POSITAL
FRABA

ABSOLUTE ROTARY ENCODER WITH INTERBUS INTERFACE

USER MANUAL

FRABA Inc.
1800 East State Street, Suite 148, Hamilton, NJ 08609
Phone +1 609 750 8705, Fax. +1 609 750 8703
www.posital.com, info@posital.com

POSITAL
FRABA

INTERBUS
USER MANUAL

Imprint

FRABA Inc.

1800 East State Street, Suite 148

Hamilton, NJ 08609

USA

Phone +1 609 750 8705

Fax. +1 609 750 8703

www.posital.com

info@posital.com

Copyright

The company FRABA POSITAL GmbH claims copyright on this documentation. It is not allowed to modify, extend, copy, or hand over to a third party this documentation without written approval by the company FRABA POSITAL GmbH. Nor is any liability assumed for damages resulting from the use of the information contained herein. Further, this publication and features described herein are subject to change without notice.

Disclaimer of Warranty

FRABA POSITAL GmbH makes no representations or warranties, either express or implied, by or with respect to anything in this manual. And shall not be liable for any implied warranties of merchantability and fitness for a particular purpose or for any indirect, special, or consequential damages.

Alteration of Specifications reserved

Technical specifications, which are described in this manual, are subject to change due to our permanent strive to improve our products.

Document information

Documentname:	UMUS OCD IB.doc
Versionnumber:	05/09
Author:	Reiner Bätjer

Phone Service

For technical support, questions and suggestions for improving our products and documentations call our telephone line: +49 (0) 221 96213-0.

POSITAL
FRABA

INTERBUS
USER MANUAL

Contents

1 Introduction

1.1 The Absolute Rotary Encoder

Absolute rotary encoders provide a definite value for every possible position. All these values are reflected on one or more code discs. The beams of infrared LEDs are sent through the code discs and detected by arrays of photo transistors. The output signals are electronically amplified and the resulting value is transferred to the interface.

The absolute rotary encoder has a maximum resolution of 8192 steps per revolution (13 Bit). The multi-turn version can detect up to 4096 revolutions (12 Bit). Therefore the largest resulting resolution is 25 Bit = 2^{25} = 33,554,432 steps. The standard single-turn version has 12 Bits, the standard multi-turn version 24 Bits.

The absolute rotary encoder meets all specifications according to INTERBUS profile No. 71, given by the user organization *ENCOM*, part of the INTERBUS club. This is the profile of the European fieldbus norm (INTERBUS norm, EN50170). The implemented interface is ready for the connection to the remote bus of the INTERBUS system. The software supports all functions of the encoder classes K1, K2 and K3. The process data is generally transmitted in binary code.

The following parameters of the absolute rotary encoder can be directly programmed via the INTERBUS network without any extra device:

- counting direction (complement)
- measuring units per number of Revolutions
- Needed number of Revolutions for given number of steps
- Preset value
- Zero point displacement
- Velocity output
- Cam functionality

To reduce the installation time significantly, the encoder supports the Windows version of the CMD software "Configurating - Monitoring - Diagnostics" (Version G4). This software is available for all INTERBUS masters from *Phoenix Contact* in D-32819 Blomberg (Phone: +49 (0) 5235 34 02 22) and is in accordance with the specifications in this manual.

1.2 The bus system INTERBUS

The trend to a higher degree of automation in general and the complex requests in production and process engineering ask for efficient sensors and actuators for every application. INTERBUS is a fast, universal, and open sensors/actuators bus system with one master and many slaves. INTERBUS moves the I/O area away from the controlling unit to a decentralized periphery directly in the machine. Only one serial bus cable connects the controlling unit with the I/O points. This reduces the costs for the installation of the cables to a minimum. Also, INTERBUS doesn't specify the transmission medium, allowing for the standard 9 wire INTERBUS cable, the cost saving 2 wire Loop cable, and most recently, a fibre optic cable (LWL).

The INTERBUS is an open system which is supported by more than 500 periphery suppliers with a strong increase in the number of suppliers and applicants. Therefore there is a broad variety to choose the best product for your specific needs.

1.3 Definitions and abbreviations

Bus participant device which can send, receive and amplify data.

Diagnostics detecting, localization, classification, display and interpretation of errors and messages

INTERBUS fieldbus, European fieldbus norm, according to INTERBUS norm (EN 50254).

The following abbreviations are used in this manual:

API absolute position value
CW clockwise
CCW counter clockwise
PW preset value
PI process value
VC velocity
MSB most significant bit
LSB least significant bit

2 Encoder Classes

The Absolute Rotary Encoders with INTERBUS interface transmit the process value in binary code. There are encoders which are able to output the position value only (K1, K2) and encoders which can be programmed according to the application (K3). The different profiles are regulated by the user group (ENCOM) which ensures the communication between the peripheral devices. One encoder type can be used in very different applications because the software in the encoder allows to adapt it to the different needs.

2.1. Profile Overview

Three different ENCOM profiles are supported by FRABA's INTERBUS encoders. These profiles differ in the number of in and out bytes allocated by the master for the device. Profiles K1 and K2 have no out bytes (from the master) and cannot be programmed. They differ only in the number of in bytes, 2 for K1 and 4 for K2. Profile K3 has 4 bytes of in and out data, hence it can be programmed.

