PMA Prozeß- und Maschinen-Automation GmbH

Modular Controller System
KS vario

Interface Description
PROFIBUS - DP
9499 040 69718

valid from: 7/2004

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

Content

1 General

Various fieldbus interfaces can be connected to Modular Controller System KS VARIO. For this, the relevant bus coupler is used as a central station for the controller system.

One of these bus couplers serves to support the Modbus protocol via a front-panel RS485/422 interface, which permits transmission of all process, parameter and configuration data.
The serial communication interface can be used for communication with supervisory systems, visualization tools, etc.

Another interface, which is always provided as standard, is on the KS VARIO controller modules. This full RS232 interface is used for connection of the 'BlueControl' tool, which runs on a PC. Communication is according to the master/slave principle. KS VARIO is always slave.

Characteristic data of the cable medium and physical and electrical interface properties are :

Network topology

Linear bus with active bus termination at both ends. With transfer rates ? 1,5 Mbit/s, tap lines must be avoided by all means.

Transfer medium

Screened, twisted 2-wire cable. The bus cable properties are specified in standard EN 50 170 . All transfer rates up to 12 Mbits/s can be used with type A cable. In addition to the standard cable, underground, overhead suspension and trailing cables are also available.
The recommended cable parameters are:

	Type A cable
Characteristic impedance in Ω	135 ... 165 at 3 ... 20 MHz
Operating capacity (pF/m)	<30
Loop resistance[Ω/km)	< 110
Core diameter (mm)	> 0,64
Core cross section (mm2)	> 0,34

Cable lengths

The max. cable length is dependent of transfer rate. The Baudrate is determined by the master configuration and detected automatically by KS VARIO. The cable length can be extended by means of repeaters.

Baudrates

Automatic Baudrate detection

Baudrate	**Maximum cable length**
9,6 / 19,2 / 31,25 / 45,45 / 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	100m

Interface

RS485 ; can be mounted at the site

Addressing 0 ... 126 (factory setting: 01)

Adjustment is via the 2 bus coupler rotary switches or via the BlueControl engineering tool. With address setting via BlueControl, the 2 rotary switches for address setting must be in position "00".

In position OFF, the bus function is switched off (operation as if no interface was provided, including bus error message switch-off). When changing from OFF to a valid address, the DP system is reset (DP processor system reset).

Number of stations 32 instruments in a segment. Can be extended to 126 by means of repeaters.

Diagnosis and monitoring Fast process value transmission

Display of bus error
- Error message
- LED

Particularities Configurable process data modules

Direct input and output reading and writing
Input forcing
Back-up controller function
Easy connection even to small control systems

1.1 Scope of delivery

The engineering set comprises:
Floppy

Pma_1030.gsd GSD file
Pma_sup.arj Step7® function module as S7 libr.
KSvario.zip project example in Step7® for S7-300

Dev_KSVario-1.bct Configuration example for BlueControl®

Interface description for PROFIBUS-DP – process data and parameter data

- Document "KS VARIO PROFIBUS-DP" (9499-040-69718) provides basic explanations for connection of KS VARIO to PROFIBUS-DP networks. It contains hints for cyclic data exchange.
- Moreover, additional possibilities for transfer of process values, parameters and configuration data via the parameter channel and via multiplex operation are described.

GSD file The GSD file is available as a standard file with English text (PMA_1030.gsd) and with German texts (PMA_1030.gsg). For working with your PROFIBUS master configuration using German texts, please, install file PMA_1030.gsg.

2 Hints for operation

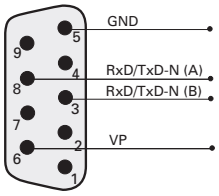
2.1 Interface connection

The PROFIBUS is connected to the front-panel sub-D interface of the bus coupler.
The physical signals are based on RS485 interfaces.

Suitable cabling must be provided by the user, whereby the general cable specifications to EN 50170 vol.2 must be followed.
Cable laying
When installing the cables, the general hints for cable laying provided by the master module supplier must be taken into account:
Cable laying inside buildings (inside and outside cabinets)
Cable laying outside buildings
Potential compensation
Cable screening
Measures against interference voltage
Tap line length
Bus termination resistors are not contained in the VARIO bus coupler, but must be realized via the external connector, if necessary.
Special hints for PROFIBUS cable laying are given in technical guideline "Mounting guidelines for PROFIBUS-DP/FMS (order no. 2.111 [Germ.]; 2.112 [Engl.] of the PROFIBUS user organization.

Sub-D pin allocation

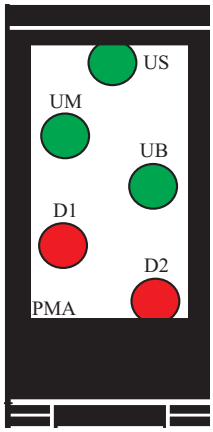
Connection is via a 9-pole sub-D socket.



2.2 Signification of indicator LEDs

LEDs

LED no.	LED colour	Function
US	green	Segment voltage U_provided
UM	green	Module voltage U_provided
UB	green	Coupler voltage U_provided
D1	red	Bus coupler-to-PLC connection: - ok = LED off - configuration error = LED blinks - no connection = LED on
D2	red	Bus coupler-to-controller connection: - ok = LED off - error = LED blinks



2.3 Forcing

Inputs

All physical inputs can be overwritten via the PROFIBUS-DP (configurable). Thus e.g. process value measurement via remote I/O (e.g. VARIO I/O system) and entry via the bus are possible.

Outputs

With output forcing, the fail-safe function setting must be taken into account. If "zero" fail-safe behaviour was adjusted, all outputs are set to zero in case of bus error or master stop, otherwise, their old value remains unchanged.

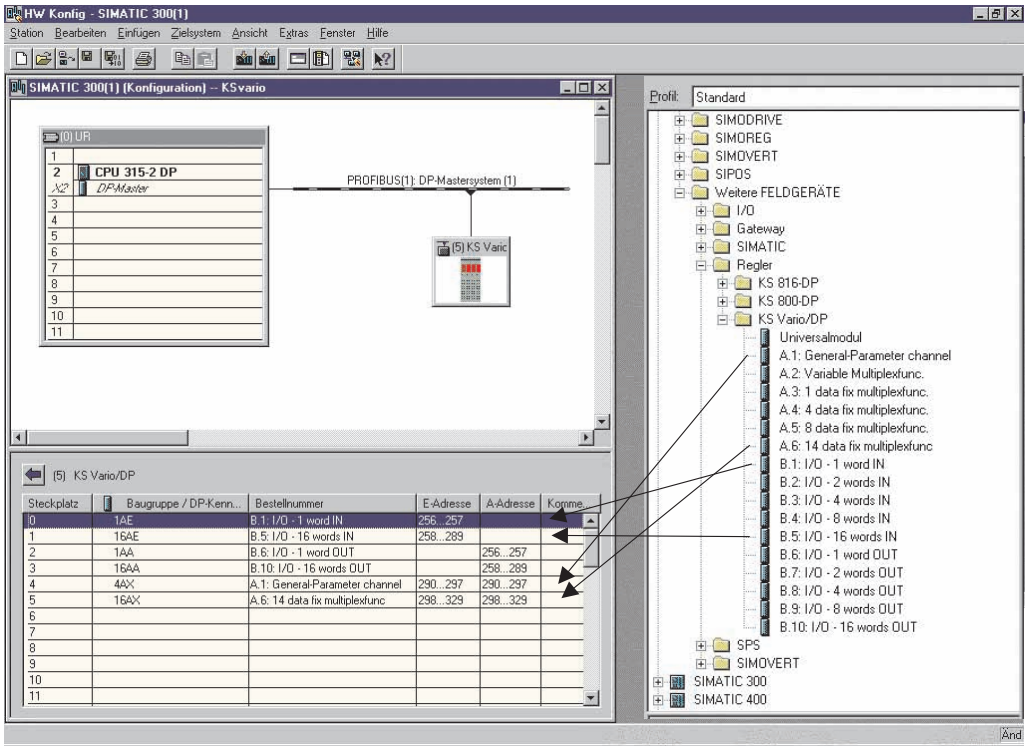
2.4 Fail-safe

User parameter setting 'fail-safe' determines the instrument behaviour in case of master bus failure or 'bus stop' .
Bus failure
In case of bus failure, the instrument operates according to the following rules.

Fail-safe	Reaction in case of bus failure or master stop
Last value	Continue operation with the values sent last
	Forced analog inputs are set to FAIL
zero	Forced analog inputs are set to FAIL 1) .
	Forced digital inputs are set to zero
	Forced outputs are set to zero

3 Process data

For flexible realization of his requirements on transfer values, memory space and transfer time, the user can compose the process data transmission of a defined selection of modules. Configuration is via the relevant bus master configuration tool.



