PMA Prozeß- und Maschinen-Automation GmbH

Industrial controller KS 94

KS94
PROFIBUS-DP

PROFI PROCESS FIELD BUS BUS

Interface description

PROFIBUS - DP

9499 040 49311

vailid from: 8363

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

Contents

1. General

The industrial controller KS94 (9407-92x-3xxx1) versions are equipped with a PROFIBUS-DP interface for transmission of process, parameter and configuration data. Connection is on the controller rear. The serial communication interface permits connection to supervisory systems, visualization tools, etc.

Another interface always provided as standard is the PC interface on the front panel. It can be used for connecting an engineering tool which runs on a PC.

Communication is according to the master/slave principle. KS94/DP is always slave

Lead and physical and electrical properties of the interface are as follows:

- Network topology
  Linear bus with active bus terminating resistor (→ p. 8) at both ends. Stub lines are possible (the maximum possible overalltap line length- with 1,5Mbit/s is 6,6m dependent of cable type).
- Cable for transmission
  screened, twisted 2-wire cable (→ EN 50170 vol.2).
- Baudrates and cable lengths (without repeater)
  The maximum cable length is dependent of the Baudrate.
  The Baudrate is determined by the master configuration.

Automatic Baudrate detection

Baudrate		Maximum cable length
9,6 / 19,2 / 93,75	kbit/s	1200 m
187,5	kbit/s	1000 m
500	kbit/s	400 m
1,5	Mbit/s	200 m
3/6/12	Mbit/s[1)]	100m

- Interface
  RS485 with AMP flat-pin connector; on-site mounting possible
- Addressing: 0 ... 126 (factory setting: 126)
  Remote addressing is possible
  Address setting via front-panel operation:
  Press ⏀ ≥ during 3 sec. → Para blinks → press ⏀ shortly → Setpt is displayed
  Press ⏀ ≥ during 3 seconds → Conf blinks → press ⏀ shortly → Contr is displayed
  Press▲ until AUX is displayed → press ⏀ shortly → ADR is displayed. After pressing key ⏀ shortly the adjusted addres blinks and can be changed by pressing key ▲▼. Press key ⏀ shortly to store the new address.
- 32 units in a segment. Can be extended up to 127 by means of repeater.

KS94 with PROFIBUS-DP interface offers many advantages referred to handling and integration into a PROFIBUS network.

- Diagnosis and monitoring
  Display of bus errors
  - Plain text display
  - LED

  Bus error signalling via e.g. relay
  Freeze and Synch commands possible

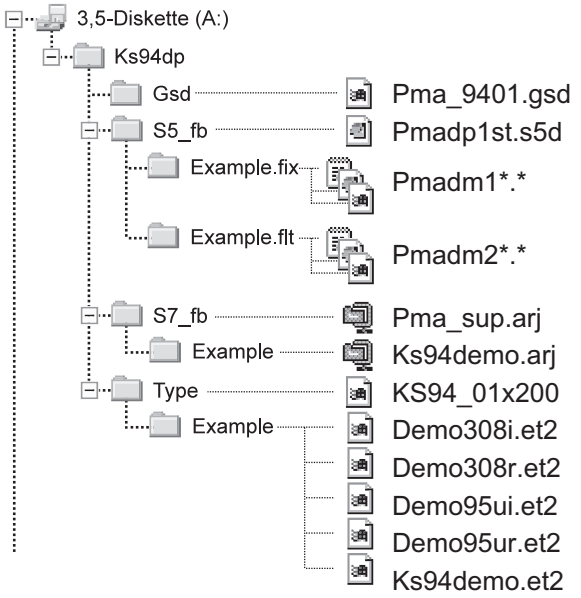
- Particularities
  Configurable process data modules
  Direct reading and writing of inputs and outputs
  Input forcing
  Output disabling
  Back-up controller function
  Simple connection even to small PLCs

[1)]The screw-on adaptor (9407 998 00021) is not suitable for this transmission speed.

1.1. Scope of delivery

The Engineering Set comprises:

- Floppy



- Interface description for PROFIBUS-DP

2. Hints for operation

2.1. 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

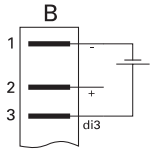
- via operator dialogue (front):
  press ⏎ ≥ during 3 sec. → Para flashes
  press ▲ until CBus flashes → ⏎ confirm briefly.
  ---
  press ⏎ ≥ during 3 sec. → Para flashes
  press ▲ until CFrnt flashes → ⏎ confirm briefly.

display CBus	≙ switch over to rear interface
display CFrnt	≙ switch over to front-panel interface

- or by activating 'REMOTE' (⊗ page 7). 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.

2.2. Remote/local



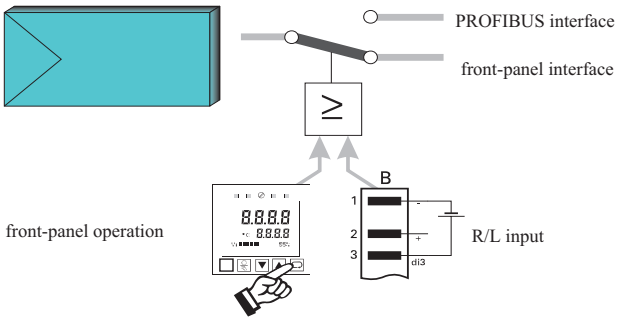
Units with serial interface are fitted with a hardware input (di3) for switch-over between REMOTE and LOCAL operation (R/L).

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.



2.2.1. *Local switch-over via interface*

Local switch-over is possible via the field bus interface. Conditions:

- Unit is in REMOTE mode (contact di3 closed)
- Switch-over to local is possible by interface command. This is also possible inversely (return to Remote mode).
- Default setting is Remote.
- With HW mode Local, this setting is ineffective.
- When changing over to HW mode Local or during Power ON, the default value is set.

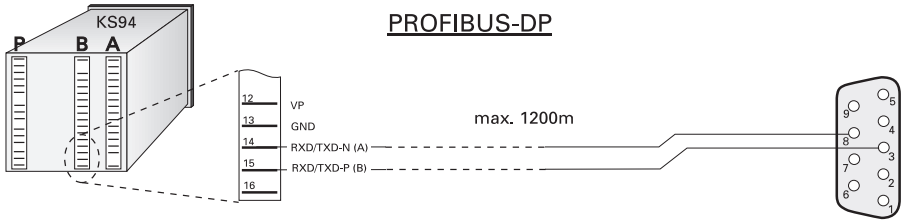
⚠ Caution! When switching over to the front-panel interface, returning to Remote is not possible.

When returning from Local to Remote, all cyclic write data must be written again with PROFIBUS, to ensure the last defined bus master condition.

2.2.2. *Connecting the interface*

PROFIBUS connection is to connector B.
Rear serial interface, RS485-based physical signals.

Fig.1: PROFIBUS-DP connection

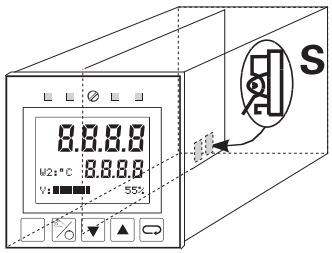


Cable construction must be done by the user. Thereby, the general cable specifications to EN 50170 vol.2 must be followed.

2.2.3. *Laying cables*

During cable laying, the general hints for cable laying made by the supplier of the master module must be followed:

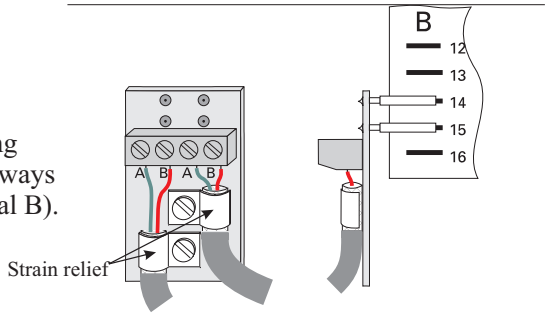
- Cable run within buildings (inside and outside cabinets)
- Cable run outside buildings
- Potential equalization
- Cable screening
- Measures against interference voltages
- Length of stub line
- The bus terminating resistor can be activated in KS94 by 2 wire hook switches (S). Both wire hook switches must always be open or closed (terminating resistor active).



For special hints for installation of PROFIBUS cables, see PNO Technical guideline "***Installation guidelines for PROFIBUS-DP/FMS***" (Order no. 2.111 [dt]; 2.112 [engl.]).

2.2.4. *Bus connector*

Fig.2: Screw-on adaptor



❑ Screw-on adaptor
For connecting the bus cable to the bus connector (order no. 9407 998 00021), strip the cable end as shown in Fig.: . Subsequently, mount the bus cable by connecting the same conductors at the same terminal A or B (e.g. Always use a green wire for terminal A and a red wire for terminal B). Note that the cable screen must be blank below the strain relief.

Fig.3: Stripped cable ends

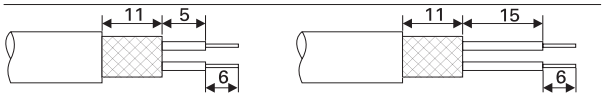
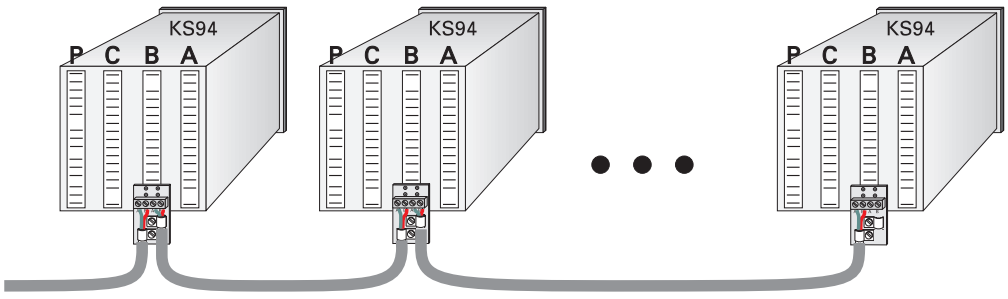
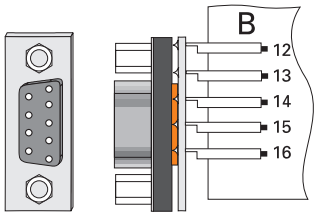


Fig.4: Wiring example with bus connector



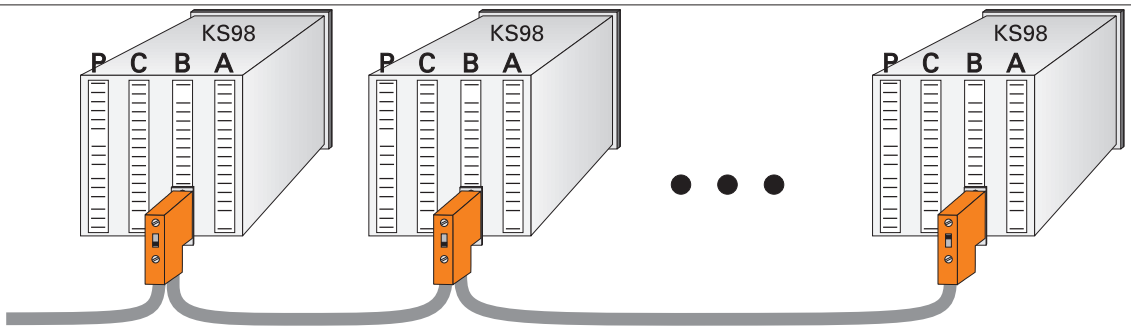
❑ Sub-D bus adaptor:
For using standard PROFIBUS connectors (9-pole sub-D) in an installation, the sub-D bus adaptor must be used (order no. 9407 998 00031).

Fig.5: Sub-D bus adaptor



⚠ When using bus connectors **with** terminating resistors, the KS 94 terminating resistors must be switched off (open S.I.L. switches).

Fig.6: Wiring example with sub-D bus adaptor and bus connector



2.3. PROFIBUS status display

The two PROFIBUS status display modes, which can be adjusted during instrument configuration (C.bUS) are:

- **Plain text message on the lower display line**
  C.bUS ; UsrTx = 2 (user-specific text via function statuses)
- **Display via LED**
  C.bUS ; LED = 4 (PROFIBUS-DP error messages)

Select the configuration word C.bUS :
Press ⏎ ≥ during 3 sec. → Para blinks → press ⏎ shortly → Setpt is displayed
Press ⏎ ≥ during 3 sec. → Conf blinks → press ⏎ shortly → Contr is displayed
Press ▲ as long as DISP is displayed → press ⏎ shortly → the actual setting of C.bUS is displayed.
Now, the relevant configuration can be selected by pressing key ⏎ shortly.