Profile	IN Bytes	OUT Bytes
K1	2	0
K2	4	0
K3	4	4

2.1.1 Profile K1

The Absolute Rotary Encoder with profile K1 transmits 16 bits of process data. These data bytes are binary coded and right justified. The encoders are not programmable. This profile is used often for single turn encoders (example: 12 bit single turn encoder).

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Meaning	0	0	0	0	MSB	x	x	x	x	x	x	x	x	x	x	LSB

2.1.2 Profile K2

The Absolute Rotary Encoder with profile K2 transmits 32 bits process data. These data bytes are binary coded and right justified. The encoders are not programmable. This profile is used for multi turn encoders (example: 24 bit multi turn encoder).

Bit number	31	30	29	28	27	26	25	24	23	22	21	...	3	2	1	0
Meaning	0	0	0	0	0	0	0	0	MSB	x	x	x	x	x	x	LSB

2.1.3 Profile K3

The Absolute Rotary Encoder with profile K3 is programmable. It transmits 32 bits of process data (7 status and command bits and 25 bits for the position value). Bits 0 to 24 are binary coded and right justified. In case of parameterization the master sends 32 bit process data to the encoder (3 status bits, 4 parameter bits and 25 data bits).

	Status bits			Parameter bits				Data bits									
Bit number	31	30	29	28	27	26	25	24	23	22	21	...	...	3	2	1	0
Meaning	0	0	0	0	0	0	0	MSB	x	x	x	X	x	x	x	x	LSB

The parameterization is done via the process data channel. In the command word (bits 25 to 31, status bits and parameter bits) the master sends the command to the encoder. In the status word the encoder transmits messages to the master. Bits 0 to 24 are reserved for the position value or the parameter value.

2.2 ID-Codes

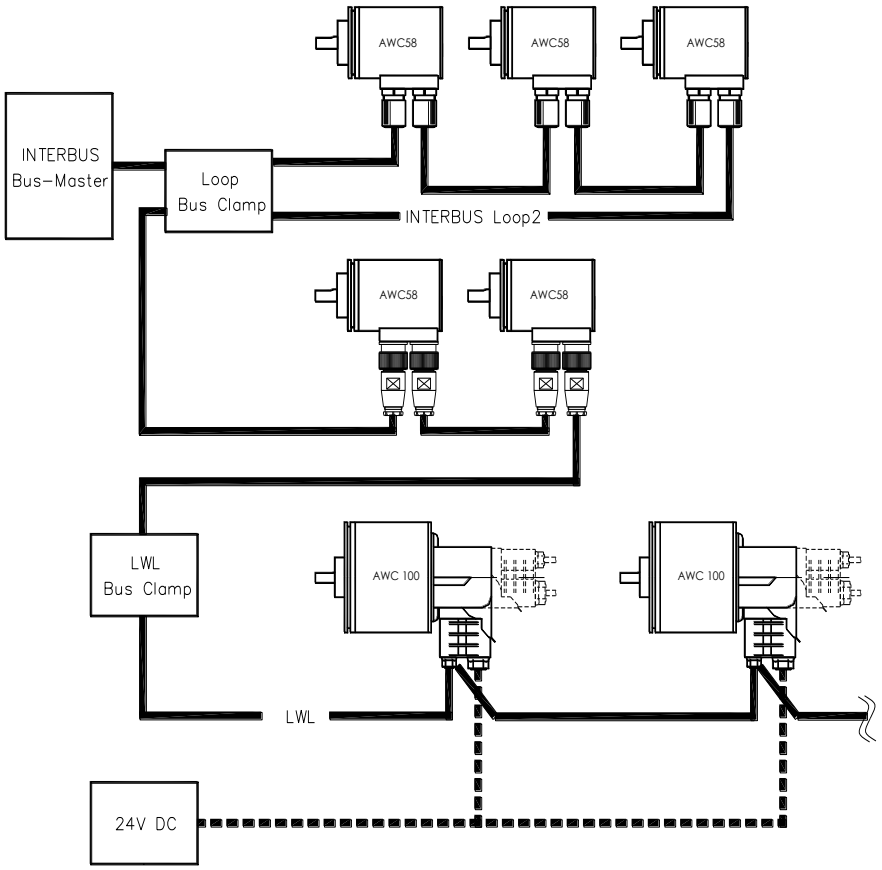
The following specified ID codes result according to the different interfaces and profiles

Interface	ID-Code for K1		ID-Code for K2		ID-Code for K3	
INTERBUS	36h	54d	36h	54d	37h	55d
INTERBUS LWL	36h	54d	36h	54d	37h	55d
INTERBUS Loop2	B2h	178d	B2h	178d	B3h	179d