Process data and selected parameter data are written and read cyclically.
User input values are stored by the instrument only in case of value changing.

Data format Values as e.g. process values and set-points are transmitted in 16-bit FixPoint format with one digit behind the decimal point.

Parameter channel All process, parameter and configuration data are accessible additionally via the parameter channel. These data are transmittewd over several cycles on request. FloatFormat accesses are also possible. These accesses are described below in this documentation.

3.1 Configurable process data modules

For flexible realization of his requirements on transfer values, memory space and transmission time, the user can compose the process data transmission from a defined module selection. Configuration is via the relevant bus configuration tool of the bus master. Selection which process data and parameters are to be transmitted is via the BlueControl engineering tool.

Available modules

The following modules are offered:

No.	Module	Description	Numb. bytes	Type	Identification [hex]	Identification [dec]
1	A.1	General: Parameter –Channel	8/8		0xF3	243
2	A.2	Variable multiplexer module	4/4		0x71	113
3	A.3	Multiplexer module for 1 data	6/6	Integer / Fix-Point 1	OxF2	242
4	A.4	Multiplexer module for 4 data	12/12	Integer / Fix-Point 1	0xF5	245
5	A.5	Multiplexer module for 8 data	20/20	Integer / Fix-Point 1	0xF9	249
6	A.6	Multiplexer module for 14 data	32/32	Integer / Fix-Point 1	0xFF	255
7	B.1	I/O: 1 word input	2	Integer / Fix-Point 1	0x50	80
8	B.2	I/O: 2 words input	4	Integer / Fix-Point 1	0x51	81
9	B.3	I/O: 4 words input	8	Integer / Fix-Point 1	0x53	83
10	B.4	I/O: 8 words input	16	Integer / Fix-Point 1	0x57	87
11	B.5	I/O: 16 words input	32	Integer / Fix-Point 1	0x5F	95
12	B.6	I/O: 1 word output	2	Integer / Fix-Point 1	0x60	96
13	B.7	I/O: 2 words output	4	Integer / Fix-Point 1	0x61	97
14	B.8	I/O: 4 words output	8	Integer / Fix-Point 1	0x63	99
14	B.9	I/O: 8 words output	16	Integer / Fix-Point 1	0x67	103
16	B.10	I/O: 16 words output	32	Integer / Fix-Point 1	0x6F	111

Min. one module must be selected.

The following modules are available to the user:

- n spacekeepers for 1 word, 2, 4, 8, 16 words (word-consistent) for input and output (1, 2, 4, 8, 16 values) → module B
- one variable multiplexer module with a variable number of transmitted data. 4 input/output bytes → module A.2
- Four multiplexer module with a fix number of transmitted data → module A.3 … A.6
- one parameter channel module with 8 input/output bytes → module A.1
- max process data input length: 244 bytes
- max.process data output length: 244 bytes
- max. number of modules: 50

Unless a “variable multiplexer module” is defined, or if I/O modules are defined prior to a "variable multiplexer module" the I/O module (3 - 12) structure corresponds to the structure of the bus coupler data cache.

The following restrictions for module definition are applicable:

Modules “Parameter channel” and “variable multiplexer module” can be inserted only once.
Module “Parameter channel” cannot be used behind module “variable multiplexer module”.
Behind module “variable multiplexer module” , min. one I/O module (3 – 12) for transmission of actual process data or parameters must be defined.

Module B

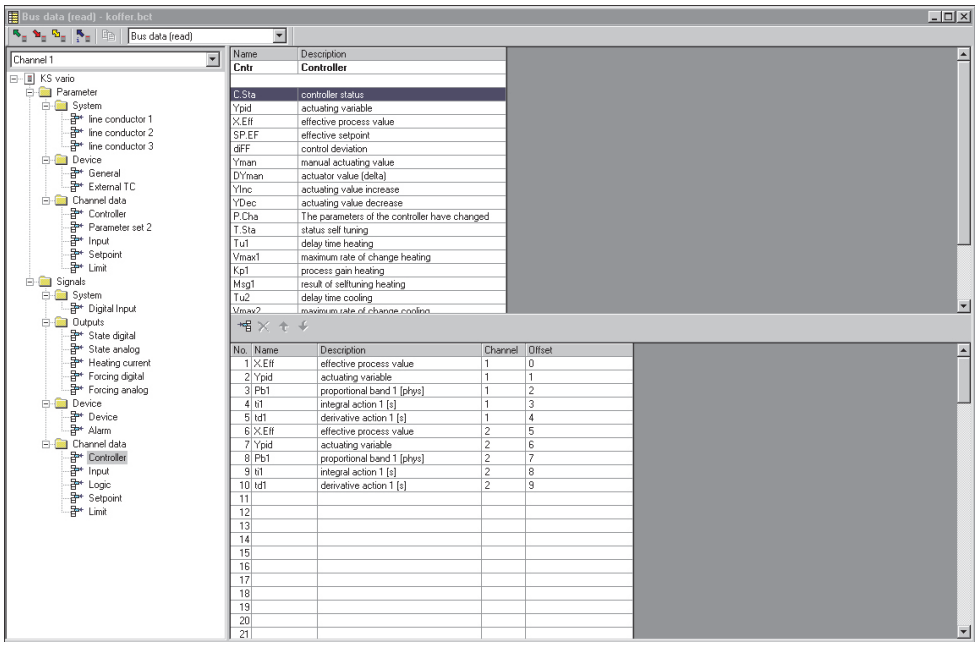
The BlueControl engineering tool is used to select the process data and parameters to be transmitted for reading and writing (see following chapter). Modules B can be selected up to the memory space limit or to the number of permitted modules.

- max. process data input length: 115 bytes
- max. process data output length: 115 bytes
- max. number of modules: 57

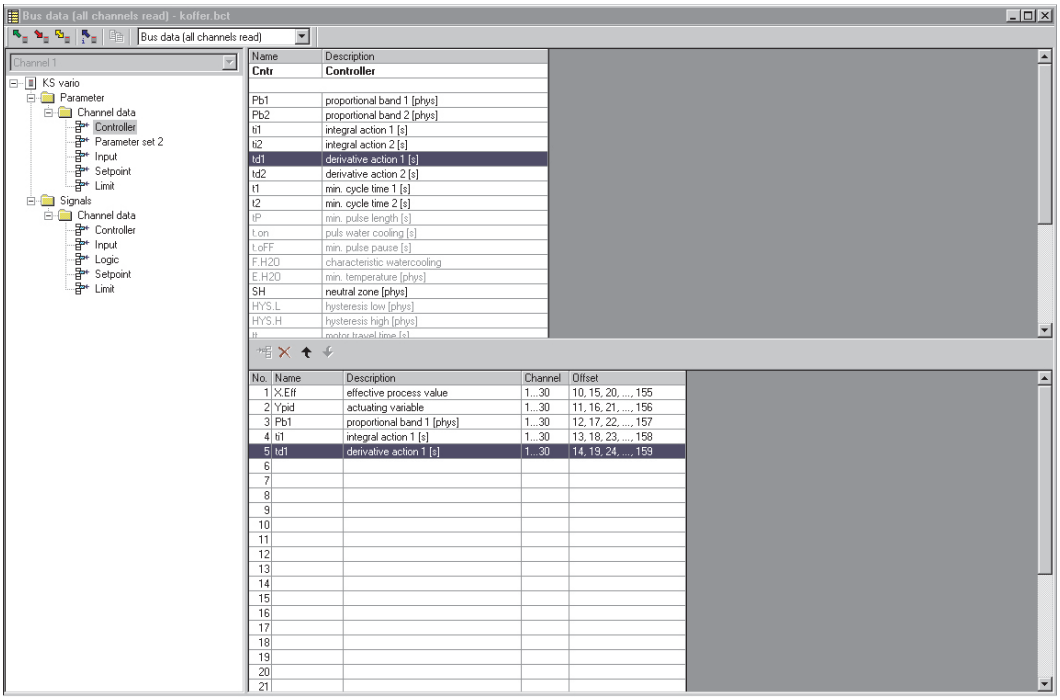
3.2 Definition of values to be transmitted in the "BlueControl" engineering tool

BlueControl offers 2 selection modes for the data to be transmitted:

- Max.120 parameters and process data from any channels for writing and max. 120 for reading. Data exchange to the Profibus master is by means of the process data modules listed above (I/O modules no. 3 – 12). The order of transmission is determined by the module position.



- Additionally or alternatively, up to 32 parameter and process data of any type can be selected in common for all channels. E.g. the process values of all channels (max. 30) can be transmitted by selecting one datum. In total, up to 960 write and 960 read data can be defined (32 data x 30 channels).



These selected data (max. 1080 write and 1080 read data) are available in the bus coupler as cache memories in the order defined in BlueControl.