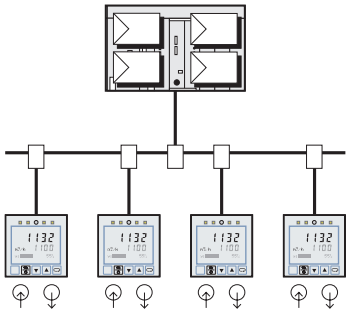
2.3.1. Display signification

Plain text display	LED	LED= on / plain text display active	LED= off / no plain text display
DP:no bus access	1	No access by bus master	Access by bus master realized
DP:param.error	2	inadmissible parameter setting telegram was sent	——
DP:config.error	3	Faulty configuration	Configuration successful
DP:no data exchg	4	no Data communication *	Data communication o.k.
	1...4	PROFIBUS controller trouble	Faulty bus operation

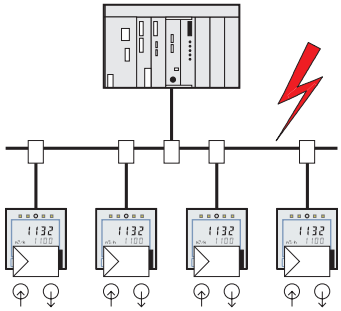
* Possible causes:
- Bus cable error or master failure
- front panel interface selected
- device address does not correspond with bus configuration

3. Special functions

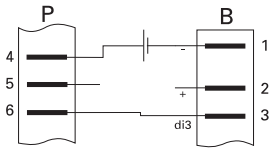
3.1. *'Back-up'* controller operation



Normally, calculation of the controller outputs is in the PLC. The controllers are used for measuring the process values and output of the correcting values (incl. duty cycle conversion and display).



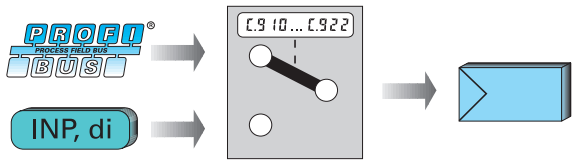
In case of trouble, i.e. with failure of PLC or bus communication, control is taken over by the KS 94 controllers independently and bumplessly.
Determination that the controller switches to automatic mode automatically when switching over from remmote to local operation is by configuration. Selection 'back-up operation' is in configuration word C.191 (S A/M = 4).
If the application requires that the controller goes to automatic operation, i.e. that control is taken over by the controller in automatic mode, the following measures must be taken:



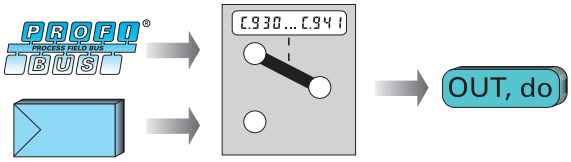
- definition of an alarm dependent of DP status (e.g. C.600 ...; Src = 23; Fnc = 6)
- output of this alarm on a relay or logic output
- external wiring of this output to the remote/local input (di3)

3.2. Forcing

Analog inputs can be determined for the controller via PROFIBUS by configuration (C.910, C.911), also if they are not provided physically. Digital inputs can also be set (C.920 ... C.922).

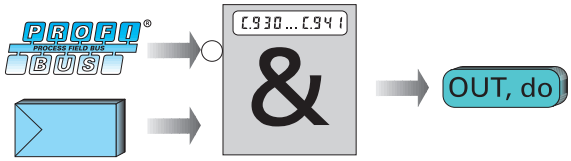


Analog outputs can be written directly after they have been configured accordingly (C.930, C.931). Digital outputs can also be set directly (C.940, C.941).



3.3. Controller output locking

Via PROFIBUS the digital outputs of KS94 can be locked. (C.930, C.941; FOUT1, 2, 4, 5/Fdo1...6 = 2)
FOUT1 can be locked only when configured as a relay or logic output.



4. Process data

Data transmission comprises cyclically transmitted process data and acyclically transmitted parameter and configuration data. The I/O data area is modular for matching to the requirements of the control task. Selection of process data module_process data module and data format (FIX point format for small PLCs; REAL format for full resolution) is via master interface configuration tools (e.g. via COM PROFIBUS with Siemens S5).

The following process data modules can be configured:

Process data module A:	read (8/14 bytes)* Process value, output value, set-point, status	write (6/10 bytes)* Set-point, output value, auto/manual	without param. channel
Process data module B:	read (16/22 bytes)* Process value, output value, set-point, status	write (14/18 bytes)* Set-point, output value, auto/manual	with parameter channel
Process data module C:	read (30/46 bytes)* Process value, output value, set-point, status, inputs	write (20/28 bytes)* Set-point, output value, auto/manual, outputs	with parameter channel
Process data module D:	read (30/46 bytes)* Process value, output value, set-point, status, inputs	write (26/40 bytes)* Set-point, output value, auto/manual, inputs(”Forcing”)	with parameter channel
Process data module E:	Read (42/76 bytes)* Any 16 analog values and 2 status words	Write (42/76 bytes)* Any 16 analog values and 2 control words	with parameter channel
Process data module F:	Read (24/36 bytes)* Any 6 analog values and 2 status words	Write (24/36 bytes)* Any 6 analog values and 2 control words	with parameter channel
Process data module G:	Read (8/10 bytes)* Multiplexed values and 2 status words	Write (8/10 bytes)* Multiplexed values and 2 status words	without param. channel

* number of required bytes in the I/O area (FIX point /REAL format)

The parameter channel is used for sequential transmission of parameter and configuration data.

Dependend of configuration, the analog value transmission is in REAL format (range: -29999 .. 200000) or in 16-bit fixpoint format (FIX). In FIX point format, all values are interpreted with one digit behind the decimal point (range -3000,0 to 3200,0).

The values to be adjusted and data significations are adjusted in the following tables:

❑ **Module A** (I/O)

No.	Descr.	R/W	REAL format			FIX point format			Rem.
			Type	Value		Type	Value		
				Hex	COMPROFIBUS		Hex	COMPROFIBUS	
0	Process value (x_{eff})	R	Float	D1	209	Int16	50	1AE	
1	Output value (y_{eff})	R	Float	D1	209	Int16	50	1AE	
2	Set-point (w_{eff})	R	Float	D1	209	Int16	50	1AE	
3	Status	R	status	10	8DE	status	10	8DE	**A**
4	Status-1	R	status	10	8DE	status	10	8DE	**B**
5	Set-point (wvol)	W	Float	E1	225	Int16	60	1AA	
6	Output value (Y_{man})	W	Float	E1	225	Int16	60	1AA	
7	Auto/manual	W	Int16	60	1AA	Int16	60	1AA	

❑ **Module B** (I/O + parameters)

No.	Descr.	R/W	REAL format			FIX point format			Rem.
			Type	Value		Type	Value		
				Hex	COM PROFIBUS		Hex	COM PROFIBUS	
0	Process value (x_{eff})	R	Float	D1	209	Int16	50	1AE	
1	Output value (y_{eff})	R	Float	D1	209	Int16	50	1AE	
2	Set-point (w_{eff})	R	Float	D1	209	Int16	50	1AE	
3	Status	R	status	10	8DE	status	10	8DE	A
4	Status-1	R	status	10	8DE	status	10	8DE	B
5	Set-point (wvol)	W	Float	E1	225	Int16	60	1AA	
6	Output value (Y_{man})	W	Float	E1	225	Int16	60	1AA	
7	Auto/manual	W	Int16	60	1AA	Int16	60	1AA	
8	Parameter channel	R/W	8Byte	F3	4AX	Byte8	F3	4AX	

❑ **Module C** (I/O + parameters + forcing outputs)

No.	Descr.	R/W	REAL format			FIX point format			Rem.
			Type	Value		Type	Value		
				Hex	COM PROFIBUS		Hex	COM PROFIBUS	
0	Process value (x_{eff})	R	Float	D1	209	Int16	50	1AE	
1	Output value (y_{eff})	R	Float	D1	209	Int16	50	1AE	
2	Set-point (w_{eff})	R	Float	D1	209	Int16	50	1AE	
3	Status	R	status	10	8DE	status	10	8DE	A
4	Status-1	R	status	10	8DE	status	10	8DE	B
5	INP1	R	Float	D1	209	Int16	50	1AE	
6	INP3	R	Float	D1	209	Int16	50	1AE	
7	INP4	R	Float	D1	209	Int16	50	1AE	
8	INP5	R	Float	D1	209	Int16	50	1AE	
9	INP6	R	Float	D1	209	Int16	50	1AE	
10	State_di1	R	status	10	8DE	status	10	8DE	C
11	State_di2	R	status	10	8DE	status	10	8DE	D
12	State_inpf	R	status	10	8DE	status	10	8DE	E
13	State_switch	R	status	10	8DE	status	10	8DE	F
14	Set-point (wvol)	R	Float	E1	225	Int16	60	1AA	
15	Output value (Y_{man})	R	Float	E1	225	Int16	60	1AA	
16	auto/manual	R	Int16	60	1AA	Int16	60	1AA	
17	FOut1	R	Float	E1	225	Int16	60	1AA	
18	FOut3	R	Float	E1	225	Int16	60	1AA	
19	Fdo	W	ICMP	60	1AA	ICMP	60	1AA	G
20	Parameterchannel	R/W	8Byte	F3	4AX	Byte8	F3	4AX	

❑ **Module D** (I/O + parameters + forcing inputs)

No.	Descr.	R/W	REAL format			FIX point format			Rem.
			Type	Value		Type	Value		
				Hex	COM PROFIBUS		Hex	COM PROFIBUS	
0	Process value (x_{eff})	R	Float	D1	209	Int16	50	1AE	
1	Output value (y_{eff})	R	Float	D1	209	Int16	50	1AE	
2	Set-point (w_{eff})	R	Float	D1	209	Int16	50	1AE	
3	Status	R	status	10	8DE	Status	10	8DE	A
4	Status-1	R	status	10	8DE	Status	10	8DE	B
5	INP1	R	Float	D1	209	Int16	50	1AE	
6	INP3	R	Float	D1	209	Int16	50	1AE	
7	INP4	R	Float	D1	209	Int16	50	1AE	
8	INP5	R	Float	D1	209	Int16	50	1AE	
9	INP6	R	Float	D1	209	Int16	50	1AE	
10	State_di1	R	status	10	8DE	status	10	8DE	C
11	State_di2	R	status	10	8DE	status	10	8DE	D
12	State_inpf	R	status	10	8DE	status	10	8DE	E
13	State_switch	R	status	10	8DE	status	10	8DE	F
14	Set-point (wvol)	W	Float	E1	225	Int16	60	1AA	
15	Output value (Y_{man})	W	Float	E1	225	Int16	60	1AA	
16	Auto/manual	W	Int16	60	1AA	Int16	60	1AA	
17	FInp1	W	Float	E1	225	Int16	60	1AA	
18	FInp3	W	Float	E1	225	Int16	60	1AA	
19	FInp4	W	Float	E1	225	Int16	60	1AA	
20	FInpt5	W	Float	E1	225	Int16	60	1AA	
21	FInpt6	W	Float	E1	225	Int16	60	1AA	
22	Fdi	W	ICMP	60	1AA	ICMP	60	1AA	H
23	Parameter channel	R/W	8Byte	F3	4AX	Byte8	F3	4AX	

For the following process data modules (module E - G), the cyclic transmission data must be selected by means of the engineering tool (‘ET/KS94) via **Change → Parameter Miscellaneous → Fieldbusdata** (→ Fig.:7).

Fig.: 7 Assignment of processdata for fieldbus

Max. 16 data for reading and 16 data for writing can be selected. Dependend of the process data module used, sixteen data (module E), the first six data (module F) or all selected data (module G) are used.