3 Hardware topology and INTERBUS network

3.1 Network topology

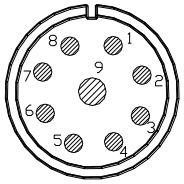
The physical structure of an INTERBUS system is that of a ring. The INTERBUS is installed as a compact cable in your plant, following one direction. Starting at a PLC-module or an IPC-master card the bus system connects the control systems with the peripheral Input- and Output-modules (INTERBUS participants). The part of the bus that leads through the whole plant is called Remote Bus (RB) and it bridges distances up to 12,8 km between the peripheral sub stations. To this main bus the local bus is connected. The local bus is called Installation remote bus or local bus – depending on the type. The structure of the Installation remote bus corresponds to the structure of the remote bus, but there is the option to carry the power supply for the sensors in the bus cable (hybrid cable structure). The Installation remote bus is suitable for the contion of systems with different sub stations which have a direct connection to sensors and actuators. That results in an optimally short and cheap solution for the connection of sensors and actuators. The local bus is designed for the cheap and flexible implementation of peripheral sub stations in control cabinets and terminal boxes. The different peripheral bus participants are connected via local bus and bus clamp. The bus clamp connects the local bus and remote bus. The encoders with standard and LWL connections are remote bus participants. The Loop2 encoder is a local bus participant



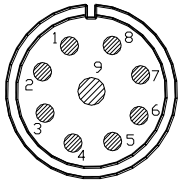
Pinning information for standard 9 pin round connector:

Male (IB-In)	Signal	Female (IB-Out)
1	DO	1
2	$\overline{DO}$	2
3	DI	3
4	$\overline{DI}$	4
5	GND	5
6	PE	6
7	+ 12-30 V DC	7
8	GND (0V)	8
9	RBST	9 *

* bridge is with version A1 no longer needed



From Soldering side:
Connector plug in / Counterpart respectively

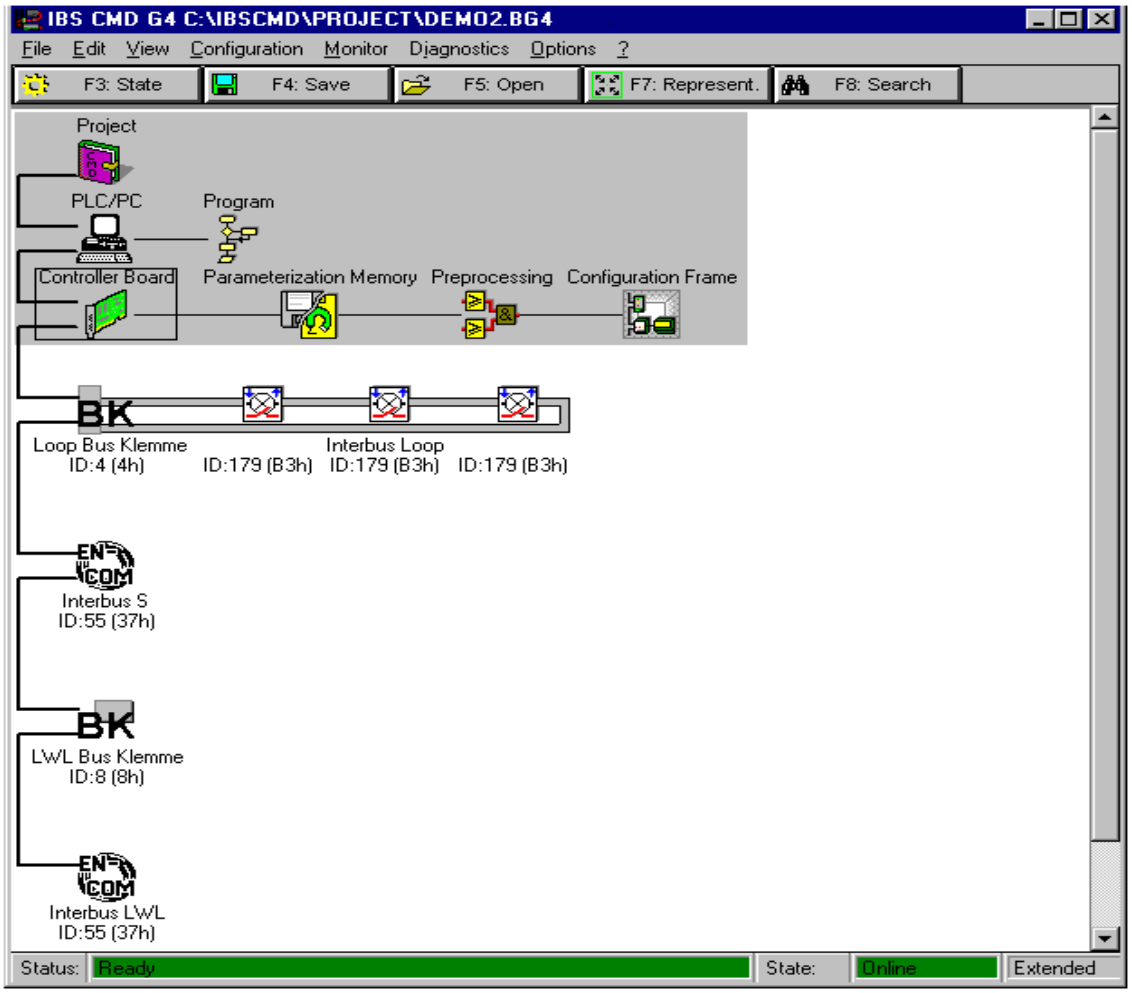


3.2 Connecting the bus participants

The different bus participants are connected with a hybrid cable. This cable carries bus wires coming from the master and bus wires back to the master.
The standard INTERBUS encoder is connected to the bus cable with two 9 pin connectors. The incoming bus cable is female (connector at the encoder: male), the outgoing cable is male (connector at the encoder: female) as it carries the power supply.