The indexes or offsets of the data are displayed or can be printed out via the BlueControl tool.

3.3 Structure of the data cache memory in the bus coupler

Index Read cache **Content**

0 **up to 120**	any data of any channels
from 121 **up to 1080**	selected data (identical for all channels): all data channel 1 all data channel 2 ... all data channel 30

Index Write cache **Content**

0 **up to 120**	any data of any channels
from 121 **up to 1080**	selected data (identical for all channels): all data channel 1 all data channel 2 ... all data channel 30

Read and write access to the cache memory data in the bus coupler is possible also via parameter channel and DPV-1 message.
Data which are written permanently or via the “multiplexer module” of the process data channel cannot be changed using parameter channel and DPV-1 message, because they are overwritten during the following data exchange on the Profibus. Problems of this two-point operation must be taken into account by the user.

3.4 Data exchange via variable multiplexer module

Module A.2

In KS Vario, max. 1080 process data can be defined for read and write access. As this large number of data cannot be provided on one Profibus process area, the variable multiplexer module permits multiplexing in the I/O modules (3 - 12) . For this, 4-byte index information is transmitted in both directions.

Application example:

In the bus configuration, 2 16-word I/O modules (e.g. I/O module 12) are specified behind the variable multiplexer module.
In BlueControl, e.g. 32 process data/parameters are selected for each channel. The data per channel are arranged successively in the bus coupler cache memory. Via the variable multiplex module, e.g. all 32 data of a channel per access can be transmitted. Subsequently, the data of the next channel can be transmitted by means of a new entry in IndexOUT.

Module eA.2

The variable multiplexer module data structure is:

Data direction	Word	Value range	Description
IndexOUT	Index_Read	0 – 1080	Index of cache memory of the 1st datum of the data block requested by the PLC
	Index_Write	0 – 1080	Index of cache memory of the 1st datum of the data block defined by the PLC
IndexIN	Index_Read	0 – 1080	Index of the cache memory of the 1st datum of the data block provided by the bus coupler
	Index_Write	0 – 1080	Index of the cache memory of the 1st datum of the data block taken over by the bus coupler

IndexOUT

The data contain the information from which cache memory index master data shall be sent, or bus coupler cache memory data shall be requested/read.

IndexIN

The data contain the mirrored information which bus coupler data are provided to the master, or which data were taken over by the bus coupler.

3.4.1 Data transfer from the Profibus master to the bus coupler

Write

For data transfer from the master to the bus coupler, the master enters the data into the data area (I/O modules 3 - 12) behind the variable multiplexer module and sets the index write word from IndexOUT to the index of the corresponding1st datum of the data cache (bus coupler).
Unless a master can ensure consistent output of data entry and relevant address in the index word on the Profibus, the index write word must be set to 0 at the beginning of the operation for signalling that the data cannot be taken over by the bus coupler instantaneously. Finally, the value for the 1st datum of the transferred area is entered by the master.
The bus coupler stores the data successively in its data cache from the position specified in the Index Write word. The bus coupler signals data storage by entry of the index for the 1st datum into the index write word of the indexIN area. The bus coupler continues taking over the data cyclically, until the value in datum index write word of indexOUT changes.
If the index write word defines a data range beyond the max. number of 1080 process data, take-over by the bus coupler is not confirmed. 0 is returned in the index write word of indexIN.

3.4.2 Data transfer from the bus coupler to the Profibus master

Read

The master requests data from the bus coupler by entry of the 1st datum of the data cache (bus coupler) into the index read word of indexOUT.
The bus coupler enters the required data successively, starting with the 1st datum defined in the index read word of indexOUT, into the data area of the variable multiplexer module. Subsequently, the index of the 1st datum of the transmitted data area is entered into the index read word of index IN.
Unless the bus coupler can ensure that the data entry and the relevant confirmation in the index read word are output consistently on the Profibus, the index read word must be set to 0 at the beginning of the operation, for signalling that data take-over by the master is not possible instantaneously. Finally, the value for the 1st datum of the transferred range is entered by the bus coupler.
The bus coupler updates the required data in each cycle, until the data value of the index read word of indexOUT changes.
If the index read word defines a data range exceeding the max.1080 process data, takeover by the bus coupler is not confirmed. 0 is returned in the index read word of the indexIN.

3.5 Data exchange via multiplexer modules with fixed data length

Modules A.3...A.6

4 multiplex modules with fixed data length are defined (1 datum, 4 data, 8 data or 14 data for each direction). These modules include the multiplex function and a fixed number of data pertaining to this module. The data are defined as consistent data and must also be transmitted in this form. If a master can ensure these only in conjunction with consistent data models (Siemens S7), these modules must be used.

Several multiplex modules can be used. Single data and the module for the parameter channel must be used prior to the first multiplex module.

The data structure of the multiplexer module with fixed data length is:

e.g. module A.4

Data direction	Word	Range	Description
IndexOUT	Index_Read	0 – 1080	Index of the cache memory of the 1st datum of the data block, which is requested from the PLC
	Index_Write	0 – 1080	Index of the cache memory of the 1st datum of the data block, which is defined by the PLC
	Read Value 1	2 bytes	
	Read Value 2	2 bytes	
	Read Value 3	2 bytes	
	Read Value 4	2 bytes	
IndexIN	Index_Read	0 – 1080	Index of the cache memory of the 1st datum of the data block, which is provided by the bus coupler
	Index_Write	0 – 1080	Index of the cache memory of the 1st datum of the data block, which was taken over by the bus coupler
	Write value 1	2 bytes	
	Write value 2	2 bytes	
	Write value 3	2 bytes	
	Write value 3	2 bytes	

IndexOUT

The data contain the information from which index of the cache memory master data are sent or data from the cache memory of the bus coupler shall be requested/read.

IndexIN

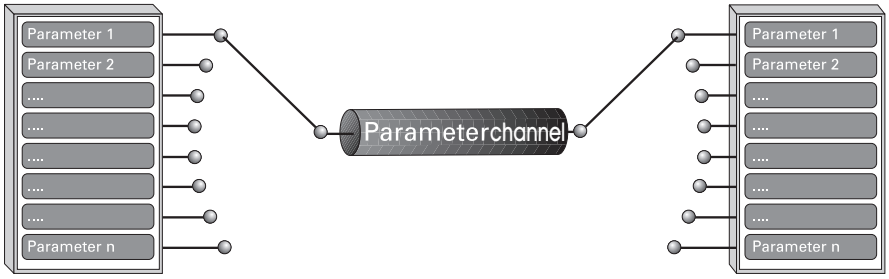
The data contain the mirrored information which bus coupler data are provided by the master or which data were taken over by the bus coupler.

The multiplex modules are supported by a function module in the "KS VARIO Engineering Set" --> FB 108 (see chapter "Function blocks for Simatic S7").

4 Parameter transmission via parameter channel in DP-V0

For parameter transmission, a "parameter channel" for transparent data exchange independent of the selected process data is available, whereby all possible protocol access modes are supported (single or block access, integer or real data format). Communication to the controller is transparent, i.e. the user himself is responsible for monitoring of value ranges, operating modes (remote/local) etc.
The parameter channel is designed for higher data quantities with low requirements on the transfer speed. This message is provided by cyclic process data exchange (also DP-V0 (=standard) operation).

Fig.: 1 Parameter channel schematic diagram



Parameter channel

Access to process, parameter and configuration data is via the parameter channel. These data are transmitted over several cycles on request.
The user can set up the parameter channel transmission by selecting module A.1 in his PROFIBUS configuration. Configuration is via the relevant bus configuration tool of the bus master.

Module eA.1

General parameter channel:[1)]

Parameter channel			
Read	Number of bytes	Write	Number of bytes
Reply data	8	Requested data	8

1) Please, note that consistent data transfer is necessary !

4.1 Message elements

An explanation of several terms used in the parameter channel protocol is given below: f

Element	Description
ID	Telegram type identification
ID1	Data format of transmitted or received data
Rd.Cnt	Number of data to be read
Adr.High	High byte start address
Adr.Low	Low byte start address
Cnt. Real	Number of real data to be written
Cnt. Int	Number of integer data to be written

ID

Element ID identifies the telegram type:
ID = 0x10 (16_) ≙ start telegram
ID = 0x68 (104_) ≙ data telegram
ID = 0x16 (22_) ≙ end telegram

ID1

Element ID1 defines the data format:
ID1 = 0 ≙ integer
ID1 = 1, 3 ≙ real
ID1 = 2 ≙ single characters (char)
ID1 = 4 ≙ characters in compact format

Integer(0) format Data are transmitted as 2 integer bytes in the data telegram. The start address determines if these data are integer, FixPoint1, FixPoint2 or FixPoint3 values. A start address in the float address range is converted into the relevant integer range.

