, always after ETX.	**F**

Bem. A Address field

The adress field can be transmitted only after 'EOT' and must be generated only by the master. It is two bytes long. The address number range is 00 ... 99. If the transmitted address corresponds with the one adjusted in the unit, the message is intended for this unit.
Different address settings are possible for COM1 and COM2.

Bem. B Data field

The data field contains the parameters and data to be transmitted.

The equality sign is followed by the value of a datum (<valuex>). Several data are separated by a comma. The data type depends on the access. The last value before 'ETX' ends without ','.

With block read access with additional selection criteria, these criteria are specified only once. The data follow without further identifications. Thus, the message structure becomes more compact.

Bem. C Identification field

The identification field addesses a defined datum or a data area in the instrument. It consists of a code and of an additional selection identification with some accesses.

With a data enquiry, the identification field contains information for KS92/94 which data the unit is expected to send. This is always followed by the address field. In the reply, it is also specified for clear determination of the datum, followed by the data field with separator „=".

With data entry, STX is followed by the identification field for addressing the values to be specified. Connection of the data field is by means of character „=".

Bem. D Code

The code identification is two bytes long and the value range is ASCII '00'...'99' and 'B2'... 'B3'.

Bem. E Addition selection criteria

In order to form a purposeful sub-set from the variety of data, additional selection criteria are defined:

❑ Function block number

A function block is addressed by a function block number. It is within ‘0’ and ‘250’ and is appended to the code field by means of a comma.

‘<code>,<function block no.>’

Function block number ranges:

- 0 general data for the overall instrument
- 1 - 99 fixed function blocks
- 100 - 250 variable function blocks

❑ Function number

A function as a partial address of a function block is addressed with a function number. It is within ‘0’ and ‘99’ and is appended to the function block number by means of a comma.

‘<code>,<function block no.>,<function no.>’

Function number ranges:

- 0 - 99 Functions
- 0 Function General, Default, unless a no. is specified

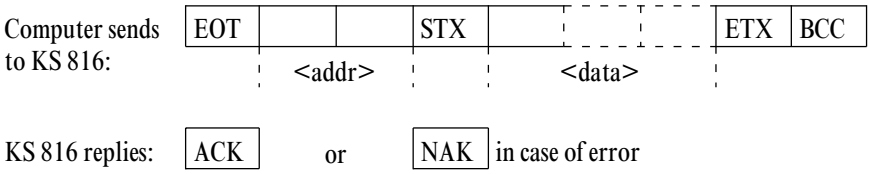
Bem. F Safety procedure

Correct transmission of a message is supported by two safety procedures:

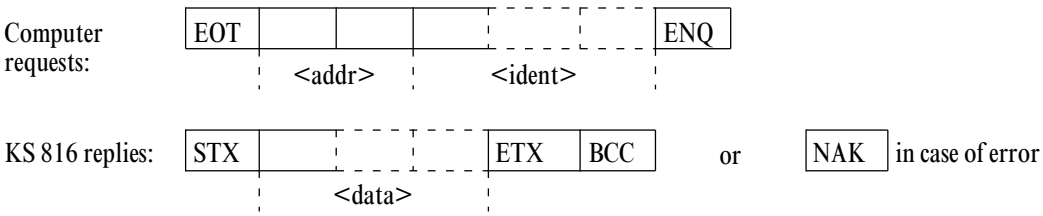
- check of each message byte by formation of parity (1 bit per 7 data bits)
- Check by Block Check Count : safety section, which connects all characters of a message between STX (excl.) and ETX (incl.) bytewisely by a logic XOR function; length 1 byte, follows always after ETX.

3.2 *Basic message structure*

Message structure with data transmission:



Message structure with data enquiry:



3.3 *Data types*

Data values are classified according to data types for transmission. Only characters which can be represented in ASCII are permitted.

- BCD
  <%-2>Floating Point number in BCD-ASCII format,
  Range: -9999 ... -0.001, 0, 0.001 ... 9999
  optional: negative polarity sign and decimal point permitted; exponent representation not permitted.
  KS816 controllers with an accuracy of max. 4 digits. With received data, number of digits and decimal point position are not fixed and depend on the FP resolution. The values are not rounded off.
  Switch-off value for BCD data is : -32000
- INT
  positive integer number in ASCII format
  Range: 0 ... 32767
  Range with configuration words: 0000 ... 9999 (→ page 13)
  Exception: switch-off value '-32000'
- ST1
  Status, bit-oriented, 1 byte length
  Range: 00H ... 3FH, transmitted: 40H...7FH
  Only 6 bits for transmission of information can be used, i.e. bit 0...5 (LSB = bit 0). Bit 6 must always be set to '1', in order to avoid confusion with the control characters. Bit 7 contains the parity bit.
- SYS16
  System identification number, 16 bytes
  Format: xx,yyyyyyyy,zzzz (→ page 11)

4 Standard protocol

The KS816 standard protocol version represents instrument-specific standard data.

