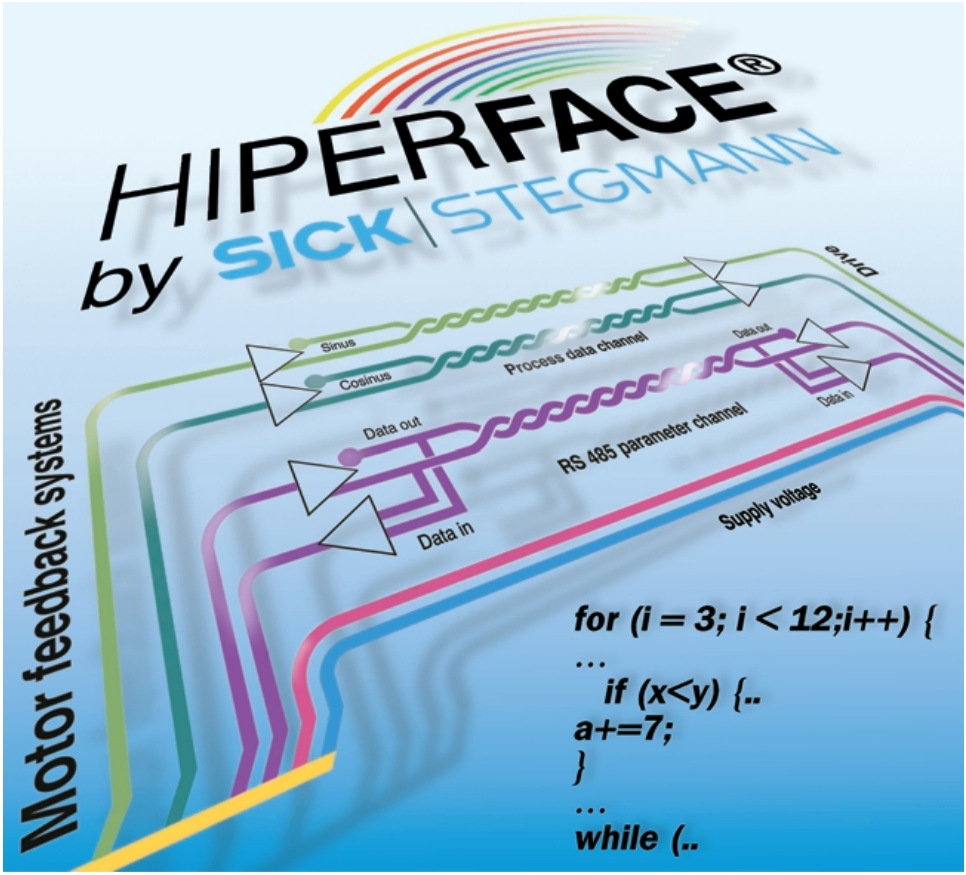


HIPERFACE®-Beschreibung

Description of the HIPERFACE® Interface

SICK|STEGMANN

Inhaltsverzeichnis

Contents

SinCos® Motor Feedback with HIPERFACE®?

HIPERFACE®

The HIPERFACE® interface has been established on the market for several years, featuring significant savings and, simultaneously, higher performance compared with conventional concepts. Conventional concepts involve the use of several encoder systems supporting the various controller functions:

- A tachogenerator to provide “speed” data
- A magnetic or optical commutating encoder for block commutation
- A final positioner for overlaid position control.

The advantage of these systems is their easy implementation due to the availability of the appropriate interface components and the low complexity of the individual components.

The disadvantage, however, is not only that the encoder systems themselves cost money. The fitting of three systems and their cabling lead to high costs and the number of components also means increased likelihood of failures.

The potential savings associated with compact low-power controllers are, of course, a major consideration with some relevant aspects being:

- the design of the commutating encoder with Hall sensors
- omitting the final positioner, if not absolutely essential
- replacing the tachogenerator by “Sensorless Control” software concepts

These savings, however, are always balanced against performance limits of the overall system.

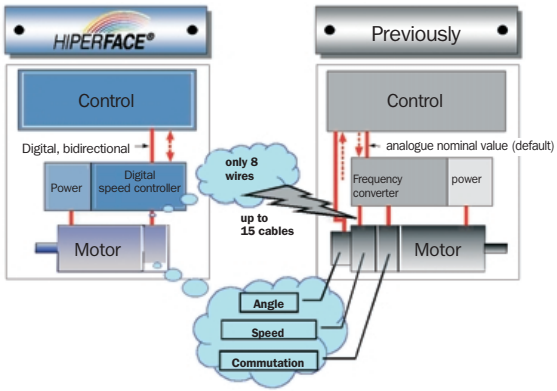


Fig. 1

The knowledge of these requirements and the wish to provide all these functions in just one device triggered the development of the innovative motor feedback by SICK-STEGMANN, equipped with a novel interface:

HIPERFACE®

It is short for **Hi**gh **Per**formance Inter**face** and is the standard interface for SICK-STEGMANN motor feedback systems.
This interface was specially developed for the requirements of digital drive control and offers the user standardised and simplified mechanical and electrical interfaces.

HIPERFACE® motor feedback – Overview of the most important features:

- Only one electrical interface on the speed controller for all applications and only one type of cable between speed controller and motor feedback system
- Implementation of both low-end and high-end applications with only one electrical interface
- Hybrid interface from
  - the analogue process data channel on which sine and cosine signals are transmitted differentially, with almost no delay
  - the bidirectional parameter channel corresponding to the RS485 specification for transmitting absolute position information and various other parameters
- Only 8 wires
- Electronic type label for identification of the motor feedback and storage of drive-related information in the motor feedback system
- Wide temperature range, high shock- and vibration resistance, immunity to electromagnetic interference and compact dimensions. The devices can be fitted inside the servo motor.
- Analogue sine/cosine signals are available for speed control. These enable both high resolution for use at low speeds and sufficiently low signal bandwidth for the control of high speeds.
- Cable lengths up to 100 m.
- Standardised mechanical interfaces.
- The encoder shaft has a rigid link to the motor shaft, enabling high amplification factors to be used.
- Easily fitted in the servo motor. Axial and radial tolerances are compensated via a flexible stator coupling (also: torque support).
- The relationship of the absolute value to the mechanical shaft position can be electronically manipulated to adjust the commutation.
- For the position control of mechanically geared applications, motor feedback systems with the same physical dimensions are also available as absolute multiturn units, for absolute positioning over up to 4096 revolutions.

Characteristics

Previously, optical encoders were classified as so-called incremental encoders or absolute encoders.

Incremental encoders are comparatively simple in design and scan a few tracks only. They offer relatively high resolutions and – due to their counting, real-time enabled interface (quadrature signals) – are well suited to the speed control of drives. Because the absolute position data is missing and position within a turn is only available after turning the shaft by up to 360° (locating zero pulse), they are unsuitable as commutating encoders.

Absolute encoders always transmit the entire position data and, therefore, are very well suited to position control. The digital transmission of the position requires a high transmission bandwidth in the cable and is only conditionally real-time enabled. Moreover, due to the major production effort involved in the conventional type, absolute encoders are relatively expensive. This conventional type reads binary data coded onto a glass disc, for which each binary digit requires a corresponding optical scan. All these scans must be adjusted to one another such that no reading errors can occur under any conditions.

HIPERFACE® motor feedback systems are a mix of incremental encoders and absolute encoders and combine the advantages of both encoder types.
Initially, the absolute value is only formed when the device is powered up and communicated – via the bus-enabled parameter interface according to the RS485 specification – to the external counter in the controller which, based on this absolute value, then incrementally counts on from the analogue sine/cosine signals.
The use of highly linear sine and cosine signals achieves the high resolution needed for speed control (the arc tangent is formated in the controller). Simultaneously, however, the signal frequencies to be transmitted remain relatively low. For example: 512 periods per revolution, even at a speed of 12,000 rpm, only produce a frequency of 102.4 kHz which can be easily transmitted, even over very long distances.

