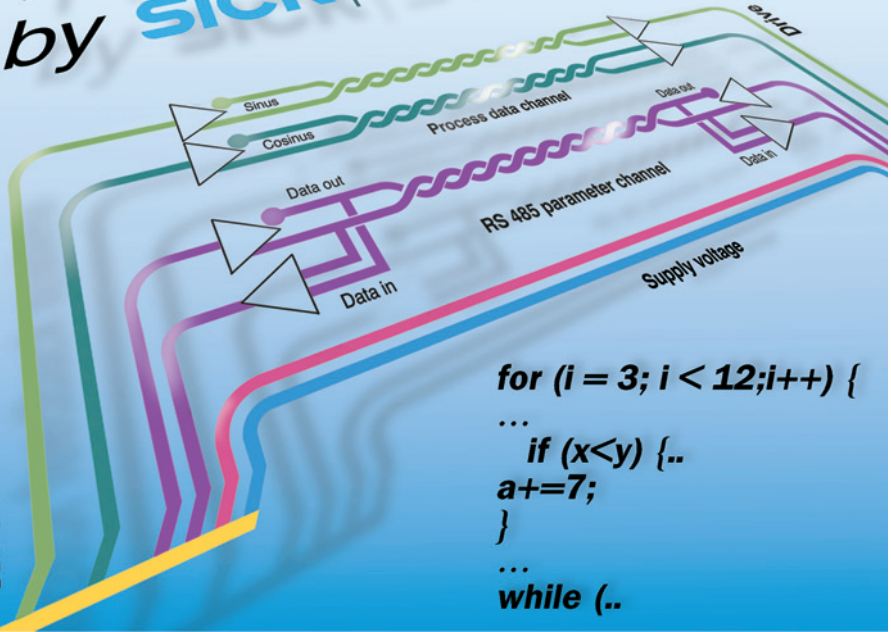


HIPERFACE[®] by SICK | STEGMANN

Motor feedback systems



```
for (i = 3; i < 12; i++) {  
  ...  
  if (x < y) {..  
    a += 7;  
  }  
  ...  
  while (..
```

BESCHREIBUNG / DESCRIPTION

HIPERFACE[®]-Beschreibung Description of the HIPERFACE[®] Interface

SICK | STEGMANN

Inhaltsverzeichnis

HIPERFACE® Motorfeedback-Systeme	3
HIPERFACE® Der Parameterkanal	15
HIPERFACE® Das Übertragungsprotokoll	16
Datenformat	16
Dialog Timing	16
ADDRESS-Format	17
Broadcastadresse FFh	17
COMMAND-Format	17
Fehlerbehandlung	17
HIPERFACE® Die Befehle	18
Position lesen	19
Position setzen	20
Analogwert lesen	21
Zählerfunktionen	22
Zähler lesen	22
Zähler erhöhen	22
Zähler löschen	22
Datenspeicherung und Verwaltung	23
Daten lesen	24
Daten speichern	24
Status eines Datenfeldes ermitteln	24
Datenfeld anlegen	25
Verfügbaren Speicherbereich ermitteln	25
Zugriffsschlüssel ändern	25
Beispiel zur Datenfelddefinition	26
Geberstatus abfragen	27
Typenschild auslesen	27
Reset	29
Adresse vergeben	30
Seriennummer & Programmversion lesen	30
Serielle Schnittstelle konfigurieren	31
Position setzen mit interner Synchronisation	32

Contents

HIPERFACE® Motor Feedback Systems	33
HIPERFACE® The parameter channel	45
HIPERFACE® The transmission protocol	46
Data Format	46
Dialogue Timing	46
ADDRESS Format	47
Broadcast Address FFh	47
COMMAND Format	47
Error Handling	47
HIPERFACE® The commands	48
Read position	49
Set position	50
Read analogue value	51
Counter functions	52
Read counter	52
Increase counter	52
Delete counter	52
Data Storage and Administration	53
Read data	54
Store data	54
Determine status of a data field	54
Create data field	55
Determine available memory area	55
Change access code	55
Example: Data Field Definition	56
Read encoder status	57
Read out type label	57
Reset	59
Allocate address	60
Read serial number and program version	60
Configure serial interface	61
Set position with internal synchronisation	62

SinCos® – Motorfeedback mit HIPERFACE®?

