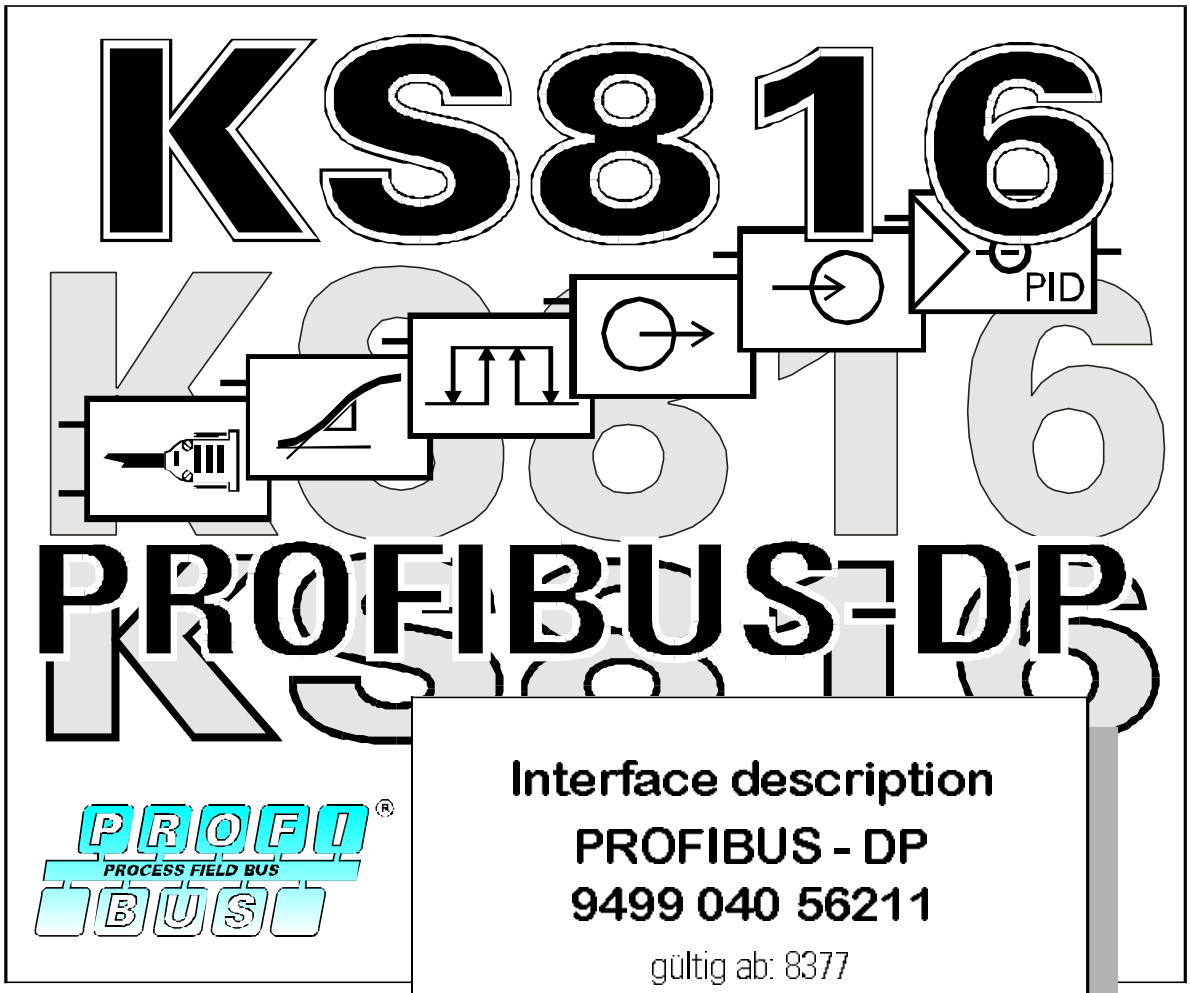
PMA Prozeß- und Maschinen-Automation GmbH

KS 816
Multi-Transmitter
Multi-Temperature Controller



Interface description

PROFIBUS - DP

9499 040 56211

gültig ab: 8377

SIMATIC® is a registered trademark of Siemens AG
STEP® is a registered trademark of Siemens AG
® is a registered trademark of the
PROFIBUS user organization (PNO)

A publication of PMA Prozeß- und Maschinen Automation
Postfach 310229
D-34058 Kassel
Germany

Contents

1 General

The KS 816 Multi-Transmitter (9407-481-30001) and Multi-Temperature Controller versions are provided with a PROFIBUS-DP interface for transmission of process, parameter and configuration data. Connection is via a 9-pole Sub-D socket. The serial communication interface permits connections to supervisory systems, visualization tools, etc.

Another interface, which is always provided as standard, is the PC interface. This interface serves for connecting an engineering tool, which runs on a PC.

Communication is according to the master/slave principle. KS 816-DP is always slave.

Cable medium as well as physical and electrical interface properties:

- Network topology
  Linear bus with active bus termination at both ends. Stub lines are possible (dependent of cable type, a maximum overall stub line length of 6,6m with 1,5Mbit/s and of 1,6 m with 3-12Mbit/s is possible).
- Transmission medium
  screened, twisted 2-wire cable (r EN 50170 vol.2).
- Baudrates and cable lengths (without repeater)
  The max. cable length is dependent of transmission rate.
  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 ... 12 Mbit/s	100m

- Interface
  RS485 with Sub-D connector (9-pole).
- Address settings
  Address setting is possible as follows:
  -Adjustment via coding switches, range 00 ... 99, default 00
  -Adjustment via software, range 0 ... 126, default 126
  With the coding switches set to '00', the adjusted software address is valid.
  A modified coding switch address is active only after switching on the supply voltage again.
- 32 instruments in one segment. Extension to 127 by means of a repeater is possible.

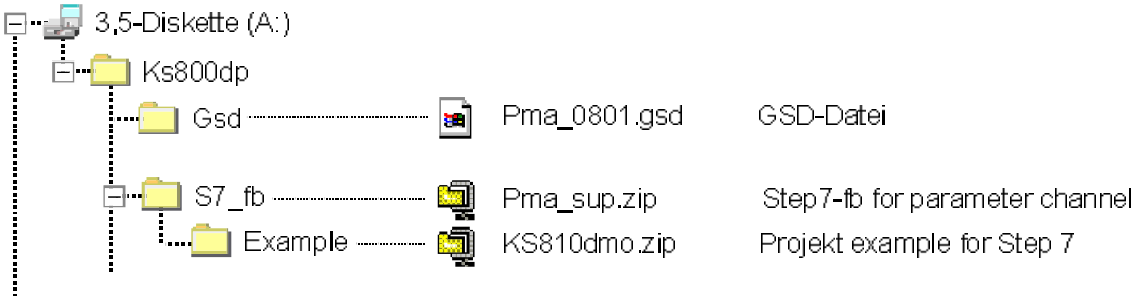
KS 816 with PROFIBUS-DP interface offers many advantages with respect to handling and integration into a PROFIBUS network.

- Diagnosis and monitoring via COM-LED
  LED off: error identification for 'no bus access' (so far not addressed by the master).
  LED on: OK, cyclic data exchange running
  LED blinks: (2Hz) data exchange interrupted
  LED blinks: (4Hz) PROFIBUS parameter setting or configuration error.
- Particularities
  Configurable process data modules
  Direct input and output reading and writing
  Easy connection to PLCs

1.1 *Scope of delivery*

The engineering set comprises:

- Disk



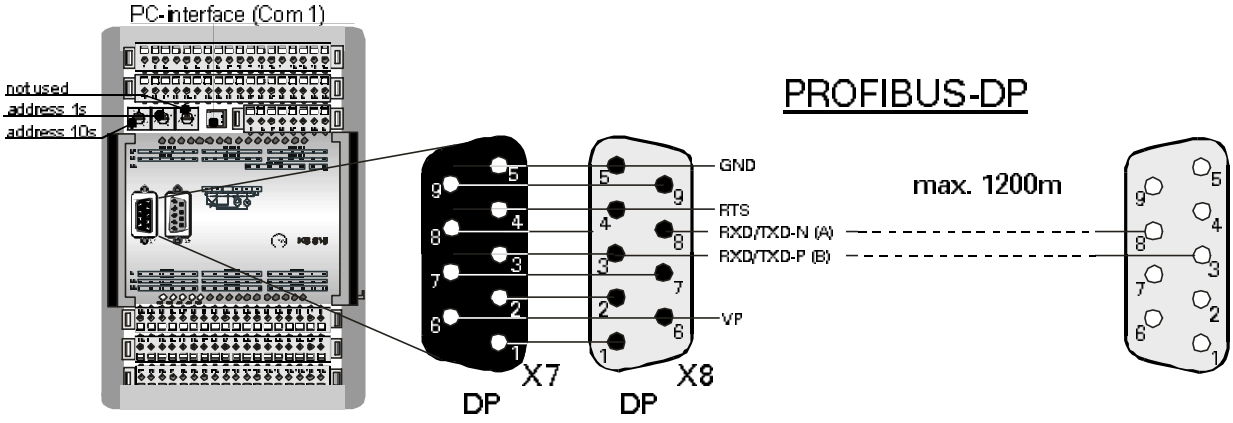
- Interface description for PROFIBUS-DP

2 Hints on operation

2.1 *Interface connection*

The PROFIBUS must be connected to the 9-pole Sub-D socket.
Serial interface, physical RS485-based signals.