4.1 *CODE table*

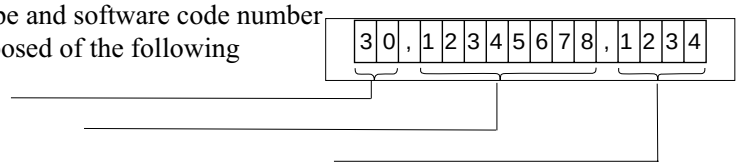
Survey of codes availabble in standard protocol

Code	Description	R/W	Type	Range	Description	Rem.
18	System ident	R	SYS16		System identification	A
80	Block 81... 83	R	Block			
81	Write Error	R	INT	0, 100 ... 127	Error of last write access	B
82	Write Error Position	R	INT	0 ... 99	Position of last write access error	
83	Read Error	R	INT	0, 100 ... 127	Error of last read access	

Bem. A ***Instrument data***

System dentifiction number (code 18)

For instrument identification, instrument type and software code number can be read via code 18. The datum is composed of the following sections:



Instrument type: (30 =KS816)
SW code number: (the last 8 digits)
Instrument version: 7th to 10th digit of 12NC (4 digits)

Bem. B ***Diagnosis access: block 8x***

For test purposes, an additional debug access which signals errors of the last write or read access is available. Presently, reading is possible for:

- error number of last write access; 0 = no error
- position of the faulty datum during the last write access;
  0 = no error or error in address
  1 = first datum is faulty (also with single accesses)
  n = nth datum is faulty (with block accesses)
- error number of last read access; 0 = no error

An independent memory for error messages is available for each interface COM1 and COM2. Presently, the following error messages are defined:

Err. no.	Description	Error name
101	unspecified error	ERR_UNSPECIFIED
102	read not permitted	ERR_RD_NOTALLOWED
103	write not defined	ERR_WR_NOTALLOWED
104	local operation / no write access	ERR_LOCOPERAT
105	non-defined key code	ERR_KEYIDENT
106	range overflow function block no.	ERR_FB_OVERFL
107	range overflow function no.	ERR_FCT_OVERFL
108	write or range overflow	ERR_WR_RANGE_OV
109	char is not a digit	ERR_NODIGIT
110	no '\0' found in the correct position	ERR_ENDDELIMITER
111	no '=' in the correct position	ERR_NO_EQUALSIGN
112	faulty ST1 format (status)	ERR_NO_ST1FORMAT
113	no ',' in the correct position	ERR_NO_COMMA
114	byte range overflow	ERR_BYTE_OVERFL
115	number of digits exceeded	ERR_DIGIT_OVERFL
116	range 9999 exceeded	ERR_RG9999_OVERFL
117	undefined protocol type	ERR_UNDEF_PRTCTYPE
118	undefined parameter reference	ERR_UNDEF_PARAMREF
119	undefined decimal point	ERR_UNDEF_DECPNT

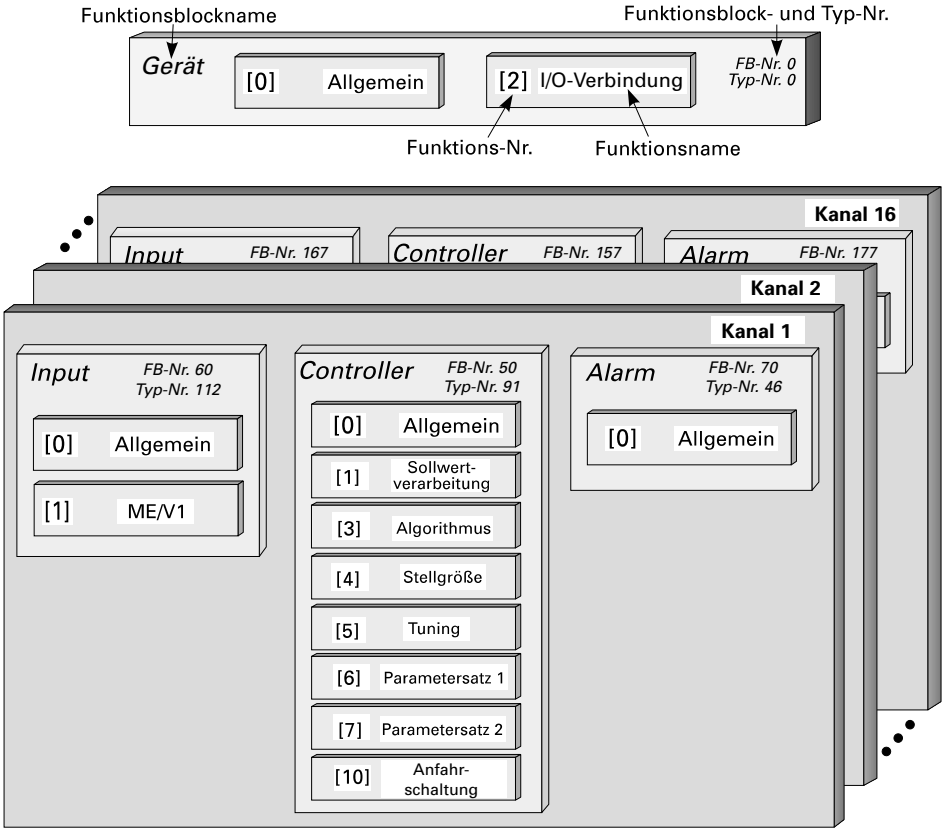
Err. no.	Description	Error name
120	no STX in the write message	ERR_NO_STX
121	number INT faulty	ERR_INT_ANZ
122	number REAL faulty	ERR_REAL_ANZ
123	faulty access type	ERR_ZUGRIFF
124	no config level	ERR_WR_NO_CONF
125	local operation	ERR_WR_LOCAL
126	error FI switch-over	ERR_WR_FU_UM

