PMA Prozeß- und Maschinen-Automation GmbH

Industrial controller KS 92/94

Interface description ISO 1745 protocol

9499 040 45011

valid from: 8344

A publication of
PMA Prozeß- und Maschinen Automation
P.O.Box 310320, D.34113 Kassel
Germany

Contents

1 Hints for operation

If industrial controller KS92/94 is equipped with module B (option), a bussable serial interface for transmission of process values, parameters and configuration data is provided. Connection is on the instrument rear. The serial communication interface permits connections to master PLC, visualization tools, etc.
Various serial interface versions are available.
As far as the hardware is concerned, a TTL and an RS485/422 interface are realized.
The protocols available on this hardware are:

- the PCI protocol according to an ISO 1745 protocol frame

Communication is according to the master/slave principle. KS92/94 is always slave. The serial interface software is implemented in the firmware as standard.

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

1.1 Operation

KS94 data can be read, or displayed and modified from the front-panel PC interface or via the serial interface.

After delivery of controller KS94, the PC interface is active. KS94 configuration and parameter setting are supposed to be done by means of the engineering tool before commissioning.

Switch-over to the serial interface is either

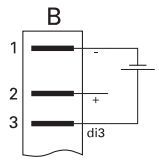
- via operator dialogue (front):

press ↵ ≥ during 3 sec. → **Para** flashes press ▲ until **CBus** flashes → ↵ confirm briefly.	display **CBus**	≙ switch over to rear interface
press ↵ ≥ during 3 sec. → **Para** flashes press ▲ until **CFrnt** flashes → ↵ confirm briefly.	display **CFrnt**	≙ switch over to front-panel interface

- or by activating 'REMOTE' (→ page 5). Switching back to LOCAL does not cause switch-over to the front-panel interface.

Switch-over to the PC interface is only possible with the R/L input set to LOCAL.

1.2 Remote/local



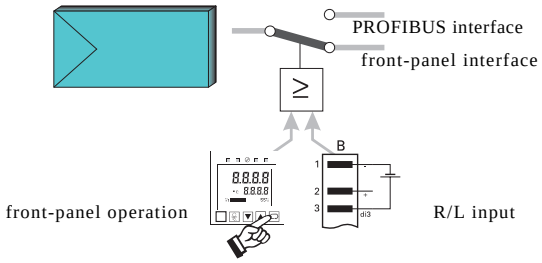
Units with serial interface are fitted with a hardware input (di3) for switch-over between REMOTE and LOCAL operation (R/L).

During **'REMOTE'**all operations via the serial interface (writing and reading) are permissible. The following operations are still possible via the keys of the local operating front panel:

- Display switch-over
- Display of parameters without modification
- Display of configuration data without modification

During remote operation, the PC interface cannot be operated. When switching over from LOCAL to REMOTE, an active PC interface is switched off.

During **'LOCAL'**, only reading of all data via the serial interface is permissible. Modifications are not possible, exception:
any data related only to the interface or which are not adjustable local via local operation.



1.3 Interface connection

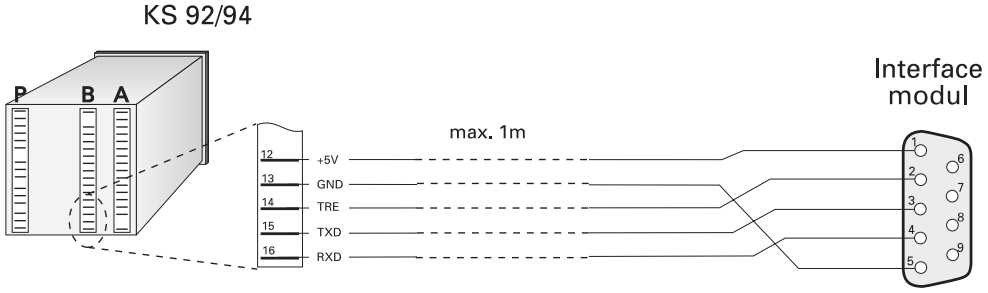
The interface hardware is implemented on module B. The serial interface is connectable via connector B.
The following options are available:

- Option 1: rear serial interface, physical
  TTL-based signals;
  Protocols: ISO1745 or MODBUS, selectable
  12 NC: 9407 xxx **1**xxxx
- Option 2: rear serial interface, physical
  RS485 / RS422-based signals, selectable;
  Protocols: ISO1745 or MODBUS, selectable
  12 NC: 9407 xxx **2**xxxx

1.3.1 Version as TTL interface

Option 'serial interface as TTL interface' is a low-priced version for connecting max. 4 units to an interface module 9404 429 980x1.

Fig.: 1 Connecting example TTL interface

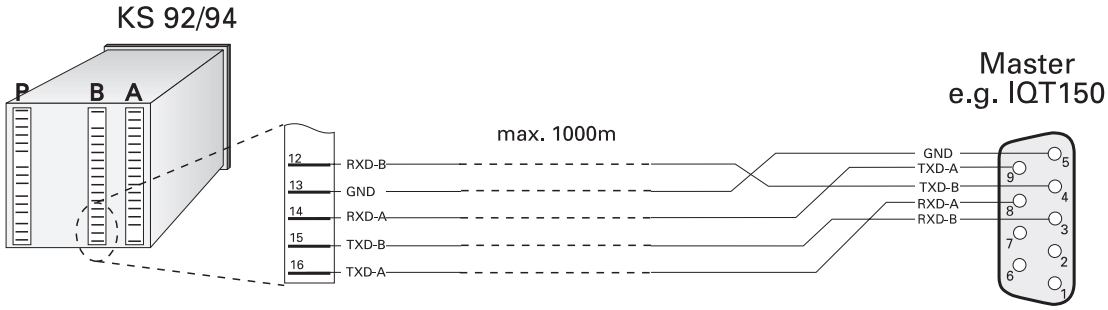


The connection is designed so that interface cable 9404 407 50011 can be used for connection to the interface box.
Inside the interface module, the connections of the instrument interfaces are galvanically connected with the bus interface and galvanically isolated from the supply voltage.

1.3.2 RS485/422 interface version

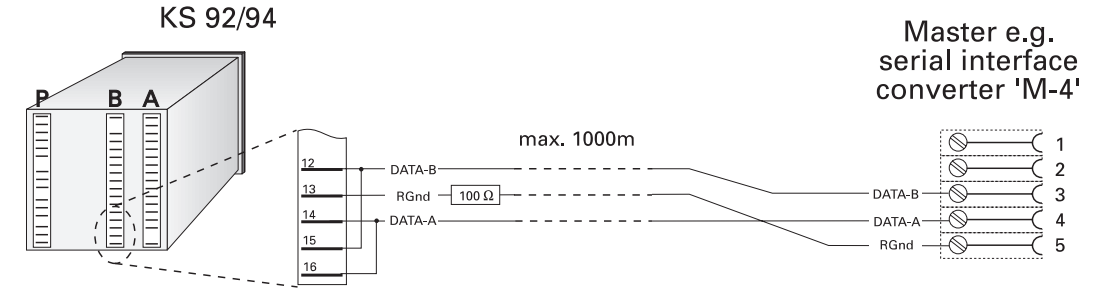
Another module B version offers an RS485 or RS422 interface. 'RS422' in the sense of this product means a 4-wire RS485 interface. A driver is available for receiving and for sending.

Fig.: 3 Connecting example RS422 interface



On the 2-wire RS485 interface, the receive and send lines must be connected galvanically by the user.

Fig.: 2 Connecting example RS485 interface



If an RGND connection is required with RS485 setting, a 100 Ohm resistor must be fitted by the user across terminal 13 (GND) and terminal 5 on the serial interface converter. The outputs are galvanically isolated.

The interface mode of operation is half-duplex.

The relevant cabling must be done by the user, whereby the general cable specifications and signal specifications according to EIA RS485 must be followed.

2 Interface protocol

2.1 Protocol level 1

Connecting the bus is done physically:

- via module B as TTL interface to the interface module (12NC: 9404 429 980x1), which is fitted with an RS422/485 interface with a 9-pole sub-D connector socket on the bus side. E.g. max. four controller can be connected to this module.
- directly via an RS485/422 connector on module B.

2.1.1 Data format

The following fixed transmission format must be used:

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

LSB is send at first, MSB is parity bit.

2.1.2 Baud rate

The baud rate for the serial interface is adjustable, e.g. via local operation. 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 92/94 can be operated at the same bus with KS 4580, KS 4770, DIGITAL 380 and PRO 96 and with the ICS 90 and ITS 90 systems. Decisive for instrument selection is the address (2 bytes).

The KS 92/94 address (0...99) is adjustable at parameter level (ADR).

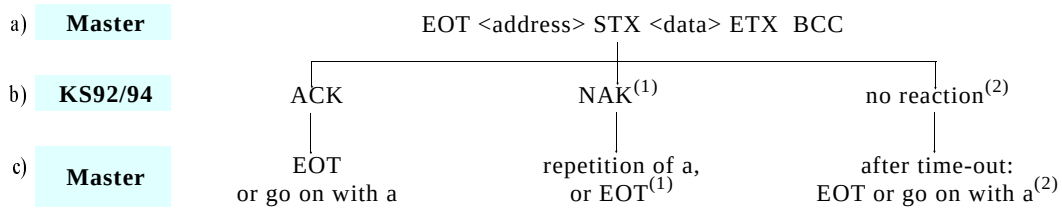
2.2 Protocol level 2

A rigid master/slave principle is applicable, whereby KS92/94 is always slave. Transmission control (communication start and cancelation by EOT) is always by the master.

Two data transmission services are available:

❑ for data sending Data sending: SDA (Send Data with Acknowledge)
Data sending with acknowledgement by KS 92/94

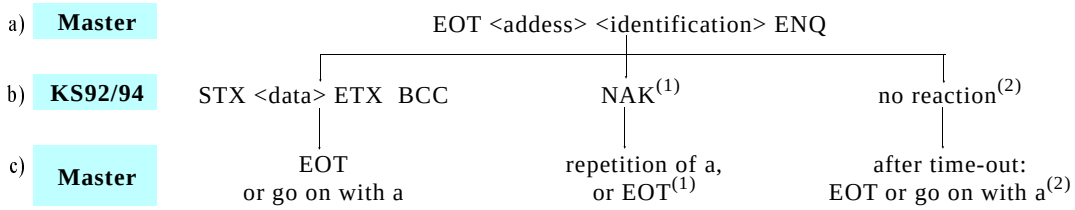
Data flow direction : Master → KS92/94



1) Can be sent after transmission error or after sending inadmissible data.
2) After KS92/94 failure, bus failure or faulty addressing.

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

Data flow direction : KS92/94 → 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 - Data end
EOT	04	End of Transmission - The interface units are reset
ENQ	05	Enquiry - request for reply
ACK	06	Acknowledge - confirmation
NAK	15	Not Acknowledge - no confirmation

2.2.2 *Character format*

The following 7-bit ASCII characters with parity (EVEN) are valid:

CHR	HEX	Description
,	2C	Comma as separator
=	3D	Separator between identification and word
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
.	2E	Decimal point

1) Can be sent after transmission error or after sending inadmissible data.
2) After KS92/94 failure, bus failure or faulty addressing.

3 Message structure

3.1 Message items

Some expressions which are used in the following text must be explained:

Item	Description	Rem.
<addr>	Address of a participating unit, always 2 bytes long, adjustable on the unit	**A**
<data>	The data field is composed of a) fields <identification> a. <value>, separated by character '=' b) several <value> with some block accesses	**B**
<identification>	The identification field comprises a) field <code> and b) additional selection criteria <selection> in some cases	**C**
<value>	Value of a date, which is addressed with the key.	
<code>	Address key of a date, 2-digit, range of decimal numbers, first digit also 'B'.	**D**
<selection>	other address 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 by an EXOR function bytewisely and output as 1 byte, always after ETX.	**F**

Bem. A ***Address field***

The address field can be transmitted only after an 'EOT'. Therefore, it may be generated only by the master. It is two bytes long. The address range is 00 ... 99. If the trnsmitted address corresponds with the address in the instrument, the message is intended for this instrument.

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 date (<valuex>). Several data are separated by comma. The data type depends on the access. The last value before 'ETX' ends without ','.

With block read access with additional selection criteria, the selection criteria are specified only once, followed by the data without further identification. Thereby, the compactness of the message structure is increased.

Bem. C ***Identification field***

The identification field addresses a defined date or a data area in the instrument. It comprises a code and an additional selection identifier with some accesses.

With a data request, the identification field is used to inform KS92/94 which data it is expected to send. In this case, it always follows on the address field. In the reply, it is also used for clear determination of the date, followed by the data field with separator ,,=".

With data sending, STX is followed by the identification field for addressing the values to be transmitted. The data field is connected with character ,,=".