Fig.: 1 Connection of PROFIBUS-DP



The construction of suitable cabling must be provided by the user, whereby the general cable specifications to EN 50170 vol.2 must be taken into account.

2.1.1 Installation of cables

When laying the cables, the general hints for cableinstallation given by the supplier of the master module must be followed:

- Cable run in buildings (inside and outside cabinets)
- Cable run inside and outside buildings
- Potential compensation
- Cablescreening
- Measures against interference voltages
- Stub line length
- Bus termination resistors are not included in the KS 816-DP scope of delivery, but must be realized via the connector, if necessary.
- Earthing

g Special hints for installation of PROFIBUS cables are given in the PNO technical guideline "***Installation guidelines for PROFIBUS-DP/FMS***" (order no. 2.111 [Germ.]; 2.112 [Engl.]).

3 Process data

During data transmission,distinction of process data to be transmitted cyclically and parameter/configuration data is made. The I/O data field is structured modularly for matching it to the requirements of the control task.
Selection of the process data module is via configuration tools of the master circuits (e.g. via COM PROFIBUS with Siemens S5).

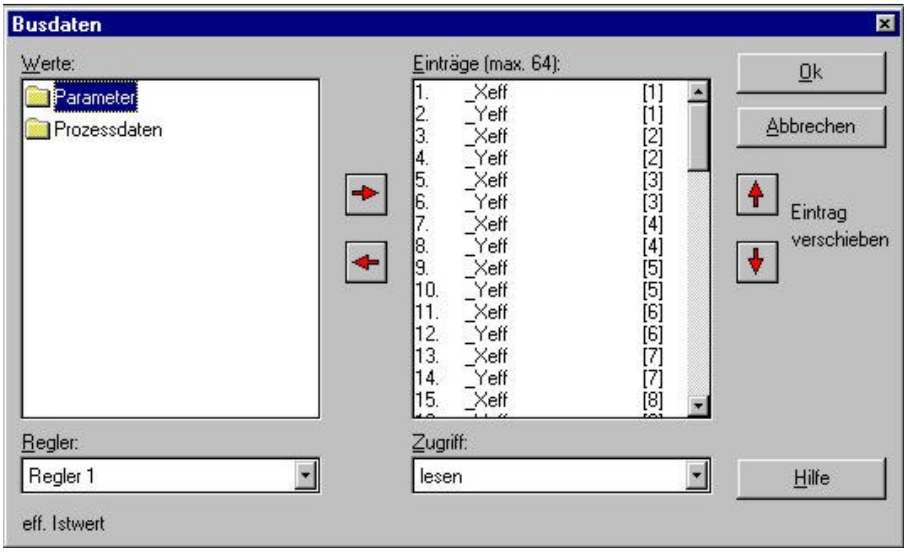
The following process data modules can be configured:

Process data module A:	Read (56 bytes) [1] Instrument status, 8 x (process value, output value, status, ..)	Write (8 bytes) [1] Instrument control, 8 x (set-point, output value, ...)	with parameter channel
Process data module B:	Read (106 bytes) [1] Instrument status, 8 x (process value, output value, status, ..)	Write (106 bytes) [1] Instrument status, 8 x (set-point, output value, ...)	with parameter channel
Process data module C:	Only parameter channel (8/8 bytes)		
Process data module D:	Read (114 bytes) [1] Instrument status, 8 x (process value, output value, status, ..)	Write (114 bytes) [1] Instrument control, 8 x (set-point, output value, ...)	with parameter channel
Process data module E:	Read (90 bytes) [1] Instrument status, (52 variable process data)	Write (90bytes) [1] Instrument control, (52 variable process data)	with parameter channel
Process data module F:	Read (14 bytes) [1] Instrument status, (40 variable process data)	Write (14 bytes) [1] Instrument control, (40 variable process data)	with parameter channel

The parameter channel is used for sequential transmission of parameter and configuration data. The values to be adjusted and the data specifications are given in the following tables:

For the process data modules (modules E + F) the cyclicaltransmission data must be selected by means of the engineering tool via `General instrument settings Communication r` **Bus data**

Fig.: 2 Instrument data assignment for the fieldbus



Max. 64 data for reading and writing can be selected. Dependent of selected process data module, the first 90 data (module E), and the first 14 data (module F), are used.

1) Number of required bytes in the required I/O field

q **Module A** (process data of all 16 channels + parameter channel - measured value acquisition)

No.	Descr.	R/W	FIX point format			Rem.
			Number of bytes	Value		
				Hex	COM PROFIBUS	
	Inputs		]L56/S8			
0	Xeff_1	R	2	50	1AE	
1	Alarm_1	R	1	10	8DE	**B**
2	Xeff_2	R	2	50	1AE	
3	Alarm_2	R	1	10	8DE	**B**
...						
30	Xeff_16	R	2	50	1AE	
31	Alarm_16	R	1	10	8DE	**B**
	Inputs/outputs					
32	Parameterchannel	R/W	8/8	F3	4AX	

q **Module B** (process data of all 16 channels + parameter channel - standard controller)

No.	Descr.	R/W	FIX point format			Rem.
			Number of bytes	Value		
				Hex	COM PROFIBUS	
	Inputs		] 106			
0	Unit_State	R	2	11	16DE	**A**
1	Xeff_1,Yeff_1, Alarm_1, Status1	R	6	52	3AE	**B C**
2	Xeff_2,Yeff_2, Alarm_2, Status_2	R	6	52	3AE	**B C**
...						
16	Xeff_16,Yeff_16, Alarm_16, Status_16	R	6	52	3AE	**B C**
	Outputs		] 106			
17	Unit_Cntr	W	2	21	16DA	**D, E**
18	Wvol_1,Yman_1,Cntrl_1	W	6	62	3AA	**F**
19	Wvol_2,Yman_2,Cntrl_2	W	6	62	3AA	**F**
...						
34	Wvol_16, Yman_16, Cntrl_16	W	6	62	3AA	**F**
	Inputs/outputs					
35	Parameterchannel	R/W	8 / 8	F3	4AX	

q **Module C** (only parameter channel)

No.	Descr.	R/W	FIX point format		
			Number of bytes	Value	
				Hex	COM PROFIBUS
	Inputs/outputs				
0	Parameterchannel	R/W	8 / 8	F3	4AX

Transmission of analog values is in 16-bit fixed point format (FIX). In FIX for mat, all values are interpreted with one di git be hind the de ci mal point (ran ge -3000,0 to 3200,0).

q **Module D** (50 variable process data and parameter channel)

No.	Descr.	R/W	FIX point format			Rem.
			Number of bytes	Value		
				Hex	COM PROFIBUS	
	Inputs		] 114			
0	Unit_State,Digital Outputs	R	6	15	16DE	**A, F**
1	IN_1 ... IN_8	R	16	57	8AE	
2	IN_9 ... IN_16	R	16	57	8AE	
...						
6	IN_41 ... IN_48	R	16	57	8AE	
7	IN_49 ... IN_50	R	4	51	4AE	
	Outputs		] 114			
8	Unit_Cntrl	R	2	21	16DA	**D, E**
9	OUT_1 ... OUT_8	R	16	67	8AA	
10	OUT_9 ... OUT_16	W	16	67	8AA	
...						
14	OUT_41 ... OUT_48	W	16	67	8AA	
15	OUT_49 ... OUT_50	W	8	61	4AA	
	Inputs/outputs					
16	Parameterchannel	R/W	8 / 8	F3	4AX	

q **Module E** (40 variable process data and parameter channel)

No.	Descr.	R/W	FIX point format			Rem.
			Number of bytes	Value		
				Hex	COM PROFIBUS	
	Inputs		] 94			
0	Unit_State,Digital Outputs	R	6	15	4DE	**A, F**
1	IN_1 ... IN_8	R	16	57	8AE	
2	IN_9 ... IN_16	R	16	57	8AE	
...						
5	IN_33 ... IN_40	R	16	57	8AE	
	Outputs		] 90			
6	Unit_Cntrl	W	2	21	16DA	**D, E**
7	OUT_1 ... OUT_8	W	16	67	8AA	
8	OUT_9 ... OUT_16	W	16	67	8AA	
...						
11	OUT_33 ... OUT_40	W	16	67	8AA	
	Inputs /Outputs					
16	Parameterchannel	R/W	8 / 8	F3	4AX	

q **Module F (multiplexing of all 64 variable process data and parameter channel)**

No.	Descr.	R/W	FIX point format			Rem.
			Number of bytes	Value		
				Hex	COM PROFIBUS	
	Inputs		] 18			
0	Unit_State,Digital Outputs	R	6	15	16DE	**A, G**
1	Index IN Read / Write	R	2	50	1AE	
2	Read Value	R	2	50	1AE	
	Outputs		] 14			
3	Unit_State	W	2	21	32DE	**D**
4	Index IN Read / Write	W	2	60	1AA	
5	Read Value	W	2	60	1AA	
	Inputs/outputs					
6	Parameterchannel	R/W	8 / 8	F3	4AX	

Procedure (read):

- Enter the index number into 'Index OUT' (read)
- After the index number is mirror-inverted in 'Index IN' (read), the read value is read in 'Read Value'.