Internal arrangement

Typical arrangement of a HIPERFACE® motor feedback system

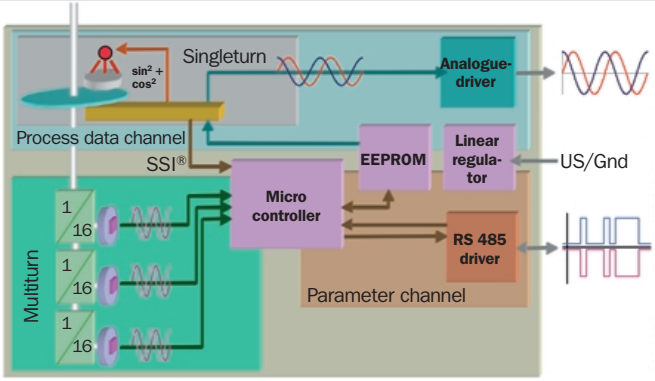


Fig. 2

As is common for standard encoders, most motor feedback systems have an internal glass code disc as a measuring scale . Using the direct-light method, an infrared diode maps the code pattern on the disc onto a fully customised highly integrated circuit.
The characteristic feature of the SinCos® motor feedback systems now is that, in addition to the absolute code track, there also is an incremental track producing sine and cosine signals.

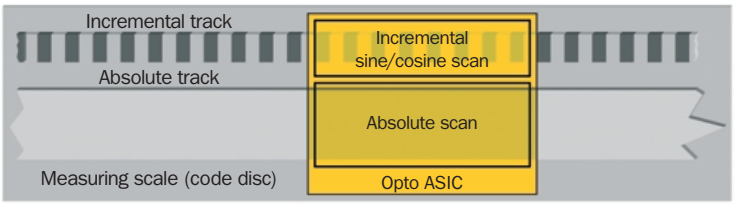


Fig. 3

Usually, in a motor feedback system, **2^n periods of sinusoidal and cosinusoidal signals** are scanned. These signals are always available via the so-called process data channel, as real-time data.
Parallel to this, the absolute position data is available via the parameter channel (RS485). It primarily serves to identify exactly one of these 2^n periods.

Processing the HIPERFACE® Signals in the Motor Controller

Controller Hardware

HIPERFACE® defines the standardised electrical interface with only 8 wires:
- 2x Supply voltage 7 … 12 V
- 4x Incremental, differentially transmitted, sine/cosine signals
- 2x Digital, bidirectional RS485 interface

A fully screened 8TP cable securely transmits the signals to the controller.

Supply voltage
At first glance it seems a disadvantage having to produce an additional, rather unusual supply of typically 8 V within the controller. This, however, ensures sufficient voltage supply to the HIPERFACE® motor feedback system even for long distances between drive and motor feedback system. More involved supplies, e.g. with sense cables, are not required.

RS485 interface
This interface can safely be described as industry standard. The physical interface requires a 130 Ω termination resistor, two biasing resistors and a standard RS485 transceiver. The protocol is handled with a standard UART, as implemented on almost all popular micro controllers/DSPs.

SinCos® interface
The sine/cosine signals are transmitted fully differentially, their amplitude varying no more than 20% in all circumstances. Since this interface determines the actual performance of the speed controller, this part of the interface must be carefully developed. Apart from differential input amplifiers with low offsets, low noise and high common-mode suppression, resistors and capacitors (low-pass) with low tolerance should also be used.
The area behind the analogue input amplifier also remains important since, here too, signal noise and crosstalk need to be kept low.
The signals lead onto the comparators for generating incremental counting signals and to a 2-channel, A/D converter which simultaneously captures the Sine and Cosine signals. The A/D converter should have a resolution of at least 10-bits.
Usually, it is not necessary to add additional external components for counting or analogue/digital conversion. Today, these functions are available in most micro controllers and DSPs.

Optional
To utilise the full functionality of the HIPERFACE® interface, a signal connection should additionally be provided between the internal RxD line and the capture input of the incremental counter. Thus, for instance in safety-related applications, it is possible to compare the counter readout with the absolute position, even at full speed (6000 rpm).
Many algorithms and/or hardware solutions are now available to perfect the speed value derived from the analogue signals, such as:
- Filtering
- Oversampling
- PLL method

The screen – the 9th line
The screen concept too is very important and has an influence on the achievable performance of the whole system. A large-area screen applied to both sides, motor and control, should achieve the best results.

Software

Serial protocol
The interface is configured to 9600 baud, as standard. In principle, the bidirectional RS485 interface is bus-enabled; therefore, each communication from the Master, i.e. the control, begins with the slave address. Binary data transmission minimises the transmission times. Each protocol is completed with an easy to calculate XOR checksum. The end of the protocol is detected using a timeout control.
The HIPERFACE® motor feedback systems have internal diagnostic functions and signal critical or faulty conditions monitoring within the response protocol.

The high resolution (arctan interpolation)
There are many known methods for calculating the high resolution angle. The following method has proved a good compromise regarding code size and speed of execution:
1. Reducing the calculation to the 1st octant (0...45°) by swapping and/or inverting the sine and cosine values, hence the following;
2. Calculation of the division sin (x)/cos (x) always has a result < = 1.
3. Table-based linearisation of the division result for approximation to the arc tangent function (32 bases approx. required).
4. Backward transformation and expansion of the result in the octants determined under 1.

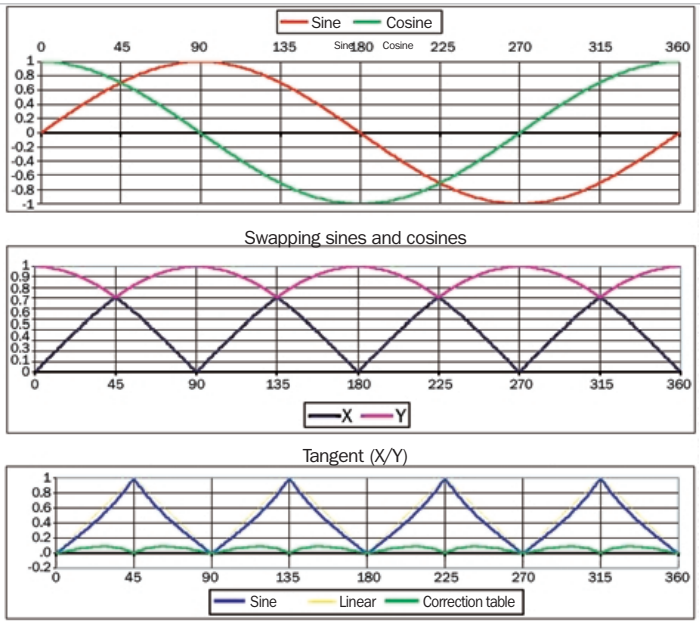


Fig. 4 Arc tangent interpolation method

Synchronisation

The overrun or the flanks on the various transmission paths will, in reality, never occur at the exactly identical point in time.
This can be explained by the following effects:

- the absolute position is determined via separate code tracks and, therefore, differs slightly from the position interpolated from the analogue signals (see figure 5). Therefore, this phase position can change by up to ± 4/32 of a period, within a turn.
- dynamic effects such as signal run times or different input filters
- hysteresis, e.g. within the comparator of the quadrature counter

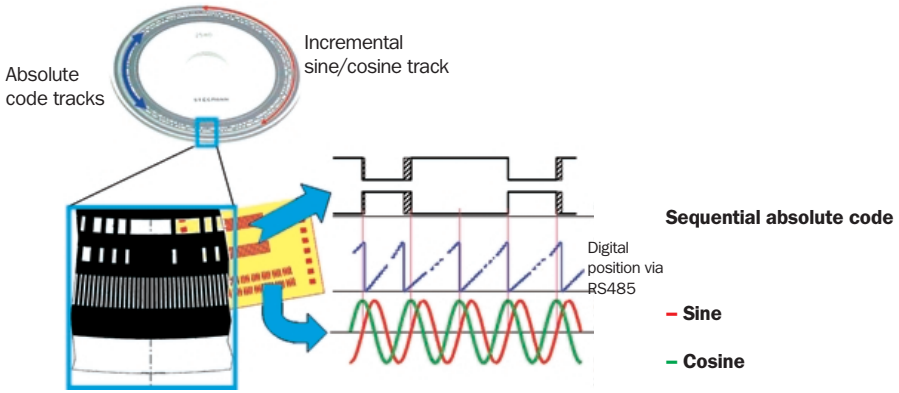


Fig. 5

The synchronisation now describes a method which can compensate these effects within wide limits (± 1/2 period). This makes it possible, even at high speeds, to assign the analogue signals to a specific period.