The addressing of the different bus participants is not necessary because the address is determined by the physical position of the sensor /actuator on the bus. When connecting the standard INTERBUS, the shield of the cable must be connected to the housing of the encoder for EMC quality. With the Loop, the housing should be connected to PE.

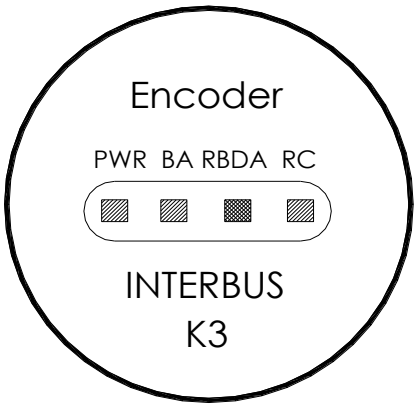
Possible structure of an INTERBUS network with multiple interface participants



3.3 Diagnose LEDs

For diagnostics of the bus status, the standard encoder has 4 coloured LEDs which reflect the state of the INTERBUS network and communication levels of the bus at the encoder. The LWL has in addition, 2 more LEDs which give information about the fibre optics. The Loop2 has only one coloured LED which yields information.

INTERBUS



The 4 LEDs have the following meaning:

UL (PWR)	Power	green
BA	Bus Active	green
RBDA (RD)	Remote Bus Disable	yellow(red)
RC	Remote Control	green

The following states are displayed if LED´s are on:

UL + RC	Power supply is on, the master of the bus system is trying to build up communication with the encoder, the bus is not running.
UL + BA + RC	Power supply is on, the configuration frame has been read in, the master has (BA blinking) detected the bus participant.
UL + BA + RC	Power supply is on, the master has placed the bus into the Run state, bus communication is active.
UL + RBDA	Power supply is on, the master has detected an error, bus communication is not active. See the diagnostic function in the CMD software for cause of error.

4 Configuration of the system with CMD software

The CMD software is a tool from Phoenix Contact for configuration, monitoring and diagnosing an INTERBUS system. It allows an interactive and PLC independent access on the bus and all its participants. With implemented service functions the parameters of the peripheral devices can be set**.** The FRABA encoders with INTERBUS interface can be programmed directly with the CMD address monitor.*

4.1 Configuration

With this part of CMD, the bus system can be structured and the participants are configured. New slaves can be inserted and their I/O´s can be addressed for the PLC program, all participants can be found with the search function in a large system, several slaves can be grouped together in a segment. The configured bus structure can be checked before the "run state" of the system. The master of the bus can be controlled with the "master function".

4.2 Monitoring

With the monitoring function the input and output of the connected devices can be displayed and changed. During run time of the bus, the status of every output bit of each device can be displayed and every input bit of the device changed. The display form depends on the manufacturer. It can be for example, a signed number, a digital matrix, or an analog beam.

4.3 Diagnostics

The diagnostics function allows a fast and effective help in case of error while the configuration and the service by detecting and localization of a bus error in the system. Defect bus participants or connecting cables and errors in the sensor/actor field are detected with the function "bus diagnostics". A further option of this part of the CMD is the possibility to compare the configured bus structure with the connected bus at the IB master.

** FRABA can supply a proprietary software to program encoders, for use only with the Phoenix Contact PC-ISA card. The software is Windows based for 95/98 and NT, and can be downloaded free of charge at our website*, www.posital.com*.*

4.4 Using the CMD software

INTERBUS-CMD is able to run on any PC with MS-WINDOWS® and can be used for all INTERBUS master (PLC- or IPC cards). The connection from the PC to the master is a special IBS V.24 interface. The CMD software is independent from the used controlling hardware and software.

After "power on" the CMD tries to get a connection to the INTERBUS master and reads out the file-name of the downloaded bus configuration. If successful the CMD loads this configuration (master and slaves) out of its memory and displays the bus network.

In the mode *Off-line* it is possible to work with a virtual bus which does not really exist. So you can configure a bus system in a projected status. The basic window provides functions for projecting, getting started and diagnosing. In status *On-line* the configured bus is compared with the real bus.

4.5 Projecting an INTERBUS system

The functions for projecting an INTERBUS system can be found in the menu *Configuration* in the main window. Normally the projecting is done in 4 steps.

Step 1: Start
Choose **Off-line** and mode **extended** in the start window.

Step 2: Configuration of the bus participants
Choose **configuration | bus structure**. CMD changes into the function **bus structure**. With functions in the menu **work** you can configure a new INTERBUS network. Here you can insert a new slave, or check and compare the virtual network with the existing network.

Step 3: Addressing of the slaves for the program
Choose **configuration | addressing**. The window **addressing** opens and allows to address each input or output bit for the further processing in the PLC program. With a Step 5 PLC, the address of the FRABA encoder has to be higher than P128 (peripheral area), because the PLC overwrites the addresses below P127.