Procedure (write):

- Enter the index number into 'Index OUT' (write)
- Enter the value to be written into 'Write Value'
- After the index number is mirror-inverted in 'Index IN' (write), the value was transmitted.

g To ensure consistent data transmission, 'Index OUT' (Write) and 'Write Value' must have been updated safely before a PROFIBUS data cycle. Unless this can be ensured, proceed as follows: '0' in 'Index OUT' (Write), write the value to be transmitted into 'Write Value' and write the index number into 'Index OUT' (Write). With entry '0' in 'Index OUT' (Read) / 'Index OUT' (Write), no data are transmitted.

3.1 *The following status bytes are defined:*

Bem. A Unit_State

MSB							LSB
D15	D14	D13	..	..	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'
D0...D3		Always "0"		
D4		Always '0'		
D5	Dex	Changed ComRead or ComWrite data [1)]	no	yes
D6, D7		Always '0'		
D8	Err1	Transmissionerror channel 1or 9	no	yes
D9	Err2	Transmissionerror channel 2 or 10	no	yes
D10	Err3	Transmissionerror channel 3 or 11	no	yes
D11	Err4	Transmissionerror channel 4 or12	no	yes
D12	Err5	Transmissionerror channel 5 or 13	no	yes
D13	Err6	Transmissionerror channel 6 or14	bo	yes
D14	Err7	Transmissionerror channel 7 or 15	no	yes
D15	Err8	Transmissionerror channel 8 or16	no	yes

Bem. B Alarm_x

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

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

Bem. C Status_x

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/w	External/internal switch-over	external	internal
D2	w/Wanf	Start-up set-point switch-over	w	Wanf
D3	Orun	Optimization active	no	yes
D4	A/M	Automatic/manual switch-over	auto	manual
D5	Coff	Controller switched off	no	yes
D6	Y1	Switching output 1	off	on
D7	Y2	Switching output 2	off	on

1) See section 3.3 page 14 'Disabling mechanism with changes'.

Bem. D Unit_Contrl

MSB							LSB
D15	D14	D13	...	...	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status'1'
D0 ... D2		Always '0'		
D3	OStartG	Start optimization all group controllers[1)]	no start	Start
D4	OStopG	Stop self-tuning of all group controllers[1)]	no stop	Stop
D5	Dval	Data valid, acknowledgement[2)]	flank '0' r '1'	
D6 .. D15		Always '0'		

Bem. E Cntrl_x

MSB							LSB
D15	D14	D13	...	...	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	A/M	Automatic/manual switch-over	auto	manual
D1	Coff	Controller switched off	no	yes
D2	w/w2	w/w2 switch-over	w	w2
D3	We/w	External/internal switch-over	external	internal
D4	OStart	Start optimization[3)]	no start	start
D5	OStop	Stop optimization[1)]	no stop	stop
D6 .. D15		Unused, always '0'		

Bem. F Digital_Outputs

MSB							LSB
D31	D30	D29	...	...	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	Y1_15	Y1-output channel 15	off	on
D1	Y2_15	Y2-output channel 15	off	on
D2	Y1_14	Y1-output channel 14	off	on
D3	Y2_14	Y2-output channel 14	off	on
D4	Y1_13	Y1-output channel 13	off	on
D5	Y2_13	Y2-output channel 13	off	on
D6	Y1_12	Y1-output channel 12	off	on
D7	Y2_12	Y2-output channel 12	off	on
D8	Y1_11	Y1-output channel 11	off	on
D9	Y2_11	Y2-output channel 11	off	on
D10	Y1_10	Y1-output channel 10	off	on
D11	Y2_10	Y2-output channel 10	off	on
D12	Y1_9	Y1-output channel 9	off	on
D13	Y2_9	Y2 output channel 9	off	on
D14	Y1_8	Y1-output channel 8	off	on
D15	Y2_8	Y2-output channel 8	off	on
D16	Y1_7	Y1-output channel 7	off	on
D17	Y2_7	Y2-output channel 7	off	on
D18	Y1_6	Y1-output channel 6	off	on
D19	Y2_6	Y2-output channel 6	off	on
D20	Y1_5	Y1-output channel 5	off	on
D21	Y2_5	Y2-output channel 5	off	on
D22	Y1_4	Y1-output channel 4	off	on
D23	Y2_4	Y2-output channel 4	off	on
D24	Y1_3	Y1-output channel 3	off	on
D25	Y2_3	Y2-output channel 3	off	on
D26	Y1_2	Y1-output channel 2	off	on
D27	Y2_2	Y2-output channel 2	off	on
D28	Y1_1	Y1-output channel 1	off	on
D29	Y2_1	Y2-output channel 1	off	on
D30	Y1_0	Y1-output channel 0	off	on
D31	Y2_0	Y2-output channel 0	off	on

D30

1) Signals are active only at a change from 0 r 1. The signal must be available, until a change of Orun (see Status_x) has occurred.
2)
3)

3.2 *Status and diagnosis messages*

For KS 816 instrument status signalling, the external (user-specific) diagnosis must be used. The format corresponds to the instrument-related diagnosis (EN50170 volume 2 PROFIBUS)

Instrument-specific diagnosis octet 1

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'	Type
D0	Online/Conf	Online/configuration	Online	Configuration	Status
D1 .. D7		Always '0'			

Instrument-specific diagnosis octet 2

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'	Type
D0	InpF1	Input Fail channell 1	no	yes	diagnosis
D1	InpF2	Input Fail channel 2	no	yes	diagnosis
D2	InpF3	Input Fail channel 3	no	yes	diagnosis
D3	InpF4	Input Fail channel 4	no	yes	diagnosis
D4	InpF5	Input Fail channel 5	no	yes	diagnosis
D5	InpF6	Input Fail channel 6	no	yes	diagnosis
D6	InpF7	Input Fail channel 7	no	yes	diagnosis
D7	InpF8	Input Fail channel 8	no	yes	diagnosis

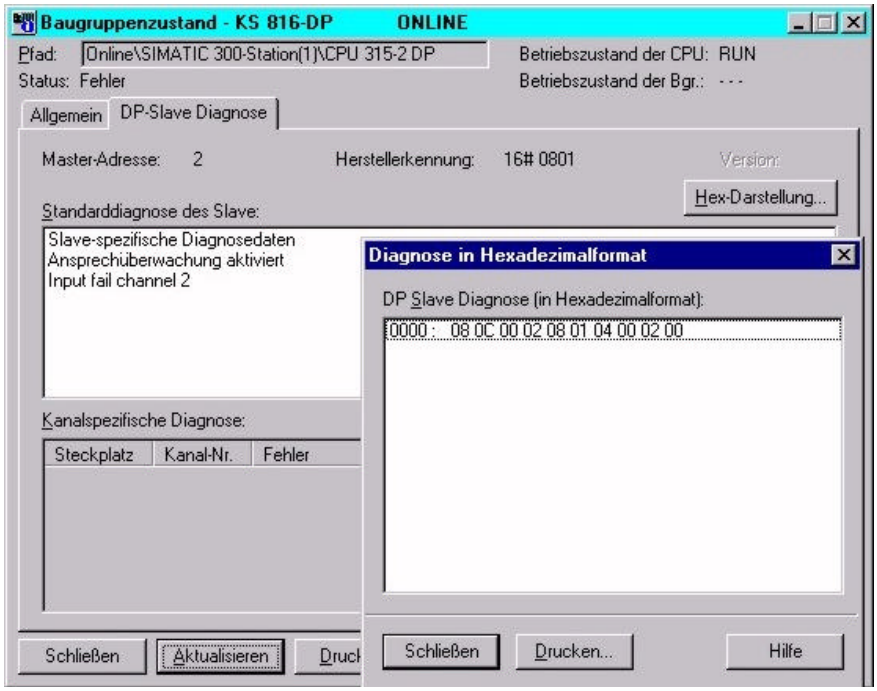
Instrument-specific diagnosis octet 3

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Bit no.	Name	Allocation	Status '0'	Status '1'	Type
D0	InpF9	Input Fail channel 9	no	yes	diagnosis
D1	InpF10	Input Fail channel 10	no	yes	diagnosis
D2	InpF11	Input Fail channel 11	no	yes	diagnosis
D3	InpF12	Input Fail channel 12	no	yes	diagnosis
D4	InpF13	Input Fail channel 13	no	yes	diagnosis
D5	InpF14	Input Fail channel 14	no	yes	diagnosis
D6	InpF15	Input Fail channel 15	no	yes	diagnosis
D7	InpF16	Input Fail channel 16	no	yes	diagnosis

Display of slave diagnosis in *STEP 7*

The following window shows the KS 816 module status and the diagnosis information in hexadecimal format.



3.3 *Disabling mechanism with changes*

When changing the reference to a datum to be transmitted during operation, e.g. via parameter channel or via the engineering interface, there is a high risk of value misinterpretation by bus master and KS 816, which shall be prevented by a disabling mechanism.