Consequently, the overlapping bits of two different pieces of position data are always oriented to the same valence, and the difference of the position values is then formed. If the two position values point to different periods, the position value with lower resolution is corrected in the direction of the smaller difference by ± 1 period.

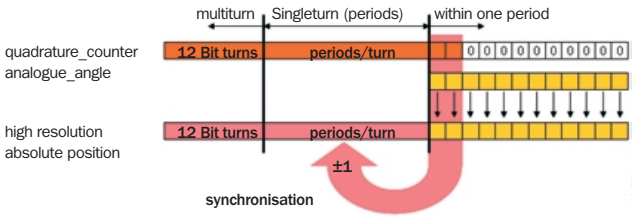


Fig. 6

quadrature_counter: the position value also within the controller with a resolution of 4* number of periods
analogue_angle: the angle calculated from the analogue signals within one period
absolute_position: the absolute position read out via the parameter channel

Commissioning/Commutation Adjustment Using a HIPERFACE® Motor Feedback System

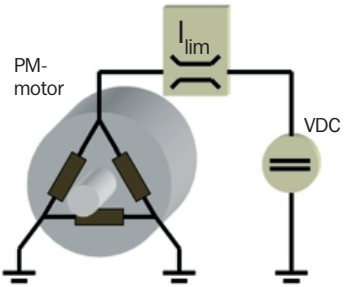


Fig. 7

When adjusting the motor commutation, one of the following methods is generally used:

1. Using the data fields
Using a static supply, a high-resolution position is calculated from the digital absolute position and the angle value interpolated from the analogue signals. This position value is then deposited in the permanent memory (EEPROM) of the motor feedback system, as a commutation offset. This method is universally applicable and permits the accurate setting of the commutation, even for high pole motors.

2. Using the MFB function “Set position“ (43h)
Using a static supply the control first captures the analogue signals (process data) and calculates the arc tangent. Then the function “Set position“ is called, where the 5 LSB of the position value must be set to match the 5 MSB from this interpolation. This commutation setting method can only be set with the resolution of whole periods of the motor feedback system. Thus, for motor feedback system with fewer periods, the setting of the commutation may be too inaccurate and hence limit the achievable performance of the drive system.

HIPERFACE® Feedback in the Control Cycle

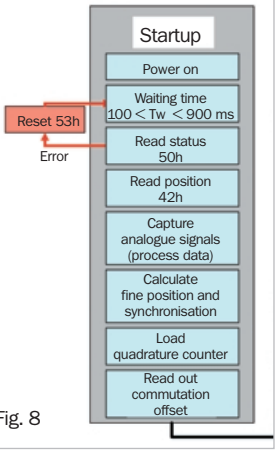


Fig. 8

Startup
Following power-on, in addition to the initialisation status of the HIPERFACE® motor feedback system, all operating parameters of the motor/controller combination can also be read out from the EEPROM, and the controller is parametrised – using the electronic type label – to match the technical data of the motor feedback system.
Then the absolute position is read out and, thus, the commutation position and the position of the axis, if required, are determined.
For the first control cycle, the analogue signals are then captured, and the incremental counter is initialised with the arc tangent calculated from this.

Speed controller
The highly linear, sinusoidal analogue signals permit a high interpolation factor and thus a high resolution of the speed. With almost no effort, this enables a wide speed adjustment range of 1:10,000.

Position controller
As regards the interface to the position controller, different concepts have already been implemented. There are solutions in which the position controller, being the second RS485 Master, directly accesses the interface of the HIPERFACE® motor feedback system to cyclically interrogate the position. More frequently, however, the incremental counting signals formed by the frequency converter are fed through to the positioning control. The initial position is thus counted out by the controller. Just as often, one finds a time-discrete implementation in which the controller makes available the absolute position calculated in each cycle, in a serial (e. g. SSI®) output register.
A further big advantage of the implementation with a HIPERFACE® motor feedback system is its availability as a multiturn device. This enables the application without limit switches.

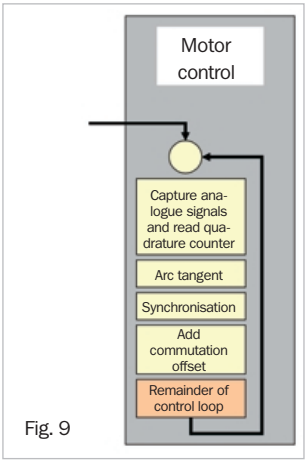


Fig. 9

Safety concepts

Although a HIPERFACE® motor feedback system is only ONE device, it does meet, due to the redundant design (not only of the interface), high safety requirements.
The position values are transmitted to two physically separated interfaces (analogue and RS485). The analogue transmission channel already has high intrinsic safety, since the signals' validity can be checked in the controller, within tight limits, e.g. using amplitude monitoring and Sin^2+Cos^2 monitoring. Mutual monitoring of data is possible due to the physical separation of the data, without additional effort.
At the heart of the motor feedback system both pieces of data are scanned by a system consisting of transmitter, code disc and receiver. Hereafter, the analogue signals – without processing by a microprocessor – are only amplified and transmitted, while the absolute value is first digitally processed, monitored and then transmitted.
If the absolute position is also read out cyclically, reliable monitoring of the incremental counter is possible. This can also provide early detection if the HIPERFACE® motor feedback system reaches critical operational states.

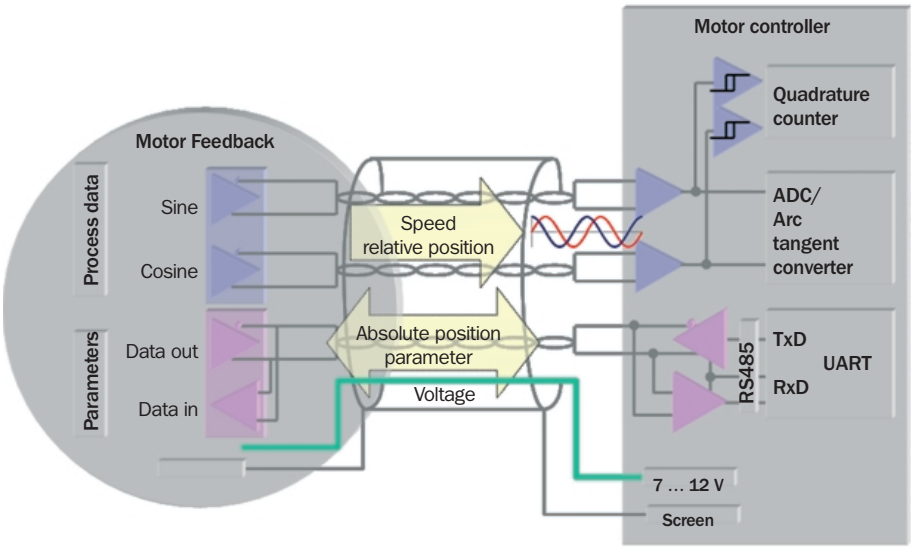


Fig. 10

General Specifications: HIPERFACE®
(if not otherwise stated on the data sheet)

Supply voltage	7 - 12 V
Current consumption	≤ 250mA
Signal specification for the process data channel	1 Vpp, differential
Signal specification for the parameter channel	according to RS 485
Interfaces-timing	see page 47 + 48
Data format for addressing and commands	see page 47 + 48

On the data sheet of a motor feedback system, there are different details regarding the error limits:
Error limits when evaluating the sine/cosine signals

- Differential non-linearity
  This value specifies the deviations within an electrical period of the process data channels and results from the sine/cosine signals deviating from their ideal shapes. Such possible error sources are, for instance, signal offsets, amplitude variations, phase errors or mapping errors.
  Above all, this parameter is relevant to the control of low speeds.