❑ **Module E** (16 selectable I/O values + parameters)

No.	Descr.	R/W	REAL format			FIX point format			Rem.
			Type	Value		Type	Value		
				Hex	COM PROFIBUS		Hex	COM PROFIBUS	
0	Status word_1	R	Status	11	16DE	Status	11	16DE	I
1	Status word_2	R	Status	11	16DE	Status	11	16DE	J
2	IN1	R	Float	D1	209	Int16	51	2AE	
3	IN2	R	Float	D1	209				
...	...								
16	IN15	R	Float	D1	209	Int16	51	2AE	
17	IN16	R	Float	D1	209				
18	Control word_1	W	Status	21	16DA	Status	21	16DA	K
19	Control word_2	W	Status	21	16DA	Status	21	16DA	L
20	OUT1	W	Float	E1	225	Int16	61	2AA	
21	OUT2	W	Float	E1	225				
...	...								
34	OUT15	W	Float	E1	225	Int16	61	2AA	
35	OUT16	W	Float	E1	225				
36	Parameter channel	R/W	8Byte	F3	4AX	Byte8	F3	4AX	

❑ **Module F** (6 selectable I/O values + parameters)

No.	Descr.	R/W	REAL format			FIX point format			Rem.
			Type	Value		Type	Value		
				Hex	COM PROFIBUS		Hex	COM PROFIBUS	
0	Status word_1	R	Status	11	16DE	Status	11	16DE	I
1	Status word_2	R	Status	11	16DE	Status	11	16DE	J
2	IN1	R	Float	D1	209	Int16	51	2AE	
3	IN2	R	Float	D1	209				
4	IN3	R	Float	D1	209	Int16	51	2AE	
5	IN4	R	Float	D1	209				
6	IN15	R	Float	D1	209	Int16	51	2AE	
7	IN16	R	Float	D1	209				
8	Control word_1	W	Status	21	16DA	Status	21	16DA	K
9	Control word_2	W	Status	21	16DA	Status	21	16DA	L
10	OUT1	W	Float	E1	225	Int16	61	2AA	
11	OUT2	W	Float	E1	225				
12	OUT3	W	Float	E1	225	Int16	61	2AA	
13	OUT4	W	Float	E1	225				
14	OUT5	W	Float	E1	225	Int16	61	2AA	
15	OUT6	W	Float	E1	225				
16	Parameter channel	R/W	8Byte	F3	4AX	Byte8	F3	4AX	

❑ **Module G** (multiplexed transmission of digital and analog values)

No.	Descr.	R/W	REAL format			FIX point format			Rem.
			Type	Value		Type	Value		
				Hex	COM PROFIBUS		Hex	COM PROFIBUS	
0	Status word_1	R	Status	11	16DE	Status	11	16DE	I
1	Status word_2	R	Status	11	16DE	Status	11	16DE	J
2	INDEX IN (Read / Write)	R	Int16	50	1AE	Int16	50	1AE	
3	Read Value	R	Float	D1	209	Int16	50	1AE	
4	Control word_1	W	Status	21	16DA	Status	21	16DA	K
5	Control word_2	W	Status	21	16DA	Status	21	16DA	
6	INDEX OUT (Read / Write)	W	Int16	60	1AA	Int16	60	1AA	
7	Write Value	W	Float	E1	225	Int16	60	1AA	

Principle of operation (read):

- Entry of the index number into 'Index OUT' (Read)
- After the index number is mirrored in 'Index IN' (Read), the value read is in 'Read Value' .

Principle of operation (write):

- Entry of the index number into 'Index OUT' (Write)
- Entry of the value to be written into 'Write Value'
- After the index number was mirrored in 'Index IN' (Write), the value was transmitted.

ⓘ To ensure consistent data transmission, updating of 'Index OUT' (Write) and 'Write Value' before a PROFIBUS data cycle must be ensured. Unless this is possible, proceed as follows: '0' in 'Index OUT' (Write), then write the value to be transmitted into 'Write Value' and the index number into 'Index OUT' (Write). Entry of a '0' in 'Index OUT' (Read) / 'Index OUT' (Write) does not cause data transmission.

Rem. A 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	off	on
D1	y2	Switching output 2	off	on
D2	Lim1	Limit 1	off	on
D3	Lim2	Limit 2	off	on
D4	Lim3	Limit 3	off	on
D5	Lim4	Limit 4	off	on
D6	R/L	Remote/Local	Local	Remote
D7	A/M	Auto/Manual	Auto	Manual

Rem. B Status-1 (previous)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	y1	Switching output 1	off	on
D1	y2	Switching output 2	off	on
D2	Lim1	Limit 1	off	on
D3	Lim2	Limit 2	off	on
D4	Lim3	Limit 3	off	on
D5	Lim4	Limit 4	off	on
D6		always '1'		
D7	WERR	Faulty value in the output area	no	yes

Rem. C 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		always '1'		
D7		0		

Rem. D 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		always '1'		
D7		0		

Rem. E State_inpf (error statuses 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	off	on
D1	'0'	always '0'	-	-
D2	if3	Error status Inp 3	off	on
D3	if4	Error status Inp 4	off	on
D4	if5	Error status Inp 5	off	on
D5	if6	Error status Inp 6	off	on
D6		always '1'		
D7		0		

Rem. F State_switch (switch-over operations)

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

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

Rem. G Fdo (digital output forcing)

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

Rem. H Fdi (digital input forcing)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wert	0	0	0	0	di12	di11	di10	di9	di8	di7	di6	di5	di4	di3	di2	di1

Rem. I Status word 1

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	Auto/Man	Automatic/Manual	Automatic	Manual
D1	Coff	Controller switched off	No	Yes
D2	y1	Switching output 1	Off	On
D3	y2	Switching output 2	Off	On
D4	Lim1	Alarm 1	Off	On
D5	Lim2	Alarm 2	Off	On
D6	Lim3	Alarm 3	Off	On
D7	Lim4	Alarm 4	Off	On
D8	Fail1	Input 1 Fail	No	Yes
D9	"0"	Always "0"	---	---
D10	Merr	Module error[1)]	No	Yes
D11	NAK	Error writing process data module	No	Yes
D12	UPD	Parameter changed	No	Yes
D13	On/Off	Online / configuratoion	Online	Configuration
D14	R/L	Remote / local	Local	Remote
D15	Dex	Changed data[2)]	No	Yes

Rem. J Status word 2

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0-D11	di1-di12	Digital inputs 1-12	Off	On
D12	Fail 3	Input 3 Fail	No	Yes
D13	Fail 4	Input 4 Fail	No	Yes
D14	Fail 5	Input 5 Fail	No	Yes
D15	Fail 6	Input 6 Fail	No	Yes

[1)] Defect in the basic instrument, or internal interface switched off
[2)] See section "Locking mechanism for protection against changes"

Rem. K ***Control word 1***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	Auto/Man	Automatic/Manual	Automatic	Manual
D1	Coff	Controller switched off	No	Yes
D2	w/W2	w/W2 switch-over	w	W2
D3	We/Wi	Wext/Wint switch-over	Wext	Wint
D4	w/dW	Set-point offset	Off	On
D5	y/Y2	Additional correcting value	Off	On
D6	0Start	Self-tuning start	Off	On
D7	"0"	Always "0"	---	---
D8-D11	do1-do4	Digital outputs 1-4	Off	On
D12	OUT4	Output OUT4	Off	On
D13	OUT5	Output OUT5	Off	On
D14	LOC	Unit switched over to	Remote	Local[3)]
D15	Dval	Data valid, acknowledgment[4)] positive flank "0"⇒"1"		

Rem. L ***Control word 2***

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0-D11	di1-di12	Digital inputs 1-12	Off	On
D12	OUT1	Output OUT1	Off	On
D13	OUT2	Output OUT2	Off	On
D14	do 5	Digital output 5	Off	On
D15	do 6	Digital output 6	Off	On

4.0.1 Blocking mechanism with changes

Changing the reference to a datum to be transmitted e.g. on-line via parameter channel or via the engineering interface during operation implies the risk that values can be misinterpreted by bus master and KS94. For solving this problem, a blocking mechanism is provided.

- When changing a reference, the controller module sets bit Dex = 1.
- Write data of message 96 won't be handled any more.
- The master must evaluate bit Dex.
- Acknowledgement and statement that valid write data are available on the master side are generated via a positive flank for bit Dval.
- When receiving a positive flank, the controller module sets Dex = 0 and stores the transmitted data.
- Resetting Dex is also possible by voltage switch-off and on.

[3)] Reversed signification as in the status information; default setting is remote (the user need not do anything)

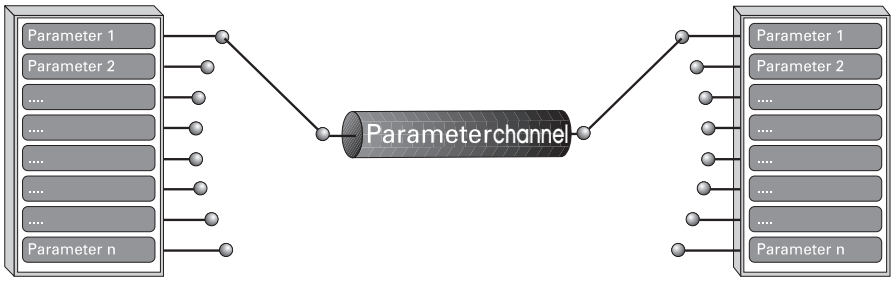
[4)] See section "Locking mechanism for protection against changes"

4.1. Process data transmission

Process data are read cyclically by the controller. Thereby, observation of the minimum poll time of 200ms is ensured, unless a simultaneous access is made via the parameter channel.
Output data sent to KS94 are compared to the values sent previously and sent to the controller in case of divergence. If the controller replies NAK, bit 7 in status-1 is set, until no faulty accesses are pending any more..
If there should be no process data exchange with the controller, e.g. by switch-over to the PC interface, bit 2 (static diagnosis) in station status 2 of the PROFIBUS slave diagnosis is set.

4.2. Parameter transmission

For transmission of parameters, the 'parameter channel' via which data can be exchanged transparently via the function block protocol independent of controller is available. Thereby, all protocol access types are supported (individual access, tens block and overall block). Communication to the controller is transparent, i.e. the user himself is responsible for monitoring adjustment ranges, operating modes (remote/local) etc.
The parameter channel is designed for large data quantities with low requirements on the Baudrate.



4.2.1. Message elements

Some terms which are explained below are used in the following paragraphs:

Element	Description	Rem.
ID	Telegram type identification	**A**
ID1	Data format of the transmitted or received data	**B**
Code	Data addressing key	**C**
FB no.	Function block number	**D**
Fct no.	Function number	**E**
Type	Function type	**F**

Rem. A ID

This element identifies the telegram type: ID = 0x10 ≙ Start telegram
ID = 0x68 ≙ Data telegram
ID = 0x16 ≙ End telegram

Rem. B ID1

This element identifies the data format: ID1 = 0 ≙ Integer
(Permissibility dependent of access type) ID1 = 1 ≙ Real
ID1 = 2 ≙ Char

Rem. C Code

The code identification is decimal within '00'...'99' and '178' ≙ B2 and '179' ≙ B3.

Rem. D FB no. (function block number)

A function block is addressed with a function block number within '0' and '250'.
Function block number range:

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

Rem. E ***Fct.-no. (function number)***

A function as a partial address of a function block is also addressed with a function number.
The number is within ‘0’ and ‘99’.
Function number ranges:

- 0 function in general
- 1 - 99 other functions

Rem. F ***Type (function type)***

A function type number is also allocated to each function block. The number is within ‘0’ and ‘111’.
Function type ranges:

- 0 function type in general
- 1 - 111 other function types

4.2.2. General communication structure

For transmitting the parameters required for a function block protocol via an 8-byte data window, the access comprises three sections:

- Order header with specification of code, FB no., fct.no., type and following real and integer values.
  Start telegram structure:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	ID1	Code	FB no.	Fct._no.	Type	Numb.real val.	Numb.integer val.

- n data blocks with the useful data to be transmitted
  Structure of data telegram:
  a) Transmission of REAL values

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	Count			Float			

  b) Transmission of fixed point integer values

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	Count					Integer	

  c) Transmission of Char values

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	Count				Char		

- an end block, provides the operation result
  Structure of end telegram:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID		Result					

Signification of Result	
0	OK
1	timeout occurred
2	parity error
3	faulty BCC
4	NAK

Reading or writing is always started by the master. If the number of real and integer values is 0, a write service, otherwise a read service is started.

The code determines the access type:

Code < 100, no multiple of 10 → individual access
Code < 100, multiple of 10 → tens block access
Code > 100 → block access overall block

4.2.3. *Data write procedure*

Start telegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	ID1	Code	FB no.	Fct._no.	Type	Number of real values	Number of integer values

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10							

Data telegrams:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	count			Value

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	count			

Thereby, the first value is sent with count = 1, for flow checking, count is mirrored by KS94 (≥ once).
The values are transmitted in the order Real Integer.

End telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x16				

	Byte 0	Byte 1	Byte 2-3	Byte 4 - 7
Controller replies:	0x16		Result	

4.2.4. *Data read procedure*

Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	ID1	Code	FB no.	Fct._no.	Type	0	0

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10						Number of real values[1)]	Number of integer values[1)]

Data telegrams:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	count			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	count			Value

Thereby the first value is sent with count = 1, for flow checking, count is mirrored by the master (≥ once). The values are transmitted in the order Real Integer.

End telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x16				

	Byte 0	Byte 1	Byte 2-3	Byte 4 - 7
Controller replies:	0x16		Result	

4.3. Examples

4.3.1. *Principle of the function block protocol*

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

The access mechanisms are:

4.3.2. *Individual access*

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

Valid values for ID1:

Configuration as FixPoint:	0, F0	Float values are transmitted as integer (without digits behind the decimal point)
	1, F1	Float values are transmitted as FixPoint (1 digit behind the decimal point)
Configuration as float:	0, F0	Transmission as integer in the 2nd data word (LSword)
	1, F1	Transmission as a 4 byte float value.