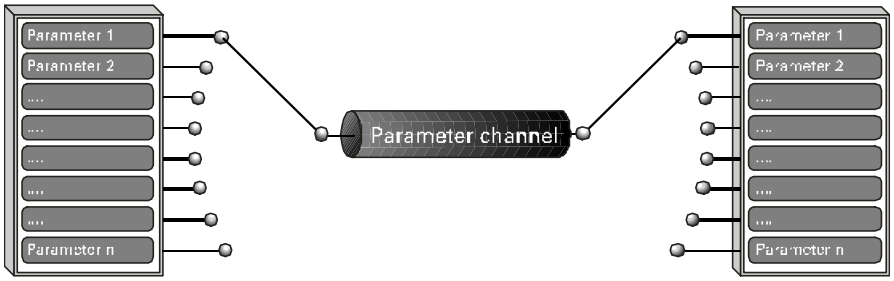
- When changing a reference, the controller module sets bit Dex = 1.
- The master must evaluate bit Dex.
- Acknowledgements and the statement that there are now valid write data also with the master are generated via a positive flank for bit Dval.
- When receiving a positive flank, the controller module sets Dex = 0 and stores the transmitted data.
- Resetting Dex is also possible by voltage switch-off and on.

3.4 *Process data transmission*

Out put data sent to KS 816 are compared with the values sent previously and processed by the controller in case of deviation. If one of the data is faulty, bit 8 in 'Unit_State' with error in channel 9, bit 9 with error in channel 2 or 10 ... or bit 15 with error in channel 8 or 16 are set, until there are no faulty accesses any more.

3.5 *Parameter transmission*

For parameter transmission, the 'Parameter channel' is available for transparent data exchange via the function block protocol, whereby all possible protocol access modes are supported (individual access, tens block and overall block). Communication to the controller is transparent, i.e. the user is responsible for monitoring the ranges, operating modes (auto/manual) etc.
The parameter channel is designed for large data quantities with low requirements on the transmission speed.



3.5.1 Message elements

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

Element	Description	Rem.
ID	Telegram mode identification	**A**
ID1	Data format of transmitted or received data	**B**
Code	Addressing code of a datum	**C**
FB no.	Function block number	**D**
Fct. no.	Function number	**E**
Type	d.c. (always '0')	

Bem. A ***ID***

This element identifies the telegram type:

ID = 0x10 = Start telegram [1]
ID = 0x68 = Data telegram
ID = 0x16 = End telegram

Bem. B ***ID1***

This element identifies the data format:

ID1 = 0 = Integer
ID1 = 1 = real value as fixpoint

Bem. C ***Code***

The code identification is decimal and the range is within '00'...'99' as well as '178' = B2 and '179' = B3.

Bem. D ***FB-no. (function block number)***

A function block is addressed with a function block number. It is within '0' and '250'. Channel addressing is also via the function block number:

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

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

A function as a partial address of a function block is also addressed with a function number, which is within '0' and '99'.

Function number range:

- 0 function general
- 1 - 99 other functions

1) 0x10 means 10 in hexadecimal

3.5.2 General communication structure

For transmission of the parameters required for the function block protocol via an 8-byte data wind ow, the ac cess is com po sed of three parts:

- Order header with specification of code, FB-no., fct. no., type and the 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 values	Num. integer values

- n data blocks with the data to be transmitted

  Data telegram structure:

  Transmission of real data as fixedpoint and of integer values

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

- an end block, provides the operationresult

  End telegram structure:

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

Re sultsignification	
0	OK
4	NAK

The read or writeoperation is always started by the master. With a number of real and integer values „ 0 , a write ser vice, ot her wise, a read ser vice is start ed.

The code determines the access type:

Code < 100, no mul tip le of 10	r	individualaccess
Code < 100, mul tip le of 10	r	tens block ac cess
Code > 100	r	block ac cess over all bloc

3.5.3 Data write sequence

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	Number real values	Number integer values

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

Datatelegrams

	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			

The re by, the first va lue is sent by = 1 . For flow con trol, count is re flec ted by KS 816 . (? once).

The values are transmitted in real-integer order.

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	

3.5.4 Data read sequence

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
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			

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

Thereby, the first value is sent by Count = 1 ,. For flow control, Count is mirrored by KS 816 (? once). The values are transmitted in the real-integer order.

End telegram:

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

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

3.6 *Examples*

3.6.1 Function block protocol principles

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

The access mechanisms are:

3.6.2 Individual access

With this access (code xx), a single value of a function can be read or written.

Valid values for ID1:

Configuration as fix point:	0 = integer	real values are transmitted as integer (without digit behind the decimal point)
	1 = real	real values are transmitted as fix point (1 digit behind the decimal point)

Example 1: (message structure with data sending)
Transmission of parameter set number (ParNr = 1) to controller (channel 2).
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	31	52	5	0	0	1
Controller replies:	0x10							

Data telegrams:

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

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

End telegram

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

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

1) if a read service was refused, these values are = 0

Example2: (message structure with data request)
Reading the error code of self-tuning heating (MSG1) from the controller (channel 2).
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	35	52	5	0	0	0
Controller replies:	0x10						0	1

Data telegrams:

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

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

End telegram:

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

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

3.6.3 Block access (tens block)

With this access (code x0), max. nine process values (always as REAL values) of a function are read.

Example: ***(message structure with data request)***
Reading the set-points (Wnvol and wvol) from the controller (channel 3).
Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0	30	53	1	0	0	0
Controller replies:	0x10						2	0

Data telegrams:

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

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

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

Controller replies:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
	0x68	2			140

End telegram:

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

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

3.6.4 Block access (overall block)

With this access, all parameters (code 178) and configuration data (code 179) can be read or written. The following conditions are applicable for this access:

- ***For writing data with code B3*** = 179', the instrument must be switched to configuration mode (r see page 24 'Op Mod'). All entered new configuration data and parameters are effective only, when the instrument 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 instrument are not used (HW and SW options), the overall message must be transmitted nevertheless. Checking the non-available data is omitted.
- The following information is valid for faulty block write accesses: a message is replied with NAK, if at least one datum is faulty. Already valid values are accepted.

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

Valid values for ID1:

Configuration as fix-point:	0, 1	transmission of real values as a Fix Point value

Example 1 (message structure with data request)
Reading the set-point parameters (**W0**, **W100**, **W2**, **Grw+**, **Grw–** and **Grw2**) from the controller (channel 7).
Start telegram:

Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	0x10	0	0xB2	57	1	0	0	0
Controller replies:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	0x10						6	0

Data telegrams:

Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7	Controller replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
	0x68	1					0x68	1			0
Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7	Control-ler replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
	0x68	2					0x68	2			700
Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7	Control-ler replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
	0x68	3					0x68	3			100
Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7	Con trol-ler replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
	0x68	4					0x68	4			-32000
Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7	Control ler replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
	0x68	5					0x68	5			-32000
Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7	Controller replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
	0x68	6					0x68	6			-32000

End telegram:

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

Example 2 : (message structure with data sending)
Writing the alarm configuration (C600, C601) to the controller (channel 1).
Start telegram:

Master sends:	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	0x10	0	0xB3	70	0	0	0	2
Controller replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	0x10						0	0

Data telegrams:

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

Master	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7	Controller replies	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
sends:	0x68	2			0110		0x68	2			

End telegrams:

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

3.7 *Data types*

Data values are classified into data types for transmission.

- FP
  Value available in the instrument as floating point number (real)
  Range: as integer (in single access) -9999 ... 0 ... 9999
  as fix point -3000,0 ... 0,0 ... 3200,0
  Exception: switch-off value '-32000'
- INT
  Positive integer value
  Range: 0 ... 32767
  Range with configuration words: 0000 ... 9999 (r page 24)
  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. bit 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.
- ***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 remaining bits are '0'
internal hex value: 0x2002, ASCII value '8194' transmitted as integer value 8194

4 Quick entrance with S7

The disk delivered with the engineering set includes the GSD file, project examples for a SIMATIC® S5 / S7, the type file and configurationexamples for COM PROFIBUS. Configuration and project can be used for easy communication build-up with a KS 816-DP.

Test environment
The following components are required for the test:

- Programming unit (recommended: PG740)
- Automation unit
  - CPU315-2 DP
- KS 816-DP
- Engineering set (order number 9407 999 09x11)
- Cable
  - PROFIBUS cable AG i KS 816-DP
  - PG i AG

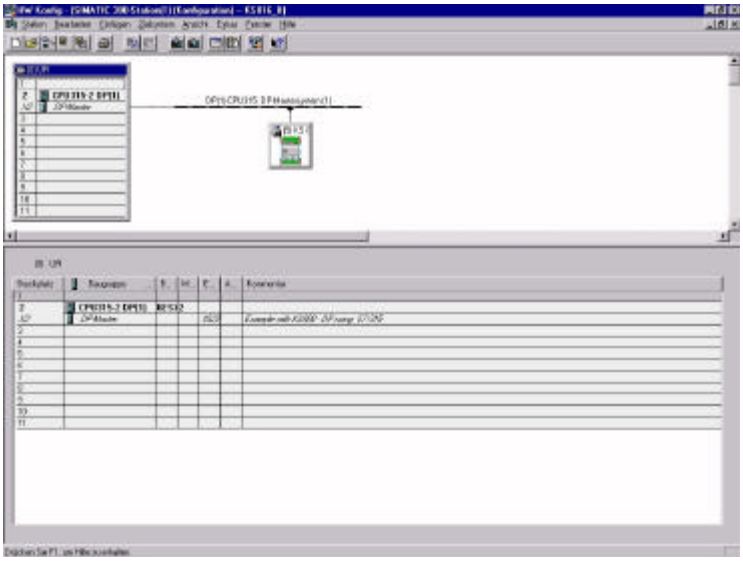
4.1 *Test environment example:*

A KS 816-DP with address 5 shall be connected to a CPU315-2 DP via PROFIBUS-DP. The process data module B is selected (16 process data channels and parameter channel). Data shall be transmitted in FixPoint format.