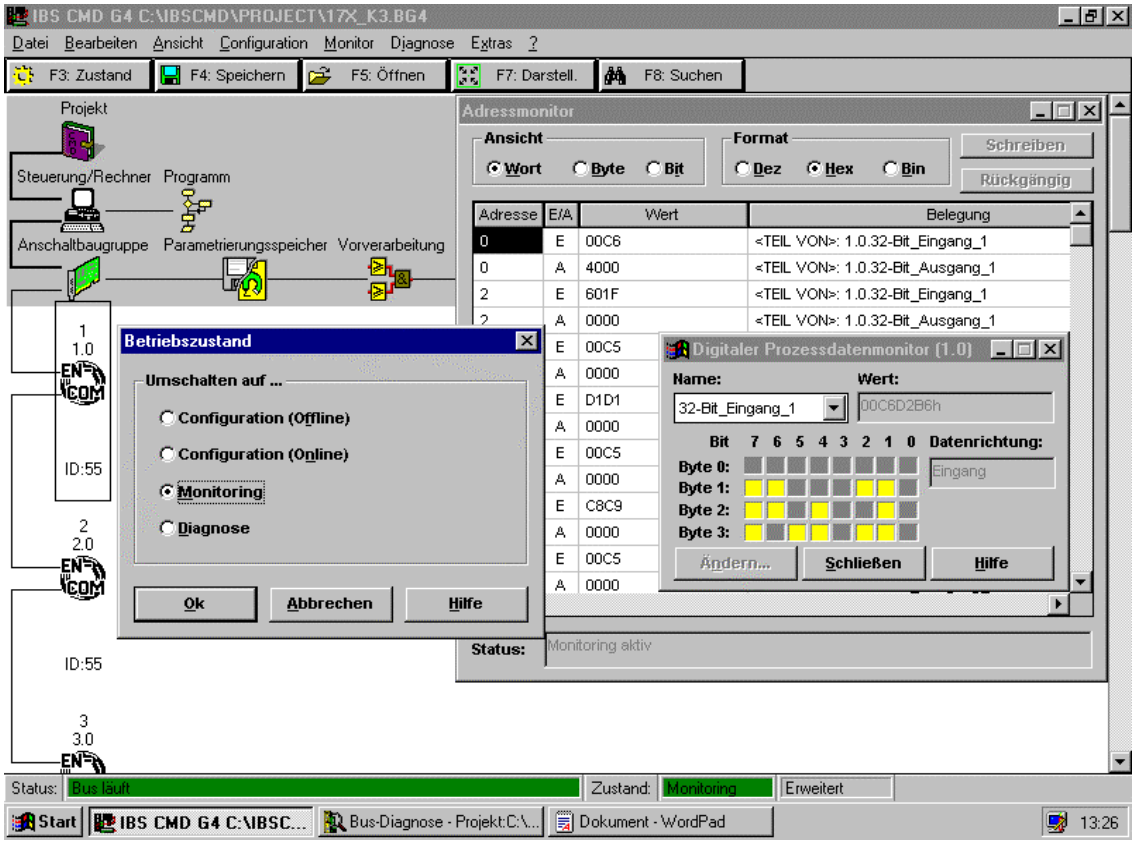
Step 4: Save and download the configured bus structure
Choose **bus structure | save**. Then choose **configuration | master**. The window **bus master** opens. The configured bus structure can be saved and downloaded in the master card.

4.4.2 Functions for getting started

For getting started with digital or analog devices you can use special monitoring tools delivered by the device manufacturer. So each device can be displayed, parameterized and checked with its specific software tool. The FRABA encoder can be parameterized simply with the CMD monitoring function.

4.4.3 Functions for diagnostics

The tool bus diagnostic (menu **diagnostic | bus diagnostic**) has the task to display and write down all messages coming from the bus master. All messages are displayed in a message window and are stored with date and time.



Screen shot of the CMD software with address monitor, digital process data monitor, and the operating mode window.

5 Programming of the encoder parameter

5.1 Inserting a FRABA INTERBUS K3 encoder

To insert a FRABA INTERBUS K3 encoder in an existing bus structure you have to go through the following steps:

- Choose **configuration | bus structure** to insert a new bus slave.
- Choose **configuration | insert with slave description**, click on ENCOM remote bus K3 encoder (RB_K3, ID code 36h = 55d).
- Insert further bus participants

5.2 Preparing the parameterization

Using the CMD software, the parameterization of an encoder can be done with a minimum of work. Please follow these steps:

- Click on the FRABA K3 encoder
- Choose **monitor | digital monitor**; the monitoring window "K3 encoder" opens
- Choose monitoring mode "32 bit input"
- Click on the FRABA K3 encoder again
- Open a further monitoring window
- Choose monitoring mode "32 bit output"
- Choose hex to display both the input and the output data. Each bit can be changed by a click in the window.
- The **writing / downloading** of parameters on the encoder is done with the menu **process data | writing** or by writing a series of hex data in the monitoring window.

Input and output monitors for the parameterization are ready now. The direction of the data is seen from the bus, i.e. "in" data are data from the encoder to the bus / bus master, "out" data are sent from the master to the encoder.

The 32 bit word in the process channel has the following meaning:

	31	30	29	28	27	26	25	24..0
Meaning	validity of position value	status of parameterization	manufacturer specific	Parameter- or error code				Parameter

In the direction master to encoder the device command word has the following meaning:

Master to encoder	31	30	29	28	27	26	25	24..0
Parameterization	0	0	0	Parameter code				Parameter data
Start	0 --> 1	0	0	0				0
Initializing of the preset	0	1	0	0				0

The programmed preset is initialized by setting bit 30.

The status of the device is displayed in the status word (bit 29 ... 31), direction encoder to master.