- Integral non-linearity
  This value describes the deviations over a mechanical rotation of the shaft.
  Deviations over a mechanical shaft rotation derive, amongst other factors, from the eccentricity of the code disc, from the transmission function of the stator coupling or from errors relating to the measuring scale.

Error limits for the digital absolute value
Added to the two above magnitudes of error, there are primarily internal quantification effects during calculation of the absolute position.
Moreover, the absolute position is determined, according to the type of encoder, from independent code tracks, which may have different error limits.

Schematic representation of the angular error over a shaft rotation

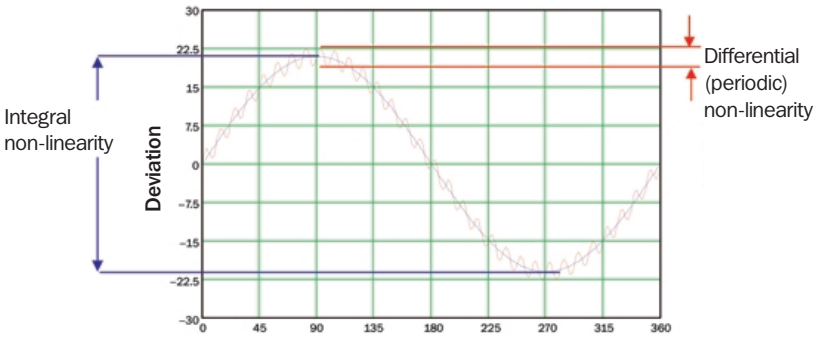


Fig. 11

Process data channel (analogue outputs)

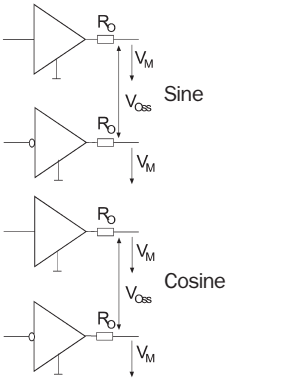


Fig. 12 Schematic diagram

Electrical parameters of the process data channels
(if not otherwise stated on the data sheet)

Signal bandwidth	0 … 150 kHz (3 dB signal amplitude)	f_{BW}
Average voltage of the individual signals	2.0 … 3.0 V	V_M
Signal amplitude peak-peak, **differential**	0.9 … 1.1 V	V_{Oss}
Output load capacity	>= ± 7 mA	I_{Omin}

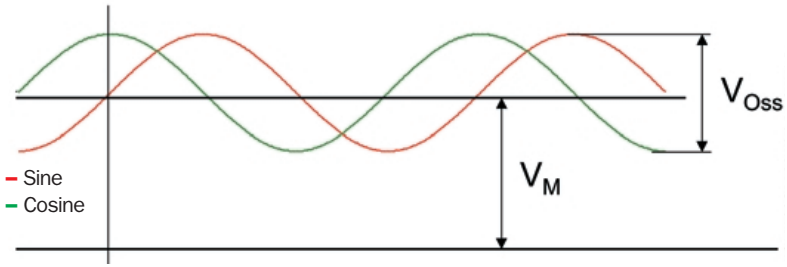


Fig. 13

Typical Input Circuit:

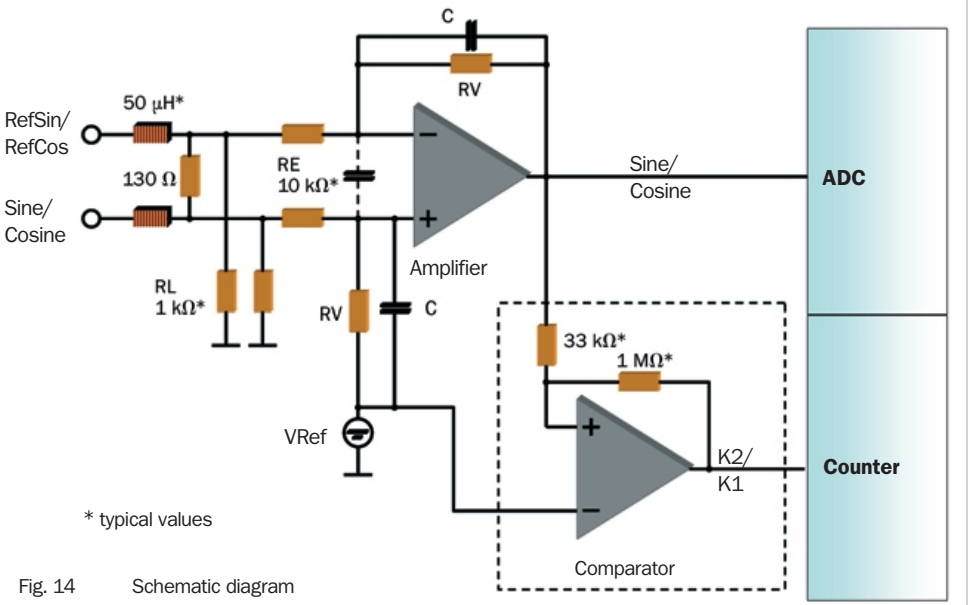


Fig. 14 Schematic diagram

The filters must be dimensioned according to the necessary bandwidth. To achieve good suppression of common-mode interference, check the PCB layout for symmetric arrangement of SIN and REFSIN.
The tolerance of the resistors used must be ≤ 1 %.

A typical dimensioning for sinusoidal input signals up to a frequency of 100 kHz is:

L	= 50 µH
RV	= determines, together with RE, the amplification (RV/RE)
C	= 22 … 47 pF
RL	= ≥ 1 kΩ (optional, additional signal loading in the zero passage of the differential signals)

The filter formed from RE and C reduces the signal noise on the analogue signals, but also determines the maximum signal frequency which can be processed.

Phase shift $\varphi \cong \arctan(\varpi * RV * C)$

Output amplitude $Ua \cong \frac{1}{\sqrt{1 + (\omega * RV * C)^2}} * Ue$

The circuit sketched here (comparator arranged behind the differential amplifier) ensures the safe switching behaviour of the comparator, since signal noise and common-mode interferences are already suppressed effectively by the differential amplifier. Moreover, the signals are available at the AD converter and at the counter (in phase), which simplifies synchronisation at high speeds.

The Parameter Channel

The parameter channel is an asynchronous, half-duplex interface which, physically, corresponds with the EIA RS485 specification.
To ensure interference-free transmission, it is necessary to provide the transmission lines with pull-up/pull-down resistors.
To also ensure independence of line transceiver manufacturers, we recommend dimensioning of the interface as follows:

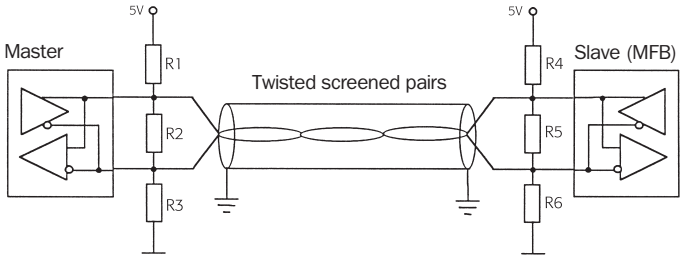


Fig. 15 RS485 wiring

Dimensioning for bus and single-ended

Mode	Master			Slave (Motor Feedback System)		
	R1	R2	R3	R4	R5	R6
Standard (single ended)	1K	130	1K	1K	130	1K
BUS (max. 8** subscribers)	*	130	*	10K	∞ not inserted	10K

* As HIPERFACE® motor feedback systems are always slaves and the number of subscribers in the BUS can vary, the BUS interface must be dimensioned by the customer.
** Please contact our distributor if there are more than 8 subscribers.