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

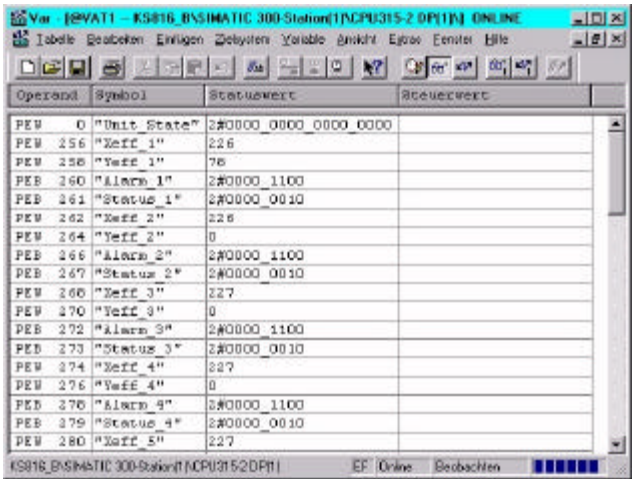
Procedure:

- Make the connections.
- Configure the units
  - Adjust address 5 at KS 816-DP (via coding switch or engineering tool) and connect the unit to the power supply. Activate the bus terminating resistors at controller connector and at PLC (S7) connector.
- PROFIBUS network configur.
  - Insert disk (engineering set) into programming unit.
  - De-archive project example (A:\KS 816DP\S7_FB \EXAMPLE \KS 816dmo.arj)
  - Open Project KS 816dmo.
  - If necessary, adapt addresses and CPU hardware configuration and transmit them to the DP master (CPU315-2 DP).
  - Switch the automation unit to Run.

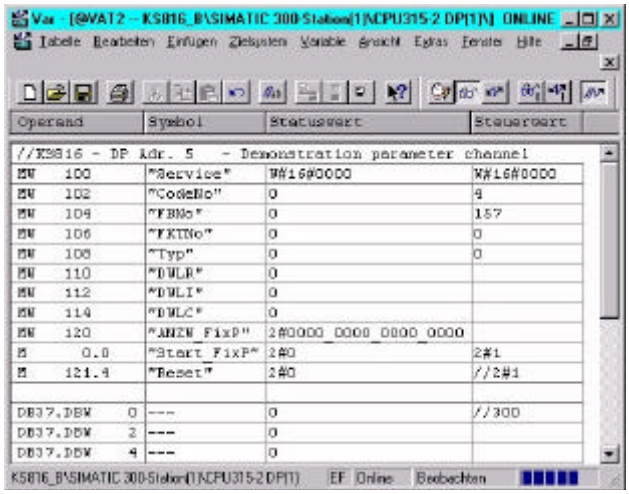


After taking the test set-up into operation, the variable tables VAT x) enclosed with the project can be used for realizing an I/O area test and for calling up the parameter channel.

VAT 1:
Shows the process data of all channels (fix point). Only channel 1 can be seen in the picture opposite.
Example channel 1:
(set-point = 30
correcting variable = 40 %
manual mode)



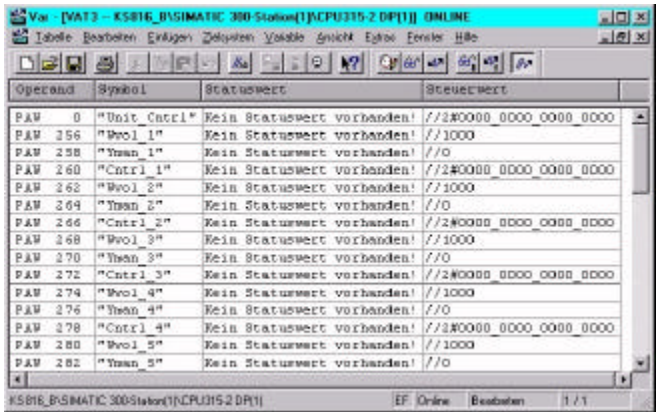
VAT 2:
This variable table can be used for access to the parameters of the function module for parameter channel mapping.



Specify e.g. when reading fix-point values:

- CodeNo, FBNo, FKTNo, Type = 0 (r section)
- Service = 0x 0001
- Start_FixP = 1
- ANZW_FixP indicates status and result after completion of function block processing.
- DWLR, DWLI, indicate the number of read values.

The picture on the bottom of the page indicates the first data of a data module for writing parameter channel data or reading values.

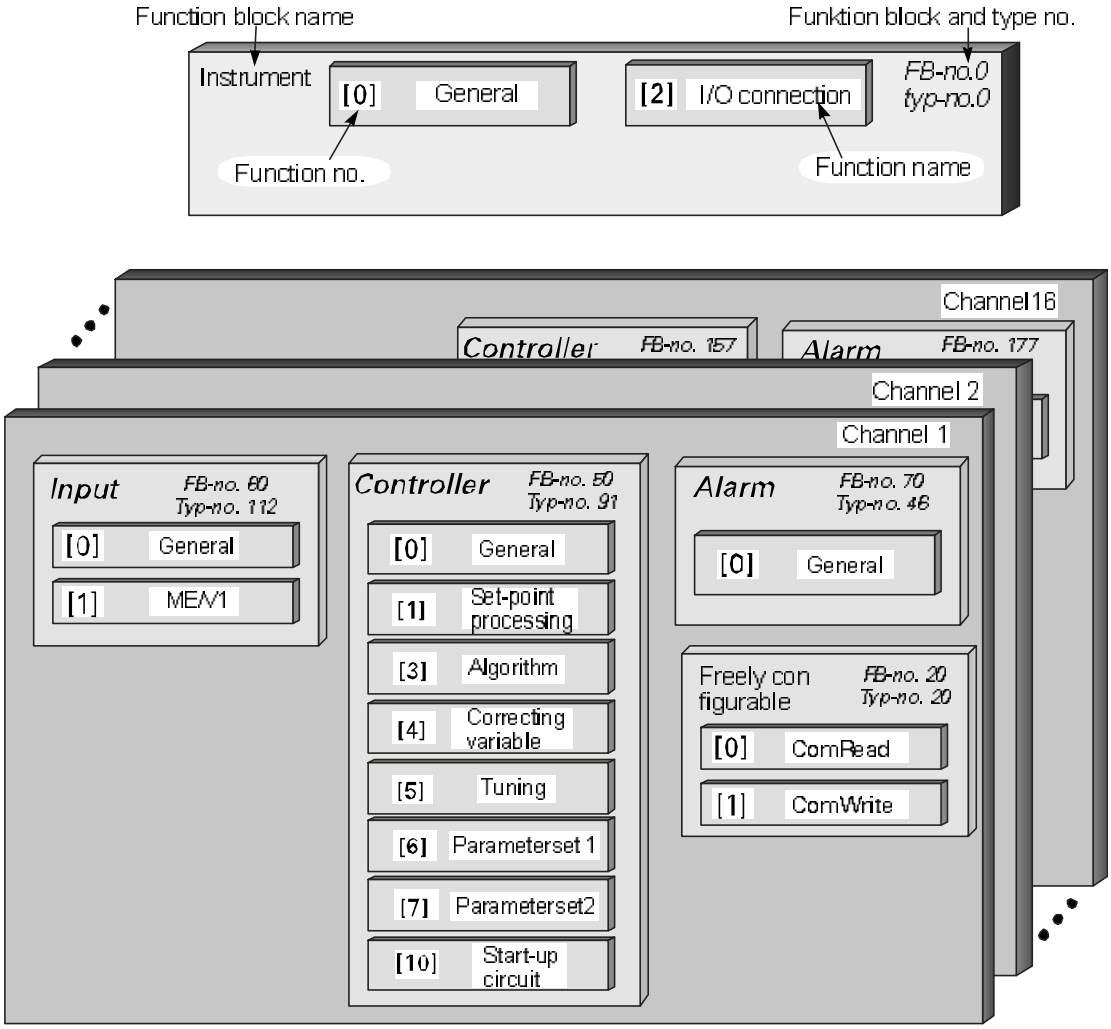


5 Function block protocol

5.1 *Data structuring*

Due to the variety of information to be processed in KS 816, logically related data and actions are grouped into function blocks. A function block has input data, output data, parameter and configuration data. They are addressed via fixed block addresses (FB no.). Each block is divided into individual functions. Functions are addressed via function numbers (fct.no.). Function number 0 addresses function block-specific data.