5 Function block protocol

5.1 *Data structure*

Due to the variety of information to be processed in KS816, logically related data and actions are grouped in function blocks. A function block has input, output data, parameters and configuration data. For KS816, 25 function blocks are defined. They are all addressed via fixed block addresses (FB no.). Each block is also divided into several functions. Functions are addressed via function numbers (Fct-no.). Function number 0 addresses function block-specific data.

Fig.: 3 Survey of KS816 function blocks and functions



5.2 *CODE tables*

5.2.1 Structure of configuration words (C.xxxx)

The configuration words mentioned in the following code tables are composed of several sections, which can be transmitted only in common.
The data in the table must be interpreted as follows:

Example (C100):

Code	Descr.	R/W	Type	Description	Range
B3	C100	R/W	INT	CFunc: Controller function (T,H) CType: Controller type (Z) WFunc:set-point function (E)	0..xxyz

Description	CFunc		CType	WFunc
	Thousands	Hundreds	Tens	Ones
Range	x	x	y	z
	00 ... 12		0...4	0...7
Example: continuous controller; standard controller; set-point-cascade with offset	1004			

For transmission of configuration words, see chapter 6.2.3 page 23.

5.2.2 INSTRUMENT (FB no.: 0 type no.: 0)

Function block 'INSTRUMENT' is used for classification of all data which are valid for the overall instrument.

Process data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
01	Unit_State 1	R	ST1	Status 1		A
10	Block 13..15, 18	R	Block			
13	Write Error	R	INT	Error during last write access	0, 100...127	→ p. 11
14	Write Error Position	R	INT	Position of last write access error	0...99	
15	Read Error	R	INT	Error of last read access	0, 100...127	
18	Type	R	INT	Type no. of function block	0	
20	Block 21...27	R	Block			
21	HWbas	R	INT	Basic HW options: module A, P		B
23	SWopt	R	INT	SW options 1		C
24	SWcod	R	INT	SW code no. 7th to 10th digit of 12NC	wxyz	D
25	SWvers	R	INT	SW code no. 11th to 12th digit of 12NC	00xy	E
26	OPVers[1)]	R	INT	Operating version		
27	EEPVers[1)]	R	INT	EEPROM version		
31	OpMod	R/W	INT	Switch over unit to configuration mode (only after 1)	0	
				Switch over unit to on-line mode (only after 0)	1	
				Cancelation of configuration mode (only after 0)	2	
32	Ostartg	R/W	INT	Self-tuning stop/start of all group controllers	0..1	
33	UPD	R/W	INT	Acknowledgement of local data change	0..1	F

Bem. A Unit_State1

MSB							LSB
D7	D6	**D5**	D4	D3	D2	**D1**	D0

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	'0'	Always '0'		
D1	CNF	Instrument status	on-line	configuration
D2...D4	'0'	Always '0'		
D5	UPD	Parameter update	no	yes
D6	'1'	Always '1'		
D7		Parity		

Bem. B HWbas

COM2		0	0
T	H	Z	E

Basic version without COM2	0	0	0	0
COM2 with CANopen	0	1	0	0
COM2 with PROFIBUS-DP	1	0	0	0
COM2 with ISO1745	1	1	0	0

Example: Value '*HWbas = 0100*' means that the addressed instrument is a COM2 interface with CANopen connection.

Bem. C SWopt

Version		0	0
T	H	Z	E

Basic version	0	0	0	0
Water cooling (so far not available)	0	1	0	0

1) For transmission of configuration words, see chapter 6.2.3 page 23.

Bem. D SWCod

T	H	Z	E
7th digit	8th digit	9the digit	10th digit

Example: Value '*SWCod= 7239*' means that the software for the addressed instrument contains code number 4012 157 239xx.

Bem. E SWvers

T	H	Z	E
0	0	11th digit	12th digit

Example: Value '*SWVers= 11*' means that the software of the addressed unit contains code number 4012 15x xxx11.

Bem. F UPD

Changing a parameter value or a configuration value via an interface is indicated in the UPD flag. This bit is also set after mains recovery. The flag, which can be read also via code UPD, can be reset (value =0).