The 130 Ω termination resistor and the 1 kΩ pull-up/pull-down resistors are fitted as standard. For bus mode (more than 1 encoder) no termination resistor is fitted in the encoder; the customer must fit this to the subscriber furthest away from the Master.

Consequently, the Motor Feedback Systems must be ordered separately as “BUS” version.

The Transmission Protocol

Data Format

The data on the asynchronous interface has the following basic format:

- 1 start bit
- 8 data bits (LSB first)
- 1 stop bit
- Programmable parity: odd, even, none
- Programmable baud rate: 600, 1200, 2400, 4800, 9600, 19200, 38400 baud

* Note: The baud rate cannot be programmed for all MFB types. Please refer to the individual data sheets.

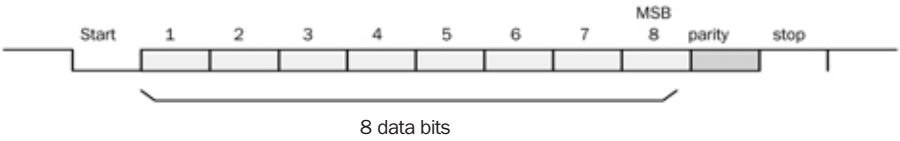


Fig. 16 General data format

When the devices are delivered, the parameters are usually preset to:
9600 baud and
parity = 1, if the sum of the data bits is odd

Dialogue Timing

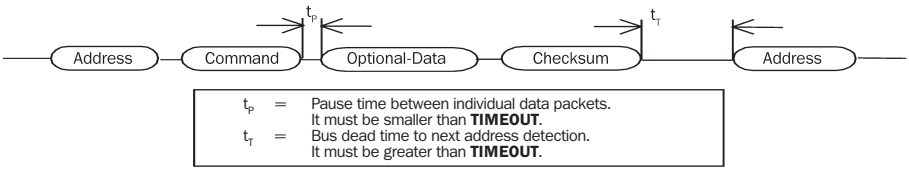


Fig. 17 Dialogue-Timing

Checksum – EXOR link of the bytes transmitted, including address and command
TIMEOUT – programmable, 11/baud rate or **44/baud rate = default**

The data transmission on the RS485 is controlled via a timeout protocol, i.e. the protocol currently received is processed only if – within a certain time (**Timeout**) – no further information arrives at the MFB.
Interruptions of a data packet by more than Timeout lead to corresponding protocol error messages (see also COMMAND 50h).
The 1st byte after a timeout is interpreted as an address.

As a rule, a HIPERFACE® motor feedback system does not occupy the interface of its own accord.

The HIPERFACE® Motor Feedback System always is a slave.

An exchange of information must always be initiated by the Master, the motor controller.

ADDRESS Format

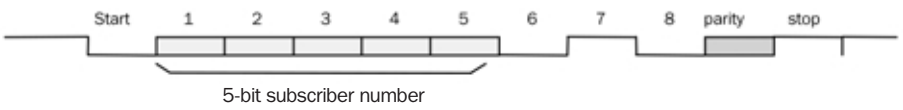


Fig. 18 Address format

ADDRESS is the subscriber number + 40h.

Broadcast Address FFh

Instead of the specific MFB address, the MFB can also be addressed with a BROADCAST, FFh. For this BROADCAST all bus-linked devices perform the selected function and acknowledge this with their specific addresses. Consequently, it must be ensured that commands resulting in an acknowledgement must only be executed with one MFB connected.

COMMAND-Format

Fig. 19 Command format

Error Handling

- Commands which cannot be processed (protocol errors, command arguments or internal MFB errors) lead to the command processing being cancelled and the MFB responding with an error protocol (see command 50h)
- During command execution, the operational state of the MFB is monitored (e.g. temperature, LED current). If this finds critical parameters (which, however, have no effect on the command currently executed) then the MFB generates the expected response protocol with the error bit set.
- The error bit is deleted by reading out the error code (command 50h) or by a MFB reset (command 53h).
- If several errors occur simultaneously, the MFB stores up to 4 error codes which, with the command 50h, can be read out successively.
  For this, the error bit remains set until all error codes have been read out.

An addressed MFB always responds to a faultlessly processed command sequence by repeating ADDRESS and COMMAND, followed by the requested data and the CHECKSUM.

Command Set

The command set, available for each motor feedback type, may vary.
The valid command set can be taken from the data sheet for the relevant motor feedback system.

Overview of the commands

Command byte	Function	Code 0[1])	maximum response time[2]) [ms]	Page
42h	Read position		10	48
43h	Set position	•	40	49
44h	Read analogue value		5	50
46h	Read counter		5	51
47h	Increase counter		30	51
49h	Delete counter	•	30	51
4Ah	Read data		30	53
4Bh	Store data		250	53
4Ch	Determine status of a data field		5	53
4Dh	Create data field		70	54
4Eh	Determine available memory area		5	54
4Fh	Change access code		40	54
50h	Read encoder status		5	56
52h	Read out type label		5	56
53h	Encoder reset		---	57
55h	Allocate encoder address	•	40	58
56h	Read serial number and program version		5	58
57h	Configure serial interface	•	40	59
63h	Set position with internal track synchronisation	•	40	60

1) The commands thus labelled contain the parameter “Code 0”. Code 0 is a byte inserted into the protocol, for additional safeguarding of vital system parameters against accidental overwriting.
When shipped, “Code 0” = 55h.

2) The response times given do NOT include transmission times and protocol timeout.
At the default setting of the RS485, the timeout is 4.7 ms approx.

All the functions listed can also be performed with the **programming tool** available as an **accessory**.

It is used for the individual parametrisation of HIPERFACE® Motor Feedback Systems and consists of:
- Programming adapter
- Link cable
- MFB cable
- Plug-in power supply
- Program CD-Rom

The Commands

Read position ➡ 42h

This function is the most important one in the whole system for setting the quadrature counter in the motor controller to the absolute value of the MFB. An important property of the MFB is that this synchronisation can even be performed at full speed without loss of information or control speed.

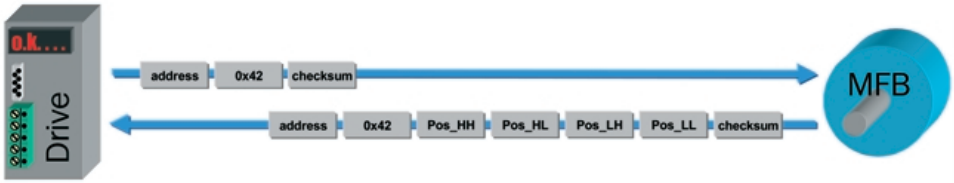


Fig. 20

In Pos_HH … Pos_LL, the position value is transmitted as **unsigned long** with right-justified LSB.
The first flank of the start bit of "ADDRESS" of the MFB response is used to synchronise the MFB position with the absolute value counted by the control.
To enable interference-free synchronisation, the RS485 driver is switched to active 2 ms approx. prior to the start of data transmission.
When executing this function, various diagnostic functions are performed in the MFB; if required, they are signalled as a warning or an error.