Bem. D ***Code***

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

Bem. E ***Additional selection criteria***

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

❑ Function block number

A function block is addressed with a function block number, which comprises range ’0’ to ’250’ and is appended to the code field using 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 (only KS98)

❑ Function number

A function as a partial address of a function block is also addressed with a function number. It comprises value range ’0’ to ’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

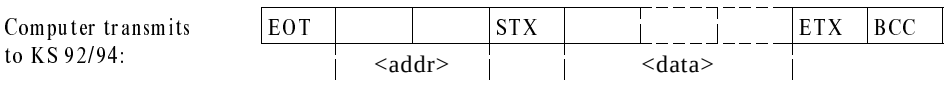
Bem. F ***Safety procedure***

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

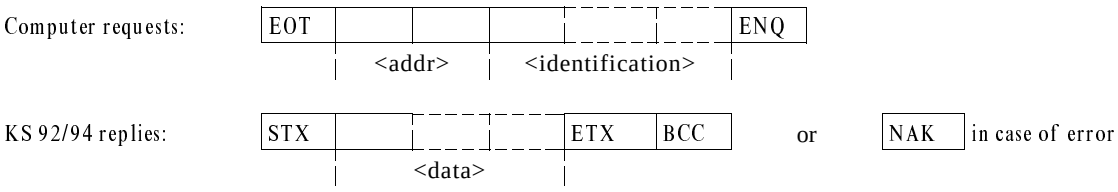
- checking each message byte by formation of the parity (1 bit per 7 data bits)
- check by Block Check Count : safety part which connects all characters of a message between STX (excl.) and ETX (incl.) bytewisely by a logic XOR function; length 1 byte, always after ETX.

3.2 Basic message structure

Message structure with data sending:



Message structure with data request:



3.3 Data types

Data values are divided into data types for transmission. Only characters which can be formatted in ASCII are permitted.

- BCD
  Floating point number in BCD-ASCII format,
  Value range: -9999 ... -0.001, 0, 0.001 ... 9999
  optional: negative polarity sign and decimal point permitted; exponents are not permitted.
- INT
  positive integer number in ASCII format
  Value range: 0 ... 32767
  Value range with configuration words: 0000 ... 9999 (→ page 23)
- ST1
  Status, bit-oriented, 1 byte length
  Value range: 00H ... 3FH, transmitted: 40H...7FH
  Only 6 bits can be used for transmission of information, bit 0...5 (LSB = bit 0). Bit 6 must always be set to ’1’ to avoid confusion with the control character. Bit 7 contains the parity bit.
- SYS16
  System identification number, 16 bytes
  Format: xx,yyyyyyyy,zzzz (→ page 17)
- CHAR16
  Text string comprising n characters, presently defined n=1, n=5, n=16
  permissible characters: 20H...7FH
- ICMP (Integer **Comp**act)
  Bit information as integer transmission, max. 15 bits
  Range: 0...32767; integer transmission is in ASCII format.

	fixed to ’0’	Bit signification														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	-	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Example:
Bit 13 = 1 and Bit 1 = 1, all remaining bits are ’0’
internal hex value: 0x2002, as integer value: 8194, transmitted as an ASCII value: ’8194’

4 Standard protocol

The KS92/94 standard protocol version shall support the user, who has already installed instruments such as KS40, KS4290 or KS4580 and who wants to add KS92/94 units without major adaptations.

4.1 CODE table

Tab.: 1 *Survey of codes availabble in standard protocol*

Code	Description	R/W	Type	Range	Description	Rem.
00	Block 01... 09	R	Block			A
01	Status 1	R	ST1		Status 1	
02	Status 2	R	ST1		Status 2	
03	Y_{PID} (L) / Y_{man} (S)	R/W	BCD		eff. correcting value	
04	W_{eff}	R	BCD		Eff. set-point	
05	X_{eff}	R	BCD		Eff. process value	
06	W_{vol} (1)	R/W	BCD	W0 ... W100	Volatile set-point	
07	X-W	R	BCD		Control deviation	
08	X2	R	BCD		Ratio	
09	X3	R	BCD		Auxiliary variable 3	
10	Block 13, 16, 18, 19	L	Block			B
13	Reset update bit(2)	R/W	INT	0..1/0	Display of local data change	
16	Wnvol	R/W	BCD	-999...9999	Non-volatile set-point	
18	System ident	R	SYS16		System identification	
19	dYman	R/W	BCD		Relative correcting variable	
20	Block 21... 29	R	Block			C
21	Xp1	R/W	BCD	0.1...999,9	Act. proportional band 1	
22	Tn1	R/W	BCD	0...9999	Act. integral time 1	
23	Tv1	R/W	BCD	0...9999	Act. derivative time 1	
24	T1	R/W	BCD	0.4...999,9	Act. min. duty cycle 1	
25	Xp2	R/W	BCD	0,1...999,9	Act. proportional band 2	
26	Tn2	R/W	BCD	0...9999	Act. integral time 2	
27	Tv2	R/W	BCD	0...9999	Act. derivative time 2	
28	T2	R/W	BCD	0.4...999,9	Act. min. duty cycle 2	
29	ParNo(3)	R/W	INT	0...3	Act. parameter set switch-over	
30	Block 31... 38	R	Block			D
31	Lim_{L1} (1)	R/W	BCD	-999...9999	Low limit value 1	
32	Lim_{H1} (1)	R/W	BCD	-999...9999	High limit value 1	
33	Lim_{L2} (1)	R/W	BCD	-999...9999	Low limit value 2	
34	Lim_{H2} (1)	R/W	BCD	-999...9999	High limit value 2	
35	Lim_{L3} (1)	R/W	BCD	-999...9999	Low limit value 3	
36	LimH3(1)	R/W	BCD	-999...9999	High limit value 3	
37	Lim_{L4} (1)	R/W	BCD	-999...9999	Low limit value 4	
38	Lim_{H4} (1)	R/W	BCD	-999...9999	High limit value 4	
40	Block 41... 48	R	Block			
41	State_di1	R	ST1		digital input di1...di6	→ S. 19
42	State_di2	R	ST1		digital input di7...di12	
43	INP1 4)	R	BCD		Signal input 1	
45	INP3 4)	R	BCD		Signal input 3	
46	INP4 4)	R	BCD		Signal input 4	
47	INP5 4)	R	BCD		Signal input 5	
48	INP6 4)	R	BCD		Signal input 6	

1) Data has switch-off function additional data value '-32000'.
2) With data sending, the remote/local restriction is not applicable.
3) The parameternumber can be entered externally via control inputs and be written in defined operation modes. Reading or writing on the instaneously active control parameter set is done via code 21-28.
4) Decimal point is varible dependend of values.

Code	Description	R/W	Type	Range	Description	Rem.
50	Block 51... 57	R	Block			
51	Grw+	R/W	BCD	0,01 ... 99,99		
52	Grw-	R/W	BCD	0,01 ... 99,99		
53	Ymin	R/W	BCD	-105<Ymax<105		
54	Ymax	R/W	BCD	-105<Ymax<105		
55	XWonx	R/W	BCD	0 ... 9999		
56	XWony	R/W	BCD	0 ... 9999		
57	Grwon	R/W	BCD	0,01 ... 99,99		
94	Operating data	R	Block		Compact format of block 00	**E**
95	Process data	R	Block		Compact format of process data	**F**

Bem. A ***Actual process data (L1) :***

Accesses to block 0x data show selected level-1 data of function block CONTR: effective correcting variable, effective set-point, effective process value, volatile set-point, X-W deviation, ratio X2, auxiliary variable X3 and instrument statuses. All data can be read, data Yabs, Wvol can also be written.

Status1: (code 01)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	Lm1	Limit 1	off	on
D1	Lm2	Limit 2	off	on
D2	Lm4	Limit 3	off	on
D3	Lm5	Limit 4	off	on
D4	CNF	Instrument	on-line	configuration
D5	UPD	Parameter changed	no	yes
D6	'1'	always '1'		
D7		Parity		

UPD has value 1, when parameter or config. data were changed by local operation or after power-on.

Status2: (code 02)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	R/L	Remote/local	Remote	Local
D1	A/M	Automatic/manual	Auto	Man
D2	We/Wi	W_{ext}/W_{int} switch-ov.	W_{ext}	W_{int}
D3	w/W2	w/W2	W	W2
D4	y/Y2	Y2 switch-over	y	Y2
D5	X Fail	Sensor fail	no	yes
D6	'1'	always '1'		
D7		Parity		

Bem. B ***Instrument data / status change : block 1***

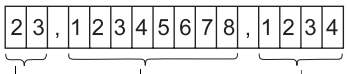
Block 1x comprises different level-1 information. Moreover, instrument-specific characteristic data are contained in this block.

Parameter update bit (code 13)

Changing a parameter value or a configuration word via local front or PC interface is indicated in the UPD flag of status 1. After power recovery, this bit is also set. The flag, which can also be read via code code 13 (FB protocol: code 33) can be reset with code 13/(33)=0. A write access is also permitted with the instrument in local mode, in order to enable a master system to detect current changes.

System identification number (code 18)

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



Instrument type: (21=KS92; 22=KS94)
SW code number: (the last 8 digits)
Instrument version: 7th to 10th digit of the 12NC (4 digits)

Bem. C ***Controller parameter set : Block 2***

Accesses of block 2x are used for reading and writing the most important controller parameters of function block CONTR. Only the instantaneously active one of the four available parameter sets can be selected.

Bem. D ***Alarm limits : block 3***

Block 3x accesses permit values of function block ALARM to be adjusted. Operable are the high and low limit value of all four alarms. Each alarm can be switched off with value '-32000'.

Bem. E ***Block 00 in FP format***

The access via code 94 is the compact format of block 00. Only the value without code is transmitted, '='- and without separator. Moreover, the floating point values in FP format are sent (see below).

Computer requests:

EOT			9	4	ENQ
	<addr>		<code>		

KS 92/94 replies:

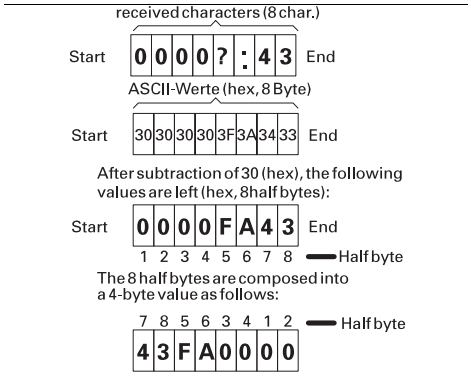
STX	1byte	1byte	8bytes	8bytes	8bytes	8bytes	8bytes	8bytes	8bytes	ETX	BCC
	<status 1>	<status 2>	<Y_{pid}>	<W_{eff}>	<X_{eff}>	<W_{vol}>	<X-W>	<X2>	<X3>		

Length of reply : 58 bytes (useful data) + 3 bytes for control character

The reply is sent immediately after reception of the request.

The 8-byte floating point message (FP format)
For a better resolution, measurement values are transmitted as 8 bytes in FP format. Transmission of a floating point number is in Intel format as a nibble hex value. A floating point value in KS92/94 comprises 4 bytes:
As a hex format cannot be transmitted due to equality with control characters, this format is converted into ASCII so that 4 bytes become 8 (bytes) to be transmitted.

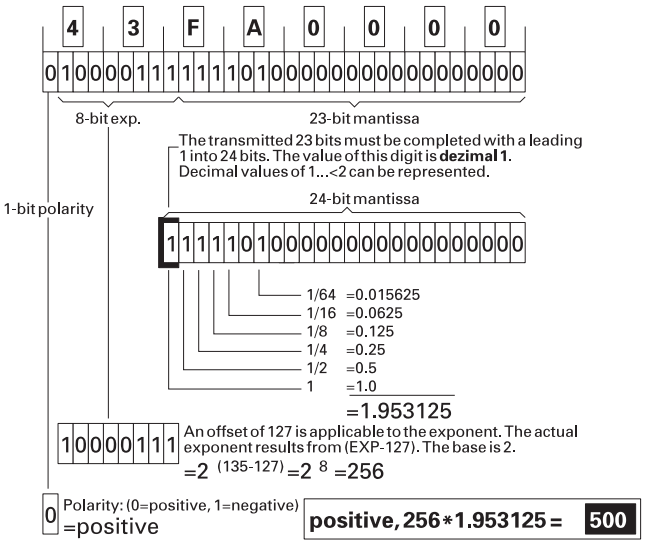
Fig.: 4 Conversion of the 8-byte message into 4-byte floating point



Conversion of received data
Fig. 4 shows the conversion of the message (8 characters ASCII) into 8 half bytes. To improve representation, the order of half bytes to IEEE 754-1985 standard (single-precision format) was selected.