Real(1,3) format Data are transmitted as 4 real bytes in the data telegram. Unless the defined start address is in the real address range, it is converted into the relevant real range.

Character(2) format Data are transmitted as 1 character byte in the data telegram. Unless the defined start address is in the integer address range, it is converted into the relevant integer range.

Compact character(4) format Data are transmitted as 4 character bytes in the data telegram. Unless the defined start address is in the integer address range, it is converted into the relevant integer range. Unused digits in the data telegram are filled with value 0x00.

Rd.Cnt Rd.Cnt defines the number of data in the selected format (ID1) to be read.

ⓘ For this purpose, data Cnt.Real and Cnt.Int must be zero.

Adr.High/ Adr.Low Fields Adr.High and Adr.Low define the start address of transmitted data. The address is split into a high byte and a low byte.

The address directory is divided into various areas. E.g. bit D15 determines, if a value is of the float or of the integer/fixpoint type.
The address survey is given in the annex and detailled in document: “Parameter Table KS Vario”.

Cnt.Real Specified value for the number of real values (ID1 = 1, 3) to be written. The value for Cnt.Int must be 0.

Cnt.Int Specified value for the number of integer values (ID1 = 0, 2, 4) to be written. The value for Cnt.Real must be 0 .

4.2 General telegram structure

For transmission of the required parameters via an 8-byte data window, an access is composed of start, data and end telegram. The structure of messages transmitted by the master is shown below.

Please, note that consistent data transmission of the 8 bytes for the parameter channel is necessary.

Start telegram ❑ Start telegram with data format, start address and number of data to be transmitted

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	ID1	Rd.Cnt	Adr.High	Adr.Low	0	Cnt. Real	Cnt. Int

Data telegram ❑ *n* data blocks with the transmitted data

Data telegram structure:

a) Transmission of REAL values (ID1 = 1,3)

Real format

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

b) Transmission of integer values (ID1 = 0)

Integer format

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

c) Transmission of char values (ID1 = 2)

Character format

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

d) Transmission of four character (ID1 = 4)

String format

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

The byte sequence for real and integer formats is according to user parameter setting as Motorola (default) oer Intel format. For this, see also document PROFIBUS-DP Process data" (9499-040-66618)

End telegram ❑ an end block, provides the operation result

End telegram structure:

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

Signification of Result	
0	OK
2	Faulty address
3	Invalid value
4	Buffer overflow

Reading or writing are always started by the master. If the values of Cnt.Real or Cnt.Int are $\neq 0$, a write service, otherwise, a read service, is started. For the latter, a value Rd.Cnt $\neq 0$ is a prerequisite.

4.3 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	0	Adr.High.	Adr.Low	0	Cnt.Real[1]	Cnt.Int[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	Count			Value

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

With Count = 1, the first value is sent, count is mirrored by KS VARIO (1 ≤ Count ≤ Cnt.Real or Cnt.Int).

❑ 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	

4.4 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	Rd.Cnt	Adr.High	Adr.Low	0	0	0

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Controller replies:	0x10						Cnt.Real[1]	Cnt.Int[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

With Count = 1, the first value is requested, Count is mirrored by the controller for flow control and the value is also sent (1 ≤ Count ≤ Cnt.Real or Cnt.Int).

❑ 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	

1) Selection: only one of values Cnt.Real and Cnt.Int may bef ≠ 0 .

4.5 Examples

4.5.1 Example 1: single access, read, integer value

Proportional band 1 of control channel 1 (**Pb 1**) of the controller shall be read out in FixPoint1 format.
Address = 0x4000 (offset for FixPoint 1) + 0x400 (channel 1) + 0x96 (Pb1) = **0x4496**

Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0x0	0x1	0x44	0x96	0	0	0

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

Data telegram 1:

	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			Value

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	

4.5.2 Example 2: block access, read, float value

The parameters of the set-point setting for control channel 2 (**SPLo , SPHi, SP2, rSP**) shall be read in real format.
Address = 0x8000 (offset for float) + 0xC00 (channel 2) + 0x69 x 2 (SPLo) = **0x8CD2 ,** 4 values.

Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0x1	0x4	0x8C	0xD2	0	0	0

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

Data telegram 1:

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

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

…

Data telegram 4:

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

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

End telegram:

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

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

4.5.3 Example 3: single access, write, integer value

Write derivative time 1 of control channel 30 **(td1**) in integer format
Address = 0x0000 (offset for integer) + 0x3E00 (channel 30) + 0x9A (td1) = **0x3E9A**

Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0x0	0x0	0x3E	0x9A	0	0	1

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

Data telegram 1:

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

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

End telegram:

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

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

4.5.4 Example 4: block access, write, float value

Write the parameters of limit value 1 of controller channel 30 **(L, H, HYS)** in real format.
Address = 0x8000 (offset for float) + 0x3C00 (channel 30) + 0x109 (L) = **0xFD09**$_{dez}$, 3 values).

Start telegram:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sends:	0x10	0x1	0x0	0xFD	0x09	0	3	0

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

Data telegram 1:

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

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

...

Data telegram 3:

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

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

End telegram:

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

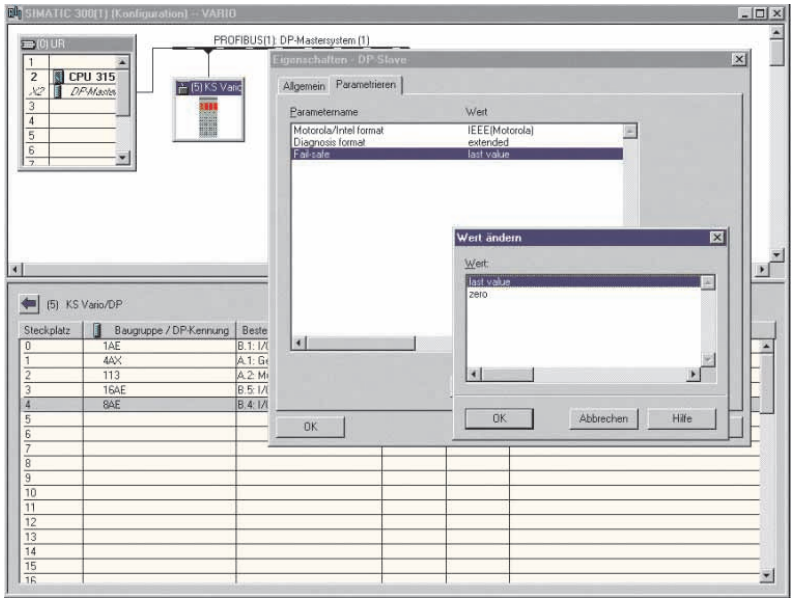
	Byte 0	Byte 1	Byte 2-3	Byte 4 - 7
Controller replies:	0x16		0	

5 Parameter transmission via DP-V1

For parameter transmission also the DPV1 functionality is realized.

6 User parameter setting, diagnosis

In addition to the standard parameter setting data, KS VARIO has also user-specific parameter setting data. Adjustment is via the bus master bus configuration tool.



The user parameter setting is valid for the complete instrument. The significations of user parameter setting data (4 bytes) are shown in the following tables.

	Bit	Descr.	Signification
1...3rd byte			Reserved for DP-V1. These bytes are set to zero for DP-V0 operation.

	Bit	Descr.	Signification	Default
4th byte	0	Motorola / Intel format	Format for float values: Motorola (IEEE 754) / Intel (0 /1) for connection also to non-conforming PLCs or PC cards.	0 (Motorola)
	1	Diagnosis format	Diagnosis extended /standard (0 / 1) Extended diagnosis: standard diagnosis plus instrument-specific diagnosis Standard diagnosis: (6 bytes) without instrument-specific information	0 (extended)
	2	Fail-safe	Last value / zero (0 / 1) Behaviour in case of bus errors:maintain existing values, or set them to zero; dependent of system concept.	0 (last value)
	3..7	reserved		0

6.1 PROFIBUS-DP diagnosis information

PROFIBUS-DP offers easy, complex functions for processing diagnosis messages due to error conditions. The KS VARIO diagnosis information comprises standard diagnosis information (6 bytes) and the additional instrument-specific diagnosis information, which can be switched off by user parameter setting.

6.1.1 Standard Diagnosis message

A standard diagnosis message consists of 6 bytes.