Example 1: (Message structure with data transmission)
Transmission of parameter set number (ParNr = 3) to the controller.
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	31	50	5	90	0	1

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10							

Data telegrams:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	1			3

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	1			

End telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x16				

	Byte 0	Byte 1	Byte 2 - 3	Byte 4 - 7
Controller replies:	0x16		0	

Example 2: (Message structure with data request)
Reading the error code of self-tuning heating (MSG1) from the controller.
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	35	50	5	90	0	0

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10						0	1

Data telegrams:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	1			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	1			2 (ok)

End telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x16				

	Byte 0	Byte 1	Byte 2 - 3	Byte 4 - 7
Controller replies:	0x16		0	

4.3.3. Block access (tens block)

Max. nine process values (always as REAL values) of a function can be read with this access (code x0).

Valid values for ID1:		
a) Configuration as Fix-Point:	0, 1	transmission is as FixPoint value.
b) Configuration as Float:	0, 1	transmission is as a 4-byte float value.

Example: (Message structure with data request)
Reading the set-points (W_{ext}, dW_{ext} and W_{lim}) from the controller.
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	20	50	0	90	0	0

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10						0	3

Data telegrams:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	1			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	1			150

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	2			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	2			10

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	3			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	3			400

End telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x16				

	Byte 0	Byte 1	Byte 2 - 3	Byte 4 - 7
Controller replies:	0x16		0	

4.3.4. 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 32 '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.

The message structure with block accesses with code B2/B3 is shown using two examples. The order of data to be transmitted is given in the relevant code table.

Valid values for ID1:

Configuration as FixPoint	0, 1, F0, F1	Float values are transmitted as FixPoint value, integer values are transmitted without change as integer.
	2, F2	Transparent transmission as a character string (for stricture, see ISO1745 message), word transmission stored in the LSByte
Configuration as float:	0, 1, F0, F1	Float values are transmitted as a 4-byte float value, integer values as an integer in the 2nd data word (LSword).
	2, F2	Transparent transmission as a character string (for structure, see ISO1745 message), word transmission stored in the LSByte

Example 1: (Message structure with data request)
Reading the set-point parameters (W0, W100, W2, Grw+, Grw− and Grw2) from the controller.
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	B2	50	1	90	0	0
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10						6	0

Data telegrams:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	1			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	1			0

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	2			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	2			700

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	3			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	3			100

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	4			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	4			-32000

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	5			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	5			-32000

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	6			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	6			-32000

End telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x16				

	Byte 0	Byte 1	Byte 2 - 3	Byte 4 - 7
Controller replies:	0x16		0	

Example 2: (Message structure with data transmission)
Writing the alarm configuration (C.600, C.620, C640 and C.660) to the controller.
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	B3	51	0	45	0	4

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10						0	0

Data telegrams:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	1			0120

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	1			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	2			0120

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	2			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	3			0241

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	3			

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x68	4			0740

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Controller replies:	0x68	4			

End telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sends:	0x16				

	Byte 0	Byte 1	Byte 2 - 3	Byte 4 - 7
Controller replies:	0x16		0	

4.4. Data types

Data values are divided into data types for transmission.

- FP[1)]
  Floating Point value
  Range: -9999 ... -0.001, 0, 0.001 ... 9999
- INT
  positive integer value
  Range: 0 ... 32767
  Range with configuration words: 0000 ... 9999 (→ page)
  Exception: switch-off value '-32000'
- ST1
  Status, bit-oriented, 1 byte length
  Range: 00H ... 3FH, transmitted: 40H...7FH
  Only 6 bits can be used for information transmission, i.e. bits 0...5 (LSB = bit 0). Bit 6 must always be set to '1', to avoid confusion with the control characters. Bit 7 contains the parity bit.
- CHAR5/16
  Text string comprising n characters, presently defined n=1, n=5, n=16
  permissible characters: 20H...7FH
- ICMP (**I**nteger **C**o**mp**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 other bits are '0'
internal hex value: 0x2002, as integer value: 8194, transmitted ASCII value: '8194'

[1)] In the following text, number with variable decimal point (floating point number) are described as FP.

5. Quick entry

The disk enclosed in the engineering set contains the GSD file, project examples for a SIMATIC® S5/S7, the type file and configuration examples for COMPROFIBUS. Communication with a KD94/DP can be built up easily by means of configuration and project.

5.1. Quick entry with S5

Test environment
The following components are required for the test set-up:

- Programming unit (PG740 recommended)
- Automation unit
  - S5-U95 / DP or
  - S5-U115, S5-U135 or S5-U155 with IM 308-C
- KS94/DP
- Engineering set (order no. 9407 999 05201)
- Cable
  - PROFIBUS cable automation unit / IM 308-C ↔ KS94/DP
  - Programming unit ↔ automation unit
  - MPI cable with S5-95U/DP ↔ Programming unit

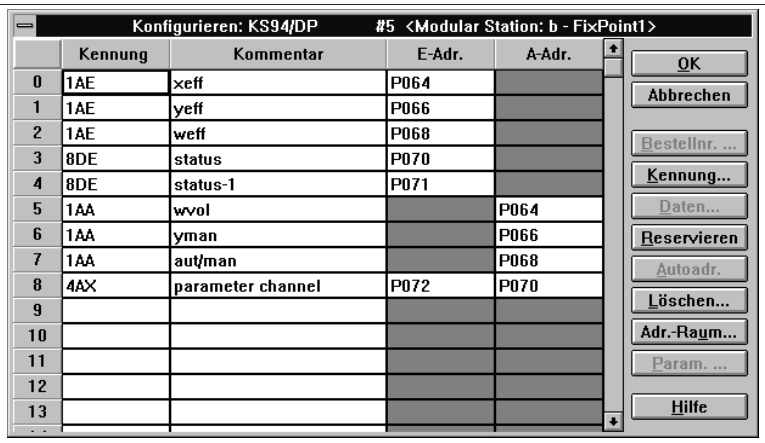
Test environment example:
A KS94/DP with address 5 shall be connected to an S5-95U/DP or IM 308-C via PROFIBUS-DP. Process data module B is selected (standard controller values and parameter channel). Data shall be transmitted in Fix Point format. The I/O addresses in the S5 start in the P-area with 64.

Before taking the test environment into operation, you should ensure that the automation units do not contain user software (“initial clear”). The same applies to the IM 308-C memory card.

Procedure:

- Make the connections
- Configure the instruments
  - Connect KS94/DP to the supply voltage and adjust address 5 (via front panel or engineering tool), df,.hndfghnfgslbmgflkn
  - Activate bus terminating resistors at controller and (S5) PLC connector.
  - Apply 24 V to Remote (di3).
- PROFIBUS network configuration
  - Insert the disk (engineering set) into the PG.
  - Call up COM ET200 and load example (A:\ks94dp\typ\example\...)
    - Select correct CPU type with IM308C.
  - If necessary, match addresses and DP network and transmit them into the DP master (→ Fig.:).
- Load the S5 program
  - Insert disk (engineering set) into PG.
  - Call up STEP® 5.
  - Load program example, e.g. (A:\ks94/dp\s5-fb\example.fix)
  - If necessary, match the addresses for send/receive window (A-A/E-A in FB) and transmit them into the automation unit.
  - Switch the automation unit to Run.

Fig.8: Configuration example module b (with COM ET200 for SIMATIC® S5-95U)



After taking the test set-up into operation, testing the I/O area and calling up the parameter channel can be done by means of the picture blocks enclosed in the project.

Picture block 1:
Includes all process data of data module b (Fix Point).
Example: (set-point = 30)
Value 300 is written into output word 64.

```
Operanden:                    Signalzustaende:
-Xeff        EW   64          KF=+265
-Yeff        EW   66          KF=+70
-Weff        EW   68          KF=+300
-Status      EB   70          KM=01000000
-Status-1    EB   71          KM=01000000

-Wvol        AW   64          KF=+300
-Yman        AW   66          KF=+0
-Auto/Man    AW   68          KF=+0
```

Picture block 2:
This picture block can be used for access to the parameters of the function block for parameter channel display.

When reading/writing values, the following data must be specified:
Example: (set-point = 30)

- Code = 32 (→ Table ‘set-point’ page)
- FBno = 50 (→ Section ‘CONTR’ page)
- FCTno = 1 (→ Table ‘Set-point’ page)
- Typ = 90 (→ Section ‘CONTR’ page)
- Read/Wr = 1
- Setting

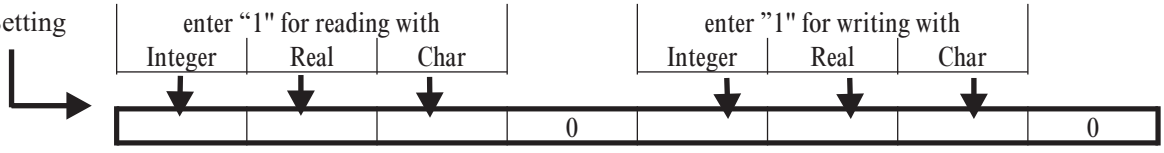
```
Operanden:                    Signalzustaende:
-DWLR        MW   52          KF=+1
-DWLI        MW   54          KF=+0
-DWLC        MW   56          KF=+0
-Read/Wr     MW   58          KH=0001

-Code        MW   60          KF=+32
-FBno.       MW   62          KF=+50
-FCTno.      MW   64          KF=+1
-Type        MW   66          KF=+90

-ANZW        MW   68          KM=00000000 00000010

-Setting     MB    0          KM=00000000

.........    DB   12
-DBval1      DW   11          KF=+300
```



- ANZW indicates status and result after completing the FB handling.
- DWLR, DWLI, DWLC indicate the number of read values.

Picture block 3:
This picture block indicates the first data of the data block, into which data of the parameter channel are written, or from which values are read.

```
Operanden:                    Signalzustaende:
.........    DB   12
-DBval1      DW   11          KF=+0
-DBval2      DW   12          KF=+9000
-DBval3      DW   13          KF=+1000
-DBval4      DW   14          KF=-32000
-DBval5      DW   15          KF=-32000
-DBval6      DW   16          KF=-32000
-DBval7      DW   17          KF=+0
-DBval8      DW   18          KF=+0
-DBval9      DW   19          KF=+0
-DBval10     DW   20          KF=+0
-DBval11     DW   21          KF=+0
-DBval12     DW   22          KF=+0
-DBval13     DW   23          KF=+0
-DBval14     DW   24          KF=+0
.........    DW   25          KF=+0
.........    DW   26          KF=+0
.........    DW   27          KF=+0
.........    DW   28          KF=+0
.........    DW   29          KF=+0
```

5.2. Quick entry with S7

Test environment
The following components are required for the test set-up:

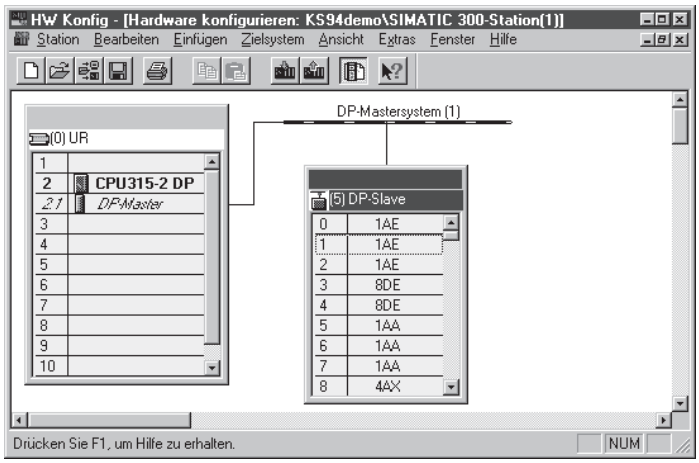
- Programming unit (PG740 recommended)
- Automation unit
  - CPU315-2 DP
- KS94/DP
- Engineering set (order no. 9407 999 05x01)
- Cable
  - PROFIBUS cable automation unit ↔ KS94/DP
  - programming unit ↔ automation unit

Test environment example:
A KS94/DP with address 5 must be connected to a CPU315-2 DP via PROFIBUS-DP. Process data module B is selected (standard controller values + parameter channel). Data shall be transmitted in Fix Point format.

Before taking the test environment into operation, you should ensure that the automation units do not contain user software (”initial clear”).