Fig. 5 shows the conversion of 8 half bytes into a decimal value. The message contains 23 bits mantissa, 8 bits exponent and 1 bit polarity. The 23rd bit of the mantissa has value 1/2, the 22nd bit has value 1/4 etc. so that decimal values within 0...<1 can be represented. For improving the accuracy, the mantissa is agreed to have a leading 24th bit which is always '1' and has decimal value 1. It is not transmitted. Thus decimal values of 1.0...<2.0 can be represented. The exponent is provided with an offset of 127 (decimal), which is exponent of base 2.
Particularity: if all 8 half bytes are '0', the decimal value is 0.

Fig.: 5



Bem. F ***Compact format process data***

Access via code 95 is the compact process data format. Only the value without code, '=' and without separator is transmitted. Moreover, the floating point values are transmitted in FP format (see Bem. E).

Computer requests:

EOT			9	5	ENQ
	<adr>		<code>		

KS 92/94 replies:

STX	1Byte	1Byte	8Byte	8Byte	8Byte	8Byte	8Byte	8Byte	8Byte	8 Byte
	<status>	<status-1>	<Y_{eff}>	<W_{eff}>	<X_{eff}>	<Inp1>	<Inp3>	<Inp4>	<Inp5>	<Inp6>

1Byte	1Byte	1Byte	1Byte	ETX	BCC
<state_di1>	<state_di2>	<state_inpf>	<state_switch>		

Length of reply : 70 bytes (useful data) + 3 bytes for control character
The reply is sent immediately after reception of the request.

Transmission in FP format includes: Yeff, Weff, Xeff, Inp1, Inp3, Inp5, Inp6
INP1 to INP6 contain the process values after pre-processing. This access is used for cyclical process data exchange to PROFIBUS-module B.

As status-byte are defined:

Status (actual)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit-no.	Name	Allocation	Status '0'	Status '1'
D0	y1	Switching output 1	of	on
D1	y2	Switching output 2	of	on
D2	Lim1	Limit 1	of	on
D3	Lim2	Limit 2	of	on
D4	Lim3	Limit 3	of	on
D5	Lim4	Limit 4	of	on
D6	'1'	always '1'		
D7		Parity		

Status-1 (beforehand)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit-no.	Name	Allocation	Status '0'	Status '1'
D0	y1	Schaltausgang 1	of	on
D1	y2	Schaltausgang 2	of	on
D2	Lim1	Limit 1	of	on
D3	Lim2	Limit 2	of	on
D4	Lim3	Limit 3	of	on
D5	Lim4	Limit 4	of	on
D6	'1'	always '1'		
D7		Parity		

State_di1 (digital input di1 ... di6)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit-no.	Name	Allocation	Status '0'	Status '1'
D0	di1	digital input 1	of	on
D1	di2	digital input 2	of	on
D2	di3	digital input 3	of	on
D3	di4	digital input 4	of	on
D4	di5	digital input 5	of	on
D5	di6	digital input 6	of	on
D6	'1'	always '1'		
D7		Parity		

State_di2 (digital inputs di7 ... di12)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit-no.	Name	Allocation	Status '0'	Status '1'
D0	di7	digital input 7	of	on
D1	di8	digital input 8	of	on
D2	di9	digital input 9	of	on
D3	di10	digital input 10	of	on
D4	di11	digital input 11	of	on
D5	di12	digital input 12	of	on
D6	'1'	always '1'		
D7		Parity		

State_inpf (Error status of analog inputs Inp1...Inp6)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit-no.	Name	Allocation	Status '0'	Status '1'
D0	if1	Error status Inp 1	of	on
D1	'0'	always '0'	-	-
D2	if3	Error status Inp 3	of	on
D3	if4	Error status Inp 4	of	on
D4	if5	Error status Inp 5	of	on
D5	if6	Error status Inp 6	of	on
D6	'1'	always '1'		
D7		Parity		

State_switch (Changeovers)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit-no.	Name	Allocation	Status '0'	Status '1'
D0	R/L	Remote/	Local	Remote
D1	A/M	Auto/Manual	Auto	Manual
D2...D4	'0'	always '0'	-	-
D5	UPD	Parameter change local	no	yes
D6	'1'	always '1'		
D7		Parity		

4.2 Message examples in standard protocol

Example 1:
The computer requests status byte 2 (code 02) from KS92/94 with address 01.

Computer requests:

EOT	0	1	0	2	ENQ
	<addr>		<code>		

KS 92/94 replies:

STX	0	1	=	D	ETX	BCC
	<code>			<val>		

<val> = 'D' means: remote, automatic, W_{int}, W2/ramp/programmer is off,
Y2 switch-over not active, no sensor fault.

Example 2:
The computer requests the active controller parameter set (code 20) from KS 92/94 with address 04.

Computer requests:

EOT	0	4	2	0	ENQ
	<addr>		<code>		

KS 92/94 replies:

STX	2	1	=	3	2	,	2	2	=	5	,	2	3	=	5	,	2	4	=	1	,		
	<code>			<X_{p1}>			<T_{n1}>					<T_{v1}>					<T1>						

2	5	=	3	2	,	2	6	=	5	,	2	7	=	5	,	2	8	=	1	ETX	BCC
<X_{p2}>						<T_{n2}>					<T_{v2}>					<T2>					

Example 3:
The computer transmits the W_{vol} (code 06) to KS 92/94 with address 02.

Computer transmits to KS 92/94:

EOT	0	2	STX	0	6	=	1	2	6	.	5	ETX
	<addr>			<code>			<val>					

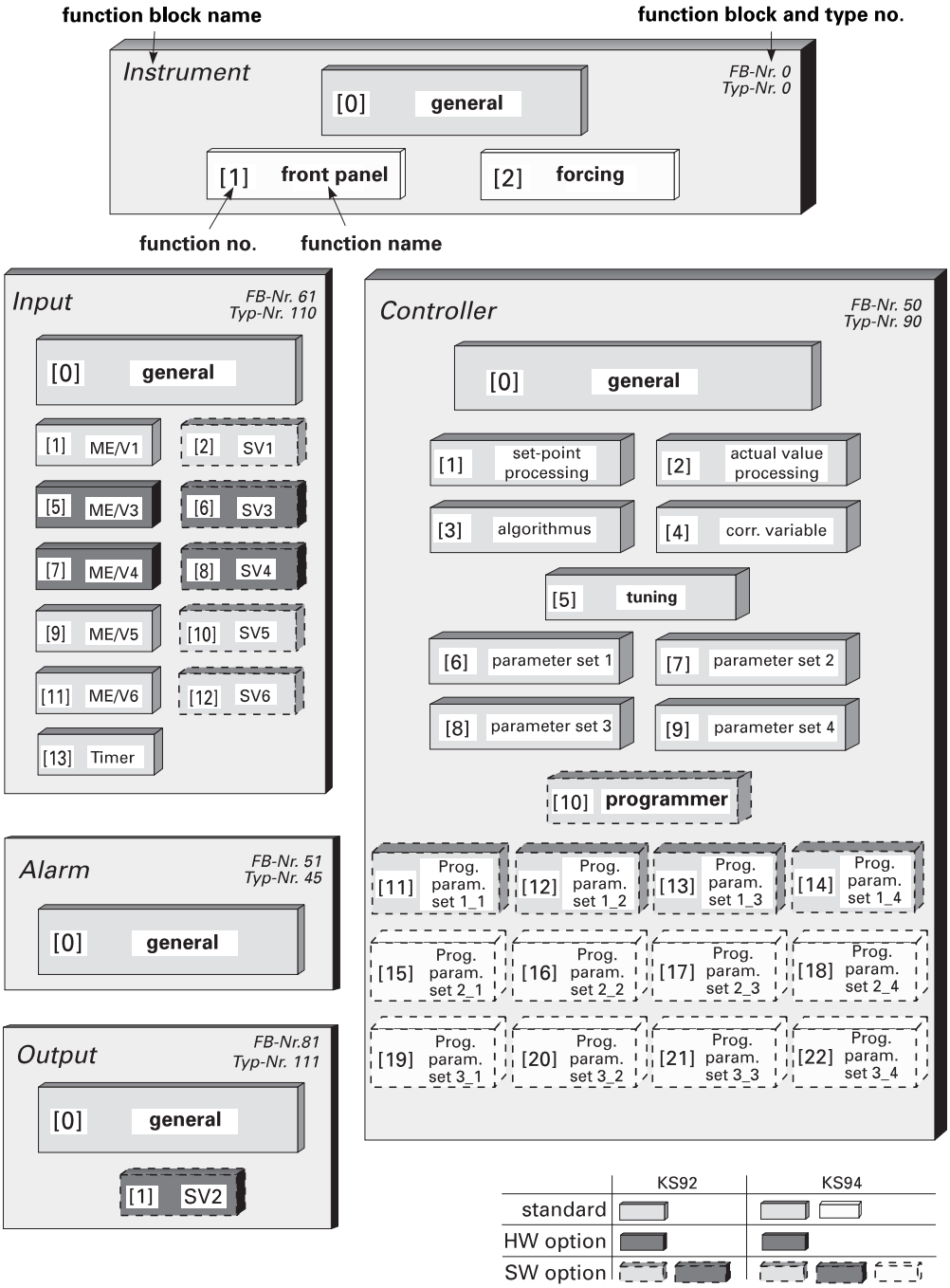
KS 92/94 replies: ACK or NAK in case of error

5 Function block protocol

5.1 Data structure

Due to the large variety of information to be processed in KS92/94, logically related data and actions are grouped into function blocks. Five function blocks are defined for KS92/94. They are addressed via fixed block addresses. Each block is divided into individual functions, which are provided dependent of HW or SW options. Functions are identified by numbers according to function blocks. Function number 0 addresses function block-specific data.

Fig.: 6 Survey of KS92/94 function blocks and functions



5.2 Function block protocol principles

A function block has input and output data (process data) as well as parameter and configuration data. It is addressable via a block number. A block type defines the relevant function.

The access mechanisms are:

5.2.1 *Individual access*

With this access (code xx), a single process value of a function can be read or written. Individual accesses to parameter and configuration data are not possible.

Example 1: (message structure with data sending)
Transmission of the absolute correcting variable (Y_{man}) to the controller.

Computer transmits data to KS92/94:

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

KS 92/94 replies: **ACK** or **NAK** in case of error

Example 2: (message structure with data request)
Reading the controller position feedback (Yp)

Computer requests:

EOT	0	2	1	3	,	5	0	,	0	**ENQ**
	Addr		code			FB no			fct-no.	

KS 92/94 replies:

STX	1	3	=	7	9	**ETX**	**BCC**
	code			value			

5.2.2 *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 the controller set-points (W_{nvol}, W_{vol} and dW).

Computer requests:

EOT	0	2	3	0	,	5	0	,	1	**ENQ**
	Addr		code			FB no.			fct.no.	

KS 92/94 replies:

STX	3	1	=	5	0	,	3	2	=	7	9	,	3	3	=	5	0	**ETX**	**BCC**
	code			value1			code			value2			code			value3			

5.2.3 *Block access (overall block)*

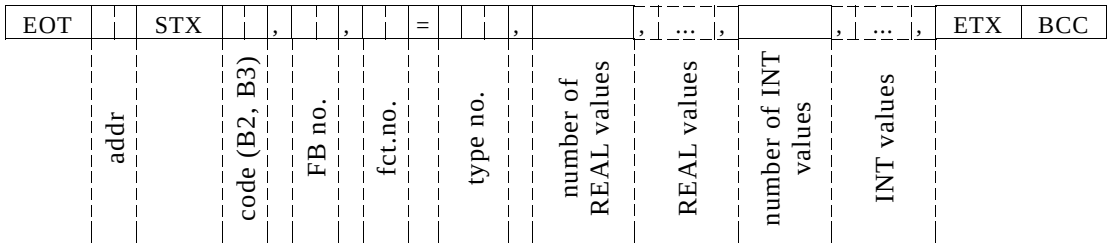
This access can be used for reading or writing all parameter (code 178) and configuration data (code 179) of a function. The following conditions are valid for this access:

- For writing data with 'Code B3', the instrument must be switched to the configuration mode (→ see page 24 'OpMod'). All newly entered configuration data and parameters are only effective, when the instrument is switched back to on-line.
- All data of a message must be defined, omissions are not permissible.
- If parts of a message in the instruments are not used (HW and SW options), the complete message must be transmitted. Checking of the non-existing data is omitted.
- With faulty block write accesses, the following rule is applicable: a message is replied with NAK, if at least 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 in the following. The exact message structure (between ***STX*** and ***ETX***) for the individual functions is given in conjunction with the relevant code table.