Es ist nun schon mehrere Jahre her, dass sich die Schnittstelle HIPERFACE® am Markt etabliert hat. Der Grund hierfür liegt in den deutlichen Einsparungen bei gleichfalls höherer Performance gegenüber konventionellen Konzepten. Diese beinhalten meist mehrere Gebersysteme um die verschiedenen Funktionalitäten am Regler zu erfüllen,

- Einen Tacho-Generator zur Bereitstellung der Information „Geschwindigkeit“
- Einen magnetischen oder optischen Kommutierungsgeber zur Blockkommutierung
- Einen Positionsgeber für die überlagerte Positioniersteuerung.

Vorteil dieser Systeme ist deren einfache Implementierung aufgrund der Verfügbarkeit der entsprechenden Interface-Bausteine und die geringe Komplexität der Einzelkomponenten.

Nachteilig ist demgegenüber nicht nur, dass die Gebersysteme selbst jeweils Geld kosten, auch die Montage dreier Systeme und deren Verkabelung, verursacht hohe Kosten. Außerdem steigt mit der Anzahl der Komponenten auch deren Ausfallwahrscheinlichkeit.

Besonders bei Kompaktreglern kleiner Leistungsklassen stehen natürlich die Einsparpotentiale im Vordergrund. Ansatzpunkte sind hier:

- Die Ausführung des Kommutierungsgebers mit Hall-Sensoren
- Der Wegfall des Positionsgebers, falls nicht unbedingt erforderlich
- Ersatz des Tachogenerators durch Software-Konzepte der „Sensorless Control“

Diese Einsparungen werden jedoch stets mit Einschränkungen der Performance des Gesamt-Systems erkauft.

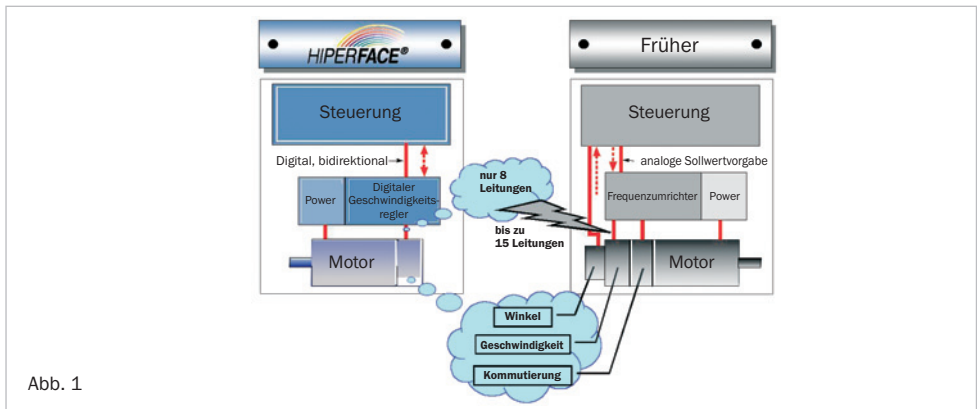


Abb. 1

Das Wissen um diese Anforderungen und der Wunsch, alle diese Funktionen mit nur einem Gerät zu erfüllen, waren Auslöser zur Entwicklung der innovativen Motorfeedback-Systeme von SICK-STEMMANN, ausgestattet mit einer neuartigen Schnittstelle:

HIPERFACE®

steht für **High Performance Interface** und ist die Standard-Schnittstelle für Motorfeedback-Systeme von SICK-STEMMANN.