Encoder to master	31	30	29	28	27	26	25	24..0
Run time	0	0	X	0				
Parameterrization	1	1	X	Parameter code				
Error	1	0	X	Error code				

5.3 Parameterization

An encoder with implemented K3 profile is programmable with the following functions: steps per revolution, counting direction, preset and zero point shifting. These functions are started by special parameter codes sent from the master to the encoder. The encoder is able to send the position value, the parameter confirmation and error codes to the master. The following table shows the meaning of the codes: **ATTENTION:** Please note that for the standard INTERBUS encoder, the number of storing cycles are limited to about 200. The parameterization should not be done in a cyclic way, for example with each power on of the machine. After a maximum storing number of 500, a correct saving of the parameters can not be guaranteed.

In the status parameterization the bits 25 ... 28 have the following meaning:

Parameter code	28	27	26	25	Function
	0	0	0	0	Output of the position value while run time
	0	0	0	1	Subparameter: Steps (see also subparameter: revolutions, these 2 parameters build the gearing factor).
	0	0	1	0	Subparameter: Revolutions
	0	0	1	1	Counting direction
	0	1	0	0	Preset-value
	0	1	0	1	Zero point shifting

Additionally to the parameter codes the parameter data (bit 0 ... 24) are sent to the encoder (e.g. steps = d4095 = h0FFF).

For the counting direction the following meaning is given:

	28	27	26	25	Function
CW counting direction	0	0	1	1	0 0000 0000 0000 0000 0000 0011
CCW counting direction	0	0	1	1	0 0000 0000 0000 0000 0000 0100

In status error the bits 25 ... 28 have the following meaning:

Error code	28	27	26	25	Function
	0	0	0	0	No error
	0	0	0	1	Invalid parameter from master
	0	0	1	0	Invalid parameter code
	0	0	1	1	**Parameters lost**

The whole parameterization is proceeded with the following handshake:

	Master to encoder			Encoder to master			Meaning
	Command word		Data word	Status word		Data word	
	31	28 ... 25	24 ... 0	31 ... 30	28 ... 25	24 ... 0	
1	0	0	X	0 0	0	Position value	Normal running
2	0	P-Code	Parameter	0 0	0	Position value	Master sends parameter to encoder, encoder does not react
3	0	P-code	Parameter	0 0	0	Position value	Master waits for quitting of encoder
4	0	P-code	Parameter	0 0	0	X	Encoder takes over parameters and starts processing
5	0	P-code	Parameter	0 0	0	X	Processing of parameters in the encoder
6	0	P-code	Parameter	1 1	P-code	X	Processing of parameters in the encoder ready, encoder stays in status "parameterization"
7	1	0	0	1 1	P-code	X	Command "run" from master to encoder, encoder doesn’t react
8	1	0	0	0 0	0	Position value	Encoder is running
9	0	0	0	0 0	0	Position value	Normal running of master and slave

Repeat steps 4 to 6 for the transmission of several parameters.

In the following table you see an example of parameterization (all numbers are given in hex). An encoder is set which has 255 steps per turn, counting direction clockwise and a preset of 0. The preset is initialized with bit 30 (see above) at any place.

	Master to encoder	Encoder to master	Meaning
1	0 0 0 0 0 0 0 0	X X X X X X X X (Position value)	Normal running
2	0 2 0 0 0 0 F F (Parameter code: steps, parameter: d255 = h0FF)	X X X X X X X X (Position value)	Master sends parameter to encoder, encoder does not react
3	0 2 0 0 0 0 F F	X X X X X X X X (Position value)	Master waits for quit of encoder
4	0 2 0 0 0 0 F F	C 2 0 0 0 0 F F	Encoder takes over parameter and starts processing
5	0 2 0 0 0 0 F F	C 2 0 0 0 0 F F	Processing runs in encoder
6	0 2 0 0 0 0 F F	C 2 0 0 0 0 F F	Processing of parameters in encoder is ready, encoder stays in mode "parameterization"
4	0 4 0 0 0 0 0 1 (Parameter code: turns, parameter: d1 = h1)	C 2 0 0 0 1 0 0	Encoder sends old confirmation
5	0 4 0 0 0 0 0 1	C 4 0 0 0 0 0 1	Processing runs in encoder
6	0 4 0 0 0 0 0 1	C 4 0 0 0 0 0 1	Processing of parameters in encoder is ready, encoder stays in mode "parameterization"
4	0 6 0 0 0 0 0 3 (Parameter code: direction, parameter: d3 = h3)	C 4 0 0 0 0 0 1	Encoder sends old confirmation
5	0 6 0 0 0 0 0 3	C 6 0 0 0 0 0 3	Processing runs in encoder
6	0 6 0 0 0 0 0 3	C 6 0 0 0 0 0 3	Processing of parameters in encoder is ready, encoder stays in mode "parameterization"
4	0 8 0 0 0 0 0 0 (Parameter code: preset, parameter: d0 = h0)	C 6 0 0 0 0 0 3	Encoder sends old confirmation
5	0 8 0 0 0 0 0 0	C 8 0 0 0 0 0 0	Processing runs in encoder
6	0 8 0 0 0 0 0 0	C 8 0 0 0 0 0 0	Processing of parameters in encoder is ready, encoder stays in mode "parameterization"
7	8 0 0 0 0 0 0 0	C 8 0 0 0 0 0 0	Command "run" from master to encoder, encoder does not react yet
8	8 0 0 0 0 0 0 0	X X X X X X X X (Position value)	Encoder in status running
9	0 0 0 0 0 0 0 0	X X X X X X X X (Position value)	Both master and encoder are in status running, encoder sends new calculated position value