Message structure with data sending:

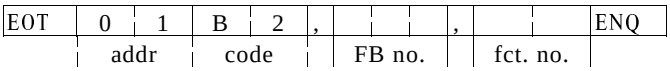
Computer transmits data to KS92/94:



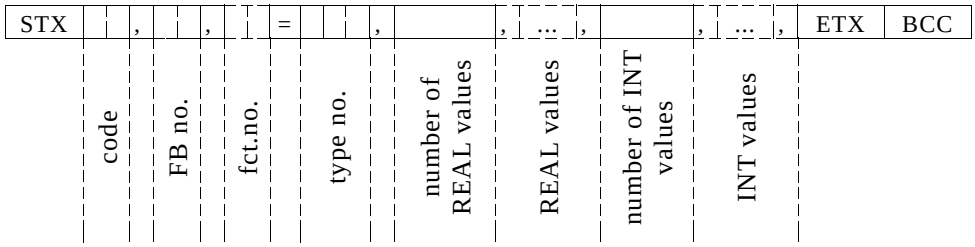
KS 92/94 replies: ACK or NAK in case of fault

Message structure with data request:

Computer requests:



KS92/94 replies:



5.2.4 Configuration words (Cxxx)

The configuration words given in the following code table comprise several components, which can be transmitted only in common.
The table data 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	**H**undreds	**Z**ens	**O**nes
Range	x	x	y	z
	00 ... 12		0...4	0...7
Example: continuous controller; standard controller;Set-point/cascade with offset	1 0 04			

5.3 CODE tables

5.3.1 *INSTRUMENT (FB no.: 0 type no.: 0)*

Function block 'INSTRUMENT' includes 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	RST1		Status 1		A
10	Block 13...15, 18	R	Block			
13	Write error	R	INT	Error of the last write access	0, 100...127	
14	Write error position	R	INT	Position of the last write error	0...99	
15	Read error	R	INT	Error of the last read access	0, 100...127	
18	Type	R	INT	Function block type no.	0	
20	Block 21...27	R	Block			
21	HWbas	R	INT	Basic HW options: module A, P		B
22	HWext	R	INT	Ext. HW options: module B, C		C
23	SWopt	R	INT	SW options 1		D
24	SWcod	R	INT	SW code no. 7th-10th digit of 12NC	wxyz	E
25	SWvers	R	INT	SW code no. 11th-12th digit of 12NC	00xy	F
26	OPVers	R	INT	Operating version		
27	EEPVers	R	INT	EEPROM version		
31	OpMod	R/W	INT	Switch-over instrument to config mode (only after 1)	0	
				Switch-over instrument to on-line mode (only after 0)	1	
				Preaking off the config mode (only after 0)	2	
33	UPD	R/W	INT	Reset of local data change	0..1	→ S. 16

Bem. A ***Unit_State1***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	R/L	Instrument status	remote	local
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***

	Instrument type (module A)		Output HW (module P)		
	T	H	Z	E	
KS92	01		01		relay: Out1,2,4,5
KS94	11		11		current: Out1, relay: Out2,4,5

Example: Value '*HWbas = 1111*' means that the addressed instrument is a KS94 with 3 relays and 1 current outputs. (12NC e.g. 9407 924xx xxx or 9407 928xx xxx).

Bem. C ***HWext***

Module B		Module C		
T	H	Z	E	
not fitted	00*	00*		not fitted
TTL interface(1)	01	01		A) 1 analog output (continuous) (OUT3)
RS485/422 interface(1)	02	02		B) 2 analog inputs (INP3, INP4)
OpenBus(Profibus)	10	04		C) 5 digital inputs (DI8...DI12 and 2 digital outputs (DO5...DO6)
		05		A + C
		06		B + C
		07		A + B + C

* Default setting

Example: Value '*HWext = 104*' means that the addressed instrument is fitted with a module B as TTL interface without real-time clock and a version C module C (12NC e.g. 9407 9xx16 xxx).

Bem. D ***SWopt conversion 12NC - 10th digit***

T				H				Z				E			
0	EXT	0	0	0	0	0	0	0	0	0	SOPT	0	PRG	MWK	SV

Descr.	**Status '0'**	**Status '1'**
SV	Signal processing disabled	Signal processing enabled
MWK	Measurement value correction f. temperature measurement disabled	Measurement value correction f. temperature measurement enabled
PRG	Programmer disabled	Programmer enabled
SOPT	Optimiz.when reaching set-p. disabled	Optimiz.when reaching set-p.enabled
EXT	Standard	Extension enabled

Example: Value '*SWopt = 13*' means that options signal processing, measurement value correction and optimization when reaching the set-point are enabled for the addressed instrument. (12NC e.g. 9407 9xxxx 3xx).

Bem. E ***SWCod***

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

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

Bem. F ***SWvers***

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

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

Bem. G ***UPD***

UPD is 1, if parameters or configuration data were changed by local operation or after Power On.

Forcing (function no.: 2)

Code	Descr.	R/W	Type	Description	Range	Rem.
30	Block 31...39	R	Block			
31	FInp 1	R/W	BCD	Forced Inp 1 (signal input before measurement value correction for INP1)		
32	FInp 3	R/W	BCD	Forced Inp 3 (signal input before signal pre-processing)		
33	FInp 4	R/W	BCD	Forced Inp 4 (signal input before signal pre-processing)		
34	FInp 5	R/W	BCD	Forced Inp 5 (signal input before signal pre-processing)		
35	FInp 6	R/W	BCD	Forced Inp 6 (signal input before signal pre-processing)		
36	Fdi	R/W	ICMP	Forced digital inputs di1...di12		**H**
37	FOut 1	R/W	BCD	Forced Out 1		**I**
38	FOut 3	R/W	BCD	Forced Out 3 (signal input before post-processing)		
39	Fdo	R/W	ICMP	Forced digital outputs Out 1...Out5; do1...do6 (serves also for output disabling with corresponding configuration. 0 = enabled; 1 = disabled)		**J**

Bem. H ***Data structure (Fdi)***

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Signification	0	0	0	0	di12	di11	di10	di9	di8	di7	di6	di5	di4	d.c.	di2	di1

Bem. I ***Range***

Dependent of configuration the force values for FOut1 and FOut3 are within the following ranges:

	Relay	Logic	Continuous
OUT1 (BCD)	d.c.	d.c.	-999 ... 9999
OUT1 (bit)	0 .. 1	0 .. 1	d.c.
OUT3 (BCD)	--	d.c.	-999 ... 9999
OUT3 (bit)	--	0 .. 1	d.c.

Bem. J ***Data structure (Fdo)***

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Signification	0	0	0	0	0	do6	do5	do4	do3	do2	do1	Out5	Out4	Out3	Out2	Out1

Parameter and configuration data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	F_{Key}	R/W	INT	Fct. of front-panel function key	0 .. 4	
	Lock	R/W	INT	Set-point blocking	0 .. 1	
	Disp2	R/W	INT	Selection display value Text2		
B3	C900 [1]	R/W	INT	Prot: Protocol type (T) Baud: Baud rate (H,Z)	0..xyy0	
	Addr[1]	R/W	INT	Instrument address	0..99	
	C902	R/W	INT	Freq: Mains frequency 50/60 (T)	0..x0yz	
	C800	R/W	INT	Text2: display Text2 (T) LED: Fct. front LEDs (H) Langu: Text language selection (Z)	0..xyz0	
	C801	R/W	INT	Unit: Unit selection (T,H)	00...xx00	

Message structure for function 'General'

Block access to parameter data																max. eff. length: 26 bytes	
STX	B2	,	0	,	0	=	0	,	0	,	3	,	FKey	,	Lock	,	Disp2	**ETX**	**BCC**

Block access to configuration data																			max. eff. length: 38 bytes	
STX	B3	,	0	,	0	=	0	,	0	,	5	,	C900	,	Addr	,	C902	,	C800	,	C801	**ETX**	**BCC**

[1] Baudrate and address setting are only effective after initialization, e.g. protocol switch-over.

Display texts			user-definable display texts only for KS94 [1]		(function no.: 1)	
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	String1	R/W	CHAR16	Display text 1	0x20...0x7F	
	String2	R/W	CHAR16	Display text 2	0x20...0x7F	
	String3	R/W	CHAR16	Display text 3	0x20...0x7F	
	String4	R/W	CHAR16	Display text 4	0x20...0x7F	
	String5	R/W	CHAR16	Display text 5	0x20...0x7F	
	String6	R/W	CHAR16	Display text 6	0x20...0x7F	
	String7	R/W	CHAR16	Display text 7	0x20...0x7F	
	String8	R/W	CHAR16	Display text 8	0x20...0x7F	
	String9	R/W	CHAR16	Display text 9	0x20...0x7F	
	String10	R/W	CHAR16	Display text 10	0x20...0x7F	
	String11	R/W	CHAR16	Display text 11	0x20...0x7F	
	String12	R/W	CHAR16	Display text 12	0x20...0x7F	
	Unit	R/W	CHAR5	User unit	20h...7Fh	

Message structure for function 'display texts'

Block access to parameter data																	max. eff. length: 224 bytes	
STX	B2	,	0	,	1	=	0	,	0	,	13	,	String1	,	String2	,	...	,	String12	,	Unit	**ETX**	**BCC**

String 1 to string 12 must contain 16 characters each and 'Unit'must contain 5 characters each (fixed data length!).

Forcing			Input and output forcing			(function no.: 2)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B3	C910	R/W	ICNF	Forcing INP1 (T) Forcing INP3 (Z) Forcing INP4 (E)	0..x0yz	
	C911	R/W	ICNF	Forcing INP5 (T) Forcing INP6 (H)	0...xy00	
	C920	R/W	ICNF	Forcing di1 (T) Forcing di2 (H) Forcing di4 (E)	0...wx0z	
	C921	R/W	ICNF	Forcing di5 (T) Forcing di6 (H) Forcing di7 (Z) Forcing di8 (E)	0...wxyz	
	C922	R/W	ICNF	Forcing di9 (T) Forcing di10 (H) Forcing di11 (Z) Forcing di12 (E)	0...wxyz	
	C930	R/W	ICNF	Forcing OUT1 (T) Forcing OUT2 (H) Forcing OUT3 (Z) Forcing OUT4 (E)	0...wxyz	
	C931	R/W	ICNF	Forcing OUT5 (T)	0...x000	
	C940	R/W	ICNF	Forcing do1 (T) Forcing do2 (H) Forcing do3 (Z) Forcing do4 (E)	0...wxyz	
	C941	R/W	ICNF	Forcing do5 (T) Forcing do6 (H)	0...wx00	

Block access to configuration data																			max. eff. length: 38 bytes	
STX	B3	,	0	,	0	=	0	,	0	,	9	,	C910	,	C911	,	C920	,	C922	,	C930	,	C931	,	C940	,	C941	**ETX**	**BCC**

[1] For transmitting the user texts via PROFIBUS-DP, a data block of min. 216 DW + management data is required.

Extended operating level				Parameter entry at the extended operating level (function no.: 3)		
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	Entry 1	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 2	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 3	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 4	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 5	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 6	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 7	R/W	INT	Parameter identification number	0 ... 9999	**K**
	Entry 8	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 9	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 10	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 11	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 12	R/W	INT	Parameter identification number	0 ... 9999	
	Hold	R/W	INT		0 ... 13	**L**

Message structure for function 'Extended operating level'

Block access to parameter data															max. eff. length: ??? bytes		
STX	B2	,	0	,	3	=	0	,	0	,	13	,	Entry1	,	Entry2	,	...	,	Entry12	,	Hold	**ETX**	**BCC**

Bem. K ***Entry 1 ... 12***

Value = 0 means 'unused entry'

Bem. L ***Hold***

Value = 0 means 'Hold on main operating page'
Value = 1 means 'Hold at status display at extended operating level'
Value = 2 ... 13 means ' Hold on entry 1 ... 12'

From firmware version 3.3 (October 1997), parameter setting of the extended operating level is possible via interface. The parameters are checked by the interface so that only valid parameters can be marked. Note, however, that a valid, marked parameter may be not displayed, because it is not displayed in the actual controller configuration by the operation. Example: `LimL1` can be marked via interface, but is only displayed, provided that alarm 1 is also configured.
The written parameters are effective immediately. After writing, an automatic change to the main operating page and after 1 minute to the entry marked with Hold is made.
For entry identification, see the following tables.