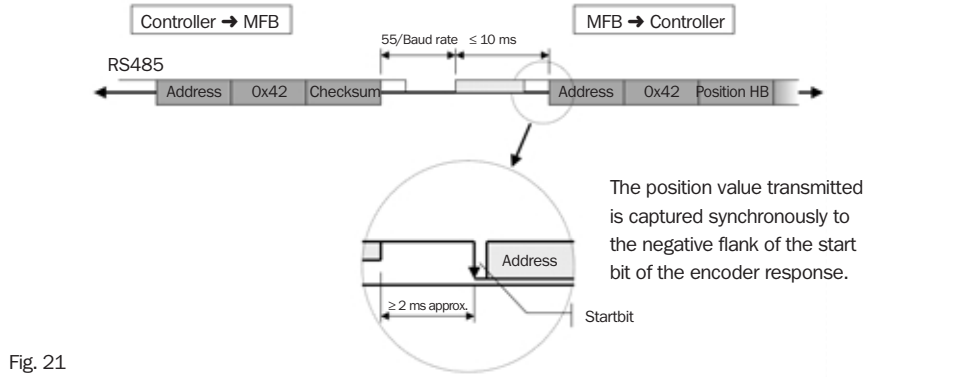


Fig. 21

The absolute position value serves to uniquely identify one of the analogue periods and, therefore, always has a resolution of 5 bits/period. To check the position counted within the controller, the absolute position of the MFB can be cyclically read out. Consequently, it is recommended to use only the position read out for verification of the count, but not for its correction. **A neatly arranged and screened system does not usually miscount**.
If the monitoring of incremental counting and position also needs to be performed at high speeds with higher accuracy, then the starting flank of the position response must be used for storing the counter readout (see figure).

Set position ➡ 43h

For synchronous motors, the absolute data per revolution is also used for the commutation of the motor. Since improper use will impair the function of the motor, **Code 0** must be given correctly.
This function should only be called by the motor manufacturer.

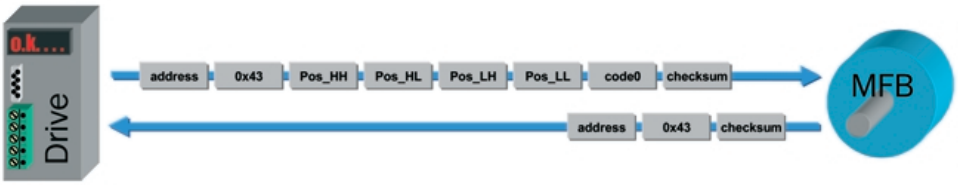


Fig. 22

With this function, the mechanical shaft position can be assigned any value within the measuring range. For this, the momentary encoder position is calculated internally, and the position offset calculated from this will be stored.
Caution:
As no synchronisation mechanism is provided for this, this function must be called **at standstill**.

CAUTION:
This function changes the assignment between the absolute position and the analogue signals.

When shipped, the phase position is defined according to the following diagram:

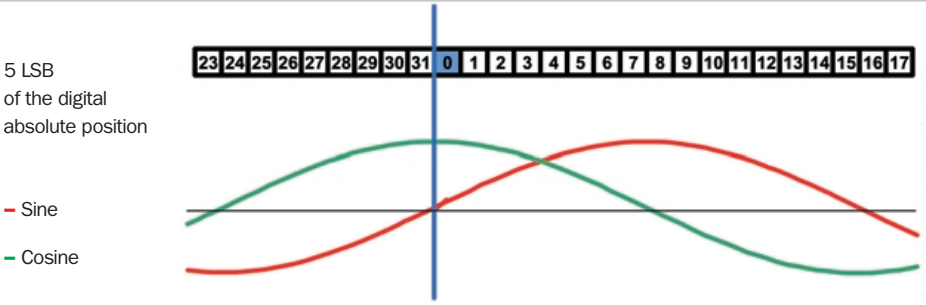


Abb. 23

To ensure maximum possible accuracy always when the motor controller is switched on, it is necessary for the assignment of the analogue signals to the digital position to be maintained in the manner shown since only then will the unambiguous identification of a period be possible.

Read analogue value ➡ 44h

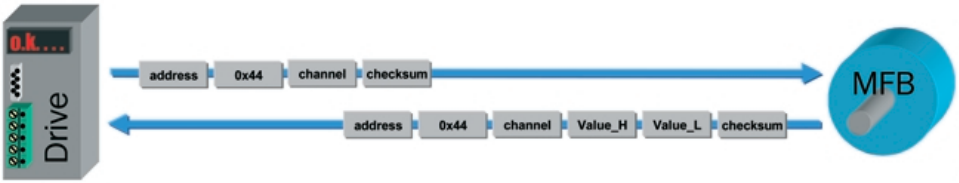


Fig. 24

The analogue signal selected with “CHANNEL“ is digitally converted and is usually sent back right-justified, signed with a resolution of 10 bits/5 V (SIGNED INT) in Value_xx.

The available analogue signals, their encryption and resolution are always given on the data sheet of the individual device.

Counter Functions

An incremental 24-bit counter is implemented in the encoder firmware.
The commands listed below permit simple implementation of a pseudo hour meterr. Counting the power-ups can also be implemented just as easily.

Consequently, ensure that the counting register in the EEPROM of the encoder is administered, whereby the number of counts is limited to a maximum of 1 million approx.

Read counter ➡ 46h

Fig. 25

Reads out the current count.

Increase counter ➡ 47h

Fig. 26

The internal 24-bit counter is increased by 1. Any overflow is shown by the ERROR bit.

Delete counter ➡ 49h

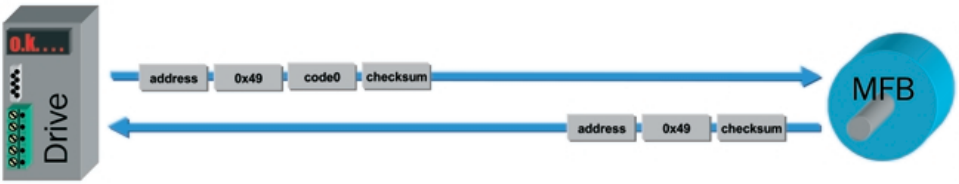


Fig. 27

The counter is set to “000“. CODE 0 must be given to protect from accidental deletion.

Data Storage and Administration

In many applications it is useful to be able to deposit parameterisation data of a motor in the MFB fitted. This, for instance, makes it much easier to exchange a motor.

To store data in the MFB, the EEPROM must first be divided into individual data fields with block sizes between 16 and 128 bytes (command 4Dh). This also defines the assignation to one of the 4 access codes deposited in the MFB. As regards the access codes (default = 55h), they can be modified with the function “4Fh“.
Furthermore, each data field can be separately write-protected.

Essentially, access to a data field is determined by three quantities:
1. Data field number, 1 byte, 0 … x
2. Start address, 1 byte, always within a data field of 0 … n
3. Amount of data to be written/read, 1 byte, 1 … 128

Accordingly, the following errors are detected and acknowledged with an ERROR message:
- wrong access code
- non-initialised data field (the given field number is too large)
- the address range of the data field selected, is exceeded

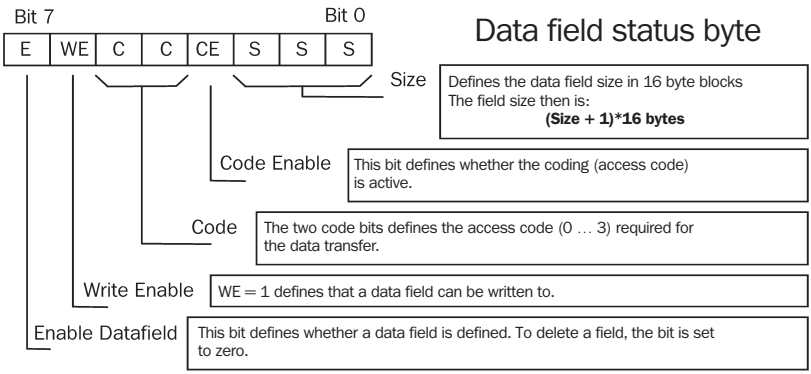


Fig. 28 Data field status byte

Read data ➡ 4Ah

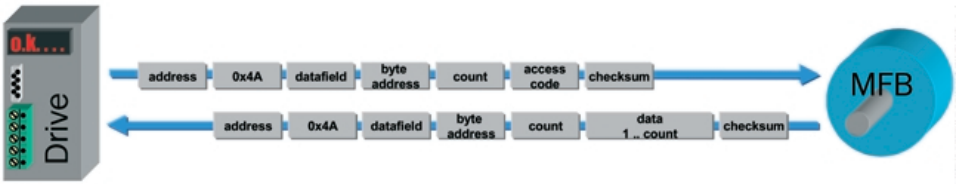


Fig. 29

From the data field with the number **Datafield** and the access code **Access Code**, the number **count** of data is read out and transmitted from the address **byte address** on. The code is ignored if the CE bit (see also read data field status, command byte 4Ch) is not set.