Standard diagnosis

	Bit	Descr.	Signification
1st byte	0	Diag.station	Does not exist (sets master)
	1	Diag.station_not_ready	Slave is not ready for data exchange
	2	Diag.cfg_Fault	No correspondence of configuration data
	3	Diag.ext_diag	Slave has external diagnosis data
	4	Diag.not_supported	Requested function is not supported in slave
	5	Diag.invalid_slave_response	Sets slave firmly to 0
	6	Diag.prm_fault	Faulty parameter setting (ident number etc.)
	7	Diag.master_lock (sets master)	Slave parameter setting by other master possible

	Bit	Descr.	Signification
2nd byte	0	Diag.Prm_req	New slave parameter setting required The application has detected a condition which requires a restart with corresponding new parameter setting and reconfiguration. Following this diagnosis, the master realizes a start-up with defined parameter setting and configuration.
	1	Diag.Stat_diag	Static diagnosis (byte diag bits) Due to an application condition, the slave cannot provide valid data. The master requests only diagnosis information, until the slave resets this bit. As the PROFIBUS-DP is in data exchange status, however, data exchange can be continued immediately after resetting the static diagnosis.
	2	fixed to 1	
	3	Diag.WD_on	Response monitoring active
	4	Diag.freeze _mode	Freeze command received
	5	Sync_Mode	Sync command received
	6	reserved	
	7	Diag.deactivated	(set by the master)

	Bit	Descr.	Signification
3rd byte	0..6	Reserved	
	7	Diag.ext_ overflow	This bit is set by the slave, if the quantity of diagnosis data exceeds the diagnosis data area.

	Bit	Descr.	Signification
4th byte	0..7	Diag.master _add	Master address after parameter setting (0xFF without parameter setting)

	Bit	Descr.	Signification
5th byte	0..7		Ident number (high byte); KS VARIO: 0x10

	Bit	Descr.	Signification
6th byte	0..7		Ident number (low byte); 0x30

6.1.2 Instrument-specific diagnosis

The following instrument-specific diagnosis can be switched off via user parameter setting (→ 2.5 S.11). Thus switch-over to the standard diagnosis is possible, e.g. for earlier DP masters which do not support all functions, of if displayed diagnosis information isn't of interest.

	Bit	Descr.	Signification
7th byte	0..5	Revision number	
	6, 7		Always '1'

	Bit	Descr.	Bedeutung
8th byte	0..7	sign length	0x08: block length 8 bytes

	Bit	Descr.	Signification
9th byte	0..7	status type	0x81: type status message

	Bit	Descr.	Signification
10th byte	0..7	Slot number	0x00: slot: instrument

	Bit	Descr.	Signification
11th byte	0..7	specifier	0x00: no status distinction

	Bit	Descr.	Signification
12th byte	0..6	Firmware version	PROFIBUS software version

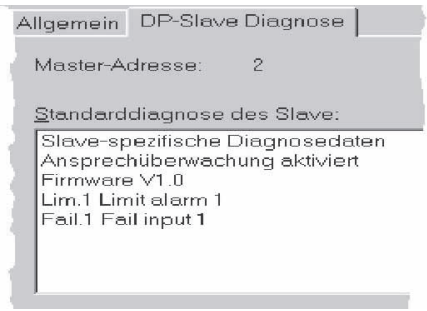
Instrument-specific diagnosis

	Bit	Descr.	Signification	Cause
13th byte	0	E.1	Internal error, cannot be corrected	e.g. EEPROM defective
	1	E.2	Internal error, can be reset	e.g. EMC trouble
	2	E.3	Configuration error, can be reset	E.g. faulty or missing configuration
	3	E.4	Hardware error	Code number and hardware not identical
	4	E.5	Internal error in DP module	
	5..7		Reserved	

	Bit	Descr.	Signification	Cause
14th byte	0	Lim.	Limit value alarm	Adjusted limit value exceeded
	1	HCA	Heating current alarm	Heating circuit break, heater band destroyed
	2	SSR	Heating current short circuit	Current flow in heating circuit with controller off, SSR defective, conglutinated
	3	Loop	Control loop alarm	Control loop broken (input or output)
	4..7		Reserved	

	Bit	Descr.	Signification	Cause
15th byte	0	Fail	Sensor error	Sensor defective, wiring error
	1..7		Reserved	

Please, note that diagnosis values are not displayed correctly on earlier Simatic® S7 master versions.



7 Quick entrance, at the example of SIMATIC® S7

The floppy delivered with the engineering set contains the GSD file and project examples for a SIMATIC® S7. By means of configuration and project, communication with a KS VARIO multi-controller system can be build up easily.

Test environment

- For the example test set-up, the following components are required:
- Programming unit or PC with PC adaptor
- Programming tool STEP®7 ³ V5.0
- Automation unit

e.g. CPU S7 315-2 DP, new version

Components

- KS VARIO BK DP/V1 (order no.: KSVC-101-00111)
- e.g. KS VARIO T8/UTH (order no.: KSVC-104-00441)
- Engineering set (order no. KSVC-109-20001)
- Cable

PROFIBUS cable automation unit <-> KS VARIO
connector with integrated terminating resistors
programming unit <-> automation unit

Test environment example:

Task

- Connect a KS VARIO with address 5 to a CPU315-2 DP via PROFIBUS-DP.
- Display or entry of process value, set-point, heating current, manual correcting variable as well as of several status messages and control words of some channels shall be possible. For this, 17 process data for reading and 17 process data for writing are required. The data are selected by means of the BlueControl engineering tool..
- For this purpose, process data module B.5 (16 words IN) and B.1 (1 word IN) for reading can be used, whereas modules B.10 (16 words OUT) and B.6 (1 word OUT) can be used for writing.

Before taking the test environment into operation, ensure that the automation unit does not contain a different user software ("initial delete").

Procedure:

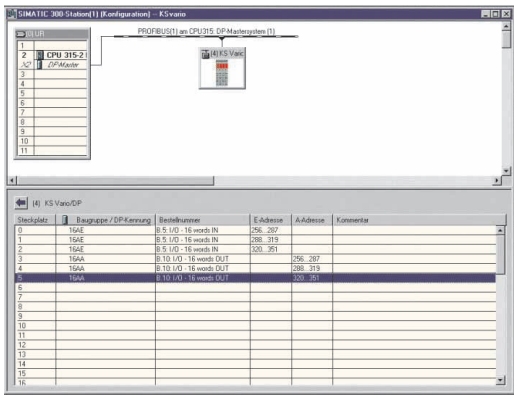
- Establishing the communication (PROFIBUS)

Load the engineering example into KS VARIO using BlueControl.
Adjust address 5 at KS VARIO bus coupler (via front panel or BlueControl) and connect it to the bus network.
Activate bus terminating resistors.

Procedure

- PROFIBUS network configuration

Insert floppy (engineering set) into programming unit.
De-archive project example (KSVARIO.zip)
Open project KS VARIO
If necessary, adapt addresses and CPU hardware configuration and transmit them to the DP master (CPU315-2 DP).
Switch automation unit to Run.



After taking the test set-up into operation, testing the I/O area by means of the variable tables (VAT 3) enclosed in the project is possible.

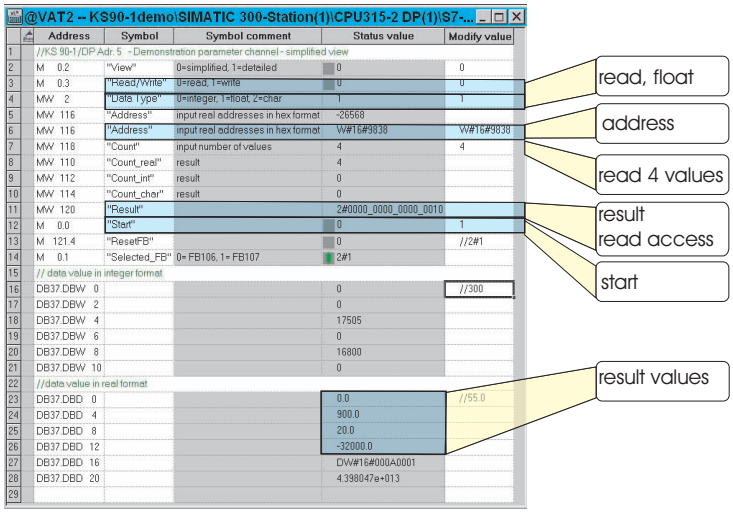
7.1 Using the parameter channel

7.1.1 Simplified access

Access to the parameter channel is facilitated via variable table VAT2. In the program example, the relevant conversions are in FC104, i.e. only the following values must be specified for manual data transmission:

- Read / write selection
- Data type selection

Fig. 2 :



- Start address
- Number of data to be transmitted
- Handling start

Procedure

For data transmission testing, proceed as follows:

- Select if you want to read or write data (0 or 1)
- Select the transmission type (integer, real, character; 0,1,2)
- Specify a start address. If you have adjusted real transmission, the address should be specified in hex format.
- Select the number of data to be transmitted by means of Count.
- Specify write values within DB37.Dxxx; as word or double word, dependent of selected data type.
- Start message handling by means of value Start = 1.
- The message is terminated with a result of 0x0002 (positive) or 0x0004 (negative). Read values are within range DB37.Dxxx.