Data write/read are via the function block protocol, whereby the access mechanisms are:

Set-point	
Kennung	**Parameter**
257	w0
258	w100
259	w2
260	Grw+
261	Grw-
262	Grw2
263	LC-
264	LC+

Limit 1	
Kennung	**Parameter**
513	LimL1
514	LimH1
515	Lxsd1

Limit 2	
Kennung	**Parameter**
769	LimL2
770	LimH2
771	Lxsd2

Limit 3	
Kennung	**Parameter**
1025	LimL3
1026	LimH3
1027	Lxsd3

Limit 4	
Kennung	**Parameter**
1281	LimL4
1282	LimH4
1283	Lxsd4

Tuning	
Kennung	**Parameter**
1537	YOptm
1538	dYopt
1539	POpt
1540	Oxsd
1541	Trig1
1542	Trig2
1543	Trig3
1544	ORes1
1545	ORes2
1546	Tu1
1547	Vmax1
1548	Tu2
1549	Vmax2

Addit. param.	
Kennung	**Parameter**
1793	Xsh
1794	Tpuls
1795	Tm
1796	Xsd1
1797	LW
1798	Xsd2
1799	Xsh1
1800	Xsh2
1801	Y2
1802	Ymin
1803	Ymax
1804	Y0
1805	ParNr
1806	ParNr (read-only)

Valid parameter	
Kennung	**Parameter**
2049	Xp1
2050	Xp2
2051	Tn1
2052	Tv1
2053	T1
2054	T2

Parameter set 0	
Kennung	**Parameter**
2305	Xp1 0
2306	Xp2 0
2307	Tn1 0
2308	Tv1 0
2309	T1 0
2310	T2 0

Parameter set 1	
Kennung	**Parameter**
2561	Xp1 1
2562	Xp2 1
2563	Tn1 1
2564	Tv1 1
2565	T1 1
2566	T2 1

Parameter set 2	
Kennung	**Parameter**
2817	Xp1 2
2818	Xp2 2
2919	Tn1 2
2820	Tv1 2
2821	T1 2
2822	T2 2

Parameter set 3	
Kennung	**Parameter**
3073	Xp1 3
3074	Xp2 3
3075	Tn1 3
3076	Tv1 3
3077	T1 3
3078	T2 3

Istwert	
Kennung	**Parameter**
3329	Tdz
3330	N0
3331	a
3332	b

Signl. process. INP 1	
Kennung	**Parameter**
3585	X1in
3586	X1out
3587	X2in
3588	X2out
3589	m
3590	b
3591	gain
3592	Tf

Signl. process. INP 3	
Kennung	**Parameter**
3841	m 3
3842	b 3
3843	gain 3
3844	Tf 3

Signl. process. INP 4	
Kennung	**Parameter**
4097	m 4
4098	b 4
4099	gain 4
4100	Tf 4

Signl. process. INP 5	
Kennung	**Parameter**
4353	m 5
4354	b 5
4355	gain 5
4356	Tf 5

Signl. process. INP 6	
Kennung	**Parameter**
4609	m 6
4610	b 6
4611	gain 6
4612	Tf 6

Sonstiges	
Kennung	**Parameter**
4865	FKey
4866	Blck1
4867	Blck2

Timer	
Kennung	**Parameter**
5121	TS.Y
5122	TS.MD
5123	TS.HM
5124	TE.Y
5125	TE.MD
5126	TE.HM

Analog prog. Recp 1	
Kennung	**Parameter**
6401	Wmode
6402	Pmode
6403	Pnext
6404	LC-
6405	LC+
6406	Wp0
6407	----
6408	Tp1
6409	Wp1
6410	Tp2
6411	Wp2
...	...
6446	Tp20
6447	Wp20

Digital prog. Recp 1	
Kennung	**Parameter**
6657	D0
6658	----
6659	Td1
6660	D1
6661	Td2
6662	D2
...	...
6697	Td20
6698	D20

Analog prog. Recp 2	
Kennung	**Parameter**
6913	Wmode
6914	Pmode
6915	Pnext
6916	LC-
6917	LC+
6918	Wp0
6919	----
6920	Tp1
6921	Wp1
6922	Tp2
6923	Wp2
...	...
6958	Tp20
6959	Wp20

Digital prog. Recp 2	
Kennung	**Parameter**
7169	D0
7170	----
7171	Td1
7172	D1
7173	Td2
7174	D2
...	...
7209	Td20

Digital prog. Recp 2	
Kennung	**Parameter**
7210	D20

Analog prog. Recp 3	
Kennung	**Parameter**
7425	Wmode
7426	Pmode
7427	Pnext
7428	LC-
7429	LC+
7430	Wp0
7431	----
7432	Tp1
7433	Wp1
7434	Tp2
7435	Wp2
...	...
7470	Tp20
7471	Wp20

Digital prog. Recp 2	
Kennung	**Parameter**
7681	D0
7682	----
7683	Td1
7684	D1
7685	Td2
7686	D2
...	...
7721	Td20
7722	D20

Signals	
Kennung	**Parameter**
7937	----
7938	----
7939	----
7940	Clock
5125	TE.MD
5126	TE.HM

Set-point signals	
Kennung	**Parameter**
8193	Wint
8194	Wext
8195	dWext
8196	dW (Dec. point = 1)
8197	dW (Dez. Punkt = 2)
8198	Wsel
8199	Weff

Controller signals	
Kennung	**Parameter**
8449	Y
8450	Yp
8451	xw
8452	X1
8453	X2
8454	X3
8455	z
8456	OVC
8457	Xeff

Input signals	
Kennung	**Parameter**
8705	INP1
8706	INP1r
8707	INP3
8708	INP3r
8709	INP4
8710	INP4r
8711	INP5
8712	INP5r
8713	INP6
8714	INP6r

Prog. Signale	
Kennung	**Parameter**
8961	Wp
8962	tBrut
8963	tNet
8964	tRest
8965	PNr

Rapid recovery	
Kennung	**Parameter**
9217	XwOnY
9218	XwOnX
9219	GrwOn

Calibration INP1	
Kennung	**Parameter**
9473	x0c (PT100)
9474	x0c
9475	x100c

Calibration INP6	
Kennung	**Parameter**
9729	x0c
9730	x100c

5.3.2 INPUT (FB no.: 61 Type no.: 110)

All data concerning the acquisition and processing of all input values (analog/digital) are grouped in function block 'INPUT'.

Process data

General — Input processing of analog, digital signals (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block access (1, 3, 5...8)		
1	Input_x_Fail	R	ST1	Signal Input x Fail		A
3	INP1	R	BCD	Signal input 1		
5	INP3	R	BCD	Signal input 3		
6	INP4	R	BCD	Signal input 4		
7	INP5	R	BCD	Signal input 5		
8	INP6	R	BCD	Signal input 6		
10	Block	R	Block	Block access (13...18)		
11	State_di1	R	ST1	digital input di1 ... di6		B
12	State_di2	R	ST1	digital input di7 ... di12		C
13	INP1A [1)]	R	BCD	Signal input 1 physical value		
14	INP3A [1)]	R	BCD	Signal input 3 physical value		
15	INP4A [1)]	R	BCD	Signal input 4 physical value		
16	INP5A [1)]	R	BCD	Signal input 5 physical value		
17	INP6A [1)]	R	BCD	Signal input 6 physical value		
18	Function type	R	INT	Function block type no.	110	

Bem. A ***Status byte 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	'0'	always '0'		
D2	INP3F	Input 3 fail	no	yes
D3	INP4F	Input 4 fail	no	yes
D4	INP5F	Input 5 Fail	no	yes
D5	INP6F	Input 6 Fail	no	yes
D6	'1'	always '1'		
D7		Parity		

Bem. B ***State_di1 (digital inputs di1 ... di6)***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	di1	Digital input 1	off	on
D1	di2	Digital input 2	off	on
D2	di3	Digital input 3	off	on
D3	di4	Digital input 4	off	on
D4	di5	Digital input 5	off	on
D5	di6	Digital input 6	off	on
D6	'1'	Always '1'		
D7		Parity		

Bem. C ***State_di2 (digital inputs di7 ... di12)***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	di7	Digital input 7	off	on
D1	di8	Digital input 8	off	on
D2	di9	Digital input 9	off	on
D3	di10	Digital input 10	off	on
D4	di11	Digital input 11	off	on
D5	di12	Digital input 12	off	on
D6	'1'	Always '1'		
D7		Parity		

[1)] Values before signal pre-processing or measurement value correction

ME/V1				Measurement value INP1 : Detection and processing		(function no.: 1)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
31	X0c	R/W	INT	Trigger f. calibration X0	0..1	
32	X100c	R/W	INT	Trigger f. calibration X100	0..1	

ME/V6				Measurement value INP6 : Acquisition and processing		(Function no.: 11)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
31	X0c	R/W	INT	Trigger f. calibration X0	0..1	
32	X100c	R/W	INT	Trigger f. calibration X100	0..1	

Timer				Timer function(1)		(function no.: 13)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Bereich**	**Rem.**
01	State_Clock	L	ST1	Status 1		B
30	Block	R	Block	Block access (code 31...36)		
31	ClkH(2)	R/W	INT	Time hours	0...23	
32	ClkMi(2)	R/W	INT	Time minutes	0...59	
33	ClkD(2)	R/W	INT	Time day	1...31	
34	ClkMt(2)	R/W	INT	Time month	1...12	
35	ClkY(2,3)	R/W	INT	Time year	0...255	
36	ClkDW(2,4)	R/W	INT	Time weekday	0...6	

Bem. D ***Status byte State_Clock***

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

Bit no.	**Name**	**Allocation**	**Status '0'**	**Status '1'**
D0	ClkEr	Clock error	no	yes
D1	T1En	Timer 1 enable		
D2	T1Out	Timer 1 state		
D3...D5	'0'	always '0'		
D6	'1'	always '1'		
D7		Parity		

T1Out	T1En	
0	0	timer not active
0	1	enabled
1	0	output active
1	1	not defined

Parameter a. configuration data

General				Input processing analog, digital signals		(function no.: 0)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B3	C180	R/W	INT	S X2: signal source for S2 (T) SWext: signal source for Wext (H) S dW: signal source for dW (Z) S z: signal source for z (E)	0..xyz0	
	C190	R/W	INT	SWi/e: signal source for W_{int}/W_{ext} . (T) STrac: signal source for WTrac (H) SWdon:signal source for dw on/off (Z) Sw/W2:signal source for w/w2 (E)	0..wxyz	
	C191	R/W	INT	S A/M: signal source for auto/manual (T) SPI/P: signal source for FB on/off (H) SY2on: signal source for Y2 (Z) SCoff: signal source for controller off (E)	0..wxyz	
	C192	R/W	INT	Prog: signal source for start/stop (T)	0.x000	

Message structure for function 'General'

Block access to configuration data																max. eff. length: 35 bytes	
STX	B3	,	61	,	0	=	110	,	0	,	4	,	C180	,	C190	,	C191	,	C192	**ETX**	**BCC**

(1) Only valid with module B with real-time clock, e.g. RS485 (d.c. for PROFIBUS)
(2) The actual, internally available time is specified when reading. The order 'year-month-day-hour-minute' must be followed for correct checking.
(3) Calculation of the actual year: data range 70...169, corresponds to 1970...2069; example: value 96 corresponds to year 1996, value 101 corresponds to year 2001.
(4) Signification: 0=Monday (first weekday), 1=Tuesday,...6= Sunday: values can be specified freely

ME/V1				Measurement value INP1 : acquisition and processing		(function no.: 1)
Code	**Descr.**	**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 with sensor fail	-999..9999	
	T_{fm}	R/W	BCD	Filter time constant. meas.value proc..	0 .. 999.9	
	T_{kref}	R/W	BCD	Reference TC	-99.9..100 °C	
	C200	R/W	INT	Type: sensor type (T,H) Unit: unit (Z) Dp: decimal point (E)	0..xxyy	
	C205	R/W	INT	Fail: sensor break behaviour (T) STk: source Tk (H) XKorr: Process value corr. enable (Z) Dp: decimal point (E)	1..wxyz	

Message structure for function 'ME/V1'

Block access to parameter data																	max. eff. length: 43 bytes
STX	B2	,	61	,	1	=	110	,	4	,	X1in	,	X1out	,	X2in	,	X2out	,	0	**ETX**	**BCC**

Block access to configuration data																					max. eff. length: 60 bytes
STX	B3	,	61	,	1	=	110	,	5	,	X0	,	X100	,	XFail	,	Tfm	,	Tkref	,	2	,	C200	,	C205	**ETX**	**BCC**

Sv1				Signal processing stage for INP1		(function no.: 2)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	m	R/W	BCD	Scaling: gradient m	0..999.9	
	b	R/W	BCD	Scaling: offset b	-999..9999	
	gain	R/W	BCD	Square root extraction: gain	0 .. 9.999	
	Tf	R/W	BCD	Pre-processing: filter time constant	0 .. 999.9	
B3	xs1	R/W	BCD	segment point 1: X value	-999..9999	
	ys1	R/W	BCD	segment point 1: Y value	-999..9999	
	xs2	R/W	BCD	segment point 2: X value	-999..9999	
	ys2	R/W	BCD	segment point 2: Y value	-999..9999	
	xs3	R/W	BCD	segment point 3: X value	-999.. 9999[1]	
	ys3	R/W	BCD	segment point 3: Y value	-999.. 9999	
	xs4	R/W	BCD	segment point 4: X value	-999.. 9999[1]	
	ys4	R/W	BCD	segment point 4: Y value	-999.. 9999	
	xs5	R/W	BCD	segment point 5: X value	-999.. 9999[1]	
	ys5	R/W	BCD	segment point 5: Y value	-999.. 9999	
	xs6	R/W	BCD	segment point 6: X value	-999.. 9999[1]	
	ys6	R/W	BCD	segment point 6: Y value	-999.. 9999	
	xs7	R/W	BCD	segment point 7: X value	-999.. 9999[1]	
	ys7	R/W	BCD	segment point 7: Y value	-999.. 9999	
	xs8	R/W	BCD	segment point 8: X value	-999.. 9999[1]	
	ys8	R/W	BCD	segment point 8: Y value	-999.. 9999	
	C220	R/W	INT	Func1: function selection 1 (T) Func2: function selection 2 (H) LDP: decimal point (E)	0..wx0z	

Message structure for function 'Sv1'

Block access to parameter data																	max. eff. length: 43 bytes
STX	B2	,	61	,	2	=	110	,	4	,	m	,	b	,	gain	,	Tf	,	0	**ETX**	**BCC**

Block access to configuration data																										max. eff. length: 133 bytes
STX	B3	,	61	,	2	=	110	,	16	,	xs1	,	ys1	,	xs2	,	ys2	,	...	xs8	,	ys8	,	1	,	C220	**ETX**	**BCC**

[1] Datum with switch-off function; additional data value '-32000'

The functions for measurement value processing and acquisition of inputs INP3, INP4, INP5, INP6 are structured identically. INP3 and INP4 are only available with options p.c.b. C fitted.