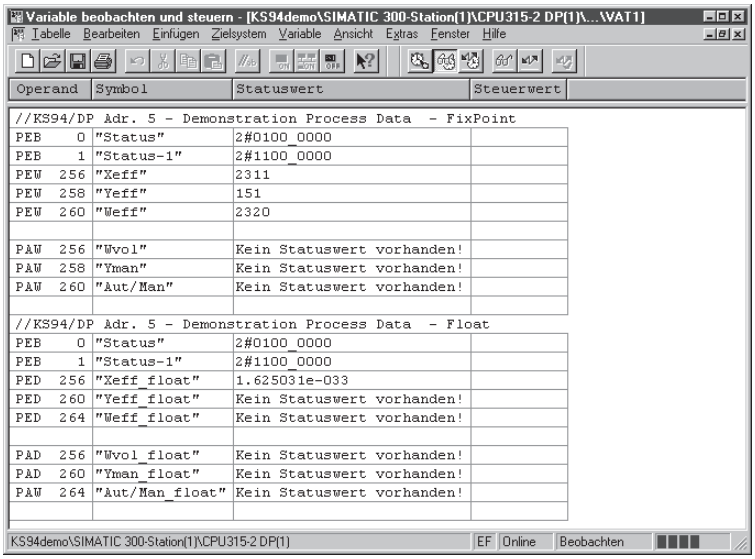
Procedure:

- Make the connections (PROFIBUS)
- Configure the instruments
  - Adjust address 5 at KS94/DP (via front panel or engineering tool) and connect to network.
  - Activate the bus termination resistors at controller and at PLC (S7) connector.
  - Apply 24 V to Remote (di3).
- PROFIBUS network configuration
  - Insert floppy (engineering set) into the programming unit.
  - De-archive project example (A:\KS94DP\S7_FB \EXAMPLE \K94demo.arj)
  - Open project KS94demo
  - If necessary, match addresses and CPU hardware configurations and transmit them into the DP master (CPU315-2 DP).
  - Switch the automation unit to Run.

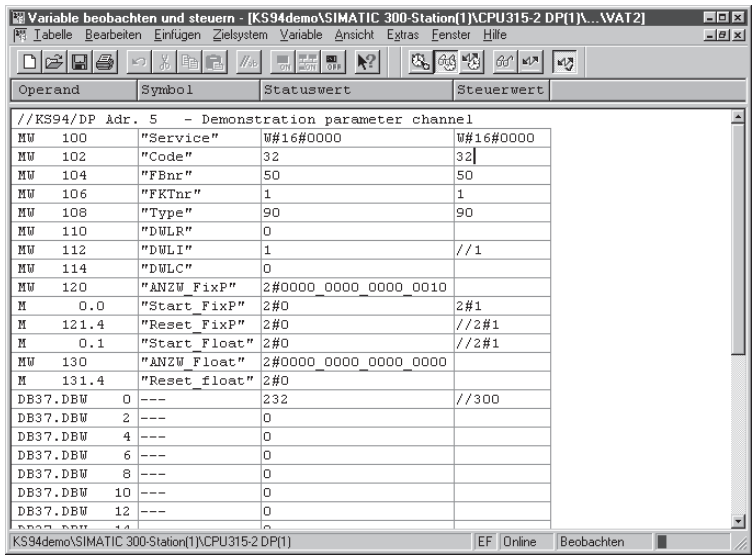


After taking the test set-up into operation, testing the I/O area and calling up the parameter channel can be done by means of the picture block variable table_variable tables (VAT x)

VAT 1:
The process data of data module b are listed.



VAT 2:
This variable table permits access to the parameters of the function block for parameter channel mapping. The first data of a data block into which data of a parameter channel are written or from which values are read are shown in the lower part of the picture.



The following values must be specified e.g. when reading fixpoint values:

Example: (set-point entry = 30)

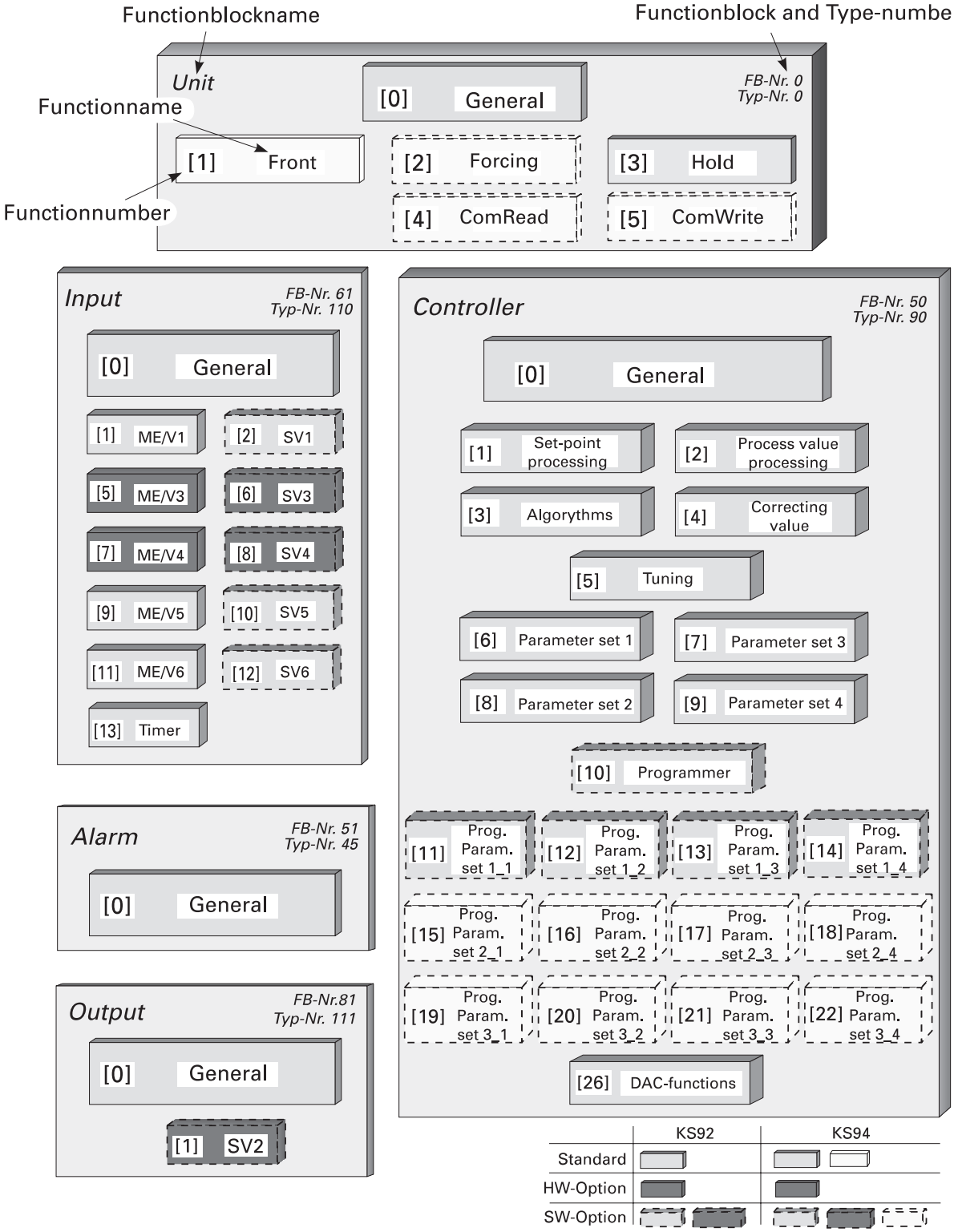
- Service = 0x 0001
- Code = 32 (→ table ‘Set-point’ page 45)
- FBnr = 50 (→ chapter ‘CONTR’ page 44)
- FKTnr = 1 (→ table ‘Set-point’ page 45)
- Type = 90 (→ chapter ‘CONTR’ page 44)
- ANZW indicates status and result after completing the FB handling.
- DWLR, DWLI, DWLC indicates the number of read values.
- Start_FixP = 1
- ANZW_FixP indicates status and result after completing the FB handling.

6. Function block protocol

6.1. Data structure

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

Fig.9: Survey of KS92/94 function blocks and functions



6.2. CODE tables

6.2.1. *Configuration words (C.xxxx)*

The configuration words listed in the following code tables comprise several partial components, 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	**H**undreds	**T**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 0 4**

For transmission of configuration words, see section Page 23.

6.2.2. *INSTRUMENT (FB no.: 0 Type no.: 0)*

All data which are valid for the overall instrument are grouped in function block ‘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	
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 option: 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 configuration mode (only after 1)	0	
				Switch over instrument to on-line mode (only after 0)	1	
				Cancellation of configuration mode (only after 0)	2	
33	UPD	R/W	INT	Acknowledgement of local data change	0..1	**G**

Rem. A Unit_State1

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

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

Rem. 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 output. (12NC e.g. 9407 924xx xxx or 9407 928xx xxx).

Rem. C HWext

Module B		Module C	
T	**H**	**Z**	**E**

not fitted	**00***	**00***	not fitted
TTL interface	**01**	**01**	A) 1 analog output (continuous) (OUT3)
RS485/422 interface	**02**	**02**	B) 2 analog inputs (INP3, INP4)
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 equipped with a module B as TTL interface without real time clock and a C module. (12NC e.g. 9407-9xx-16xxx).

Rem. 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	Self-tuning with the process lined out disabled	Self-tuning with the process lined out enabled
EXT	Standard	extension enabled

Example: Value '*SWopt = 13*' means that options signal processing, measurement value correction and self-tuning with the process lined out are enabled for the addressed instrument. (12NC e.g. 9407 9xxxx 3xx).

Rem. E SWCod

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

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

Rem. F SWvers

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

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

Rem. G Local-Switch

Switch-over to local operation (only valid with digital input di3 REMOTE closed):
0: switch off local (default)
1: switch on local

Rem. H UPD

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

Rem. I BSAct

INTERBUS – process data structure – activate switch-over
Internal datum reset to 0 is automatic (trigger function).

Command writing can lead to INTERBUS standstill and affect the overall system safety in a non-intended manner.

Forcing (function no.: 2)

Code	Descr.	R/W	Type	Description	Range	Rem.
30	Block 31...39	R	Block			
31	FInp 1	R/W	FP	Forced Inp 1 (signal input before measured value correction for INP1)		
32	FInp 3	R/W	FP	Forced Inp 3 (signal input before signal pre-processing)		
33	FInp 4	R/W	FP	Forced Inp 4 (signal input before signal pre-processing)		
34	FInp 5	R/W	FP	Forced Inp 5 (signal input before signal pre-processing)		
35	FInp 6	R/W	FP	Forced Inp 6 (signal input before signal pre-processing)		
36	Fdi	R/W	ICMP	Forced digital inputs di1...di12		**J**
37	FOut 1	R/W	FP	Forced Out 1		**K**
38	FOut 3	R/W	FP	Forced Out 3 (signal input before post-processing)		
39	Fdo	R/W	ICMP	Forced digital outputs Out 1...Out5; do1...do6 (also used for disabling outputs when configured accordingly Configuration 0 = enabled; 1 = disabled)		**L**

Rem. J Data structure of '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

Rem. K Range

Dependend 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.

Rem. L Data structure of '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

ComRead (function no.: 4)

Code	Name.	R/W	Type	Description	Range	Default	Rem.
20	Block 21...29	R	Block	Block Access	21...29		
21	Val 1	R	BCD	Value 1			
22	Val 2	R	BCD	Value 2			
23	Val 3	R	BCD	Value 3			
24	Val 4	R	BCD	Value 4			
25	Val 5	R	BCD	Value 5			
26	Val 6	R	BCD	Value 6			
27	Val 7	R	BCD	Value 7			
28	Val 8	R	BCD	Value 8			
29	Val 9	R	BCD	Value 9			
30	Block 31...37, 39	R	Block	Block Access	31...37, 39		
31	Val 10	R	BCD	Value 10			
32	Val 11	R	BCD	Value 11			
33	Val 12	R	BCD	Value 12			
34	Val 13	R	BCD	Value 13			
35	Val 14	R	BCD	Value 14			
36	Val 15	R	BCD	Value 15			
37	Val 16	R	BCD	Value 16			
39	ResetRead	R/W	INT	Configuration assignment reset	0:d.c. / 1: reset		

ComWrite (function no.: 5)

Code	Name.	R/W	Type	Description	Range	Default	Rem.
20	Block 21...29	R	Block	Block Access	21...29		
21	Val 1	(R)/W	BCD	Value 1			
22	Val 2	(R)/W	BCD	Value 2			
23	Val 3	(R)/W	BCD	Value 3			
24	Val 4	(R)/W	BCD	Value 4			
25	Val 5	(R)/W	BCD	Value 5			
26	Val 6	(R)/W	BCD	Value 6			
27	Val 7	(R)/W	BCD	Value 7			
28	Val 8	(R)/W	BCD	Value 8			
29	Val 9	(R)/W	BCD	Value 9			
30	Block 31...39	R	Block	Block Access	31...39		
31	Val 10	(R)/W	BCD	Value 10			
32	Val 11	(R)/W	BCD	Value 11			
33	Val 12	(R)/W	BCD	Value 12			
34	Val 13	(R)/W	BCD	Value 13			
35	Val 14	(R)/W	BCD	Value 14			
36	Val 15	(R)/W	BCD	Value 15			
37	Val 16	(R)/W	BCD	Value 16			
38	BusConf	R/W	INT	Selection of bus process data module	0...3	1	**M**
39	ResetRead	R/W	INT	Configuration assignment reset	0:d.c. / 1: reset	0	

Rem. M IBS structure assignments

0: Structure A.0 ”standard function”
1: Structure A.1 ”standard function with PCP 2 word” (default)”
2: Structure B ”extended function with PCP 2 word””
3: Structure C ”Flexible selection function”

A change of this structure is activated only after switching on the voltage again, or setting the activation flag (see next page).