Fig.: 3 Survey of KS 816 function blocks and functions



5.2 *CODE tables*

5.2.1 Structure of configuration words (C.xxxx)

The configuration words given in the following code tables comprise several partial components which can be transmitted only in common.
The data in the tables must be interpreted as follows:

Example (C100):

Code	Descr.	R/W	Type	Description	Range
71	C100	R/W	INT	CFunc: controller function (T,H) WFunc:set-point function (E)	0..xx0z

Description	CFunc		WFunc
	Thousands	Hundreds	Ones
Range	x	x	z
	00 ... 07		0...1
Example: 2-pnt.contr.; set-point/ cascade	0	2	0 4

g - The possible configuration word settings are given in the KS 816 function description (order no.: 9499 040 55918)

5.2.2 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_State1	R	ST1	Status1		A
10	Block 13..15, 18	R	Block			
13	WriteError	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 during last read access	0, 100...127	
16	DPErr	R	INT	Error messages from DP module		B
17	DPAdr_eff	R	INT	Effective PROFIBUS address	0...126	
18	Type	R	INT	Type no. of function block	0	
20	Block 21...27	R	Block			
21	HWbas	R	INT	Basic HW options: module A, P		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 [1)]	R	INT	Operating version		
27	EEPVers [1)]	R	INT	EEPROM version		
31	OpMod	R/W	INT	Instrument switch-over to configuration mode (only after 1)	0	
				Instrument switch-over to on line mode (only after 0)	1	
				Cancellation of configuration mode (only after 0)	2	
32	Ostartg	R/W	INT	Stop/start self-tuning for all group controllers	0..1	
33	UPD	R/W	INT	Acknowledgement of local data change	0..1	G

1) Data are reserved for distinction of internal versions in future applications.

Bem. A ***Unit_State1***

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

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

Bem. B ***DPErr***

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		Bus access not successful	no error	error
D1		Faulty parameter setting telegram	no error	error
D2		Faulty configuration	no error	error
D3		No more data exchange	no error	error
D4...D15		Always '0'		

Bem. C ***HWbas***

COM2		0	0
T	H	Z	E

Basic version without COM2	0	0	0	0
COM2 with CA No pen	0	1	0	0
COM2 with PROFIBUS-DP	0	2	0	0
COM2 with ISO1745	0	3	0	0

Example: Value '*HWbas = 0200*' means that the addressed instrument has a COM2 interface with PROFIBUS connection.

Bem. D ***SWopt***

Version		0	0
T	H	Z	E

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

Bem. 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 15**7 239**xx.

Bem. 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 xxx**11**.

Bem. G ***UPD***

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

I/O connection

(function no.: 2)

Code	Description	R/W	Type	Description	Range	Rem.
20	Block 21...24	R	Block			
21	SnOEMOpt	R	INT	Serial number OEM field		
22	SnFabMonth	R	INT	Serial number Production month		
23	SnCntHi	R	INT	Serial number Counter High		
24	SncntLo	R	INT	Serial number Counter Low		

Parameter and configuration data

General

(function no.: 0)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	lim_wk_enable	R/W	INT	Cooling function enabling for all channels	-999,9 ... 999,9	
B3	71	C900 [1)] COM1	R/W	INT	Prot: protocol type (T) Baud: Baudrate (H,Z)	0..xyy0	1)
	72	Adr1 [1)]	R/W	INT	COM1: instrument address:	0..99	1)
	73	C904	R/W	INT	Freq: mains frequency 50/60 (T)	0..x000	
	74	C902 [1)] COM2	R/W	INT	Prot: protocol type (T) Baud: Baudrate (omitted with PROFIBUS) (H,Z)	0..wxyz	
	75	Adr2 [1)]	R/W	INT	COM2: instrument address: ISO1745 (def. 0) CAN-BUS PROFIBUS (def. 126)	0..99 0..255 0..126	

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

5.2.3 Freely configurable

(FB no.: 20 ... 27 for controllers 1 - 8; 120 ...127 for controllers 9 - 16 type no.: 20)

The data for the freely definable modules D ... F are defined via these accesses. Specification is with the ComRead block for data to be read and with the ComWrite block for data to be written.

Parameter and configuration data

ComRead (function no.: 0)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	ComReadBlock1	R/W	INT	Function block number for value 1	0 ... 177	A
	42	ComReadFctKey1	R/W	INT	Function number and code for value 1	0 ... 2999	
	43	ComReadBlock1	R/W	INT	Function block number for value 2	0 ... 177	
	44	ComReadFctKey1	R/W	INT	Function number and code for value 2	0 ... 2999	
	45	ComReadBlock1	R/W	INT	Function block number for value 3	0 ... 177	
	46	ComReadFctKey1	R/W	INT	Function number and code for value 3	0 ... 2999	
	47	ComReadBlock1	R/W	INT	Function block number for value 4	0 ... 177	
	48	ComReadFctKey1	R/W	INT	Function number and code for value 4	0 ... 2999	

ComWrite (function no.: 1)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	ComWriteBlock1	R/W	INT	Function block number for value 1	0 ... 177	B
	42	ComWriteFctKey1	R/W	INT	Function number and code for value 1	0 ... 2999	
	43	ComWriteBlock1	R/W	INT	Function block number for value 2	0 ... 177	
	44	ComWriteFctKey1	R/W	INT	Function number and code for value 2	0 ... 2999	
	45	ComWriteBlock1	R/W	INT	Function block number for value 3	0 ... 177	
	46	ComWriteFctKey1	R/W	INT	Function number and code for value 3	0 ... 2999	
	47	ComWriteBlock1	R/W	INT	Function block number for value 4	0 ... 177	
	48	ComWriteFctKey1	R/W	INT	Function number and code for value 4	0 ... 2999	

Bem. A Data structure

For definition to which data an access is made, the following entries are required:

- W Function block number r ComReadBlock or ComWriteBlock
- W Function number + single code r ComReadFctKey or ComWriteFctKey

Example:

If the wvol value of controller 2 (controller description 1 - 8) for ComRead must be selected, the values are composed as follows:

Function block number	Controller 2	= 51	ComReadBlock = 51
Function number	wvol	= 01	ComReadFctKey = 0132
Single code	wvol	= 32	

5.2.4 INPUT

(FB no.: 60 ... 67 for controllers 1 - 8; 160 ...167 for controllers 9 - 16 type no.: 112)
All data which concern acquisition and processing of all input values (analog/digital) are grouped in function block 'INPUT'. The data are available once per controller channel.

Process data

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

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

Bem. A Status byte Input_X_Fail:

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

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

Parameter and configuration data

ME/V1 — Measured value INP1 : acquisition and processing (function no.: 1)

Code		Descr	R/W	Type	Description	Range	Rem.
B2	41	$X1_{in}$	R/W	FP	Measured value correction X1 input	-999..9999	
	42	$X1_{out}$	R/W	FP	Measured value correction X1 output	-999..9999	
	43	$X2_{in}$	R/W	FP	Measured value correction X2 input	-999..9999	
	44	$X2_{out}$	R/W	FP	Measured value correction X2 output	-999..9999	
B3	71	X0	R/W	FP	Phys. value at 0%	-999..9999	
	72	X100	R/W	FP	Phys. value at 100%	-999..9999	
	73	X_{Fail}	R/W	FP	Substitute value at sensor fail	-999..9999	
	74	T_{fm}	R/W	FP	Filter time constant measured value processing	0.0 .. 999.9	
	75	T_{kref}	R/W	FP	ReferenceTC	0...60 °C / 32...140°F	
	76	C200	R/W	INT	Typ: sensor type (T,H) Unit: unit (Z)	0..xxy0	
	77	C205	R/W	INT	Fail: sensor break behaviour. (T) STk: temp. compens. source (int./ext.) (H) XKorr: enable process value correction (Z)	1..wxy0	
	78	C190	R/W	INT	Digital signal allocation: Controller off (Z) w/w2 (E)	0...00xy	

5.2.5 CONTR

(FB no.: 50 ... 57 for con trol lers 1 - 8; FB no. 150 ... 157 for con trol lers 9 - 16 type no.: 91)
All data which concern the controller are grouped in function block 'CONTR' . They are available once for each controllerchannel.