I/O connection (Funktions-Nr: 2)

Code	Des.	R/W	Typ	Description	Range	Rem.
20	Block 21...25	R	Block	Blockzugriff		
21	H1_K4	R	INT	Heizen/Kühlen - Signale	0...255	G
22	H5_K8	R	INT	Heizen/Kühlen - Signale	0...255	
23	H9_K12	R	INT	Heizen/Kühlen - Signale	0...255	
24	H13_K16	R	INT	Heizen/Kühlen - Signale	0...255	
25	A1_3	R	INT	Alarm - Signale	0...7	H

Bem. G H1_K4 … H13_K16

MSB															LSB
D15	D14	D13	D12	D11	D10	D9	D8	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**

Bit-Nr.	Name	Belegung	Zustand '0'	Zustand '1'
D0	H1, 5, 9, 13	Heizen Kanal 1, 5, 9, 13	aus	ein
D1	K1	Kühlen Kanal 1, 5, 9, 13	aus	ein
D2	H2	Heizen Kanal 2, 6, 10, 14	aus	ein
D3	K2	Kühlen Kanal 2, 6, 10, 14	aus	ein
D4	H3	Heizen Kanal 3, 7, 11, 15	aus	ein
D5	K3	Kühlen Kanal 3, 7, 11, 15	aus	ein
D6	H4	Heizen Kanal 4, 8, 12, 16	aus	ein
D7	K4	Kühlen Kanal 4, 8, 12, 16	aus	ein
D8...D15	'0'	immer '0'		

Bem. H A1_3

MSB															LSB
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	**D2**	**D1**	**D0**

Bit-Nr.	Name	Belegung	Zustand '0'	Zustand '1'
D0	A1	Alarm 1		
D1	A2	Alarm 2		
D2	A3	Alarm 3		
D3...D15	'0'	immer '0'		

Parameter a. configuration data

General (function no.: 0)

Code	Des.	R/W	Type	Description	Range	Rem.
B3	C900 (1) COM1	R/W	INT	Prot: protocol type (T) Baud: Baud rate (H,Z)	0..xyy0	
	Adr1(1)	L/S	INT	COM1: instrument address:	0..99	
	C904	R/W	INT	Freq: mains frequency 50/60 (T)	0..x000	
	C902(1) COM2	R/W	INT	Prot: protocol type (T) Baud: Baud rate (H,Z)	0..wxyz	
	Adr2(1)	R/W	INT	COM2: Instrument address: ISO1745 CAN-BUS	0..99 0..255	

I/O connection (function no.: 2)

Code	Des.	R/W	Type	Description	Range	Rem.
B3	C530	R/W	ICNF	Main configuration do17 ... do19 mode_alarm1 (T) mode_alarm2 (H) mode_alarm3 (Z)	0...xyz0	

5.2.3 INPUT (FB no.: 60 ... 67 and 160 … 167 Type no.: 112)

All data which concern acquisition and processing of all input values (analog/digital) are grouped in function block 'INPUT'. The data are provided once per controller channel.

Process data

General Input processing of analog signals (Function no.: 0)

Code	Des.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block access (1, 3)		
1	Input_x_Fail	R	ST1	Signal Input x Fail		A
3	x1	R	BCD	Main variable		
10	Block	R	Block	Block access (13, 18)		
13	INP1	R	BCD	Raw meas. value before meas. value correction		
18	Function Type	R	INT	Type no. of function block	112	

Bem. A Statusbyte Input_X_Fail:

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	**D0**

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	INP1F	Input 1 Fail	no	yes
D1...D5	'0'	Always '0'		
D6	'1'	Always '1'		
D7		Parity		

Parameter a. configuration data

ME/V1 Measurement value INP1 : acquisition and processing (Function no.: 1)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	$X1_{in}$	R/W	BCD	Measurement value correction X1 Input	-999..9999	
	$X1_{out}$	R/W	BCD	Measurement value correction X1 Output	-999..9999	
	$X2_{in}$	R/W	BCD	Measurement value correction X2 Input	-999..9999	
	$X2_{out}$	R/W	BCD	Measurement value correction X2 Output	-999..9999	
B3	X0	R/W	BCD	Phys. value at 0%	-999..9999	
	X100	R/W	BCD	Phys. value at 100%	-999..9999	
	X_{Fail}	R/W	BCD	Substitute value at sensor fail	-999..9999	
	T_{fm}	R/W	BCD	Filter time const. meas. value processing.	0.0 .. 999.9	
	T_{kref}	R/W	BCD	Customer-specified TC	0...60 °C / 32...140°F	
	C200	R/W	INT	Type: sensor type (T,H) Unit: unit (Z)	0..xxy0	
	C205	R/W	INT	Fail: sensor fail behaviour (T) STk: TC source (H) XKorr: process value correction enable (Z)	1..wxy0	
	C190	R/W	INT	Allocation of digital signals: controller off (Z) w/w2 (E)	0...00xy	

5.2.4 CONTR (FB no.: 50 ... 57 and 150 … 157 Type no.: 91)

All data concerning the controller are grouped in function block 'CONTR'. They are provided once for each controller channel.