The preset ("0" in the example) is initialized with the following sequence:

1	0 0 0 0 0 0 0 0	X X X X X X X X (Position value)	Encoder sends position value
2	4 0 0 0 0 0 0 0	X X X X X X X X (Position value)	Master sends parameter to encoder, encoder does not react
3	4 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Encoder sends new calculated position value
4	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Both master and encoder are in status running, encoder sends new calculated position value

5.4 Manufacturer-specific functions

The FRABA INTERBUS encoder offers a multiplicity of manufacturer-specific functions, which are suggested by the K3-Profil, but not supported by the profile or other manufacturers.

5.4.1 FRABA Preset

To set the preset-value in the K3-profile, first a preset-value must be programmed, this value can then be actived by sending “active zero point shifting”. However each preset-value can only be activated once. If one wants to activate the same preset-value a second time, one must first program the value again. With the command “FRABA Preset”, a once programmed preset-value can be activated arbitrarily. It is to be noted that due to the nature of the INTERBUS (a command transmitted by the SPS once is repeatedly transferred with every buscycle) after transmitting the command “FRABA Preset” another command (e.g. “Run”) should be sent.

5.4.2 Velocity mode

In the velocity mode the FRABA rotary encoder no longer outputs its current position value, but rather the angular velocity in revolutions per minute.

5.4.3 Read-Out mode

In this mode all parameters, all registers of the INTERBUS chip Supi3-Opc and, if the encoder is equipped with a temperature sensor, the temperature level may be read out via the INTERBUS.

5.4.4 Cam mode

The FRABA INTERBUS encoder also offers integrated cam functionality. The command used to switch the encoder to Cam mode is (in hexadecimal notation) 2080000xh[1], whereby the x stands for the desired Cam program. The encoder offers 8 programs (x of 1-8) with 8 available cams each. With the command 20800000h the encoder is switched back into the position mode.

In each of the 8 cam programs up to 8 cams may be activated, deactivated and programmed independently. In order to activate or deactivate cams the instruction 2040xxxxh is used, whereby xxxx corresponds to the desired cam configuration. Each place in binary code corresponds to a cam, if e.g. the cams 0,1,4 and 7 should be active and all other cams should be inactive, the resulting command is 20400093h. The binary value of the last 2 places of the hexadecimal command is 1001 0011b. One can clearly see that the bits which are set correspond to the desired cam selection. This configuration may be read out via the bus, by using the command 20200000h. In response to this command the encoder sends an output word, which contains a one in each of the last 16 bits, if the appropriate cam is activated, otherwise the bit contains a zero.

[1] Hexadecimal values are denoted by a ’h’ immediately following the number, binary values by a ’b’.

The parameterization of the values, used to define the cam is also made by an output word to the INTERBUS containing: the instruction (either 010b, switching on value, or 011b, switching off value), the cam identifier (4 bits, 0-7 - > 0000b to 0111b) and the 25 bit position value, at which the cam is switched on respectively off. For example if cam 3 should be activated from 15h to 213h the parameterization is done by the following sequence of instructions:

Binary value (Bit 31 = MSB first)	Hex value	Meaning
00100000100000000000000000000001b	20800001h	Switch encoder from position output mode to cam program 1. All following instructions refer thereby to cams in the program 1.
01000110000000000000000000010101b	46000015h	set start value of cam 3 to 15h
01100110000000000000001000010011b	66000213h	set stop value of cam 3 to 213h
00100000001000000000000000001000b	20400008h	Activate cam 3, deactivate all other cams.
10000000000000000000000000000000b	80000000h	Switch encoder from parameterization to cam mode

As soon as a parameterization takes place the encoder is switched into the parameterization mode. In this mode the commands send via the bus are acknowledged. This mode is terminated with the run command (80000000h), which switches the encoder back into cam mode. To check the start and stop values they can be read out with an additional command.

While the encoder is in cam mode, the last 8 places (bit 0-7) correspond to the 8 cams. If the encoder position is in the area of cam 5 and cam 5 is activated, then bit 5 of the output word is set to 1. Cams that are not activated or switched off are denoted by a zero in the corresponding position of the output word. An error or a non valid cam value (e.g. because the encoder is in parameterization mode) is displayed by a 1 in bit 31 (MSB).

The user may also choose to display parts of the position value in addition to the cams. However only the bits 8 to 24 can be used. In order to ensure the maximum flexibility for the customer the position value can be shifted up within these 17 bits by up to 25 bits, so that according to the chosen value either the last 4 hexadecimal places of the position value (shift_iw to 0 set) or the first 4 places (shift_iw to 8 set) can be displayed. If one sets shift_iw to 25 then only the cams and no position value is shown. The variable shift_iw can be set via the bus with the command 201000xx and be read out with the command 20200040h. Switching between the different cam programs is possible at any time by transmitting the appropriate command. The parameterization of previously programmed cams will be saved to the EEPROM when leaving the program and read back when entering the program again.