Store data ➡ 4Bh

Fig. 30

When writing to a data field: in addition to the access authorisation and the address range overrun, the WE bit (Write Enable, see also read data field status, command byte 4Ch) in the status word of the data field definition is also monitored. The response time of the MFB depends on the number of data to be programmed and is 250 ms max.

Determine status of a data field ➡ 4Ch

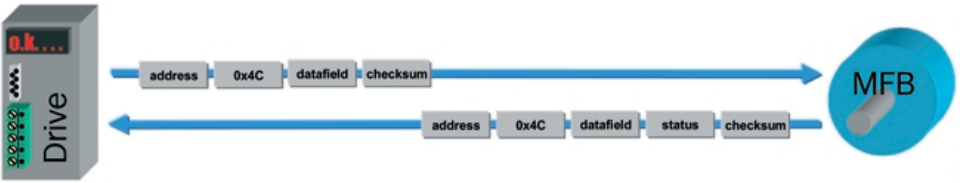


Fig. 31

This enables the determination of characteristic quantities and access modes of a data field already defined.

Create a data field ➡ 4Dh

With this function, the data fields in the EEPROM of the encoder are created. The data fields must be created in ascending order, starting with the data field number 0.

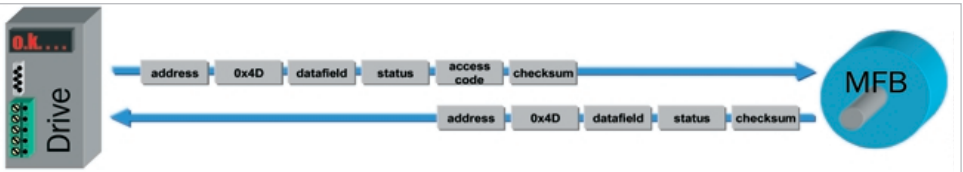


Fig. 32

Irrespective of the **CE** bit, new definitions of data fields are only possible by giving the access **Code**. For this, note that a change of the field size is only possible for the field created last (highest data field number).
The same applies to deleting a data field (**E** = 0)

Determine available memory area ➡ 4Eh

Fig. 33

The available memory area can be determined using this function. The parameter **free memory** gives the size of unallocated memory and is shown, according to the data field definition, as the number of free 16 byte blocks.
In addition, the **number of datafields** already defined, is output.

Change access code ➡ 4Fh

Fig. 34

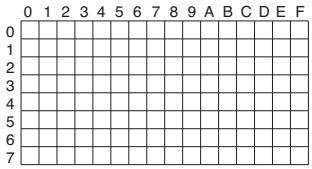
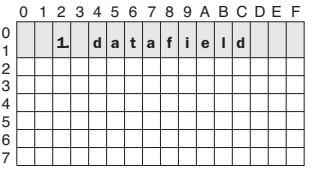
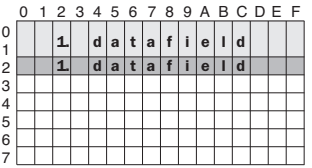
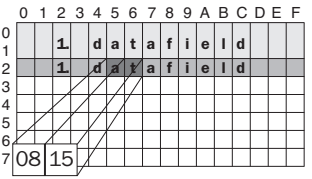
For customers, the motor manufacturer can define access words for certain memory areas. Data such as date of purchase, services, project numbers etc. can be deposited in these.

An access code can only be newly defined or redefined by giving its current value (**Old_Code**). When shipped, the four possible codes (**Code_Nr** = 0 … 3) are pre-occupied with the value **55h**.

Example: Data Field Definition

Starting set-up

- Address 40h
- Codes 0 … 3 55h
- free EEPROM 128 bytes

Communication	Map of the EEPROM	Explanation
(all numbers in HEX!) 1st line: command 2nd line: MFB response		Determine available memory area • 8 x 16 bytes free EEPROM • 0 (no) data fields created
40 4E 0E 40 4E 08 00 06		
40 4D 00 C9 55 91 40 4D 00 C9 C4		Create 1st data field • Number 0 • Size 32 bytes • Use code 0 • Write access allowed • Code must be given ➧ Status byte = **C9**h
40 4D 01 E0 55 69 40 4D 01 E0 EC		Create 2nd data field • Number 1 • Size 16 bytes • Use code 2 • Write access allowed • Code must be given ➧ Status byte = **E0**h
40 4F 02 55 66 3E 40 4F 02 0D		Change access code • Code no. 2 • previously 55h • new 66h
40 4B 01 05 02 66 08 15 76 40 4B 01 05 02 0D		Save data • in the 2nd data field (no. 1) • at address 5 • 2 bytes Data • with Code 2, (66h) • data value (08h, 15h)

Read encoder status ➡ 50h

Fig. 35

With this function, the current encoder state can be interrogated (e.g. after power-on).

This function is also needed if, when executing a command, the ERROR bit has been set, the actual command however having been processed error-free (pending error). For example, this is the case when the temperature monitoring has detected a range overrun.
The actual available errors vary according to MFB type and are always given on the data sheet.

Read out type label ➡ 52h

With this command, the current configuration of the interface such as baud rate and parity as well as the hardware configuration of the MFB are given.

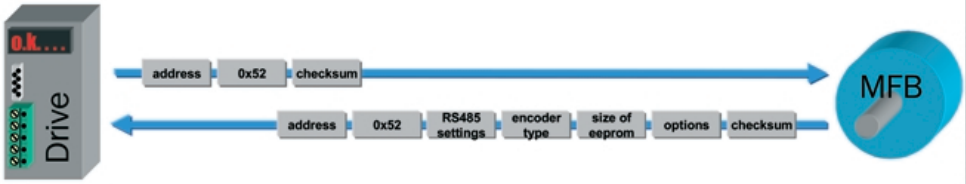


Fig. 36

- **RS485 settings**
  see command “57h“ (see p. 59)

- **Encoder type**
  Type detection, see product specific data sheets
- **Size of EEPROM:**
  The size of EEPROM used in the MFB = EEPR_Size * 16

- **Options**
  specific hardware/software upgrades
  HIPERFACE® version

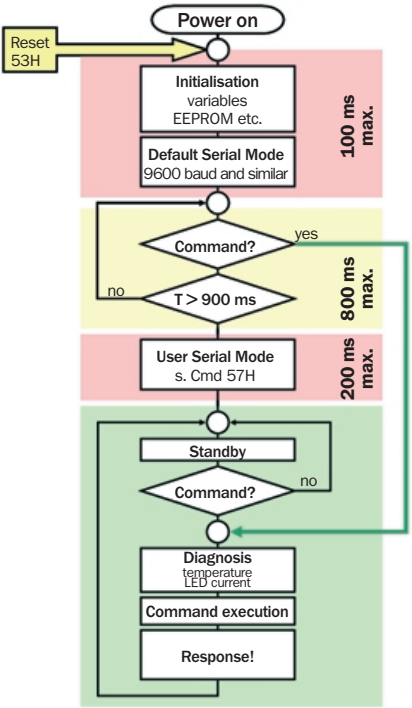
Reset ➡ 53h

Fig. 37

No MFB acknowledgement

Here, the MFB runs through the same initialisation process as it does when the supply voltage is switched on.