ME/Vx	Measurement value INPx: acquisition a. processing				(function no.: 5, 7, 9, 11)	
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B3	X0	R/W	BCD	Phys. value at 0%	-999..9999	
	X100	R/W	BCD	Phys. value at 100%	-999..9999	
	XFail	R/W	BCD	Substitute value with sensor fail	-999..9999	
	Tfm	R/W	BCD	Filter time constant meas.value processsing	0 .. 999.9	
	INPx (Cxx0)	R/W	INT	Type: sensor type (T,H) Dp: decimal point (E)	0..xxyy	
	Option (Cx05)	R/W	INT	Fail: sensor break behaviour (T)	0..x00y	

Message structure for function 'ME/Vx'

Block access to configuration data													max. eff. length: 54 bytes
STX	B3	,	61	,	<5, 7, 9, 11>	=	110	,	4	,	X0	,	X100	,	XFail	,	Tfm	,
2	,	INPx	,	Option	**ETX**	**BCC**												

The functions for signal pre-processing of inputs INP3, INP5, INP6 are structured identically. INP3 is available only with options p.c.b. C fitted.

Svx	Signal processing level for INPx				(function no.: 6,8,10,12)	
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	m	R/W	BCD	Scaling: gradient m	0 .. 999.9	(1)
	b	R/W	BCD	Scaling: offset b	-999..9999	
	gain	R/W	BCD	Square root extraction: gain	0 .. 9.999	
	Tf	R/W	BCD	Pre-processing: filter time constant	0 .. 999.9	
B3	xs1	R/W	BCD	Segment point 1 : X value	-999..9999	(2, 3)
	ys1	R/W	BCD	Segment point 1: Y value	-999..9999	
	xs2	R/W	BCD	Segment point 2: X value	-999..9999	
	ys2	R/W	BCD	Segment point 2: Y value	-999..9999	
	xs3	R/W	BCD	Segment point 3: X value	-999.. 9999	
	ys3	R/W	BCD	Segment point 3: Y value	-999.. 9999	
	xs4	R/W	BCD	Segment point 4: X value	-999.. 9999	
	ys4	R/W	BCD	Segment point 4: Y value	-999.. 9999	
	xs5	R/W	BCD	Segment point 5: X value	-999.. 9999	
	ys5	R/W	BCD	Segment point 5: Y value	-999.. 9999	
	xs6	R/W	BCD	Segment point 6: X value	-999.. 9999	
	ys6	R/W	BCD	Segment point 6: Y value	-999.. 9999	
	xs7	R/W	BCD	Segment point 7: X value	-999.. 9999	
	ys7	R/W	BCD	Segment point 7: Y value	-999.. 9999	
	xs8	R/W	BCD	Segment point 8: X value	-999.. 9999	
	ys8	R/W	BCD	Segment point 8: Y value	-999.. 9999	
	Cxx0	R/W	INT	Func1: function selection 1 (T) Func2: function selection 2 (H) NSeg: number of segments (Z) LDP: decimal point (E)	0..wxyz	(1, 2)

Message structure for function 'Svx'

Block access to parameter data																max. eff. length: 44 bytes
STX	B2	,	61	,	<6,8,10,12>	=	110	,	4	,	m	,	b	,	gain	,	Tf	,	0	**ETX**	**BCC**

Block access to configuration data																					max. eff. length: 139 bytes
STX	B3	,	61	,	<6,8,10,12>	=	110	,	16	,	xs1	,	ys1	,	xs2	,	ys2	,	...	xs8	,	ys8	,	1	,	Cxx0	**ETX**	**BCC**

(1) Defined only for
KS92: for input 5 and 6 (function 10,12)
KS94: for input 3,4 (if C option is fitted), 5 and 6 (function 6,8,10 and 12)
(2) Only defined, if the relevant SW option is enabled.
(3) Defined only for KS94, INP4, if C option is fitted

Timer				Timer function [1]		(function no: 13)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	T1SY	R/W	INT	Timer 1 start value year	0...255	(2)
	T1SMt	R/W	INT	Timer 1 start value month	1..12	
	T1SD	R/W	INT	Timer 1 start value day	1 .. 31	
	T1SH	R/W	INT	Timer 1 start value hours	0 .. 23	
	T1SMi	R/W	INT	Timer 1 start value minutes	0 .. 59	
	T1EY	R/W	INT	Timer 1 end value year	0...255	(2)
	T1EMt	R/W	INT	Timer 1 end value month	1..12	
	T1EH	R/W	INT	Timer 1 end value hours	0 .. 23	
	T1EMi	R/W	INT	Timer 1 end value minutes	0 .. 59	
B3	C905	R/W	INT	TmMd: Mode timer 1 (T)	0..x000	

Message structure for function 'Timer'

Block access to parameter data																max. eff. length: 68 Bytes
STX	B2	,	61	,	13	=	110	,	0	,	10	,	T1SY	,	T1SMt	,
T1SD	,	T1SH	,	T1SMi	,	T1EY	,	T1EMt	,	T1ED	,	T1EH	,	T1EMi	**ETX**	**BCC**

Block access to configuration data												max. eff. length: 22 bytes
STX	B3	,	61	,	13	=	110	,	0	,	1	,	C905	**ETX**	**BCC**

[1] Only defined for module B with real-time clock (d.c. for PROFIBUS).
[2] Calculation of actual year: data range 70...69, corresponding to 1970...2069; example: value 96 corresponds to year 1996, value 01 corresponds to year 2001.

5.3.3 CONTR (FB no.: 50 Type no.: 90)

All data concerning the controller are grouped in function block 'CONTR'.

Process data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block access (1…9)		
1	Status 1	R	ST1	Status 1		A
2	Status 2	R	ST1	Status 2		B
3	W	R	BCD	Eff. set-point		
4	X	R	BCD	Eff. process value		
5	Y	R	BCD	Eff. correcting variable		
6	xw	R	BCD	Control deviation		
7	x1	R	BCD	Variable 1		
8	x2	R	BCD	Auxiliary variable 2		
9	x3	R	BCD	Auxiliary variable 3		
10	Block	R	Block	Block access (11, 13…16, 18)		
11	Status 3	R	ST1	Status 3		C
13	Yp	R	BCD	Position feedback		
14	z	L	BCD	Auxiliary variable		
15	OVC+	R	BCD	Override control +		
16	OVC-	R	BCD	Override control -		
18	Type	R	INT	Function block type no.	90	
20	Block	R	Block	Block access (21…23)		
21	Wext	R	BCD	Ext. set-point		
22	dW_extern	R	BCD	Set-point offset		
23	Wlim	R	BCD	Set-point limiting W min/max		
30	Block	R	Block	Block access (31…38)		
31	y/Y2	R/W	INT	Addition. corr. value on/off	0..1	
32	PI/P	R/W	INT	PI/P-switch-over	0..1	
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	
37	w/dW	R/W	INT	Correction set-point off/on	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	Auto/manual	auto	manual
D3	y/Y2	y/Y2 switch-over	y	Y2
D4	Coff	Controller switched off	no	yes
D5	XFail	Sensor fail	no	yes
D6	'1'	always '1'		
D7		Parity		

(1) Datum with switch-off function; additional data value '-32000'

Bem. B ***Status2: (code 02)***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	GRW	Gradient function	inactive	active
D1	BAND	Bandwidth regulation	inactive	active
D2	RCV	Rapid recovery	no	yes
D3	'0'	always '0'		
D4	PI/P	Status PI/P	PI	P
D5	CFail	Controller status	Ok	not Ok
D6	'1'	always '1'		
D7		Parity		

Bem. C ***Status 3: (code 11)***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	Xtrk	Int. set-point with X	off	on
D1	DOVC-	Override control- with 3-pnt.step.controller	off	on
D2	DOVC+	Override control+ with 3-pnt.step.controller	off	on
D3...D5	'0'	always '0'		
D6	'1'	always '1'		
D7		Parity		

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

Code	Descr.	R/W	Type	Description	Range	Rem.
1	WState	R	ST1	set-point status		D
30	Block	R	Block	Block access (31...33)		
31	Wnvol	R/W	BCD	Int. set-point, non volatile	-999..9999	
32	Wvol	R/W	BCD	Int. set-point, volatile	-999..9999	
33	WdW	R/W	BCD	Offset set-point / correction value	-999..9999	

Bem. D ***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		
D2	Wp/Wi	Wprog/Wint		
D3	w/dW	set-point correction active		
D4	w/dWe	ext. set-point correction active		
D5	'0'	always '0'		
D6	'1'	always '1'		
D7		Parity		

wp/wi	we/wi	Set-point
0	1	external
1	0	Programmer
1	1	internal

w/dwe	w/dw	set-point correction
0	0	inactive
0	1	active
1	0	active, external

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

Code	Descr.	R/W	Type	Description	Range	Rem.
30	Block	R	Block	Block access (31, 32)		
31	dYman	R/W	BCD	differenc. correcting variable	-210..210	
32	Yman	R/W	BCD	absolute correcting variable	-105..105	

Tuning				Self-tuning		(function no.:5)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
00	Block	R	Block	Block access (1, 3)		
1	State_Tune1	R	ST1	Status Tuning		E
3	ParNeff	R	INT	Eff. parameter set number	0...3	
30	Block	R	Block	Block access (31...39)		
31	ParNr	R/W	INT	Parameter set number effective	1 .. 4	
32	Tu1	R	BCD	Delay time heating	0...9999	
33	Vmax1	R	BCD	Rate of increase heating	0,000...9,999	
34	Kp1	R	BCD	Process gain heating	0,000...9,999	
35	MSG1	R	INT	Error code self-tuning heating	0...8	
36	Tu2	R	BCD	Delay time cooling	0...9999	
37	Vmax2	R	BCD	Rate of increase cooling	0,000...9,999	
38	Kp2	R	BCD	Process gain cooling	0,000...9,999	
39	MSG2	L	INT	Error code self-tuning cooling	0...8	

Bem. E ***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	Operation self-tuning	off	on
D2	Oerr	Result self-tuning	Ok	error
D3...D5	'0'	always '0'		
D6	'1'	always '1'		
D7		Parity		

Programmer				Programmer processing		(function no.: 10)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
00	Block	R	Block	Block access (1...9)		
1	State_Prog1	R	ST1	Programmer status 1		F
2	State_Prog2	R	ST1	Programmer status 2		G
3	PNreff	R	BCD	Eff. program number	1..3	
4	Tnet	R	BCD	Net program time	0 .. 9999	
5	Tbrut	R	BCD	Gross program time	0 .. 9999	
6	Wp	R	BCD	Programmer set-point	-999 .. 9999	
7	Trest	R	BCD	Programmer rest time	0 .. 9999	
8	Wend	R	BCD	Act. segment end value	-999 .. 9999	
9	Seg AD	R	INT	Segment no. analog/digital	0000 .. 1919	
30	Block	R	Block	Block access (31...35)		
31	Pnr	R/W	INT	Program number effective	1 .. 3	
32	PRun	R/W	INT	Program stop/run	0 .. 1	
33	PRset	R/W	INT	Programm continue / reset	0 .. 1	
34	PSearch	R/W	INT	Start program search mode	0 .. 1	
35	PSet	R/W	BCD	Program preset value Pmode = Seg Pmode = Time	1...20 0...9999(min)	
36	LC-	R/W	BCD	min. limit bandwidth	-999...9999	(1)
37	LC+	R/W	BCD	max. limit bandwidth	-999...9999	