Process data

General (Function no.: 0)

Code	Des.	R/W	Type	Description	Range	Rem
00	Block	R	Block	Block access (1...9)		
1	Status 1	R	ST1	Status 1		A
3	W	R	BCD	Eff. set-point		
4	X	R	BCD	Eff. process value		
5	Y	R	BCD	Effective output variable		
6	xw	R	BCD	Control deviation		
18	Type	R	INT	Type no. of function block	90	
30	Block	R	Block	Block access (31...38)		
33	A/M	R/W	INT	Automatic/manual switch-over	0..1	
34	OStart	R/W	INT	Self-tuning start	0..1	
35	We/i	R/W	INT	Wext/Wint switch-over	0..1	
36	w/W2	R/W	INT	w/W2 switch-over	0..1	
38	Coff	R/W	INT	Controller off/on	0..1	

Bem. A Status1: (code 01)

MSB							LSB
D7	D6	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	Y1	Switching output	off	on
D1	Y2	Switching output	off	on
D2	A/M	Autom/manual	Auto	Manual
D3	CFail	Controller status	ok	not ok
D4	Coff	Controller switched off	no	yes
D5	XFail	Sensor Fail	no	yes
D6	'1'	Always '1'		
D7		Parity		

Set-point — Set-point processing (Function no.:1)

Code	Des.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block access (1, 3)		
01	WState	R	ST1	Set-point status		B
03	Wint	R	BCD	Effective internal set-point		
30	Block	R	Block	Block access (31...32)		
31	Wnvol	R/W	BCD	Int. set-point, non-volatile	-999..9999	
32	Wvol	R/W	BCD	Int. set-point, volatile	-999..9999	

Bem. B WState: (code 01)

MSB							LSB
D7	D6	D5	**D4**	**D3**	**D2**	**D1**	**D0**

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	w/W2	w/W2 switch-over	w	W2
D1	We/Wi	Wext/Wint	Wext	Wint
D2	w/Wanf	w/Wanfahr	w	Wanf
D3	GRW	Gradient function active	no	yes
D4	Weff_fail	Error effective set-point	no	yes
D5	'0'	Always '0'		
D6	'1'	Always '1'		
D7		Parity		

Output variable — Output variable processing (function no.:4)

Code	Des.	R/W	Type	Description	Range	Rem.
30	Block	R	Block	Block access (31, 35)		
31	dYman	R/W	BCD	Difference output variable	-210..210	
32	Yman	R/W	BCD	Absolute output variable	-105..105	
33	Yinc	R/W	INT	Increment. output variable	0, 1	
34	Ydec	R/W	INT	Decrement. output variable	0, 1	
35	Ygrw_ls	R/W	INT	Speed for incr./decr. output variable offset	0, 1	

Tuning — Self-tuning (Function no.:5)

Code	Des.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block access (1, 3)		
1	State_Tune1	R	ST1	Status Tuning		C
3	ParNeff	R	INT	Eff. parameter set number	0...1	
30	Block	R	Block	Block access (31...39)		
31	ParNr	R/W	INT	Parameter set number effective	0 .. 1	
32	Tu1	R	BCD	Delay time heating	0...9999 s	
33	Vmax1	R	BCD	Rate of increase heating	0,000...9,999 %/s	
34	Kp1	R	BCD	Process gain heating	0,000...9,999	
35	MSG1	R	INT	Error code of self-tuning heating	0...8	
36	Tu2	R	BCD	Delay time cooling	0...9999 s	
37	Vmax2	R	BCD	Rate of increase cooling	0,000...9,999 %/s	
38	Kp2	R	BCD	Process gain cooling	0,000...9,999	
39	MSG2	R	INT	Error code of self-tuning cooling	0...8	

Bem. C Status 1 Tuning 'State_Tune1'

MSB							LSB
D7	D6	D5	D4	D3	**D2**	**D1**	**D0**

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	OStab	Process at rest	no	yes
D1	Orun	Self-tuning mode	off	on
D2	Oerr	Self-tuning result	Ok	error
D3...D5	'0'	Always '0'		
D6	'1'	Always '1'		
D7		Parity		

Parameter a. configuration data

General

(Function no.: 0)

Code	Des.	R/W	Type	Description	Range	Rem.
B3	C100	R/W	INT	CFunc: controller function (T,H) CType: controller type) WFunc:set-point function (Z) (E)	0..xxyz	
	C101	R/W	INT	CMode:controller output action (T) CDiff: x/x-w differentiation (H) CFail: behaviour with sensor fail (Z) CAnf: start-up circuit (E)	0..wxyz	
	C700	R/W	INT	OMode: self-tuning mode (T) OCond: process at rest (H) OGrp: allocation group self-tuning (Z) OCntr: controlled adapt. mode (E)	0..wxyz	
	C180	R/W	INT	SWext: source for Wext (T)	0..x000	

Set-point

Set-point processing(Function no.: 1)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	W0	R/W	BCD	Lower set-point limit f. Weff	-999..9999	
	W100	R/W	BCD	Upper set-point limit f. Weff	-999..9999	
	W2	R/W	BCD	Additional set-point	-999..9999	
	Grw+	R/W	BCD	Set-point gradient	>0..9.999	
	Grw-	R/W	BCD	Set-point gradient minus	>0..9.999	
	Grw2	R/W	BCD	Set-point gradient W2	>0..9.999	

Algo

Control algorithm (Function no.: 3)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	Xsh	R/W	BCD	Neutral zone	0.2 .. 20,0 %	
	Tpuls	R/W	BCD	Min. pulse length	0.1..2,0 s	
	Tm	R/W	BCD	Actuator response time	10..300 s	
	Xsd1	R/W	BCD	Switching difference signaller	0,1..9999 %	
	LW	R/W	BCD	Trigger point separation addit. contact	-999..9999	
	Xsd2	R/W	BCD	Switching difference addit. contact	0,1..9999 %	
	Xsh1	R/W	BCD	Neutral zone	0.0 .. 999.9%	
	Xsh2	R/W	BCD	Neutral zone	0.0 .. 999.9 %	