The parameters previously changed by the user, such as the MFB address, are retained!



CAUTION:
No RS485 communication is possible during the phases highlighted in red!

Fig. 38

Allocate address ➡ 55h

Allocation of a new address to a subscriber. After executing this command, the encoder must be addressed with its new address **New_Address**.

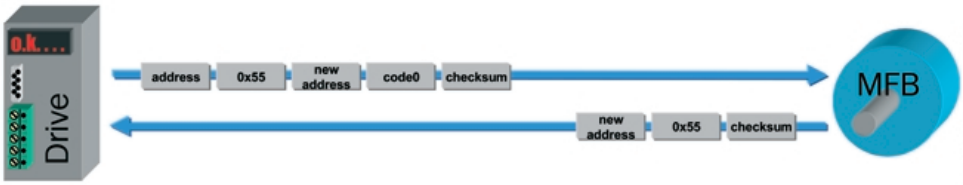


Fig. 39

Read serial number and program version ➡ 56h

With this function, the 9-digit serial number of the MFB, as well as version and date of the internal program, are read out.

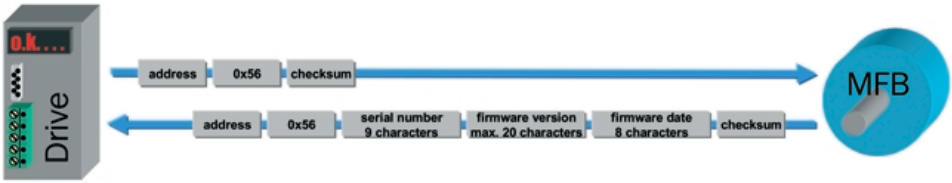


Fig. 40

The program version “**firmware version**“ consists of **up to** 20 characters.
The 8-digit “**firmware date**" specifies the time of program creation and is given in the "DD.MM.YY" format.

Configure serial interface ➡ 57h

With this function, the transmission parameters of the asynchronous parameter channels can be adapted to the requirements of the application, within wide limits.

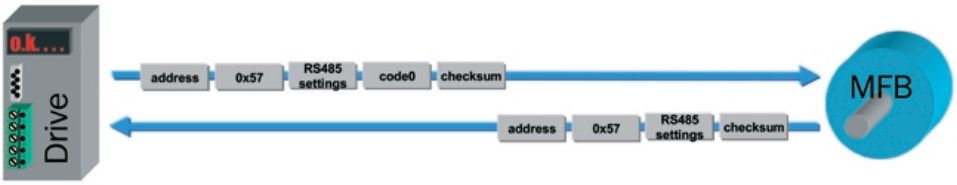


Fig. 41

Structure of RS485 settings

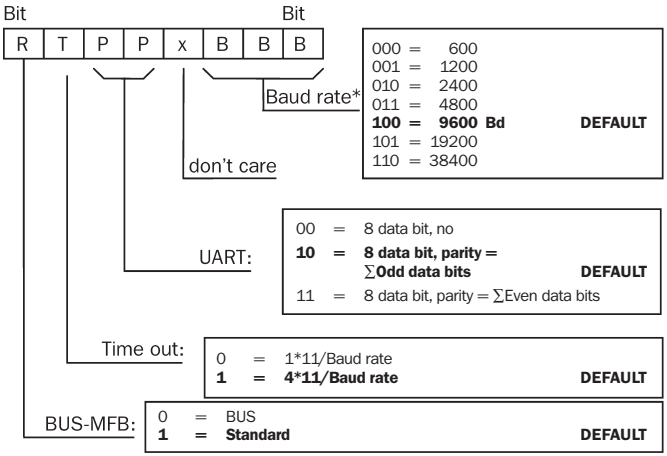


Fig. 42 RS485 settings

* Note: The baud rate cannot be programmed for all MFB types. Please refer to the individual data sheets.

Basically, each HIPERFACE® MFB, after power-up, is set to the default values for a time of 1 second approx. If, during this second, a plausible piece of information arrives, this setting is retained.
If this is not the case, the setting defined via this function is adopted. This procedure is necessary to be able to communicate with devices whose setting is unknown.

The baud rate set as well as the definition of the parity only become active through an ENCODER RESET (53h) or switching the encoder on and off.

Set position with internal synchronisation ➡ 63h

For particular applications there are also encoders without their own bearings, so-called KITs, in the SICK-STEGMANN product portfolio. For this, the code disc (measuring scale), the evaluation electronics as well as (if required) the multiturn device, are supplied separately and fitted by the user himself. In addition to the correct mechanical alignment of the parts, which is easily ensured by the design of the encoder systems, it is also necessary to electronically match and align the various code tracks, to synchronise the tracks. This is performed by calling this command.

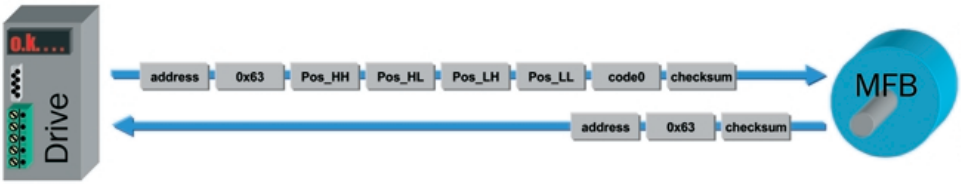


Fig. 43

The MFB determines the values of the internal code tracks and uses this to calculate appropriate internal angle offsets, with a virtual 0 being generated.
Together with these angle offsets, the position offset (Pos_HH .. Pos_LL) given in the protocol is stored in the EEPROM of the MFB, where it cannot be lost.
The position offset value is transmitted as unsigned long with the LSB right-justified in Pos_LL.

Caution:
This command must **only** be called **when the MFB shaft is stationary**, otherwise faulty angle offset values might be obtained.

Notizen/Notes

Contact:

Australia
Phone +61 3 9497 4100
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Belgium / Luxembourg
Phone +32 (0)2 466 55 66
E-Mail info@sick.be

Brasil
Phone +55 11 5091-4900
E-Mail sac@sick.com.br

Ceská Republika
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China
Phone +852-2763 6966
E-Mail ghk@sick.com.hk

Danmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Deutschland
Phone +49 (0)2 11 53 01-250
E-Mail info@sick.de

España
Phone +34 93 480 31 00
E-Mail info@sick.es

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Great Britain
Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

Italia
Phone +39 011 797965
E-Mail info@sick.it

Japan
Phone +81 (0)3 3358 1341
E-Mail info@sick.jp

Korea
Phone +82-2 786 6321/4
E-Mail kang@sickkorea.net

Nederlands
Phone +31 (0)30 229 25 44
E-Mail info@sick.nl

Norge
Phone +47 67 81 50 00
E-Mail austefjord@sick.no

Österreich
Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska
Phone +48 22 837 40 50
E-Mail info@sick.pl

Schweiz
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Singapore
Phone +65 6744 3732
E-Mail admin@sicksgp.com.sg

Suomi
Phone +358-9-25 15 800
E-Mail sick@sick.fi

Sverige
Phone +46 8 680 64 50
E-Mail info@sick.se

Taiwan
Phone +886 2 2365-6292
E-Mail sickgrc@ms6.hinet.net

Türkiye
Phone +90 216 388 95 90 pbx
E-Mail info@sick.com.tr

USA
Phone +1 937-454-1956
E-Mail sales@stegmann.com

More representatives and agencies
in all major industrial nations at
www.sick.com

SICK | STEGMANN

SICK AG • Industrial Sensors • Waldkirch • Germany • www.sick.com
SICK STEGMANN GmbH • Donaueschingen • Germany • www.sick-stegmann.de