1)Value is reset to 0 after handling
2)Structure changing causes changing of the data length on the INTERBUS. This means short-time interruption of the overall bus and may be done by the user only purposefully, whereby all the consequences must be taken into account.

Parameter and configuration data

General

(function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	F_{Key}	R/W	INT	Function of front-panel key	0 .. 2	
	Lock	R/W	INT	EBloc: blocks change of extended operating level (T) HBloc: blocks the auto/manual key (H) CBloc: blocks the controller switch-off (Z) WBloc: blocks set-point adjustment (E)	0 .. wxyz	
	Disp2	R/W	INT	PBloc: blocks program preset (T) RBloc: blocks programmer Run/Stop/Reset (H) OBloc: blocks the self-tuning (Z)	0 .. xyz0	
B3	C900	R/W	INT	Prot: Protocol type (T) Baud: Baudrate (d.c. with PROFIBUS) (H,Z)	0..xyy0	
	Addr[(1)]	R/W	INT	Device address: ISO1745 PROFIBUS	0..99 1..127	
	C902	R/W	INT	Freq: mains frequency 50/60 (T)	0..x0yz	
	C800	R/W	INT	Text2: signification of display field Text2 (T) UsrTx: user text selection (H) LED: function of front-panel LEDs (Z) Langu: language selection for plain text display (E)	0..wxyz	
	C801	R/W	INT	LUnit: unit selection (T,H) xDisp: process value display (Z)	0...xxy0	

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-defined unit	20h...7Fh	

String 1 to string 12 must always contain 16 characters and unit must always comprise 5 characters (fixed!).

[1)] Baudrate and adress setting are only effective after an initialization, e.g. protocol switch-over
[2)] For transmitting the user texts via PROFOBUS-DP, a data module of min. 216 DW + management data is necessary.

Forcing Input and output forcing function no.: 2)

Code	Descr.	R/W	Type	Description		Range	Rem.
B3	C910	R/W	INT	Forcing INP1 Forcing INP3 Forcing INP4	(T) (Z) (E)	0....x0yz	
	C911	R/W	INT	Forcing INP5 Forcing INP6	(T) (H)	0...xy00	
	C920	R/W	INT	Forcing di1 Forcing di2 Forcing di4	(T) (H) (E)	0...wx0z	
	C921	R/W	INT	Forcing di5 Forcing di6 Forcing di7 Forcing di8	(T) (H) (Z) (E)	0...wxyz	
	C922	R/W	INT	Forcing di9 Forcing di10 Forcing di11 Forcing di12	(T) (H) (Z) (E)	0...wxyz	
	C930	R/W	INT	Forcing OUT1 Forcing OUT2 Forcing OUT3 Forcing OUT4	(T) (H) (Z) (E)	0...wxyz	
	C931	R/W	INT	Forcing OUT5	(T)	0...x000	
	C940	R/W	INT	Forcing do1 Forcing do2 Forcing do3 Forcing do4	(T) (H) (Z) (E)	0...wxyz	
	C941	R/W	INT	Forcing do5 Forcing do6	(T) (H)	0...wx00	

Extended operating level Parameter entry for field bus data (function no.: 3)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Entry 1	R/W	INT	Parameter identification number	0 ... 9999	N
	Entry 2	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 3	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	O

Write field bus data Parameter entry at the extended operating level (function no.: 4)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Entry 1	R/W	INT	Parameter identification number	0 ... 9999	N
	Entry 2	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 3	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	

Read field bus data Parameter entry for field bus data (function no.: 5)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Entry 1	R/W	INT	Parameter identification number	0 ... 9999	N
	Entry 2	R/W	INT	Parameter identification number	0 ... 9999	
	Entry 3	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	

Rem. N ***Entry 1 ... 12***
Value = 0 means ‘unused entry’

Rem. O ***Hold***
Value = 0 means ‘Hold on main operating page’
Value = 1 means ‘Hold on status display at extended operating level’
Value = 2 ... 13 means ‘ Hold on entry 1 ... 12’

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

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

Tuning

Kennung	Parameter
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

Parameter set 0

Kennung	Parameter
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

Process value

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

Other

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

Digital prog. Recp 2

Kennung	Parameter
7172	D1
7173	Td2
7174	D2
...	...
7209	Td20
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

Other

Kennung	Parameter
9985	Status 1
9986	Finp1
9987	Finp3
9988	Finp4
9989	Finp5
9990	Finp6
9991	Fdi
9992	Fout1
9993	Fout3
9994	Fdo
9995	XFail
9996	Status di1
9997	Status di2
9998	Clock hour
9999	Clock Minute
10000	Clock Day
10001	Clock Month
10002	Clock Year
10003	Clock Weekday
10004	Contr Status 1

Other

Kennung	Parameter
10005	Contr Status 2
10006	Contr Status 3
10007	Y/Y2
10008	PI/P
10009	A/M
10010	0Start
10011	We/Wi
10012	W/W2
10013	W/dW
10014	Coff
10015	Wstate
10016	Wnvol
10017	Wvol
10018	Tune Status
10019	ParNeff
10020	Prog Status 1
10021	Prog Status 2
10022	Wend
10023	Seg_AD
10024	PRun
10025	PRset
10026	Pset
10027	Alarm Status 1
10028	Yman

6.2.3. INPUT (FB-Nr.: 61 Typ-Nr.: 110)

All data containing the acquisition and processing of all input values (ana./dig.) 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 inputs di1...di6		**B**
12	State_di2	R	ST1	digital inputs 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	Type no. of function block	110	

Rem. 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		

Rem. 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		

Rem. 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 : detection 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	Range	Rem.
01	State_Clock	R	ST1	Status 1		**D**
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	70...169	
36	ClkDW[(2,4)]	R/W	INT	Time weekday	0...6	

Rem. 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	Timer1 enabled		
D2	T1Out	Timer1 status		
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 of 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..wxyz	
	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	

[1] only valid with module B with real-time clock, e.g. RS485 (d.c. for PROFIBUS)
[2] The current internally available time is specified when reading. When specifying, the correct order 'year-month-day-hour-minute' for correct checking must be met..
[3] Calculation of the current 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 day of the week), 1=Tuesday,...6=Sunday; values are freely adjustble and are not evaluated presently.

ME/V1	Measurement value INP1 : detection a. 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 const. input value processing	0.0 .. 999.9	
	T_{kref}	R/W	BCD	specified TC	0...60 $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: Enable meas. value correction (Z)	1..wxy0	

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: correction b	-999..9999	
	gain	R/W	BCD	Square root extraction: gain	0 .. 9.999	
	Tf	R/W	BCD	Pre-processing: filter time constant	0.0 .. 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	

The functions for input value processing and detection of inputs INP3, INP4, INP5, INP6 are structured identically. INP3 and INP4 are available only with option card C fitted.

ME/Vx	Measurement value INPx: detection 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 const. input value processing	0.0 .. 999.9	
	INPx (Cxx0)	R/W	INT	Type: Sensor type (T,H) Dp: decimal point (E)	0..xx0y	
	Option (Cx05)	R/W	INT	Fail: Sensor break behaviour. (T)	0..x000	

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

The functions for signal pre-processing of inputs INP3, INP4, INP5, INP6 are structured identically. INP3 and INP4 are available only with option card C fitted.

Svx	Signal processing stage 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: correction b	-999..9999	
	gain	R/W	BCD	Square root extraction: gain	0 .. 9.999	
	Tf	R/W	BCD	Pre-processing.: filter time constant	0.0 .. 999.9	
B3	xs1	R/W	BCD	Segment point1 : 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)

Timer	Timer function [4]				(function no.: 13)
Code	**Descr.**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	T1SY	R/W	INT	Timer 1 start value year	0...255	(5)
	T1SMt	R/W	INT	Timer 1 start value month	1..12	
	T1SD	R/W	INT	Timer 1 start value day	1 .. 31	
	T1SH	R/S	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	(5)
	T1EMt	R/W	INT	Timer 1 end value month	1..12	
	T1ED	R/W	INT	Timer 1 end value day	1 .. 31	
	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: Operating mode time 1 (T)	0..x000	

1) Defined only for KS92: for input 5 and 6 (function 10, 12)
KS94: for inpfut 3, 4 (if module option C is provided), 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 module option C is provided.
4) Defined only for module B with realtimeclock (e.g. For PROFIBUS)
5)Calculation of the current year: data range 70...69, corresponds to 1970...2069;
example: value 96 corresponds to year 1996, value 01 corresponds to year 2001.

6.2.4. 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	Effective 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	R	BCD	Auxiliary variable		
15	OVC+	R	BCD	Override control +		
16	OVC-	R	BCD	Override control -		
18	Type	R	INT	Type no. of function block	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	Additional correcting value on/off	0..1	
32	PI/P	R/W	INT	PI/P switch-over	0..1	
33	A/M	R/W	INT	Manual/automatic 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	Set-point offset off/on	0..1	
38	Coff	R/W	INT	Controller off/on	0..1	

Rem. 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	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		

Rem. 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	not active	active
D1	BAND	Bandwidth control	not active	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		

Rem. 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.stepp.contr.	off	on
D2	DOVC+	Override control+ with 3-pnt.stepp.contr.	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	set-point offset	-999..9999	

Rem. 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 offset active		
D4	w/dWe	Ext. set-point offset active		
D5	'0'	Always '0'		
D6	'1'	Always '1'		
D7		Parity		

wp/wi	we/wi	Setpoint entry
0	1	external
1	0	programmer
1	1	internal

w/dwe	w/dw	Set-point offset
0	0	not active
0	1	active, int. set-point
1	0	active, ext. set-point

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

Code	Desc.	R/W	Type	Description	Range	Rem.
30	Block	R	Block	Block access (31, 32)		
31	dYman	R/W	FP	Differ. output variable	-210..210	
32	Yman	R/W	FP	Absolute output variable	-105..105	
33	DAC©	R/W	FP	Start of automatic calibration of position feedback Yp	0...1	

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 of 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	R	INT	Error code of self-tuning cooling	0...8	

Rem. 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	Status 1 programmer		**F**
2	State_Prog2	R	ST1	Status 2 programmer		**G**
3	PNreff	R	BCD	Eff. program number	1..3	
4	Tnet	R	BCD	Program time net	0 .. 9999	
5	Tbrut	R	BCD	Program time gross	0 .. 9999	
6	Wp	R	BCD	Set-point programmer	-999 .. 9999	
7	Trest	R	BCD	Rest time programmer	0 .. 9999	
8	Wend	R	BCD	End value act. segment	-999 .. 9999	
9	Seg AD	R	INT	Segment no. analog/digital	0101 .. 2020	
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	Program continue / reset	0 .. 1	
34	PSearch*	R/W	INT	Start program search	0 .. 1	
35	PSet	R/W	BCD	Program preset value Pmode = Seg Pmode = time	1...20 0...9999 (min)	
36	LC-	R/W	BCD	Bandwidth lower limit	0...9999 a. '____'	(1)
37	LC+	R/W	BCD	Bandwidth upper limit	0...9999 a. '____'	

* Function is presently not supported.

1) LC-/LC+ are used as set-point gradient. These data are effective only with software option for programmer without configuring a programmer. These data are identical with LC-/LC+ of the first program.

Rem. 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	$P_{\underline{Run}}$	Progr. run	stop	running
D1	$P_{\underline{End}}$	Progr. end	no	yes
D2	$P_{\underline{Res}}$	Progr. reset	off	on
D3...D5	'0'	Always '0'		
D6	'1'	Always '1'		
D7		Parity		

Rem. 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 a. 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	C103(Xn0)	R/W	BCD	Lower calibration limit X1	-999..9999	
	C104(Xn100)	R/W	BCD	Upper calibration limit X1	-999..9999	
	C108(X_{min})	R/W	BCD	Lower process value limit	-999..9999	
	C109(X_{max})	R/W	BCD	Upper process value limit	-999..9999	
	C110	R/W	BCD	S factor	0.01 .. 99.99	
	C100	R/W	INT	CFunc: Controller function (T,H) CType: Controller type (Z) WFunc: Set-point function (E)	0..xxyz	
	C101	R/W	INT	CMode: Controller output action (T) CDiff: x, Xw-different. (H) CFail: Behaviour with sensor fail (Z)	0..xyz0	
	C102	R/W	INT	XnDp: Decimal point f. X standard (E)	0..000x	
	C105	R/W	INT	C Aux: Auxiliary variable (T,H) C OVC: Output limiting (Z)	0.. xxy0	
	C106	R/W	INT	WTrac: Tracking int. set-point (T) WD: Set-point offset (H) WSel: Set-point selection (Z)	0..xyz0	
	C107	R/W	INT	Ratio: Ratio control function (T) XDp: Deciimal point for process value (E)	0..x00y	
	C700	R/W	INT	OMode: Type of self-tuning (T) OCond: Process at rest (H) OCntr: Operating mode contr. tuning (E)	0..xy0z	

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

Code	Descr.	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 plus	>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	

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

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	N0	R/W	BCD	Zero offset/ratio	-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	Differentiation time constant for auxiliary variable	0...9999	

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	Switching difference signaller	0..9999 %	
	LW	R/W	BCD	Trigger point separation additional p.c.b.	-999..9999	
	Xsd2	R/W	BCD	Switching difference additional p.c.b.	0..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 action time 1	0..9999	
	Tv1	R/W	BCD	Act. derivative action 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 action 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	

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

Code	Descr.	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	
	Y2	R/W	BCD	Additional correcting value	-105..105	
	Y0	R/W	BCD	Working point for correcting variable	-105..105	

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

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	YOptm	R/W	BCD	Correcting variable during process at rest	-105..105	
	dYopt	R/W	BCD	Change of output when self-tuning runs	5..100	
	OXsd	R/W	BCD	Hysteresis with 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 which shall be optimized	0...3	

1) Datum has switch-off function; additional data value '-32000'.
2) The user must ensure that condition Trig1 < Trig2 < Trig3 is met for the trigger points.