Bem. F ***Status 1 programmer 'State_Prog1'***

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

Bit no.	**Name**	**Allocation**	**Status '0'**	**Status '1'**
D0	PRun	Program run	Stop	running
D1	PEnd	Program end	no	yes
D2	PRes	Program reset	off	on
D3...D5	'0'	alwas '0'		
D6	'1'	always '1'		
D7		Parity		

(1) LC-/LC+ are used as set-point gradient. These data are effective only if programmer is not configured although a software option for programmer is provided. These data are identical with LC-/LC+ of the first program (Program x_1)

Bem. G ***Status 2 programmer 'State_Prog2'***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	Sp1	Output 1	off	on
D1	Sp2	Output 2	off	on
D2	Sp3	Output 3	off	on
D3	Sp4	Output 4	off	on
D3	'0'	always '0'		
D6	'1'	always '1'		
D7		Parity		

Parameter and configuration data

General

(function no.: 0)

Code	Descr.	R/W	Type	Description		Range	Rem.
B2	Xwonx	L/S	BCD	X-W limit value (X-W < Xwony → Y tracking)		0 .. 9999	
	Xwony	L/S	BCD	X-W limit value (X-W > Xwonx → X tracking)		0 .. 9999	
	Grwon	L/S	BCD	Set-point gradient with X tracking active		0,01 .. 99,99 /min	
B3	C100	R/W	INT	CFunc: Controller function	(T,H)	0..xxyz	
				CType: Controller type	(Z)		
				WFunc:set-point functions	(E)		
	C101	R/W	INT	CMode:controller output action	(T)	0..xyz0	
				CDiff: x, Xw-differenciat.	(H)		
				CFail: behaviour with sensor fail	(Z)		
	C102	R/W	INT	XnDp: Decimal point f. X-reference	(E)	0..000x	
	C103(Xn0)	R/W	BCD	Min. reference X1		-999..9999	
	C104(Xn100)	R/W	BCD	Max. reference limit X1		-999..9999	
	C105	R/W	INT	C Aux: auxiliary variable	(T,H)	0.. xxy0	
				C OVC:corr.variable limiting.	(Z)		
	C106	R/W	INT	WTrac: Tracking int. set-point	(T)	0..xyz0	
				WD:set-point offset	(H)		
				WSel: set-point selection	(Z)		
	C107	R/W	INT	Ratio: ratio control fct.	(T)	0..xyyz	
				XDp: decimal point f. process value	(E)		
	C108(Xmin)	R/W	BCD	Min. process value limit		-999..9999	
	C110	R/W	BCD	S factor		0.01 .. 99.99	
	C700	R/W	INT	OMode:Self-tuning mode	(T)	0..xy0z	
				OCond:Process at rest	(H)		
				OCntr: Controlled adaptation mode.	(E)		

Message structure for function 'General'

Block access to parameter data														max. eff. length: 35 bytes			
STX	B2	,	50	,	1	=	90	,	3	,	Xwonx	,	Xwony	,	Grwon	,	0	**ETX**	**BCC**

Block access to configuration data																max. eff. length: 84 bytes			
STX	B3	,	50	,	0	=	90	,	5	,	C103	,	C104	,	C108	,	C109	,	C110	,
	7	,	C100	,	C101	,	C102	,	C105	,	C106	,	C107	,	C700		**ETX**		**BCC**

Set-point

Set-point processing (function no.: 1)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	W0	R/W	BCD	min. set-point limit f. Weff	-999..9999	
	W100	R/W	BCD	max. 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	(1)
	Grw-	R/W	BCD	Set-point gradient minus	>0..9.999	
	Grw2	R/W	BCD	Set-point gradient W2	>0..9.999	

Message structure for function 'set-point'

Block access to parameter data																	max. eff. length: 76 bytes		
STX	B2	,	50	,	1	=	90	,	6	,	W0	,	W100	,	W2	,	Grw+	,	Grw-	,
	Grw2	,	0		**ETX**		**BCC**													

(1) Datum with switch-off function; additional data value '-32000'

Process value — Measurement value processing (function no.: 2)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	N0	R/W	BCD	Zero offset/behaviour	-999..9999	
	a	R/W	BCD	Factor a / 3-element control	-9.99..99.99	
	b	R/W	BCD	Factor b / Mean value control	0..9.999	
	Tdz	R/W	BCD	Differenciation time constant for signal	0..9999	

Message structure for function 'process value'

Block access to configuration data																max. eff. length: 44 bytes
STX	B2	,	50	,	2	=	90	,	4	,	NO	,	a	,	b	,	Tdz	,	0	**ETX**	**BCC**

Algo — Control algorithm (function no.: 3)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Xsh	R/W	BCD	Neutral zone	0.2 .. 999.9%	
	Tpuls	R/W	BCD	Min. pulse length	0.1..999.9%	1)
	Tm	R/W	BCD	Actuator response time	10..9999s	
	Xsd1	R/W	BCD	Signaller switching difference	0.1..9999%	
	LW	R/W	BCD	Trigger point separation add.contact	-999..9999	
	Xsd2	R/W	BCD	Switching difference add.contact	0.1..9999%	
	Xsh1	R/W	BCD	Neutral zone	0.0 .. 999.9%	
	Xsh2	R/W	BCD	Neutral zone	0.0 .. 999.9 %	
	Xp1	R/W	BCD	act. proportional band 1	0.1..999.9	
	Tn1	R/W	BCD	act. integral time 1	0..9999	
	Tv1	R/W	BCD	act. derivative time 1	0..9999	
	T1	R/W	BCD	act. min. duty cycle 1	0.4..999.9	
	Xp2	R/W	BCD	act. proportional band 2	0.1..999.9	
	Tn2	R/W	BCD	act. integral time 2	0..9999	
	Tv2	R/W	BCD	act. derivative time 2	0..9999	
	T2	R/W	BCD	act. min. duty cycle 2	0.4..999.9	

Message structure for function 'Algo'

Block access to parameter data																										max. eff. length: 127 bytes
STX	B2	,	50	,	3	=	90	,	16	,	Xsh	,	Tpuls	,	Tm	,	Xsd1	,	LW	,	Xsd2	,	Xsh1	,		
	X_{sh2}	,	X_{p1}	,	T_{n1}	,	T_{v1}	,	T_1	,	X_{p2}	,	T_{n2}	,	T_{v2}	,	T2	,	0	**ETX**	**BCC**					

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

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Y2	R/W	BCD	Additional correcting value	-105..105	
	Y_{min}	R/W	BCD	min. correcting variable limiting	-105..105	
	Y_{max}	R/W	BCD	max. correcting variable limiting	-105..105	
	Y0	R/W	BCD	Working point f. correcting variable	-105..105	

Message structure for function 'correcting variable'

Block access to parameter data																		max. eff. length: 42 bytes
STX	B2	,	50	,	4	=	90	,	4	,	Ymin	,	Ymax	,	Y2	,	Y0	,	0	**ETX**	**BCC**

Tuning — Self-tuning (function no.: 5)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	YOptm	R/W	BCD	Corr.variable during process at rest	-105..105	
	dYopt	R/W	BCD	Step width with identification	5..100	
	OXsd	R/W	BCD	Hysteresis at parameter switch-over	0.0..9999	
	Trig1 2)	R/W	BCD	Trigger point 1	0.0..9999	
	Trig2 2)	R/W	BCD	Trigger point 2	0.0..9999	
	Trig3 2)	R/W	BCD	Trigger point 3	0.0..9999	
	POpt	R/W	INT	Parameter set to be optimized	0...3	

Message structure for function 'Tuning'

Block access to parameter data																						max. eff. length: 128 bytes
STX	B2	,	50	,	5	=	90	,	6	,	YOptm	,	dYopt	,	OXsd	,	Trig1	,	Trig2	,	Trig3	,
1	,	POpt	**ETX**	**BCC**																		

1) Datum with switch-off function; additional data value '-32000'
2) The user must ensure that the conditions for triggerpoints are met.

Paramset x	Control parameter set 1...4				(function no.: 6,7,8,9)	
Code	**Descr.**	**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. duty cycle 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. duty cycle 2	0.4..999.9	

Message structure for function 'Paramset x'

Block access to parameter data														max. eff. length: 70 bytes
STX	B2	,	50	,	6	=	90	,	8	,	Xp1	,	Tn1	,	Tv1	,	T1	,	Xp2	,	Tn2	,	Tv2	,
T2	,	0	**ETX**	**BCC**																				

Programmer	Programmer operation				(function no.: 10)	
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B3	C120	R/W	INT	PSel: source for program selection (T)	0...1	
				PwrUp: Behaviour at mains recovery (H)	0...4	
				Pend: Behav. at ProgEnd (Z)	0...3	
				PStrt: Function of run/stop (E)	0...1	

Message structure for function 'Programmer'

Block access to configuration data												max. eff. length: 21 bytes
STX	B3	,	50	,	10	=	90	,	0	,	1	,	C120	**ETX**	**BCC**

The values sent for a program (max. 20 segment points and 4 control outputs) are divided into 3 functions due to the limited transmission buffer.

Program x_1	Programmer parameter set part 1				(function no.: 11, 15,19)	
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	Tp1[1]	R/W	INT	Segment time analog	0 .. 9999	
	Tp2[1]	R/W	INT	T value segment 1	0 .. 9999	
	Tp3[1]	R/W	INT	T value segment 2	0 .. 9999	
	Tp4[1]	R/W	INT	T value segment 3	0 .. 9999	
	Tp5[1]	R/W	INT	T value segment 4	0 .. 9999	
	Tp6[1]	R/W	INT	T value segment 5	0 .. 9999	
	Tp7[1]	R/W	INT	T value segment 6	0 .. 9999	
	Tp8[1]	R/W	INT	T value segment 7	0 .. 9999	
	Tp9[1]	R/W	INT	T value segment 8	0 .. 9999	
	Tp10[1]	R/W	INT	T value segment 9	0 .. 9999	
	Wp0	R/W	BCD	Reset value W0	-999 .. 9999	
	LC-	R/W	BCD	Band width min. limit	-999 .. 9999	
	LC+	R/W	BCD	Band width max. limit	-999 .. 9999	
	Wp1	R/W	BCD	W value segment	-999 .. 9999	
	Wp2	R/W	BCD	W value segment 1	-999 .. 9999	
	Wp3	R/W	BCD	W value segment 2	-999 .. 9999	
	Wp4	R/W	BCD	W value segment 3	-999 .. 9999	
	Wp5	R/W	BCD	W value segment 4	-999 .. 9999	
	Wp6	R/W	BCD	W value segment 5	-999 .. 9999	
	Wp7	R/W	BCD	W value segment 6	-999 .. 9999	
	Wp8	R/W	BCD	W value segment 7	-999 .. 9999	
	Wp9	R/W	BCD	W value segment 8	-999 .. 9999	
	Wp10	R/W	BCD	W value segment 9	-999 .. 9999	
	Wmode	R/W	INT	Change mode	0 .. 2	
	Pmode	R/W	INT	Preset mode	0 .. 1	
	PNext[1]	R/W	INT	Following program	0 .. 3	

Message structure for function 'Program x_1'

Block access to parameter data																						max. eff. length: 174 bytes
STX	B2	,	50	,	<11, 14, 17, 20>	=	90	,	13	,	Tp1	,	Tp2	,	...	,	Tp10	,	Wp0	,		
LC-	,	LC+	,	13	,	Wp1	,	Wp2	,	...	,	Wp10	,	Wmode	,	Pmode	,	Pnext	**ETX**	**BCC**		

[1] Datum with switch-off function; additional data value '-32000'