Diese Schnittstelle wurde speziell auf die Anforderungen der digitalen Antriebsregelung entwickelt und bietet dem Anwender vereinheitlichte und vereinfachte mechanische und elektrische Schnittstellen.

HIPERFACE® – Motorfeedback Die wichtigsten Features im Überblick:

- Nur eine Schnittstelle am Drehzahlregler für alle Anwendungen und nur eine Art von Leitung zwischen Drehzahlregler und Motorfeedback.
- Realisierung von Low- als auch High-End-Anwendungen mit nur einer elektrischen Schnittstelle.
- Hybride Schnittstelle aus
 - dem analogen Prozessdatenkanal, auf dem Sinus- und Cosinussignale differentiell, nahezu verzögerungsfrei übertragen werden
 - dem bidirektionalen Parameterkanal, der RS 485-Spezifikation entsprechend, zur Übertragung der absoluten Position und weiterer unterschiedlichster Parameter.
- Nur 8 Leitungen
- Elektronisches Typenschild zur Identifikation des Motorfeedback und zur Speicherung antriebsrelevanter Informationen im Motorfeedback
- Großer Temperaturbereich, hohe Schock- und Vibrationsfestigkeit, unempfindlich gegen elektromagnetische Störungen und kompakt. Die Geräte können direkt in den Servomotor eingebaut werden.
- Zur Drehzahlregelung stehen analoge Sinus-/Cosinussignale zur Verfügung. Diese ermöglichen sowohl hohe Auflösungen zur Regelung geringer Drehzahlen, als auch ausreichend geringe Signalbandbreite zur Regelung hoher Drehzahlen.
- Kabellängen bis 100 m.
- Einheitliche mechanische Schnittstellen.
- Die Encoderwelle wird fest mit der Motorwelle verbunden, damit sind hohe Regelverstärkungen möglich.
- Einfacher Einbau in den Servomotor. Axiale und radiale Toleranzen werden über eine elastische Statorkupplung (auch: Drehmomentstütze) ausgeglichen.
- Zur Einstellung der Kommutierung kann der Absolutwert der mechanischen Wellenstellung elektronisch zugeordnet werden.
- Zur Positionsregelung mechanisch unteretzter Anwendungen stehen die Motorfeedback-Systeme in der gleichen mechanischen Bauform auch als absolute Multiturns, zur absoluten Positionserfassung über 4096 Umdrehungen, zur Verfügung.

Das Besondere

Bisher wurden optische Drehgeber als so genannte Incrementalgeber oder Absolutwertgeber klassifiziert.

Incrementalgeber sind vergleichsweise einfach aufgebaut und tasten nur wenige Spuren ab. Sie bieten relativ hohe Auflösungen, und eignen sich aufgrund ihrer zählenden, echtzeitfähigen Schnittstelle (Quadratur-Signale) gut zur Drehzahlregelung von Antrieben. Aufgrund der fehlenden absoluten Positionsinformation, bzw. erst nach Drehung der Welle um bis zu 360°, sind sie jedoch als Kommutierungsgeber ungeeignet.

Absolutwertgeber übertragen stets die gesamte Positionsinformation und sind deshalb sehr gut zur Positionsregelung geeignet. Die digitale Übertragung der Position benötigt eine hohe Übertragungsbandbreite im Kabel und ist nur bedingt echtzeitfähig. Außerdem sind die Absolutwertgeber wegen ihres großen Aufwandes zur Herstellung nach konventioneller Art relativ teuer. Diese konventionelle Art ist das Lesen, einer auf einer Glasscheibe codierten binären Information, wobei für jede binäre Stelle eine entsprechende optische Abtastung nötig ist. Alle diese Abtastungen müssen zueinander so eingestellt sein, damit unter allen Bedingungen kein Lesefehler auftreten kann.