Output variable

Output variable processing (Function no.: 4)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	Y_{min}	R/W	BCD	Min. output limiting	-105..105 %	
	Y_{max}	R/W	BCD	Max. output limiting	-105..105 %	
	Y0	R/W	BCD	Working point f. output variable	-105..105 %	
	Yh	R/W	BCD	Maximum mean output value	5..100%	
	LYh	R/W	BCD	Limit for mean value formation	0,1 .. 10,0	

Tuning

Self-tuning (Function no.: 5)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	YOptm	R/W	BCD	Output variable during process at rest	-105..105	
	dYopt	R/W	BCD	Step height with identification	5..100	
	POpt	R/W	INT	Parameter set to be optimized	0...1	
	OXsd	R/W	BCD	Hysteresis with parameter selection	0.0..9999	
	Trig1	R/W	BCD	Trigger point 1	0.0..9999	

Paramset x — Control parameter set 1 / 2(Function no.: 6,7)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	Xp1	R/W	BCD	Proportional band 1	0.1..999.9	
	Tn1	R/W	BCD	Integral time 1	0..9999	
	Tv1	R/W	BCD	Derivative time 1	0..9999	
	T1	R/W	BCD	Min. cycle time 1	0.4..999.9	
	Xp2	R/W	BCD	Proportional band 2	0.1..999.9	
	Tn2	R/W	BCD	Integral time 2	0..9999	
	Tv2	R/W	BCD	Derivative time 2	0..9999	
	T2	R/W	BCD	Min. cycle time 2	0.4..999.9	

Start-up circuit — (Function no.: 10)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	Ya	R/W	BCD	Max. output value	5 .. 100 %	
	Wa	R/W	BCD	Start-up set-point	-999 .. 9999	
	TPa	R/W	BCD	Start-up holding time	0 .. 9999 min	

5.2.5 ALARM (FB-Nr.: 70 ... 77 and 170 … 177 Typ-Nr.: 46)

Function block ‘ALARM’ defines the overall alarm processing of the relevant controller. The data are provided once per controller channel.

Process data

General — (Function no.: 0)

Code	Des.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block access (1 .. 3)		
1	Status_Al1	R	ST1	Alarm status 1		A
3	HC	R	BCD	Heating current meas. value		
18	Type	R	INT	Type no. of function block	46	

Bem. A Status_Al1

MSB							LSB
D7	D6	D5	**D4**	**D3**	**D2**	**D1**	**D0**

Bit no.	Name	Allocation	Status ‘0’	Status ‘1’
D0	Lim HH	Alarm HH	off	on
D1	Lim H	Alarm H	off	on
D2	Lim L	Alarm L	off	on
D3	Lim LL	Alarm LL	off	on
D4	Fail	Fail	no	yes
D5	‘0’	Always ‘0’		
D6	‘1’	Always ‘1’		
D7		Parity		

Parameter a. configuration data

General

(Function no.: 0)

Code	Des.	R/W	Type	Description	Range	Rem.
B2	LimL	R/W	BCD	Low limit alarm	-999..9999	
	LimH	R/W	BCD	High limit alarm	-999..9999	
	xsd1	R/W	BCD	Switching difference low and high alarms	0..9999	
	LimLL	R/W	BCD	Low low limit alarm	-999..9999	
	LimHH	R/W	BCD	High high limit alarm	-999..9999	
	LimHC	R/W	BCD	Heating current limit value	0..HC100	
B3	C600	R/W	INT	Src: signal source (T,H) Fnc: function (Z) DestFail: fail Destination (E)	0..xxyz	
	C601	R/W	INT	DestLL : (T) DestL : (H) DestH : (Z) DestHH : (E)	0..wxyz	

6 Examples

6.1 *Message examples in standard protocol*

Example:
The computer requests system identification (code 18) from KS816 with address 01.

Computer requests:

EOT	0	1	1	8	ENQ
	<addr>		<code>		

KS 816 replies:

STX	1	8	=	3	0	,	1	5	7	2	7	5	1	0	,	0	0	0	0	ETX	BCC
	<code>			<val>																	

<val> = 30 means a KS816 is concerned
15727510 ≙ Software code no. 4012-157-27510
0000 ≙ Order no. 9407-480-00001

6.2 *Principles of the function block protocol*

A function block comprises input and output data (process data) as well as parameter and configuration data. It can be addressed via a block number. An allocated block type defines the relevant function.

The various access mechanisms are:

6.2.1 Single access

This access (code xx) can be used for reading or writing a single process value of a function. Single accesses to parameter and configuration data are not possible.

Example: (message structure with specified data)<P255%-2>
Transmission of the absolute output value (Y_{man}) to controller channel 1.

Computer transmits data to KS816:

EOT	0	2	**STX**	3	2	,	5	0	,	4	=	5	0	**ETX**	**BCC**
	Addr			code			FB no.			Fct-no.		Value			

KS 816 replies: **ACK** or **NAK** in case of failure

6.2.2 Block access (tens block)

This access (code x0) can be used for reading max. nine process values of a function.