Result

Display word Result gives the current status of transmission for the selected data area. The display word structure is:

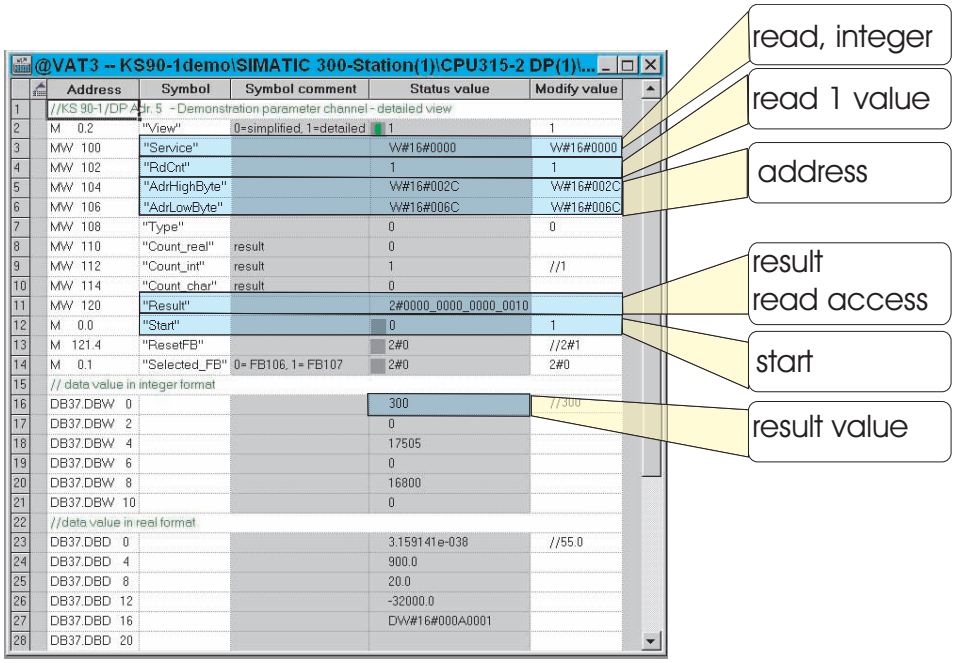
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timeout (FB)	Faulty service				000 : ok 010 : faulty address 011 : invalid value 100: buffer overflow				Wait for end telegram	Service (0=Read; 1=Write)	Reset order	wait for acknowl.	Order finished with error	Order finished without error	Order running

7.2 Direct access

Alternatively, direct access to function modules FB106, FB107 (see below) is possible via the variable table.

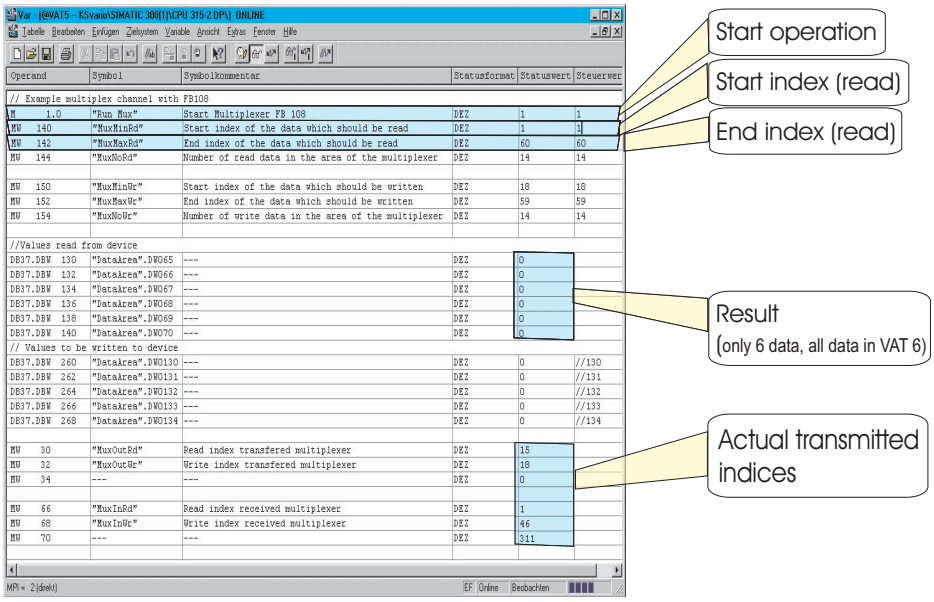
For parameter description, see below

Fig. 3 : Example of direct access to FBs to parameter channel



7.3 Using the multiplexer function

Access to the up to 1080 defined data via variable table VAT5 using the multiplex functionality is possible. After entry of the relevant parameters (e.g. start and end address of the data to be read/written), the data are stored in data block 37 / or transmitted from data block 37 to the KS Vario bus coupler.
VAT6 is used to display the data read by KS Vario. Data can be written in KS Vario via VAT7.



8 Function modules for SIMATIC® S7

The S7 function modules FB106 (PMA-FIX) and FB107 (PMA-FLoat) facilitate the access to the KS VARIO parameter and configuration data.
FB call-up is conditional 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. With a CPU315-2 DP and the on-board DP interface used, SFC modules 14 and 15 must be used for consistent data transmission. SFC modules 14 and 15 copy the I/O areas into the marker or data block 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 called up.
Every FB is provided with an instance DB, which must be specified when calling up the FB.

8.1 FB106, FB107 structure

The call parameters of function modules FB106, FB107 are listed below. With FB106, the data are stored in word width in the allocated data block, and stored in double word width with FB107.

Name	Type	Description / function
O-start	Pointer	Start of address area of output words (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).
I-start	Pointer	Start of address area of input words (e.g. address of 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 data block with parameter data. Entry includes the data block no. and the data word no. at which parameter data start. Thereby, an Offset need not be taken into account. The data are interpreted as parameter data by the specified address. The DB must be specified in the following form, e.g. DB6.DBX10.0
Service	WORD	Service (Read/Write)
Code_nr	WORD	Read access: number of values to be read
FB_nr	WORD	Address, high byte
FKT_nr	WORD	Address, low byte
Typ	WORD	d.c. (always '0')
Timeout	DWORD	Timeout value, decremented at each call. If the value = 1, the order is cancelled with error message 'timeout' .
DWLR	WORD	Length of real values; specified when writing
DWLI	WORD	Length of integer values; specified when writing
DWLC	WORD	Length of char values; specified when writing
ANZW	W	The actual 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 (FB)	Faulty service				000 : ok 010 : faulty address 011 : invalid value 100: buffer overflow				wait for end telegram	Service (0=Read; 1=Write)	Reset order	wait for acknowledgement	Order finished wih error	Order finished without error	Order running

The function modulei reads or writes KS VARIO parameter/configuration data
O-start, I-start.
The parameter channel input addresses or output addresses are entered into these parameters. The addresses are determined during configuration of the unit connected on the PROFIBUS (STEP 7 - Hardware configuration)

- DB-Para
  DB-Para is a pointer to the data block into which read data are written,
  or from which data are used when writing.

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

write access:				read access:			
	F0	≙	integer	0	≙	integer
	F1	≙	real	1	≙	real
	F2	≙	char	2	≙	char

- Code_nr
  With read accesses, Code_nr only defines the number of data to be read. In this case, data DWLR, DWLI, DWLC must be zero. When writing, Code_nr = 0 must be set.

- FB_nr
  Specifies the high byte of the parameter to be addressed.
  Example: parameter L. 1 ,address 2100_{dez} ≙ 0834_{hex}
  FB_nr contains: 08_{hex} ≙ 8_{dez}
  FKT_nr contains: 34_{hex} ≙ 52_{dez}

- FKT_nr
  Specifies the low byte of the parameter to be addressed.

- DWLR (real), DWLI (integer), DWLC (char)
  After a read access, these parameters contain the number of received data. With a write access, these parameters contain the number of data to be transmitted. Only one of the data may contain a value ≠ 0 .

- ANZW
  This display word gives the actual transmission status. Bit 4 can be used as an input for resetting FB 106 / FB 107.

8.2 FB108 structure

Function block FB 108 supports the data exchange via multiplexer modules. It writes and reads data cyclically. Max. 1080 write and read data are available in the bus coupler. As the complete multiplex data must be consistent, using functions SFC 14 or SFC 15 for data transfer from and to the process area is necessary.