RETURN- AND ERROR CODES

6 Appendix

6.1 Profile Functions

	binary																																hex
bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
steering bits and data bits	15	14	13	12	11	10	9	MSB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LSB	
Normal running	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0xxxxxxx
run!	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80000000
set presetvalue	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40000000
subparameter steps	0	0	0	0	0	0	**1**	parameter data																									02xxxxxx
subparameter turns	0	0	0	0	0	**1**	0	parameter data																									04xxxxxx
direction	0	0	0	0	0	**1**	**1**	parameter data																									06xxxxxx
presetvalue	0	0	0	0	**1**	0	0	parameter data																									08xxxxxx
activate zero point shifting	0	0	0	0	**1**	0	**1**	parameter data																									0Axxxxxx
set encoder to default (!)	0	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0E000000

6.2 Manufacturer specific Functions

		binary																																hex
bit position		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
steering bits and data bits		15	14	13	12	11	10	9	MSB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LSB	
FRABA Preset		0	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60000000
Velocity:	velocity start	0	0	**1**	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	20400001
	velocity stop	0	0	**1**	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20400000

RETURN- AND ERROR CODES

Manufacturer specific functions (contd.)

	binary																																hex
bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
steering bits and data bits	15	14	13	12	11	10	9	MSB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LSB	
Read out data: serial number	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20210000
steps	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	20210001
revolutions	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	20210002
complement	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	20210003
preset	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	20210004
zero point shifting	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	**1**	20210005
preset_flag	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	20210006
software version	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	20210007
operating time [10 min]	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	20210008
startup_mode [1]	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	**1**	20210009
flag_mt_stepsm1 [2]	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	**1**	0	2021000A
temperature [Celsius/10] [3]	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	2021000C
read out OPC-register	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	adress				**1**	**1**	0	**1**	202100xD
read out stop	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	xx0xxxxx

[1] answer of encoder: K2 = 0h, K3 = 1h, velocity mode = 2h, cam program 1 to 8 = 3h to 10h

[2] answer of the encoder to this command:

Bit 0 = 0: Normal-mode

= 1: steps_minus_1-mode (100% compatible to old Standard-INTERBUS-encoder)

Bit 1 = 0: Single-Turn-encoder

= 1: Multi-Turn-encoder

[3] as signed in

RETURN- AND ERROR CODES

6.3 Error codes

	binary																															hex
																1					1	1										
bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	6	15	14	13	2	1	10	9	8	7	6	5	4	3	2	1	0	
steering bits and data bits	15	14	13	12	11	10	9	MSB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LSB	
no valid position value is given out	**1**	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	8xxxxxxx
wrong parameter data	**1**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82000000
unknown parameter number	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84000000
parameters lost	**1**	0	0	0	0	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86000000
unknown parameter (in read out)	**1**	0	**1**	0	0	**1**	0	wrong parameter																									A4xxxxxx
manufacturer specific error code	**1**	0	0	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	980000xx

6.4 Miscellaneous return codes

	binary																															hex
																1					1	1										
bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	6	15	14	13	2	1	10	9	8	7	6	5	4	3	2	1	0	
steering bits and data bits	15	14	13	12	11	10	9	MSB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LSB	
normal running	0	0	0	0	0	0	0	position value																									0xxxxxxx
processing parameter	**1**	**1**	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Cxxxxxxx
parameter received(not checked yet)	**1**	**1**	0	P.-Nr.				parameter data																									Cxxxxxxx
Read out mode or auto test	**1**	0	**1**	0	0	0	0	read out value																									Axxxxxxx

INTERBUS CAM COMMAND WORDS

6.5 Cam functions

bit position	binary 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	hex
steering bits and data bits	15 14 13 12 11 10 9 MSB - LSB	
Normal running	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 x x x x x x x x	0000xxxx
run/exit cam-parameterization mode	**1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	80000000
set encoder to cam mode, program no. x	0 0 **1** 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 x x x x	2080000x
set encoder to position mode	0 0 **1** 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	20800000
set cam configuration	0 0 **1** 0 0 0 0 0 0 **1** 0 0 0 0 0 0 bit x=1: activate cam; bit x=0: deactivate cam	2040xxxx
set shift_iw	0 0 **1** 0 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 shift_value	201000xx
set start value for a cam	0 **1** 0 Cam-No. start position value	4xxxxxxx
set end value for a cam	0 **1** **1** Cam-No. end postion value	6xxxxxxx
read out cam configuration	0 0 **1** 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	20200000
read out start value for a cam	0 0 **1** 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **1** Cam-No.	2020001x
read out end value for a cam	0 0 **1** 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **1** 0 Cam-No.	2020002x
readout shift_iw-value	0 0 **1** 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0	20200030

6.6 Cam error codes

bit position	binary 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	hex
steering bits and data bits	15 14 13 12 11 10 9 MSB - LSB	
no valid cam value is given out	**1** 0 0 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x	8xxxxxxx
wrong parameter data	**1** 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	82000000
wrong command	**1** 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	84000000
parameters lost	**1** 0 0 0 0 **1** **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	86000000
manufacturer specific error code	**1** 0 0 **1** **1** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 x x x x x x x x	980000xx
Cam-start/stop-value higher than max. Position value	**1** 0 0 **1** **1** 0 0 0 **0** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **1** 0 0	98000004