Example: (message structure with data request)
Reading of set-points (W_{nvol} and W_{vol}) from controller channel 3 .

Computer requests:

EOT	0	2	3	0	,	5	3	,	1	**ENQ**
	Addr		Code			FB-no.			Fct-no.	

KS 816 replies:

STX	3	1	=	5	0	,	3	2	=	7	9	**ETX**	**BCC**
	Code			Value1			Code			Value2			

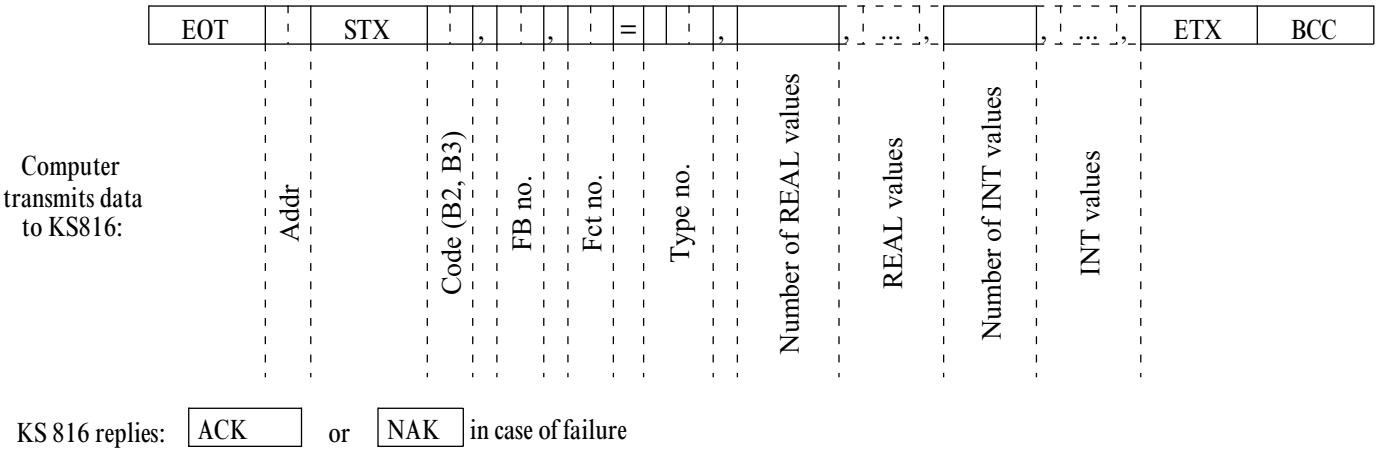
6.2.3 Block access (overall block)

This access can be used for reading and writing all parameter (code B2) and configuration data (code B3) of a function. The following conditions are applicable for this access:

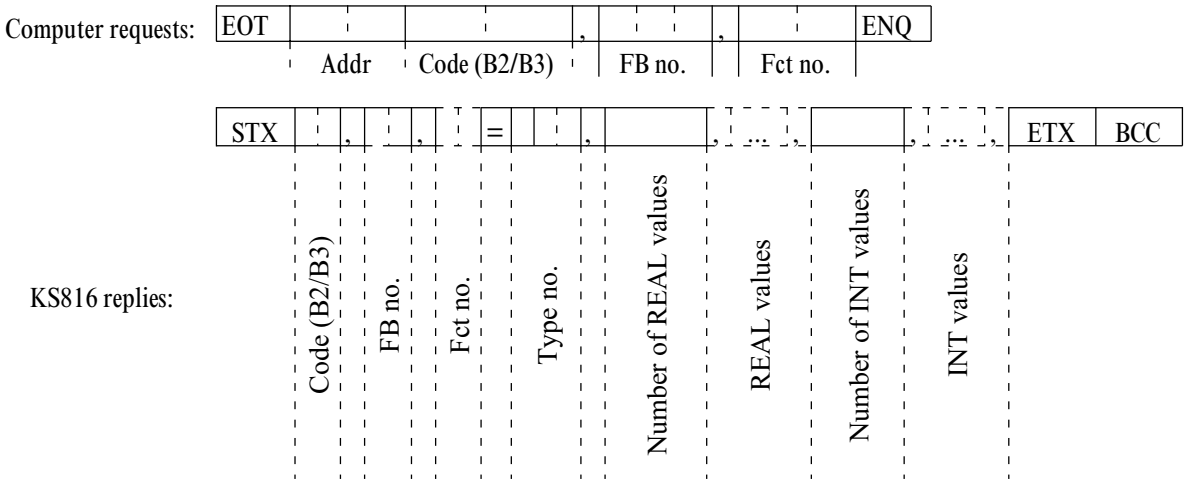
- For writing data with 'Code B3', the unit must be switched to configuration mode (→ see page 14 'OpMod'). All new configuration data and parameters entered are only effective, when the unit was switched back to on-line.
- All data of a message must be defined, omissions are not permissible.
- If all parts of a message in the instrument are unused (HW and SW options), the complete message must be transmitted nevertheless. Checking the non-available data is omitted.
- With faulty block write accesses, the following rule is applicable: A message is answered with NAK, if min. one datum is faulty. Already valid values are stored.
- If the function number is omitted, function 0 (general) is addressed.