Process data

General (Function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block access(1...9)		
1	Status1	R	ST1	Status1		**A**
3	W	R	FP	Eff. set-point		
4	X	R	FP	Eff. processvalue		
5	Y	R	FP	Effectivecorrectingvariable		
6	xw	R	FP	Controldeviation		
13	Status_Alarm_x	R	INT	Status_x and alarm_x		**B**
18	Type	R	INT	Type no. of functi on block	90	
20	Block	R	Block	Block access(21...26)		
21	Xeff	R	FP	Eff. processvalue		
22	Yeff	R	FP	Effectivecorrectingvariable		
24	Unit State	R	ICMP	In put er ror chan nel 1 ... 16	r page 11	
25	Alarm x	R	ICMP	Alarm values	r page 11	
26	Status_x	R	ICMP	Statusinformation	r page 11	
30	Block	R	Block	Block access(31...38)		
33	A/M	R/W	INT	Automatic/manualswitch-over	0..1	
34	OStart	R/W	INT	Self-tuning start	0..1	
35	We/i	R/W	INT	Wext/Wint switch-over	0..1	
36	w/w2	R/W	INT	w/w2 switch-over	0..1	
38	Coff	R/W	INT	Controller off/on	0..1	
39	Cntrl x	R/W	INT	Control word	0..65	**C**

Bem. A Status1: (Code 01)

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

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

Bem. B Status_Alarm_x: (Code 13)

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

Bit-Nr.	Name	Allocation	Status '0'	Status '1'
D0	w/w2	w/w2 switch-over	w	w2
D1	We/w	External/internalswitch-over	external	internal
D2	w/Wanf	Start-up set-point switch-over	w	Wanf
D3	Orun	Self-tuningactive	nein	yes
D4	A/M	Automatic/manualswitch-over	auto	manual
D5	Coff	Controllerswitched off	no	yes
D6	Y1	Switching output 1	off	on
D7	Y2	Switching output 2	off	on
D8	Lim HH	Alarm HH	off	on
D9	Lim H	Alarm H	off	on
D10	Lim L	Alarm L	off	on
D11	Lim LL	Alarm LL	off	on
D12	Fail	Alarm Sensor Fail	no	yes
D13	'0'	Always '0'		
D14, D15	'0'	Always '0'		

Bem. C Cntrl_x: (Code 39)

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	A/M	Auto/manual	Auto	Manual
D1	Coff	Switch off controller	no	yes
D2	w/w2	w/w2 switch-over	w	w2
D3	We/w	Wext/Wint	Wext	Wint
D4	OStart	Start self-tuning [1)]	no start	start
D5	OStop	Stop self-tuning [1)]	no stop	stop
D6..D15	'0'	Always '0'		

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

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

Bem. D WState: (Code 01)

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

Bit no.	Name	Allocation	Status '0'	Status '1'
D0	w/w2	w/w2 switch-over	w	w2
D1	We/Wi	Wext/Wint	Wext	Wint
D2	w/Wanf	w/Wanfahr	w	Wanf
D3	GRW	Gradient function active	no	yes
D4	Weff_fail	Error effective set-point	no	yes
D5	'0'	Always '0'		
D6	'1'	Always '1'		
D7		Parity		

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

Code	Descr.	R/W	Type	Description	Range	Rem.
30	Block	R	Block	Block access (31, 35)		
31	dYman	R/W	FP	Different. correcting variable	-210..210	
32	Yman	R/W	FP	Absolute correcting variable	-105..105	
33	Yinc	R/W	INT	Increment. correcting variable	0, 1	
34	Ydec	R/W	INT	Decrement. correcting variable	0, 1	
35	Ygrw_ls	R/W	INT	Speed for incr./decr. correcting variable offset	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...1	
30	Block	R	Block	Block access (31...39)		
31	ParNr	R/W	INT	Parameter set number active	0 .. 1	
32	Tu1	R	FP	Delay time heating	0...9999 s	
33	Vmax1	R	FP	Rate of increase heating	0,000...9,999 %/s	
34	Kp1	R	FP	Process gain heating	0,000...9,999	
35	MSG1	R	INT	Error code of self-tuning heating	0...8	
36	Tu2	R	FP	Delay time cooling	0...9999 s	
37	Vmax2	R	FP	Rate of increase cooling	0,000...9,999 %/s	
38	Kp2	R	FP	Process gain cooling	0,000...9,999	
39	MSG2	R	INT	Error code of self-tuning cooling	0...8	

1) Signals are only active with transition 0 r 1. The signal must be available, until a change from Orun (see Status_Alarm_x) has occurred.

Bem. E Status 1 Tuning 'State_Tune1'

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

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

Parameter and configuration data

General

(function no: 0)

Code		Descr.	R/W	Type	Description		Range	Rem.
B3	71	C100	R/W	INT	CFunc: controller function CType: controller type WFunc: set-point function	(T,H) (Z) (E)	0..xxyz	
	72	C101	R/W	INT	CMode: controller output action CDiff: x/x-w differentiation CFail: behaviour with sensor fail CAnf: start-up circuit	(T) (H) (Z) (E)	0..wxyz	
	73	C700	R/W	INT	OMode: self-tuning mode OCond: process at rest OGrp: allocation group self-tuning OCntr: controlled adaptation mode	(T) (H) (Z) (E)	0..wxyz	
	74	C180	R/W	INT	SWext: source for Wext	(T)	0..x000	

Set-point

Set-point processing (function no.: 1)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	W0	R/W	FP	Min. set-point limit for Weff	-999..9999	
	42	W100	R/W	FP	Max. set-point limit for Weff	-999..9999	
	43	w2	R/W	FP	Additional set-point	-999..9999	
	44	Grw+	R/W	FP	Set-point gradient plus	>0..9.999	
	45	Grw-	R/W	FP	Set-point gradient minus	>0..9.999	
	46	Grw2	R/W	FP	Set-point gradient w2	>0..9.999	

Algo

Control algorithm (function no.: 3)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	Xsh	R/W	FP	Neutral zone	0.2 .. 20,0 %	
	42	Tpuls	R/W	FP	Min. pulse length	0.1..2,0 s	
	43	Tm	R/W	FP	Actuator travel time	10..300 s	
	44	Xsd1	R/W	FP	Signaller switching difference	0,1..9999 %	
	45	LW	R/W	FP	Trigger point separation add. contact	-999..9999	
	46	Xsd2	R/W	FP	Switching difference add. contact	0,1..9999 %	
	47	Xsh1	R/W	FP	Neutral zone	0.0 .. 999.9%	
	48	Xsh2	R/W	FP	Neutral zone	0.0 .. 999.9 %	

Corr. variable

Correcting variable processing(function no.: 4)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	Y_{min}	R/W	FP	Min. correcting variable limiting	-105..105 %	
	42	Y_{max}	R/W	FP	Max. correcting variable limiting	-105..105 %	

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

	43	Y0	R/W	FP	Working point for correcting variable	-105..105 %	
	44	Yh	R/W	FP	Max. mean value of correcting variable	5..100%	
	45	LYh	R/W	FP	Limit for mean value formation	0,1 .. 10,0	

Tuning

Self-tuning (function no.: 5)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	YOptm	R/W	FP	Correcting variable during process at rest	-105..105	
	42	dYopt	R/W	FP	Step height during identification	5..100	
	43	OXsd	R/W	FP	Hysteresis with parameter switch-over	0.0..9999	
	44	Trig1	R/W	FP	Trigger point 1	0.0..9999	
	45	POpt	R/S	INT	Parameter set to be optimized	0...1	

Parameter set x

Control parameter set 1 / 2 (function no.: 6,7)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	Xp1	R/W	FP	Proportional band 1	0.1..999.9	
	42	Tn1	R/W	FP	Integral time 1	0..9999	
	43	Tv1	R/W	FP	Derivative time 1	0..9999	
	44	T1	R/W	FP	Min. cycle time 1	0.4..999.9	
	45	Xp2	R/W	FP	Proportional band 2	0.1..999.9	
	46	Tn2	R/W	FP	Integral time 2	0..9999	
	47	Tv2	R/W	FP	Derivative time 2	0..9999	
	48	T2	R/W	FP	Min. cycle time 2	0.4..999.9	

Start-up circuit

(function no.: 10)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	Ya	R/W	FP	Max. correcting value	5 .. 100 %	
	42	Wa	R/W	FP	Start-up set-point	-999 .. 9999	
	43	TPa	R/W	FP	Start-up holding time	0 .. 9999 min	

5.2.6 ALARM

(FB no.: 70 ... 777 for con trol lers 1 - 8; FB no. 170 ... 177 for con trol lers 9 - 16 type no.: 91)
Function block 'ALARM' defines the overall alarm processing of the relevantcontroller. The data are available once per controller channel.

Process data

General (function no.: 0)

Code	Descr.	R/W	Type	Description	Range	Rem.
00	Block	R	Block	Block ac cess (1 .. 3)		
1	Status_AI1	R	ST1	Alarm sta tus 1		**A**
18	Type	R	INT	Type no.of functi on block	46	

Bem. A ***Status_AI1***

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

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

Parameter and configuration data

General (function no.: 0)

Code		Descr.	R/W	Type	Description	Range	Rem.
B2	41	LimL	R/W	FP	Low alarm	-999..9999	1)
	42	LimH	R/W	FP	High alarm	-999..9999	
	43	xsd1	R/W	FP	High/low alarm swit ching difference	0..9999	
	44	LimLL	R/W	FP	Low low alarm	-999..9999	1)
	45	LimHH	R/W	FP	High high alarm	-999..9999	

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

6 Function module for SIMATIC® S7

The handling principle of S7-FB corresponds to the S5 version. The FB must be called up conditionally when starting an order and as long as the order is active.
Dependent of S7-CPU and DP master, there are results in the I/O handling. With a CPU315-2 DP with on-board DP interface, SFC modules 14 and 15 must be used for consistent data transfer. 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 DP-SEND and DP-RECEIVE FBs must be called up at cycle start and end.
The FB has an instance DB, which must be specified when calling up the FB.