Name	Type	Description / function
Run	Bool	Activating function block 108
MinRd	Integer	Index minimum for reading
MaxRd	Integer	Index maximum for reading
NoMuxRd	Integer	Number of read data in a multiplex message
PtrMuxRd	POINTER	Start address of data copied from the PEW area bySFC 14
MinWr	Integer	Index minimum for writing
MaxWr	Integer	Index maximum for writing
NoMuxWr	Integer	Number of write data in a multiplex message
PtrMuxWr	POINTER	Start address of data copied into the PAW area by SFC 15
PtrDataRd	POINTER	Start address of the data area in which the data read by function block 108 must be stored
PtrDataWr	POINTER	Start address of the data area containing the data which must be output by function block 108
IndOutRd	Integer	First index of data which are currently requested
IndOutWr	Integer	First index of data which are currently written

The data signification of function block 108 is:

- Run
  1 activates the function block, i.e. the data exchange. 0 stops the data exchange.

- MinRd
  Min. index of data to be read. The range of the datum is within 1 – 1080.

- MaxRd
  Max. index of data to be read. The range of the datum is within I - 1080. Consequently, the number of cyclically read data is determined by MaxRd – MinRd + 1.

- NoMuxRd
  Number of read data in a multiplex message. According to the selected fixed multiplex module, this value can be 1, 4, 8 or 14.

- PtrMuxRd
  As the complete multiplex module data must be consistent, function block SFC 14 must be used for copying the data from the PEW area. The start address into which these data were copied must be transmitted to the function block.

- MinWr
  Min. index of data to be written. The range of the datum is within 1 – 1080.

- MaxWr
  Max. index of data to be written. The range of the datum is within I - 1080. Consequently, the number of cyclically written data is determined by MaxWr – MinWr + 1.

- NoMuxWr
  Number of write data in a multiplex message. According to the selected fixed multiplex module, this value can be 1, 4, 8 or 14. .

- PtrMuxWr
  As the complete data of the multiplex module must be consistent, function block SFC 15 must be used to copy the data into the PAW area. To enable function SFC 15 to copy the data into the PAW area, the start address for storing the data must be transmitted to the function block.

- PtrDataRd
  Start address of the data area for storing the data read by function block 108. Typically, this is a data block. The data block length must be appropriate for storage of the number of data to be read defined by MaxRd – MinRd + 1.

- PtrDataWr
  Start address of the data area containing the data to be output by function block 108. Typically, this is a data block. The data block length must be appropriate for containing the number of data to be written defined by MaxWr– MinWr + 1 .

- IndOutRd
  Indicates the first index of data which are currently requested. The range of the datum is within MinRd – MaxRd.

- IndOutWr
  Indicates the first index of the data which are currently written. The value of the datum is within MinWr – MaxWr.

9 Annex

9.1 System structure

9.1.1 Minimum equipment fo a PROFIBUS system

A PROFIBUS system comprises the following minimum components:
a bus master, which controls the data communication,
one or several slave units, which make data available on request by the master,
the transfer medium comprising bus cable and bus connector for connecting the individual units, one or several bus segments, which are connected with repeaters.

9.1.2 Maximum equipment of a PROFIBUS system

A bus segment comprises max. 32 field instruments (active and passive ones). The maximum number of slave units, which can be operated at a PROFIBUS master over several segments is determined by the internal memory structure of the master. Therefore you should inform yourself on the master capacity when planning a system. The bus cable can be opened at any point to include another unit by adding a bus connector. At the end of a segment, extending the bus cable up to the specified segment lengths and including new units for extensions are possible. The length of a bus segment is dependent of adjusted Baudrate. The Baudrate is determined mainly by the system constellation (segment length, number of distributed inputs and outputs) and the required polling intervals of individual units. For all units connected on the bus, the Baudrate determined by the master must be selected.
At the start and end of a segment, terminating resistors must be connected to ensure a physically clean signal level. These are already integrated in most connectors and must be activated only by means of switch.

Profibus instruments must be connected in line structure.
A PROFIBUS installation can be extended by connecting repeaters,
if more than 32 units must be connected
or for connection over longer distances than defined for the Baudrates.

A fully extended PROFIBUS system can include max. 126 stations with addresses 0 ... 125. Each repeater reduces the maximum number of stations within a segment. No PROFIBUS unit address is assigned to a passive unit. Nevertheless, its input circuitry is an additional load due to the bus driver power consumption. However, a repeater is without effect on the overall number of stations connected on the bus. The maximum connectable number of repeaters which may be connected in series can differ dependent of manufacturer. Therefore, information on any limitations should be asked for in advance from the manufacturer when planning a system.

9.1.3 Cable run inside buildings

The following hints for cable installation are applicable to a pairwisely twisted, screened two-wire cable. The screening is used for improvement of the electromagnetic compatibility. With type A PROFIBUS cable, meshed screening and foil screening are integrated into the cable. The following cable screening versions always include the two screening versions (meshed and foil). Using only the foil screening must be omitted, because it is very thin and can be interrupted easily, which may cause interruption of the potential equalization.

Both ends of the cable screening must be connected to the reference potential via a large surface of conducting material. When installing a repeater or a field unit in a cabinet, the cable screening should be connected to a screening rail via cable collar near the cable gland.

The screening must be continued up to the field instrument and connected with the conducting housing and/or the metal connector. Ensure that the ground potential of the instrument housing and of the control cabinet accommodating the field instrument are equal due to large-surface metal contact.

Mounting a screening rail on a painted surface is without effect. These measures ensure grounding of high-frequency interference via the meshed screening. With external interference voltage on the data lines despite these measures, increase the voltage potential on the two data lines regularly so that the difference voltage is normally not destroyed. In ordinary cases, safe data transmission is still ensured with a shift of the ground potential by some volts. With higher shifts (potential DGND at pin 5 against reference potential) a potential equalization lead with a minimum cross section of 10 mm² should be installed in parallel to the bus cable. With extreme interference effect, the bus cable can be installed additionally in a steel tube or a tight sheet metal duct. The tube or duct must be grounded correctly.

The minimum distance between bus cable and other cables for voltages exceeding 60 V must be 20 cm. The bus cable should also be kept separate from telephone cables and cables leading into the explosion-hazarded area. In these cases, we recommend using a separate cable duct for the bus cable.

Only conducting materials with regular connection to the reference potential should be used for the cable duct. The bus cables must not be subjected to mechanical stress or obvious damage. Unless this condition is met, special protective measures, e.g. installation in tubes, etc., are necessary.

9.1.4 Non-grounded systems

The construction of a non-grounded system may be necessary for various reasons. For this purpose, there must be a high-impedance connection between instrument ground and reference potential (e.g. by means of RC protective circuitry). When connecting bus segments by means of repeaters, we recommend using the non-grounded construction to prevent transmission of potential differences between bus segments.

9.2 Address survey

Modbus addresses

The detailled address-table you find in the document: Parameter Table for KS VARIO (9499-040-72911)

The address is coded in 2 bytes. The 2 most significant bits (D15, D14) are used for definition of the format in which the data are written or read.

The Modbus directory is divided into equally sized areas of 512 words (bit D13...D09). Each of these areas permits access to all data for one control channel (1...30 channels).

2 areas have a special status. All instrument data are stored in the lower address area (Modbus addr. 0..512). The most important process data of all 30 channels are contained additionally in the following area (addr. 512...1023). This area is intended for access by visualization facilities.

The signification of address bits is:

INTEGER/ FIX-Point Modbus addresses:

MSB		LSB
D15 - D14	**D13 - D09**	**D08 - D00**
Data format 00: Integer 01: Fix Point 1 1X: reserved for Float	**Instrument, visualization, channel X** 00000: Instrument data 00001: Visualization data 00010: Data channel 1 00011: Data channel 2 .... 11111: Data channel 30	**relevant datum**

Modbus directory (data format: Integer):
4000 hex must be added for the **Fix Point 1** area.

Integer

Addresses	**Data**
0 511 (1FF hex)	**Instrument data**
512 (200 hex) 1023 (3FF hex)	**Visualization area channel 1..30**
1024 (400 hex) 1535 (5FF hex)	**Data channel 1**
1536 (600 hex) 2047 (7FF hex)	**Data channel 2**
....	....
15872 (3E00 hex) 16383 (3FFF hex)	**Data channel 30**

FLOAT Modbus addresses:

MSB		LSB
D15	**D14 - D10**	**D09 - D00**
Data format 0: reserved for Integer and Fix Point 1 1: Float	**Instrument, visualization, channel X** 00000: instrument data 00001: visualization data 00010: data channel 1 00011: data channel 2 .... 11111: data channel 30	**relevant datum**

Modbus directory (data format: FLOAT):

Float

Addresses	Data
32768 (8000 hex) 33791 (83FF hex)	**Instrument data**
33792 (8400 hex) 34815 (87FF hex)	**Visualization area channel 1..30**
34816 (8800 hex) 35839 (8BFF hex)	**Data channel 1**
35840 (8C00 hex) 36863 (8FFF hex)	**Data channel 2**
....	....
64512 (FC00 hex) 65535 (FFFF hex)	**Data channel 30**

The data are stored in 4 bytes. The Float-Modbus addresses (related to Integer / Fix-Point 1) must be multiplied by 2.