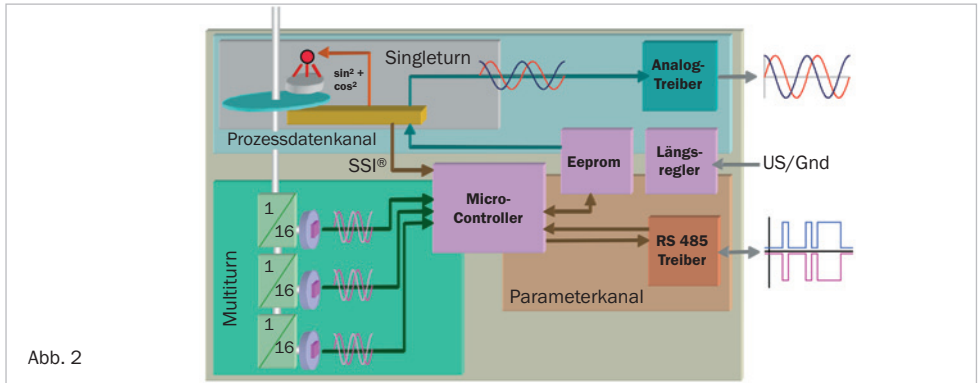
HIPERFACE®-Motorfeedback-Systeme sind eine Mischung aus Incrementalgeber und Absolutwertgeber und vereinen die Vorteile beider Geberarten.

Der Absolutwert wird hierbei zunächst nur beim Einschalten des Gerätes gebildet und über die busfähige Parameter-Schnittstelle nach RS485-Spezifikation dem externen Zähler im Regler mitgeteilt, der danach von diesem Absolutwert aus, inkrementell mit den analogen Sinus-/Cosinussignalen weiterzählt.

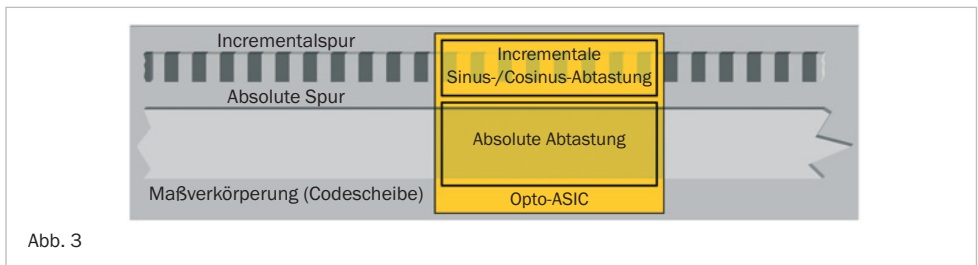
Durch die Verwendung hochlinearer Sinus- und Cosinussignale wird die zur Drehzahlregelung notwendige hohe Auflösung erreicht (Arcus tangens-Bildung im Regler). Gleichzeitig bleiben jedoch die zu übertragenden Signalfrequenzen relativ nieder, so entsteht z. B. bei einer Periodenzahl von 512 pro Umdrehung, selbst bei einer Drehzahl von 12.000 UPM nur eine Frequenz von 102,4 kHz, welche problemlos auch über weite Strecken übertragbar ist.

Interner Aufbau

Typischer Aufbau eines HIPERFACE®-Motorfeedback-Systems



Wie bei Standard-Gebern üblich, befindet sich im Inneren eines Motorfeedback-Systems meist eine Codescheibe aus Glas als Maßverkörperung. Das auf der Scheibe befindliche Codemuster wird im Durchlichtverfahren von einer Infrarot-Diode auf einem voll kundenspezifischen, hochintegrierten Schaltkreis abgebildet. Die Besonderheit der SinCos®-Motorfeedback-Systeme beruht nun darin, dass neben der absoluten Codespur auch eine inkrementelle Spur angeordnet ist, welche Sinus- und Cosinussignale erzeugt.



Üblicherweise werden in einem Motorfeedback-System **2ⁿ Perioden sinus- und cosinusförmiger Signale** abgetastet. Diese Signale stehen jederzeit über den so genannten Prozessdatenkanal als Echtzeit-Information zur Verfügung. Parallel dazu ist über den Parameterkanal (RS485) die absolute Positionsinformation verfügbar, die in erster Linie dazu dient, genau eine dieser 2ⁿ Perioden zu identifizieren.