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 action time 1	0..9999	
	Tv1	R/W	BCD	Derivative action 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 action time 2	0..9999	
	Tv2	R/W	BCD	Derivative action time 2	0..9999	
	T2	R/W	BCD	Min. duty cycle 2	0.4..999.9	

Programmer — Programmer operation(function no.: 10)

Code	Descr.	R/W	Type	Description	Range	Rem.
B3	C120	R/W	INT	PSel: Source f. program selection (T)	0...1	
				PwrUp: Behaviour at mains recovery (H)	0...4	
				Pend: Behaviour at program end (Z)	0...3	
				PStrt: Function of run/stop (E)	0...1	

The parameters for a program (max. 20 segment points a. 4 control outputs) are distributed to 4 functions due to the limited capacity of the transmission buffer.

Program x_1 — Programmer parameter set part 1 (function no.: 11,15,19)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Wp1	R/W	BCD	W value segment	0-999..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	
	Wp0	R/W	BCD	Reset value W0	-999..9999	
	LC-(1)	R/W	BCD	Bandwidth lower limit	0 .. 9999	
	LC+(1)	R/W	BCD	Bandwidth upper limit	0 .. 9999	
	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	
	Wmode	R/W	INT	Change mode	0 .. 1	
	Pmode	R/W	INT	Preset mode (0=segment; 1=time)	0 .. 1	
	PNext(1)	R/W	INT	Sequence program	1 .. 3	

[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	Wp11	R/W	BCD	W value segment 10	-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	
	Tp11[1]	R/W	INT	T value segment 10	0..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	

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	
	Td3	R/W	INT	T value segment 2	0..9999	
	Td4	R/W	INT	T value segment 3	0..9999	
	Td5	R/W	INT	T value segment 4	0..9999	
	Td6	R/W	INT	T value segment 5	0..9999	
	Td7	R/W	INT	T value segment 6	0..9999	
	Td8	R/W	INT	T value segment 7	0..9999	
	Td9	R/W	INT	T value segment 8	0..9999	
	Td10	R/W	INT	T value segment 9	0..9999	
	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	

[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	
	Td13	R/W	INT	T value segment 12	0..9999	
	Td14	R/W	INT	T value segment 13	0..9999	
	Td15	R/W	INT	T value segment 14	0..9999	
	Td16	R/W	INT	T value segment 15	0..9999	
	Td17	R/W	INT	T value segment 16	0..9999	
	Td18	R/W	INT	T value segment 17	0..9999	
	Td19	R/W	INT	T value segment 18	0..9999	
	Td20	R/W	INT	T value segment 19	0..9999	
	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	

DAC©		Motor actuator monitoring (Digital actor control)				(function no.: 4)
Code	**Name**	**R/W**	**Type**	**Description**	**Range**	**Rem.**
B2	---	R/W	FP	Always “0”	0	**1)**
	---	R/W	FP	Always “0”	0	
	---	R/W	FP	Always “0”	0	
	---	R/W	FP	Always “0”	0	
	---	R/W	FP	Always “0”	0	
	DAC	R/W	INT	DAC®function checking	0...1	
	---	R/W	INT	Always “0”	0	
	---	R/W	INT	Always “0”	0	
	---	R/W	INT	Always “0”	0	
	---	R/W	INT	Always “0”	0	

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

6.2.5. *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	Type no. function block	45	

Rem. 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 a. configuration data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
B2	Lim_{L1}	R/W	BCD	lower limit value 1	-999..9999	1)
	Lim_{H1}	R/W	BCD	Upper limit value 1	-999..9999	
	x_{sd1}	R/W	BCD	Switching difference 1	0..9999	
	Lim_{L2}	R/W	BCD	Lower limit value 2	-999..9999	1)
	Lim_{H2}	R/W	BCD	Upper limit value 2	-999..9999	
	x_{sd2}	R/W	BCD	Switching difference 2	0..9999	
	Lim_{L3}	R/W	BCD	lower limit value 3	-999..9999	1)
	Lim_{H3}	R/W	BCD	Upper limit value 3	-999..9999	
	x_{sd3}	R/W	BCD	Switching difference 3	0..9999	
	Lim_{L4}	R/W	BCD	Lower limit value 4	-999..9999	1)
	Lim_{H4}	R/W	BCD	Upper 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: Modus (E)	0..xxyz	
	C620 (ALARM2)	W/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	

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

6.2.6. 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		Type no. of function block	111	

Parameter a. configuration data

General

(Function no.: 0)

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

SV 1

Signal processing for OUT 3 (Function no.: 1)

Code	Descr.	R/W	Type	Description		Range	Rem.
B3	X0_Out3 (C570)	R/W	FP	Reference value for output of 0%		999..9999	(1)
	X100_Out3 (C571)	R/W	BCD	Reference value for output of 100%		999..9999	
	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(3)	R/W	BCD	Segment point 3: X value		-999.. 9999	(1, 2)
	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	
	ys5	R/W	BCD	Segment point 5: Y value		-999.. 9999	
	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	(1, 2)
	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..x00y	(1)

SV 2

Signal processing for OUT 1 (function no.: 2)

Code	Name	R/W	Type	Description		Range	Rem.
B3	X0_Out1 (C.510)	R/W	FP	Reference value f. Output of 0%		999...9999	**1)**
	X100_Out1 (C.511)	R/W	FP	Reference value f. Output of 100%		999...9999	
	C505	R/W	INT	Func: function selection DP: decimal point	(T (E))	0...x00y	

[1] Only defined with HW option C provided
[2] Only defined for KS94 with the relevant HW option
[3] Date with switch-off function; additional data value

7. Function block's

7.1. Function block for SIMATIC® S5

Function blocks FB206 (K94-FIX) and FB207 (K94-FLOAT) are used for easy access to the controller parameters and configuration data.

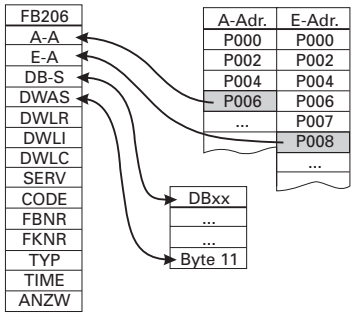
7.1.1. Structure

The parameters of the function block are:

Name	Type	Description / Function
A-A	**KF**	Start of output bytes for send window
E-A	**KF**	Start of input bytes for receive window
DB-S	**B**	Data module for parameter data
DWAS	**W**	Data word start for order in DB
DWLR	**W**	Number of real values
DWLI	**W**	Number of integer values
DWLC	**W**	Number of char values (number of all char values including the contained SUB characters!)
SERV	**W**	Service (read/write)
CODE	**W**	Code
FBNR	**W**	Function block no.
FKNR	**W**	Function no.
TYP	**W**	Type no.
TIME	**KH**	Timeout in time units, decremented at each FB call, must be higher than timeout in DP module.
ANZW	**W**	The display word gives the actual status of transmission for the selected data area. The display word structure is:

15	**14**	13	12	11	**10**	**9**	**8**	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Timeout (FB)	Service wrong				NAK (access not acknowledged by the controller)	Parity error	Timeout internal (controller)		wait for end telegram	Service (0=read; 1=write)	Reset order	waits for acknowledgement	Order finished with error	Order finished without error	Order running

The function module reads or writes KS94 parameter/configuration data.



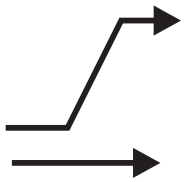
- **A-A, E-A**
  The input addresses or output addresses of the parameter channel are entered into these parameters. The addresses are determined during configuration of the PROFIBUS unit (→ Fig.: S.28)

- **DB-S**
  In DB-S, the function block-related data block in which all FB operations are executed are allocated. It must have been opened previously.
  Data words 0...3 of the data block are "provisional markers", they are free again after function block handling.

- DWAS
  DWAS marks the start of the data area in the data block. The first 4 words following DWAS are required for internal use of the function block and must not be used for other purposes.

Example: 2 data sets shall be stored in dense packing in DB25.

FB206 / FB207
A-A
E-A
DB-S
DWAS
DWLR
....
ANZW



DB25	DB-S: 25	
DW 0 ... DW3	Provisional marker	
DW6 ... DW9	Management call1	1. DWAS: 6
DW10 ... DW12	Data set 1	3 words user data
DW13 ... DW16	Management call2	2. DWAS: 13
DW17 ... DW26	Data set 2	10 words user data

- DWLR (Real), DWLI (Integer), DWLC (Char)
  These parameters contain the relevant number of received data after a read access.
  With a write access, the relevant number of data to be transmitted is specified.
- SERV
  This parameter determines the access type (write / read)

Write access:				**Read access:**			
	F0	≙	Integer (only with individual access)		0	≙	Integer (only with individual access)
	F1	≙	Real		1	≙	Real
	F2	≙	Char		2	≙	Char

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

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

Valid values for service:

Configuration as FixPoint:	0, F0	float values are transmitted as integer (without digits behind the decimal point)
	1, F1	float values are transmitted as FixPoint (1 digit behind the decimal point)
Configuration as float:	0, F0	Transmission as integer in the 2nd data word (LSword)
	1, F1	transmission as a 4 byte float value.

Block access (tens block)
With this access (code x0), max. nine process values (always as REAL values) of a function can be read.

Valid values for ID1:

Configuration as Fix-Point:	0, 1	Each transmitted as a FixPoint value.
Configuration as Float:	0, 1	Each transmitted as a 4 byte Real value.

Block access (overall block)
With this access, all parameters (code 178) and configuration data (code 179) of a function can be read or written. For this access, the following conditions are applicable:

- For writing data with 'code 179', the unit must be switched to the configuration mode previously (→ see page 'OpMod'). All newly entered configuration data and parameters are effective only when the unit was switched back to on-line.
- All data of a message must be defined, omissions are not permissible.
- If parts of a message in the unit are not used (HW and SW options), the complete message must be transmitted nevertheless. Checking the unused data is omitted.
- With faulty block write accesses, the following information is applicable: a message is replied with NAK, if min. one datum is faulty. Already valid values are stored.

Valid values for service:

Configuration as FixPoint	0, 1, F0, F1	Float values are transmitted as FixPoint values , integer values are transmitted as integer values.
	2, F2	Transparent transmission as a character string (for structure, see ISO1745 message), word transmission stored in the LSByte
Configuration as float:	0, 1, F0, F1	Float values are transmitted as 4-byte float values, integer values are transmitted as integers in the 2nd data word (LSword).
	2, F2	Transparent transmission as character string (for structure, see ISO1745 message), word transmission stored in the LSByte

- CODE
  The code identification is decimal and the value range comprises '00'...'99' as well as '178' ≙ B2 and '179' ≙ B3.
- FBNR. (function block number)
  A function block is addressed with a function block number. It includes range '0' to '250'.
  Function block number ranges:
  0 general data for the overall unit
  1 - 99 fixed function blocks
- FKTNR (function number)
  A function as a partial address of a function block is also addressed with a function number. It comprises range '0' to '99'.
  Function number ranges:
  0 function general
  1 - 99 other functions
- TYP (function type)
  A function type number is also assigned to each function block. It is within range '0' and '111'.
  Function type ranges:
  0 Function type general
  1 - 111 other function types
- TIME
  Timeout counter: range 0x0000 ≤ TIME ≤ 0x7FFF
  - is decremented with each PLC cycle (max. 32767)
  - with 0 Timeout.
  If the CPU is too fast, FB206/FB207 must be called up with delay via timer module.
- ANZW
  This display word shows the bit mapping of the current transmission state. Bit 4 can be used as an input for resetting the FB 206 / FB 207.