9.3 Explanation of terms

Abbr.	Abbreviation
AG	Abbr. For automation system, f.e. PLC
BlueControl®	Engineering tool - software for BluePort® controllers
BluePort® interface	Controller front panel interface for connection of an engineering tool
ET	Abbr. for engineering tool
Fail-safe	Instrument behaviour in case of failure of the PROFIBUS cable or bus master
FB	Abbr. for function block
Fkt	Abbr. for function
Forcing	Input or output specification via the bus
Function	A self-contained partial function of a function block seen from the interface
Function block	A self-contained processing unit
GSD file	Geräte Stammdaten Datei; standard description of communication functions
HW	Abbr. for hardware
Parameter channel	Possibility for cyclic, sequential value transmission during cyclic process data exchange
PG	Abbr. f. programming unit
PROFIBUS-DP	Standard communication protocol to EN50170 vol.2 (DP: de-central peripheral units)
RS485	Standard 2-wire connection, half duplex, (EIA RS 485)
S5 / S7	Siemens AG PLC series
Serial interface	Bussable interface on the controller rear panel
SW	Abbr. for software
TTL	Signal level at block level
VAT	Variable table: monitor view of values in STEP®7

9.4 GSD File

```
;===============================================================
; Device Database File for product  KS Vario / D P
; Copyright (C) PMA Prozeß- und Maschinen Automation GmbH 2002
; Miramstr. 87, D-34123 Kassel, Tel. +49 (0) 561/ 505 -1307
; Version : V1.0   Date: 2002-08-01
; File:   PMA_Vari.gsd
;===============================================================
#Profibus_DP
GSD_Revision=1
Vendor_Name = "PMA GmbH"
Model_Name = "KS Vario/DP"
Revision = "V 1.0"
Revision_Number = 1
Ident_Number = 0x1030
Protocol_Ident = 0
; DP
Station_Type = 0
; Slave
FMS_supp = 0
; only DP
Hardware_Release = "HV 01.00"
Software_Release = "SV 01.00"
;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= "PMA1030N"
Bitmap_Diag= "PMA1030D"
Bitmap_SF= "PMA1030S"
;
;--DP-Slave related key words -----
;
```

```
Freeze_Mode_supp = 1
; supported
Sync_Mode_supp = 1
; supported
Auto_Baud_supp = 1
; supported
Set_Slave_Add_supp = 0
; not supported
Max_User_Prm_Data_Len = 4
; supported
;User_Prm_Data = 0,0,0,0
;
Min_Slave_Intervall = 1
;minimum slave poll cycle (based on 100us)
Modul_Offset=0
Modular_Station = 1
; modular device
Max_Module=50
; max. number of modules
Max_Input_len = 244
Max_Output_len = 244
Max_Data_len = 488
;-----------------------
;User parameter text
;-----------------------
PrmText=1 ;Intel / Motorola(IEEE) - Format
Text(0)="IEEE(Motorola)"
Text(1)="Intel"
EndPrmText
;
PrmText=2 ;Diagnosis Standard / extended
Text(0)="extended"
Text(1)="standard"
EndPrmText
;
PrmText=3 ;Fail-safe behaviour last value / zero
Text(0)="last value"
Text(1)="zero"
EndPrmText
;
ExtUserPrmData=1 "Motorola/Intel format"
Bit(0) 0 0-1
Prm_Text_ref=1
EndExtUserPrmData
;
ExtUserPrmData=2 "Diagnosis format"
Bit(1) 0 0-1
Prm_Text_ref=2
EndExtUserPrmData
;
ExtUserPrmData=3 "Fail-safe"
Bit(2) 0 0-1
Prm_Text_ref=3
EndExtUserPrmData
;
;-----------------------
; Device User parameter
;-----------------------
Ext_User_Prm_data_const(0) = 0x00, 0x00, 0x00
Ext_User_Prm_data_ref(3)= 1; Motorola /Intel format
Ext_user_prm_data_ref(3)= 2; Diagnosis format
```

```
Ext_user_prm_data_ref(3)= 3; Fail-safe
;
;----------------------
;Module description
;----------------------

;A.1: General - Parameter channel
Module = "A.1: General-Parameter channel" 0xF3
1
EndModule
;
;A.2: Multiplexfunction variablen Aufbau
Module = "A.2: Variable Multiplexfunc." 0x71
2
EndModule
;
;A.3: Multiplexfunction festen Aufbau 1 Date
Module = "A.3: 1 data fix multiplexfunc." 0xF2
3
EndModule
;
;A.4: Multiplexfunction festen Aufbau 4 Daten
Module = "A.4: 4 data fix multiplexfunc." 0xF5
4
EndModule
;
;A.5: Multiplexfunction festen Aufbau 8 Daten
Module = "A.5: 8 data fix multiplexfunc." 0xF9
5
EndModule
;
;A.6: Multiplexfunction festen Aufbau 14 Daten
Module = "A.6: 14 data fix multiplexfunc." 0xFF
6
EndModule
;
;B.1: I/O input 1 word
Module = "B.1: I/O - 1 word IN" 0x50
7
EndModule
;
;B.2: I/O input 2 words
Module = "B.2: I/O - 2 words IN" 0x51
8
EndModule
;
;B.3: I/O input 4 words
Module = "B.3: I/O - 4 words IN" 0x53
9
EndModule;
;
;B.4: I/O input 8 words
Module = "B.4: I/O - 8 words IN" 0x57
10
EndModule
;
;B.5: I/O input 16 words
Module = "B.5: I/O - 16 words IN" 0x5F
11
EndModule
;
```

```
;B.6: I/O output 1 word
Module = "B.6: I/O - 1 word OUT" 0x60
12
EndModule;
;
;B.7: I/O output 2 words
Module = "B.7: I/O - 2 words OUT" 0x61
13
EndModule
;
;B.8: I/O output 4 words
Module = "B.8: I/O - 4 words OUT" 0x63
14
EndModule
;
;B.9: I/O output 8 words
Module = "B.9: I/O - 8 words OUT" 0x67
15
EndModule
;
;B.10: I/O output 16 words
Module = "B.10: I/O - 16 words OUT" 0x6F
16
EndModule
;
Slave_Family=5
Max_Diag_Data_Len=15
Fail_Safe=1
;OrderNumber="KS90-1xx-2xxx1"
;
;--Diagnosis -----
;
; Byte 1
Unit_Diag_Area=24-31
Value(10) = "Firmware V1.0"
Value(11) = "Firmware V1.1"
Value(12) = "Firmware V1.2"
Value(13) = "Firmware V1.3"
Value(14) = "Firmware V1.4"
Value(15) = "Firmware V1.5"
Value(16) = "Firmware V1.6"
Value(17) = "Firmware V1.7"
Value(18) = "Firmware V1.8"
Value(19) = "Firmware V1.9"
Value(20) = "Firmware V2.0"
Value(21) = "Firmware V2.1"
Value(22) = "Firmware V2.2"
Value(23) = "Firmware V2.3"
Value(24) = "Firmware V2.4"
Value(25) = "Firmware V2.5"
Value(26) = "Firmware V2.6"
Value(27) = "Firmware V2.7"
Value(28) = "Firmware V2.8"
Value(29) = "Firmware V2.9"
Value(30) = "Firmware V3.0"
Value(31) = "Firmware V3.1"
Value(32) = "Firmware V3.2"
Value(33) = "Firmware V3.3"
Value(34) = "Firmware V3.4"
Value(35) = "Firmware V3.5"
Value(36) = "Firmware V3.6"
```

```
Value(37) = "Firmware V3.7"
Value(38) = "Firmware V3.8"
Value(39) = "Firmware V3.9"
Value(40) = "Firmware V4.0"
Unit_Diag_Area_End
;
; Byte 2
;
Unit_Diag_Bit(32)  = "E.1 Internal error"
Unit_Diag_Bit(33)  = "E.2 Internal error/temporary"
Unit_Diag_Bit(34) = "E.3 Dev. configuration mismatch"
Unit_Diag_Bit(35) = "E.4 Hardware error"
Unit_Diag_Bit(36) = "E.5 Internal interface problem"
;
; Byte 3
;
Unit_Diag_Bit(40)  = "Lim Limit alarm"
Unit_Diag_Bit(41)  = "HCA Heating current alarm"
Unit_Diag_Bit(42)  = "SSR Heating current short circ."
Unit_Diag_Bit(43)  = "LOOP Control loop alarm"
;
; Byte 4
;
Unit_Diag_Bit(48)  = "Fail Fail input"
;
```

A4