6.1 *Structure*

The call-up parameters of the function module are:

Name	Type	Description / function
A-Anfang	Pointer	Start of address area of output words (e.g. address data area 'RECORD' of SFC 15, Ax, y when using an external CP). When specifying a data word, the DB number must also be transmitted (e.g. DB4.DBX0.0)
E-Anfang	Pointer	Start of address area of input words (e.g. address data area 'RECORD' of SFC 15, Ex, y when using an external CP). When specifying a data word, the DB number 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, whereby **no** offset needs being taken into account. The data are interpreted as parameter data by the specified address. Specification of the DB must be in the following form e.g. DB6.DBX10.0
Service	WORD	Service (Read/Write)
Code_nr	WORD	Code
FB_nr	WORD	Function block no. (channel addressing)
FKT_nr	WORD	Function no.
Typ	WORD	No function (always '0')
Timeout	DWORD	Time out value, decremented at each call-up. With value = 1, the order is cancelled with error message 'timeout'.
DWLR	WORD	Length of real values
DWLI	WORD	Length of integer value
DWLC	WORD	d.c. always '0'
ANZW	W	The current status of transmission for the selected data area is given in the display word. The display word structure is:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timeout (F3)	Service faulty				NAK (access by controller not accepted)	Parity error	Timeout internal (controller)		Wait for end telegram	Service (0=Read; 1=Write)	Reset order	Wait for acknowledgement	Order finished with error	Order finished without error	Order running

The function module reads or writes KS 816 parameter/configuration data.

- A-Anfang, E-Anfang
  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
  (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 taken when writing.

- Service
  This parameter determines the access type (write / read) cf. ID1.

Write access:	F0	=	Integer	Read access:	0	=	Integer
	F1	=	Real		1	=	Real

Single access
With this access (code xx), a single value of a function can be read or written.

Valid values for ID1:

Configuration as Fix-point:	0	Real values are transmitted as integer values (without digits behind the decimal point)
	1	Real values are transmitted as Fix Point (1 digit behind the decimal point).

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

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

- For writing data with 'code B3 = 179', the instrument must have been switched to configuration mode previously (cf. see page 24 'OpMod'). All newly entered configuration data and parameters are valid only, when the instrument was switched back to online mode.
- All data of a message must be defined. Omissions are not permissible.
- If parts of a message in the instrument are not in use (HW and SW options), the complete message must be transmitted nevertheless. Checking the non-existing data is mitted.
- With faulty block write accesses, a message is replied with NAK, if at least one datum is faulty. Already valid values are accepted.

The order of data to be transmitted is given in the relevant code table.

Valid values for ID1:

Configuration as Fix-Point:	0, 1	Transmission of real values as Fix Point value

- Code_nr
  The code identification is decimal and the range includes '00'...'99', '178' = B2 and '179' = B3.

- FB_nr. (function block number)
  A function block is addressed with a function block number, which is within '0' and '250'.
  Function block number rqnges:
  0 general data for the overall instrument
  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 '0' and '99'.
  Function number ranges:
  0 Function general
  1 - 99 other functions

- Type (function type)
  A function type number is allocated to each function block. The number is withind ‘0’ and ‘111’.
  Function type ranges:
  0 function type general
  1 - 111 other function types
- Timeout
  Timeout counter: range 0x0000 ß TIME ß 0x7FFF
  - is decremented at each PLC cycle (max. 32767)
  - timeout at 0.
  If the CPU is too fast, call up FB206/FB207 via timer module.
- DWLR (Real), DWLI (Integer)
  After a read access, these parameters include the relevant number of received data. With a write access, the relevant number of data to be transmitted is specified. DWLC is not required in KS 816.The value must be set to 0.
- ANZW
  The actual status of transmission is mapped by this display word. Bit 4 can be used as an input for resetting FB 206 / FB 207. Selection of the controller channel is via the FB_nr.

7 Annex

7.1 *Legend of terms*

COM PROFIBUS	Configurationtool (formerly COM ET200)of the Siemenscompany for PROFIBUS
FB	Abbreviation for function block
Fct	Abbreviation for function
ET	Abbreviation for engineering tool
Function	A self-contained partial function of a function block seen from the interface
Function block	Self-contained processing unit
GSD file	Devicedatabasefile
HW	Abbreviation for hardware
ISO1745	Standard communication protocol ISO 1745, ASCII-based
PC interface	Front-panel controller interface for connection of an engineering tool
PCI	Process Control Instrument
PCI protocol	Protocol based on ISO 1745, implemented for PMA controllers
PNO	**P**ROFIBUS user organization
PROFIBUS-DP	Standard communication protocol to EN50170 vol.2 (DP: decentral periphery)
RS422	Standard 4-wire connection, full duplex, (EIA RS 422); in this case: separate send/receive channels with max.32 units
RS485	Standard 2-wire connection, half duplex, (EIA RS 485)
S5 / S7	Siemens AG PLC series
Serialinterface	Bussable rear panel controllerinterface
SW	Abbreviation for software
Type file	Configuration file for COM ET200

7.2 *GSD file*

```
;=============================================================
; Device Database File for product  K S   8 1 6 - D P
; Copyright (C) PMA Prozeß-und Maschinen-Automation GmbH 2001
; D-34123 Kassel, Miramstr. 87, Tel. +49 (0) 561/ 505 -1307
; Release : V1.0
; File:     PMA_0801.gsd
;=============================================================
#Profibus_DP
GSD_Revision = 1
Vendor_Name = “PMA GmbH”
Model_Name = “KS 816-DP”
Revision = “V 1.1 "
Ident_Number = 0x0801
Protocol_Ident = 0      ; DP
Station_Type = 0        ; Slave
FMS_supp = 0
Hardware_Release = “HV 01.00"
Software_Release = “SV 01.00"
;supported baud rates:
9.6_supp = 1
19.2_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_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 = 2   ; TTL
24V_Pins = 0          ; not available
Bitmap_Device = “PMA0816N”
Bitmap_Diag = “PMA0816D”
Bitmap_SF = “PMA0816F”
;
;—DP-Slave related key words——-
;
Freeze_Mode_supp = 1    ; supported
Sync_Mode_supp = 1      ; supported
Auto_Baud_supp = 1
Set_Slave_Add_supp = 0
User_Prm_Data_Len = 0   ; no user prm data
;minimum slave poll cycle (Basis 100us):
Min_Slave_Intervall = 1
Modular_Station = 1     ; modular device
Max_Module = 0x01       ; max. number of modules
Max_Input_Len = 116
Max_Output_Len = 116
Max_Data_Len = 232
; Module description
; 1. measuring values for 16 channels + parameter channel
Module = “A:Measured data(16)+ parameter” \
        0x50,0x10,0x50,0x10,0x50,0x10,0x50,0x10,\
        0x50,0x10,0x50,0x10,0x50,0x10,0x50,0x10,\
        0x50,0x10,0x50,0x10,0x50,0x10,0x50,0x10,\
        0x50,0x10,0x50,0x10,0x50,0x10,0x50,0x10,\
        0xF3
EndModule
;
; 2. Controller values + parameter channel
Module = “B:Process data(16)+parameter” \
        0x11,\
        0x52, 0x52, 0x52, 0x52, 0x52, 0x52, 0x52, 0x52,\
        0x52, 0x52, 0x52, 0x52, 0x52, 0x52, 0x52, 0x52,\
        0x21,\
        0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62,\
        0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62, 0x62,\
        0xF3
EndModule
;
; 3. Only parameter channel
Module = “C: Parameter” 0xF3
EndModule
;
; 4. Process data for 52 Variable data + parameter channel
Module = “D: 52 Variable data + parameter” \
        0x15,\
        0x57, 0x57, 0x57, 0x57, 0x57, 0x57, 0x51,\
        0x21,\
        0x67, 0x67, 0x67, 0x67, 0x67, 0x67, 0x61,\
        0xF3
```

```
EndModule
;
; 5. Process data for 40 Variable data + parameter channel
Module = “E: 40 Variable data + parameter” \
        0x15,\
        0x57, 0x57, 0x57, 0x57, 0x57,\
        0x21,\
        0x67, 0x67, 0x67, 0x67, 0x67,\
        0xF3
EndModule
;
; 6. Multipexing of Process data for 1 Variable data + parameter channel
Module = “F: Multipexed data + parameter” \
         0x15,\
         0x50, 0x50,\
         0x21,\
         0x60, 0x60,\
         0xF3
EndModule
;
; Device related diagnostic data
Unit_Diag_Bit(0) = “Configuration state”
Unit_Diag_Bit(8) = “Input fail channel 1"
Unit_Diag_Bit(9) = “Input fail channel 2"
Unit_Diag_Bit(10) = “Input fail channel 3"
Unit_Diag_Bit(11) = “Input fail channel 4"
Unit_Diag_Bit(12) = “Input fail channel 5"
Unit_Diag_Bit(13) = “Input fail channel 6"
Unit_Diag_Bit(14) = “Input fail channel 7"
Unit_Diag_Bit(15) = “Input fail channel 8"
Unit_Diag_Bit(16) = “Input fail channel 9"
Unit_Diag_Bit(17) = “Input fail channel 10"
Unit_Diag_Bit(18) = “Input fail channel 11"
Unit_Diag_Bit(19) = “Input fail channel 12"
Unit_Diag_Bit(20) = “Input fail channel 13"
Unit_Diag_Bit(21) = “Input fail channel 14"
Unit_Diag_Bit(22) = “Input fail channel 15"
Unit_Diag_Bit(23) = “Input fail channel 16"
;
Slave_Family = 5
Max_Diag_Data_Len = 10
Fail_safe = 0
```