HIPERFACE®**Die Verarbeitung der HIPERFACE®-Signale im Motorregler****Hardware im Regler**

HIPERFACE® definiert die einheitliche elektrische Schnittstelle mit nur 8 Leitungen:

- 2x Versorgungsspannung 7 ... 12V
- 4x Incrementelle, differentiell übertragene, Sinus-/Cosinus-Signale
- 2x Digitales, bidirektionales RS485-Interface

Ein Kabel mit Gesamtschirm und 8 paarig verseilten Leitungen überträgt die Signale sicher zum Regler.

Versorgungsspannung

Auf den ersten Blick erscheint es nachteilig, im Regler eine weitere, recht ungewöhnliche Versorgung von typischerweise 8V aufbauen zu müssen. Dies gewährleistet jedoch auch bei langen Distanzen zwischen Regler und Motorfeedback eine ausreichende Versorgungsspannung am HIPERFACE®-Motorfeedback-System. Aufwändigere Versorgungen z. B. mit Sense-Leitungen werden nicht benötigt.

RS485-Interface

Diese Schnittstelle kann getrost als Industriestandard bezeichnet werden. Das physikalische Interface benötigt einen 130 Ω -Abschlusswiderstand, zwei Biasing-Widerstände und einen Standard RS485-Transceiver. Das Protokoll wird mit einer Standard-UART abgewickelt, wie sie auf fast allen üblichen Mikrocontollern/DSPs implementiert ist.

SinCos®-Schnittstelle

Die Sinus-/Cosinussignale werden voll differentiell übertragen, ihre Amplitude variiert unter allen Bedingungen um höchstens 20%. Da diese Schnittstelle die eigentliche Performance des Drehzahlreglers gewährleistet, muss dieser Teil des Interface sorgfältig entwickelt werden. Neben Differenz-Eingangverstärkern mit geringen Offsets, geringem Rauschen und hoher Gleichtaktunterdrückung, sind auch Widerstände und Kondensatoren (Tiefpass) geringer Toleranz einzusetzen.

Auch nach dem analogen Eingangsverstärker müssen Signalrauschen und Übersprechen klein gehalten werden. Hier geht es auch weiter zu den Komparatoren zur Erzeugung von inkrementellen Zählsignalen und zu einem 2-kanaligen, simultan erfassenden AD-Wandler mit mindestens 10 Bit Auflösung.

Üblicherweise sind weder zum Zählen, noch zur Analog/Digital-Wandlung zusätzliche, externe Komponenten notwendig. Diese Funktionalitäten sind heute bei Mikrocontrollern und DSP Stand der Technik.

Optionales

Um die volle Funktionalität der HIPERFACE®-Schnittstelle auszunützen, sollte zusätzlich eine Verbindung zwischen interner RxD-Leitung und dem Capture-Eingang des Incremental-Zählers vorgesehen werden. Hiermit ist es z. B. in sicherheitsrelevanten Applikationen, auch bei voller Drehzahl (6000 rpm) möglich, den Zählerstand mit der absoluten Position zu vergleichen.

Zur Vervollkommnung des aus den Analogsignalen abgeleiteten Geschwindigkeitswerts stehen inzwischen viele algorithmische und/oder Hardware-Lösungen zur Verfügung, wie z. B.

- Filterung
- Oversampling
- PLL-Verfahren

Der Schirm – die 9. Leitung

Auch das Schirmkonzept ist für die erreichbare Performance des Gesamtsystems von großer Bedeutung. Ein beidseitig, am Motor und an der Steuerung, großflächig aufgelegter Kabelschirm erzielt hier meist die besten Resultate. Der Schirm sollte mit der Schutzterde verbunden sein. Ist in Anlagen mit großen Ausgleichströmen zu rechnen, ist ein separater Potentialausgleichsleiter zu verwenden.