Program x_2				Programmer parameter set part 2		(function no.: 12,16,20)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	Tp11[1]	R/W	INT	T value segment 10	00..9999	
	Tp12[1]	R/W	INT	T value segment 11	0..9999	
	Tp13[1]	R/W	INT	T value segment 12	0..9999	
	Tp14[1]	R/W	INT	T value segment 13	0..9999	
	Tp15[1]	R/W	INT	T value segment 14	0..9999	
	Tp16[1]	R/W	INT	T value segment 15	0..9999	
	Tp17[1]	R/W	INT	T value segment 16	0..9999	
	Tp18[1]	R/W	INT	T value segment 17	0..9999	
	Tp19[1]	R/W	INT	T value segment 18	0..9999	
	Tp20[1]	R/W	INT	T value segment 19	0..9999	
	Wp11	R/W	BCD	W value segment 10	0-999..9999	
	Wp12	R/W	BCD	W value segment 11	-999..9999	
	Wp13	R/W	BCD	W value segment 12	-999..9999	
	Wp14	R/W	BCD	W value segment 13	-999..9999	
	Wp15	R/W	BCD	W value segment 14	-999..9999	
	Wp16	R/W	BCD	W value segment 15	-999..9999	
	Wp17	R/W	BCD	W value segment 16	-999..9999	
	Wp18	R/W	BCD	W value segment 17	-999..9999	
	Wp19	R/W	BCD	W value segment 18	-999..9999	
	Wp20	R/W	BCD	W value segment 19	-999..9999	

Message structure for function 'Program x_2'

Block access to parameter data															max. eff. length: 138 bytes
STX	B2	,	50	,	<12, 15, 18, 21>	=	90	,	10	,	Tp11	,	Tp12	,	...	,	Tp20	,
Wp11	,	Wp12	,	...	,	Wp20	**ETX**	**BCC**										

Program x_3				Programmer parameter set part 3		(function no.: 13, 17, 21)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	Td1	R/W	INT	T value segment	00..9999	(1)
	Td2	R/W	INT	T value segment 1	0..9999	(1)
	Td3	R/W	INT	T value segment 2	0..9999	(1)
	Td4	R/W	INT	T value segment 3	0..9999	(1)
	Td5	R/W	INT	T value segment 4	0..9999	(1)
	Td6	R/W	INT	T value segment 5	0..9999	(1)
	Td7	R/W	INT	T value segment 6	0..9999	(1)
	Td8	R/W	INT	T value segment 7	0..9999	(1)
	Td9	R/W	INT	T value segment 8	0..9999	(1)
	Td10	R/W	INT	T value segment 9	0..9999	(1)
	D1	R/W	INT	control output 1..4	0000 .. 1111	
	D2	R/W	INT	Control output 1..4	0000 .. 1111	
	D3	R/W	INT	Control output 1..4	0000 .. 1111	
	D4	R/W	INT	Control output 1..4	0000 .. 1111	
	D5	R/W	INT	Control output 1..4	0000 .. 1111	
	D6	R/W	INT	Control output 1..4	0000 .. 1111	
	D7	R/W	INT	Control output 1..4	0000 .. 1111	
	D8	R/W	INT	control output 1..4	0000 .. 1111	
	D9	R/W	INT	Control output 1..4	0000 .. 1111	
	D10	R/W	INT	control output 1..4	0000 .. 1111	
	D0	R/W	INT	Reset value control outputs 1..4	0000 .. 1111	

Message structure for function 'Program x_3'

Block access to parameter data															max. eff. length: 142 bytes
STX	B2	,	50	,	<13, 17, 21>	=	90	,	0	,	21	,	Td1	,	Td2	,	...	,	Td10	,
D1	,	D2	,	...	,	D10	,	D0	**ETX**	**BCC**										

[1] Datum with switch-off function; additional data value '-32000'

Program x_4				Programmer parameter set part 4		(function no.: 14, 18, 22)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	Td11	R/W	INT	T value segment 10	0..9999	(1)
	Td12	R/W	INT	T value segment 11	0..9999	(1)
	Td13	R/W	INT	T value segment 12	0..9999	(1)
	Td14	R/W	INT	T value segment 13	0..9999	(1)
	Td15	R/W	INT	T value segment 14	0..9999	(1)
	Td16	R/W	INT	T value segment 15	0..9999	(1)
	Td17	R/W	INT	T value segment 16	0..9999	(1)
	Td18	R/W	INT	T value segment 17	0..9999	(1)
	Td19	R/W	INT	T value segment 18	0..9999	(1)
	Td20	R/W	INT	T value segment 19	0..9999	(1)
	D11	R/W	INT	Control output 1..4	0000 .. 1111	
	D12	R/W	INT	Control output 1..4	0000 .. 1111	
	D13	R/W	INT	Control output 1..4	0000 .. 1111	
	D14	R/W	INT	Control output 1..4	0000 .. 1111	
	D15	R/W	INT	Control output 1..4	0000 .. 1111	
	D16	R/W	INT	Control output 1..4	0000 .. 1111	
	D17	R/W	INT	Control output 1..4	0000 .. 1111	
	D18	R/W	INT	Control output 1..4	0000 .. 1111	
	D19	R/W	INT	Control output 1..4	0000 .. 1111	
	D20	R/W	INT	Control output 1..4	0000 .. 1111	

Message structure for function 'Program x_3'

Block access to parameter data																max. eff. length: 137 bytes
STX	B2	,	50	,	<14, 18, 20>	=	90	,	0	,	20	,	Td11	,	Td12	,	...	,	Td20	,
D11	,	D12	,	...	,	D20	**ETX**	**BCC**												

[1] Datum with switch-off function; additional data value '-32000'

5.3.4 ALARM (FB no.: 51 type no.: 45)

Function block 'ALARM' defines the overall alarm processing of the relevant controller.

Process data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
1	Status 1	R	ST1	Alarm status x		A
18	Type	R	INT	Function block type no.	45	

Bem. A ***Status 1***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	AL1	Alarm 1	off	on
D1	Al2	Alarm 2	off	on
D2	Al3	Alarm 3	off	on
D3	Al4	Alarm 4	off	on
D4, D5	'0'	always '0'		
D6	'1'	always '1'		
D7		Parity		

Parameter and configuration data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Lim_{L1}	R/W	BCD	min. limit value 1	-999..9999	(1)
	Lim_{H1}	R/W	BCD	High limit value 1	-999..9999	
	x_{sd1}	R/W	BCD	Switching difference 1	0..9999	
	Lim_{L2}	R/W	BCD	Low limit value 2	-999..9999	(1)
	Lim_{H2}	R/W	BCD	High limit value 2	-999..9999	
	x_{sd2}	R/W	BCD	Switching difference 2	0..9999	
	Lim_{L3}	R/W	BCD	Low limit value 3	-999..9999	(1)
	Lim_{H3}	R/W	BCD	High limit value 3	-999..9999	
	x_{sd3}	R/W	BCD	Switching difference 3	0..9999	
	Lim_{L4}	R/W	BCD	Low limit value 4	-999..9999	(1)
	Lim_{H4}	R/W	BCD	High limit value 4	-999..9999	
	x_{sd4}	R/W	BCD	Switching difference 4	0..9999	
B3	C600 (ALARM1)	R/W	INT	Src: signal source (T,H) Fnc: Function (Z) Mod: Mode (E)	0..xxyz	
	C620 (ALARM2)	R/W	INT	Src: signal source (T,H) Fnc: Function (Z) Mod: Mode (E)	0..xxyz	
	C640 (ALARM3)	R/W	INT	Src: signal source (T,H) Fnc: Function (Z) Mod: Mode (E)	0..xxyz	
	C660 (ALARM4)	R/W	INT	Src: signal source (T,H) Fnc: Function (Z) Mod: Mode (E)	0..xxyz	

Message structure for function 'General'

Block access to parameter data																max. eff. length: 99 bytes
STX	B2	,	51	,	0	=	45	,	12	,	Lim_{L1}	,	Lim_{H1}	,	x_{sd1}	,
Lim_{L2}	,	Lim_{H2}	,	x_{sd2}	,	Lim_{L3}	,	Lim_{H3}	,	x_{sd3}	,	Lim_{L4}	,	Lim_{H4}	,	x_{sd4}	,	0	**ETX**	**BCC**

Block access to configuration data															max. eff. length: 34bytes
STX	B3	,	51	,	0	=	45	,	0	,	4	,	C600	,	C620	,	C640	,	C660	**ETX**	**BCC**

[1] Datum with switch-off function; additional data value '-32000'

5.3.5 OUTPUT (FB no.: 81 type no.: 111)

All data which concern the signal processing of all output values (analog/digital) are grouped in function block 'OUTPUT'.

Process data

General (function no.: 0)

Code	Descr.	Access	Type	Description	Range	Rem.
18	Type	R		Function block type no.	111	

Parameter and configuration data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
B3	C500 (OUT1)	R/W	INT	Src: output signal source (T,H) Type: output stage type (Z) Mode: actuator action (E)	0..xxyz	
	C530 (OUT2)	R/W	INT	Src: output signal source (T,H) Type: Output stage type (Z) Mode: actuator output action (E)	0..xxyz	
	C560 (OUT3)	R/W	INT	Src: output signal source (T,H) Type: output stage type (Z) Mode: actuator output action (E)	0..xxyz	
	C590 (OUT4)	R/W	INT	Src: Output signal source (T,H) Type: output stage type (Z) Mode: actuator output action (E)	0.. xxyz	
	C591 (OUT5)	R/W	INT	Src: output signal source (T,H) Type: output stage type (Z) Mode: actuator output action (E)	0.. xxyz	
	C596 (DO5)	R/W	INT	Src: output signal source (T,H) Mode: inversion of output (E)	0..xx0y	
	C597 (DO6)	R/W	INT	Src: output signal source (T,H) Mode: inversion of output (E)	0..xx0y	

Message structure for function 'General'

Block access to configuration data														max. eff. length: 50 bytes
STX	B3	,	81	,	0	=	111	,	0	,	7	,	C500	,	C530	,	C560	,	C590	,	C591
C596	,	C597	ETX	BCC																	

SV Signal processing for OUT 3 (function no.: 1)

Code	Descr.	R/W	Type	Description	Range	Rem.
B3	X0_Out3	R/W	BCD	Reference value f. output of 0%	999..9999	(1)
	X100_Out3	R/W	BCD	Reference value for output of 100%	999..9999	
	xs1	R/W	BCD	Segment point 1 : X value	-999..9999	(1, 2)
	ys1	R/W	BCD	Segment point 1: Y value	-999..9999	
	xs2	R/W	BCD	Segment point 2: X value	-999..9999	
	ys2	R/W	BCD	Segment point 2: Y value	-999..9999	
	xs3(3)	R/W	BCD	Segment point 3: X value	-999.. 9999	
	ys3	R/W	BCD	Segment point 3: Y value	-999.. 9999	
	xs4(3)	R/W	BCD	Segment point 4: X value	-999.. 9999	
	ys4	R/W	BCD	Segment point 4: Y value	-999.. 9999	
	xs5(3)	R/W	BCD	Segment point 5: X value	-999.. 9999	

1) Only defined if HW option C is fitted.
2) Only defined for KS94 with the relevant W option.
3) Datum with switch-off function; additional data value '-32000'

SV				Signal processing for OUT 3 (function no.: 1)			
Code	**Descr.**	**R/W**	**Type**	**Description**		**Range**	**Rem.**
B3	ys5	R/W	BCD	Segment point 5: Y value		-999.. 9999	(1, 2)
	xs6(3)	R/W	BCD	Segment point 6: X value		-999.. 9999	
	ys6	R/W	BCD	Segment point 6: Y value		-999.. 9999	
	xs7(3)	R/W	BCD	Segment point 7: X value		-999.. 9999	
	ys7	R/W	BCD	Segment point 7: Y value		-999.. 9999	
	xs8(3)	R/W	BCD	Segment point 8: X value		-999.. 9999	
	ys8	R/W	BCD	Segment point 8: Y value		-999.. 9999	
	C565	R/W	INT	Func: function selection DP: decimal point	(T) (E)	0..x0yz	(1)

Message structure for function 'Sv'

Block access to configuration data														max. eff. length: 147 bytes
STX	B3	,	81	,	1	=	111	,	18	,	X0	,	X100	,	xs1	,	ys1	,	xs2	,	ys2	,
xs3	,	ys3	,	...	,	xs8	,	ys8	,	1	,	C565	**ETX**	**BCC**								

1) Only defined if HW option C is fitted.
2) Only defined for KS94 with the relevant W option.
3) Datum with switch-off function; additional data value '-32000'

6 Annex

6.1 Terms

FB	Abbreviation for function block
Fct	Abbr. for function
ET	Abbr. for Engineering Tool
Function	a partial function of a function block
Function block	self-contained processing unit
HW	Abbr. for hardware
ISO1745	Standard communication protocol ISO 1745, ASCII-based ^
PC interface	front-pnel interface on KSX controller for connecting an Engineering Tool
PCI	process control instrument
PCI protocol	ISO 1745-based protocol, implemented for philips controllers
RS422	Standard 4-wire interface, full duplex, (EIA RS 422); in this case: separate send/receive channels with max. 32 units
RS485	Standard 2-wire interface, half duplex, (EIA RS 485)
SW	Abbr. for software
TTL	Signal level at module level

7 Index