7.1.2. Function block call

List:

```
      :SPA          FB 206
Name     :KS94-FIX
A-A :
E-A :
DB-S     :
DWAS     :
DWLR     :
DWLI     :
DWLC     :
SERV     :
CODE     :
FBNR     :
FKNR     :
TYP :
TIME     :
ANZW     :
```

7.2. Function module for SIMATIC® S7

The handling principle of the S7 -FB corresponds to the S5 principle. Calling up the FB is indispensable when starting an order and as long as the order is active.
Dependent of S7-CPU and DP master, there are differences in the I/O handling. When using the on-board DP interface of a CPU315-2 DP, SFC modules 14 and 15 must be used for transmitting data consistently. SFC modules 14 and 15 copy the I/O areas into the marker or data module area. When using an external CP' (CP 342-5 DP), the relevant SEND and RECEIVE FB's at the cycle start and end must be used. The FB has an instance DB which must be specified with function block call.

7.2.1. Structure

The call parameters of the function module are:

Name	Type	Description / function
A-Start	Pointer	Start of output word address area (e.g. address of data area 'RECORD' of SFC 15, Ax, y when using an external CP). When specifying a data word, the DB no must also be transmitted (e.g. DB4.DBX0.0)
E-Start	Pointer	Start of input word address area (e.g. address data area 'RECORD' of SFC 15, Ex, y when using an external CP). When specifying a data word, the DB no. must also be transmitted (e.g. DB4.DBX0.0)
DB-Para	Pointer	Specification of the data module with the parameter setting data. The entry comprises the data module no. and the data word number at which the parameter data start. An offset need **not** be taken into account. The data are interpreted as parameter data by the specified address. Specification of the data block must be as follows (e.g. DB6.DBX10.0).
Service	WORD	Service (read/write)
Code_nr	WORD	Code
FB_nr	WORD	Function block no.
FKT_nr	WORD	Function no.
Type	WORD	d.c. (always '0')
Timeout	DWORD	Timeout value, is decremented at each call. If the value is = 1, the order is cancelled with error message 'timeout'.
DWLR	WORD	Length of Real values
DWLI	WORD	Length of Integers
DWLC	WORD	Length of Char values
ANZW	W	The display word indicates the actual status of transmission for the selected data area. The structure of the display word is:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timeout (FB)	Service faulty				NAK (access not accepted by controller)	Parity error	Timeout internal (controller)		waits for end telegram	Service (0=Read; 1=Write)	Reset order	waits for acknowledgement	order finished with error	order finished without error	order running

The function module reads or writes KS94 parameter/configuration data.

- A-Start, E-Start
  Enter the input addresses or output addresses of the parameter channel in these parameters. The addresses are determined when configuring the PROFIBUS unit.
  (STEP 7 - hardware configuration)
- DB-Para
  DB-Para is a pointer to the data module into which read data are written or from which data are removed when writing.

- Service
  This parameter determines the access mode (write / read).

Write access:	F0	≙	Integer (only with individual access)	**Read access:**	0	≙	Integer (only with individual access)
	F1	≙	Real		1	≙	Real
	F2	≙	Char		2	≙	Char

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

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

Valid values for Service:

Configuration as FixPoint:	0, F0	Float values are transmitted as an integer (without digit behind the decimal point)
	1, F1	Float values are transmitted as FixPoint (1 digit behind the decimal point)
Configuration as float:	0, F0	Transmission as an integer in the 2nd data word (LSword)
	1, F1	Transmission is as a 4-byte real value.

Block access (tens block)
With this access (code x0), max. nine process values of a function can be read (always as REAL values).

Valid values for service:

Configuration as Fix-Point:	0, 1	Transmission is as a FixPoint value.
Configuration as Float:	0, 1	Transmission is as a 4-byte real value.

Block access (overall block)
All parameter (code 178) and configuration data (code 179) of a function can be read or written. For this access, the following conditions are applicable: For writing data with 'Code 179', the unit must be switched to the configuration mode previously (→ see page 'OpMod'). All newly entered configuration data and parameters are only valid when the unit was switched back to online. All data of a message must be defined, omissions are not permissible.

- If parts of a message in the instrument are not used (HW and SW options), the complete message must be transmitted nevertheless. Checking the unused data is omitted.
- The following information is applicable to faulty block write accesses: a message is replied NAK, when at least one datum is faulty. Already valid values are stored.

Valid values for service:

Configuration as FixPoint	0, 1, F0, F1	Float values are transmitted as FixPoint value, integer values are transmitted invariably as integers.
	2, F2	Transparent transmission as a character string (for structure, see ISO1745 message), word transmission stored in the LSByte
Configuration as float:	0, 1, F0, F1	Float values are transmitted as 4-byte float value, integer values are transmitted as an integer in the 2nd data word (LSword).
	2, F2	Transparent transmission as a character string (for structure, see ISO1745 message), word transmission stored in the LSByte

- Code_nr
  CodeThe code identification is decimal and the range is '00'...'99' and '178' ≙ B2 and '179' ≙ B3.
- FB_nr. (function block number)
  A function block is addressed with a function block number. It is within range '0' and '250'.
  Function block number ranges:
  0 general data for the overall unit
  1 - 99 fixed function blocks
- FKT_nr (function number)
  A function as a partial address of a function block is also addressed with a function number. It is within range '0' to '99'.
  Function number ranges:
  0 Function general
  1 - 99 other functions

- Type (function type)
  A function type number is also assigned to each function block. It is within range '0' and '111'.
  Function type ranges:
  0 function type general
  1 - 111 other function types
- DWLR (Real), DWLI (Integer), DWLC (Char)
  These parameters contain the relevant number of received data after a read access. With a write access, the relevant number of data to be transmitted is specified.
- ANZW
  This display word provides the mapping for the current transmission status. Bit 4 can be used as an input for resetting the B 206 / FB 207.

8. Annex

8.1. Terms

BCD	Data type (→ see page 24)
CHAR5/16	Data type (→ see page 24)
COM PROFIBUS	Configuration tool (also COM ET200) of the Siemens company for PROFIBUS
FB	Abbreviation of function block
Fkt	Abbreviation of function
ET	Abbreviation of engineering tool
Function	A partial function of the function block which is self-contained seen from the interface
Function block	closed sequence unit
GSD file	file of instrument data
HW	Abbreviation of hardware
ISO1745	Standard communication protocol ISO 1745, ASCII-based
PC interface	controller front panel interface for connection of an engineering tool
PCI	Process Control Instrument
PCI protocol	ISO 1745-based protocol, implemented for PMA controllers
PROFIBUS-DP	Standard communication protocol to EN50170 vol.2 (DP: decentral peripheral units)
RS422	Standard 4-wire connection, 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)
S5 / S7	PLC family of the Siemens AG
Serial interface	rear bussable controller interface
ST1	Data type (→ see page 24)
SW	Abbreviation of software
TTL	Signal level at block level
Typ	
Type file	configuration file for COM PROFIBUS

8.2. GSD file

```
;============================================================
; Device Database File for product  K S 9 4 / D P ;
Copyright (C) PMA Proze- und Maschinen Automation GmbH 1998-1999 ;
Miramstr. 87, D-34123 Kassel, Tel. +49 (0) 561/ 505 -1307 ;
Version : V2.0 ;
File:   PMA_9401.gsd ;
============================================================
#Profibus_DP
GSD_Revision= 1
Vendor_Name = „PMA GmbH”
Model_Name = „KS94/DP“
Revision = „V 2.0"
Ident_Number = 0x9401
Protocol_Ident = 0; DP
Station_Type = 0; Slave
FMS_supp = 0; only DP
Hardware_Release = “HV 01.01"
Software_Release = „SV 02.00" ;
Product supports the baud rates:
9.6_supp = 1
19.2_supp = 1 ;
31.25_supp = 1 ;
```

```
45.45_supp = 1
93.75_supp = 1
187.5_supp = 1
500_supp = 1
1.5M_supp = 1
3M_supp = 1
6M_supp = 1
12M_supp = 1 ;
max. time to answer after a request
MaxTsdr_9.6 = 60
MaxTsdr_19.2 = 60
MaxTsdr_31.25 = 60
MaxTsdr_45.45 = 60
MaxTsdr_93.75 = 60
MaxTsdr_187.5 = 60
MaxTsdr_500 = 100
MaxTsdr_1.5M = 150
MaxTsdr_3M = 250
MaxTsdr_6M = 450
MaxTsdr_12M = 800
Redundancy = 0; not supported
Repeater_Ctrl_Sig = 0; not available
24V_Pins = 0 ; not available
Bitmap_Device= "PMA9401N"
Bitmap_Diag= „PMA9401D"
Bitmap_SF= „PMA9401F" ;

;—DP-Slave related key words ——- ;

Freeze_Mode_supp = 1; supported
Sync_Mode_supp = 1; supported
Auto_Baud_supp = 1; supported
Set_Slave_Add_supp = 1; supported
User_Prm_Data_Len = 0; not supported ;
minimum slave poll cycle (based on 100us)
Min_Slave_Intervall = 1
Modular_Station = 1; modular device
Max_Module = 0x01; max. number of modules
Max_Input_Len = 76
Max_Output_Len = 76
Max_Data_Len = 152 ;
Module description ;
A.1: standard controller values
Module = „A: Standard (FixP)" 0x50,0x50,0x50,0x10,0x10,\
0x60,0x60,0x60
EndModule ; ;
A.2: standard controller values
Module = „A: Standard (Float)" 0xD1,0xD1,0xD1,0x10,0x10,\
0xE1,0xE1,0x60
EndModule ; ;
B.1: standard controller values + parameter channel
Module = „B: Standard + parameter (FixP)" 0x50,0x50,0x50,0x10,0x10,\
0x60,0x60,0x60,0xF3
EndModule ; ;
B.2: standard controller values + parameter channel
Module = „B: Standard + parameter (Float)" 0xD1,0xD1,0xD1,0x10,0x10,\
0xE1,0xE1,0x60,0xF3
EndModule ; ;
C.1: standard controller values + extended input values +
;output forcing + parameter channel
Module = „C: Ext.input+out.forcing (FixP)" \0x50,0x50,0x50,0x10,0x10,\
0x50,0x50,0x50,0x50,0x50,0x10,0x10,0x10,0x10,\
0x60,0x60,0x60,0x60,0x60,0x60,0xF3
EndModule ; ;
C.2: standard controller values + extended input values +
;output forcing + parameter channel
Module = „C: Ext.input+out.forcing (Float)" \0xD1,0xD1,0xD1,0x10,0x10,\
0xD1,0xD1,0xD1,0xD1,0xD1,0x10,0x10,0x10,0x10,\
0xE1,0xE1,0x60,0xE1,0xE1,0x60,0xF3
EndModule ; ;
```

```
D.1: standard controller values + extended input values + ;input forcing +
parameter channel
 Module = „D: Ext.input+inp.forcing (FixP)“ \0x50,0x50,0x50,0x10,0x10,\
0x50,0x50,0x50,0x50,0x50,0x10,0x10,0x10,0x10,\
0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,0x60,0xF3
EndModule ; ;
D.2: standard controller values + extended input values + ;input forcing +
parameter channel
 Module = „D: Ext.input+inp.forcing (Float)“ \0xD1,0xD1,0xD1,0x10,0x10,\
0xD1,0xD1,0xD1,0xD1,0xD1,0x10,0x10,0x10,0x10,\
0xE1,0xE1,0x60,0xE1,0xE1,0xE1,0xE1,0xE1,0x60,0xF3
EndModule ; ;
E.1: 16 selectable I/O values + parameter channel
Module = „E: 16 select. I/O values (FixP)“ \
0x11,0x11,0x51,0x51,0x51,0x51,0x51,0x51,0x51,0x51,\
0x21,0x21,0x61,0x61,0x61,0x61,0x61,0x61,0x61,0x61,0xF3
EndModule ; ;
E.2: 16 selectable I/O values + parameter channel
Module = „E: 16 select. I/O values (Float)“ \
0x11,0x11,0xD3,0xD3,0xD3,0xD3,0xD3,0xD3,0xD3,0xD3,\
0x21,0x21,0xE3,0xE3,0xE3,0xE3,0xE3,0xE3,0xE3,0xE3,0xF3
EndModule ; ;
F.1: 6 selectable I/O values + parameter channel
Module = „F: 6 select. I/O values (FixP)“ \
0x11,0x11,0x51,0x51,0x51,0x21,0x21,0x61,0x61,0x61,0xF3
EndModule ; ;
F.2: 6 selectable I/O values + parameter channel
Module = „F: 6 select. I/O values (Float)“ \
0x11,0x11,0xD3,0xD3,0xD3,0x21,0x21,0xE3,0xE3,0xE3,0xF3
EndModule ; ;G.1: Multiplexed values
Module = „G: multiplexed values (FixP)“ \
0x11,0x11,0x50,0x50,0x21,0x21,0x60,0x60
EndModule ; ;G.2: Multiplexed values
Module = „G: multiplexed values (Float)“ \
0x11,0x11,0x50,0xD1,0x21,0x21,0x60,0xE1
EndModule;
Slave_Family=5
Max_Diag_Data_Len = 8 ;
OrderNumber="9407-92x-3xxx1"
```

9. Index

A4