The general structure of a message with block accesses with code B2/B3 is shown below. The exact message structure (between ***STX*** and ***ETX***) for the individual functions can be found below the relevant code table.

Message structure with data specification:



Message structure with data request:



6.3 *Message structure in function block protocol*

6.3.1 INSTRUMENT

Message structure for function 'General'

Blockzugriff auf Konfigurationsdaten																							max. eff. Länge: 41 Bytes
STX	B3	,	0	,	0	=	0	,	0	,	5	,	C900	,	Adr1	,	C904	,	C902	,	Adr2	ETX	BCC

Message structure for function 'I/O connection'

Blockzugriff auf Konfigurationsdaten																						max. eff. Länge: 43 Bytes
STX	B3	,	0	,	2	=	0	,	1	,	HC100	,	4	,	C500	,	C530	,	C551	,	HCcycl	ETX	BCC

6.3.2 INPUT

Message structure for function 'ME/V1'

Blockzugriff auf Parameterdaten																						max. eff. Länge: 44 Bytes
STX	B2	,	6x	,	1	=	112	,	4	,	X1in	,	X1out	,	X2in	,	X2out	,	0	ETX	BCC

Blockzugriff auf Konfigurationsdaten																						max. eff. Länge: 66 Bytes
STX	B3	,	6x	,	1	=	112	,	5	,	X0	,	X100	,	XFail	,	Tfm	,	Tkref	,	3	,	C200	,	C205	,	C190	ETX	BCC

6.3.3 CONTR

Message structure for function 'General'

Blockzugriff auf Konfigurationsdaten																						max. eff. Länge: 36 Bytes
STX	B3	,	5x	,	0	=	91	,	0	,	4	,	C100	,	C101	,	C700	,	C180	ETX	BCC

Message structure for function 'Set-point'

Blockzugriff auf Parameterdaten																						max. eff. Länge: 56 Bytes
STX	B2	,	5x	,	1	=	91	,	6	,	W0	,	W100	,	W2	,	Grw+	,	Grw-	,	Grw2	,	0	ETX	BCC

Message structure for function 'Algo'

Blockzugriff auf Parameterdaten																						max. eff. Länge: 72 Bytes
STX	B2	,	5x	,	3	=	91	,	8	,	Xsh	,	Tpuls	,	Tm	,	Xsd	,	LW	,	Xsd	,	Xsh	,	Xsh	,	0	ETX	BCC

Message structure for function 'Output variable'

Blockzugriff auf Parameterdaten																						max. eff. Länge: 51 Bytes
STX	B2	,	5x	,	4	=	91	,	5	,	Ymin	,	Ymax	,	Y0	,	Yh	,	LYh	,	0	ETX	BCC

Message structure for function 'Tuning'

Blockzugriff auf Parameterdaten																						max. eff. Länge: 49 Bytes
STX	B2	,	5x	,	5	=	91	,	4	,	YOptm	,	dYopt	,	OXsd	,	Trig1	,	1	,	POpt	ETX	BCC

Message structure for function 'Paramset x'

Blockzugriff auf Parameterdaten																						max. eff. Länge: 72 Bytes
STX	B2	,	5x	,	<6, 7>	=	91	,	8	,	Xp1	,	Tn1	,	Tv1	,	T1	,	Xp2	,	Tn2	,	Tv2	,	T2	,	0	ETX	BCC

Message structure for function 'Start-up circuit'

Blockzugriff auf Parameterdaten																						max. eff. Länge: 37 Bytes
STX	B2	,	5x	,	10	=	91	,	3	,	Ya	,	Wa	,	Tpa	,	0	ETX	BCC

6.3.4 ALARM

Message structure for function 'General '

Blockzugriff auf Parameterdaten																		max. eff. Länge: 58 Bytes	
STX	B2	,	7x	,	0	=	46	,	6	,	LimL	,	LimH	,	xsd_1	,	LimLL	,	LimHH	,	LimHC	**ETX**	**BCC**

Blockzugriff auf Konfigurationsdaten															max. eff. Länge: 36Bytes	
STX	B3	,	7x	,	0	=	46	,	0	,	2	,	C600	,	C601	**ETX**	**BCC**

7 Annex

7.1 *Terms*

FB	Abbr. of function block
Fkt	Abbr. of function
ET	Abbr. of Engineering Tool
Function	A self-contained partial function of the function block seen from the interface
Function block	Self-contained processing unit
HW	Abbr. f. hardware
ISO1745	Standard communication protocol ISO 1745, ASCII-based
PC interface	Front-panel interface on KSX controller for connecting an engineering tool
PCI	Process Control Instrument
PCI protocol	Protocol based ISO 1745, implemented for Philips controllers
RS422	Standard 4-wire interface, Full duplex, (EIA RS 422); in this case: separate send/receive channels with up to 32 units
RS485	Standard 2-wire connection, Half duplex, (EIA RS 485)
SW	Abbr. f. software
TTL	Signal level at module level

8 Index