Software

Serielles Protokoll

Standardmäßig ist die Schnittstelle auf 9600 Bd konfiguriert. Die bidirektionale RS485-Schnittstelle ist prinzipiell busfähig, weshalb jede Kommunikation vom Master, also von der Steuerung, mit der Slave-Adresse beginnt. Zur Minimierung der Übertragungszeiten werden die Informationen binär übertragen. Jedes Protokoll wird mit einer einfach zu berechnenden XOR-Prüfsumme abgeschlossen. Die Protokoll-Ende-Erkennung erfolgt anhand einer Timeout-Steuerung.

Die HIPERFACE®-Motorfeedback-Systeme verfügen über interne Diagnose-Funktionen und signalisieren kritische oder fehlerhafte Zustände im Antwortprotokoll.

Die Hochofölung (Arctan-Interpolation)

Zur Berechnung des hochaufgelösten Winkels innerhalb einer Periode sind viele Verfahren bekannt. Das folgende Verfahren (Abbildung 4) hat sich dabei als guter Kompromiss bzgl. Codegröße und Ausführungsgeschwindigkeit bewährt:

1. Reduzierung der Berechnung auf den 1. Oktanten (0 ... 45°) durch vertauschen und/oder invertieren der Sinus- und Cosinuswerte, damit die nachfolgende
2. Berechnung der Division $\sin(x)/\cos(x)$ immer ein Ergebnis ≤ 1 hat.
3. Tabellengestützte Linearisierung des Divisionsergebnisse zur Annäherung an die Arcus-Tangens-Funktion (ca. 32 Stützpunkte erforderlich).
4. Rücktransformation & Erweiterung des Ergebnisses in den unter 1) ermittelten Oktanten.

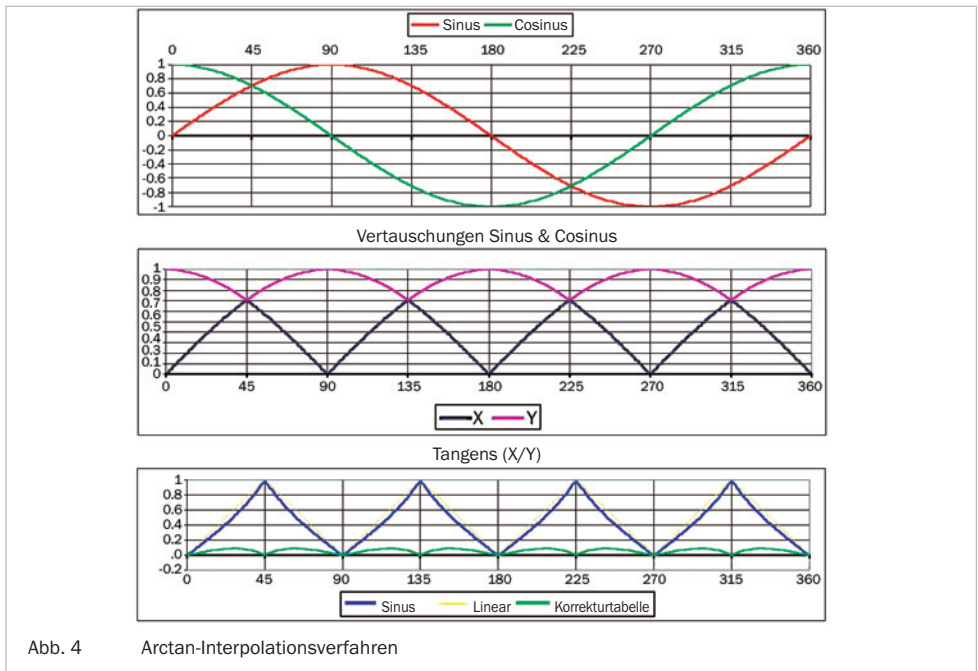


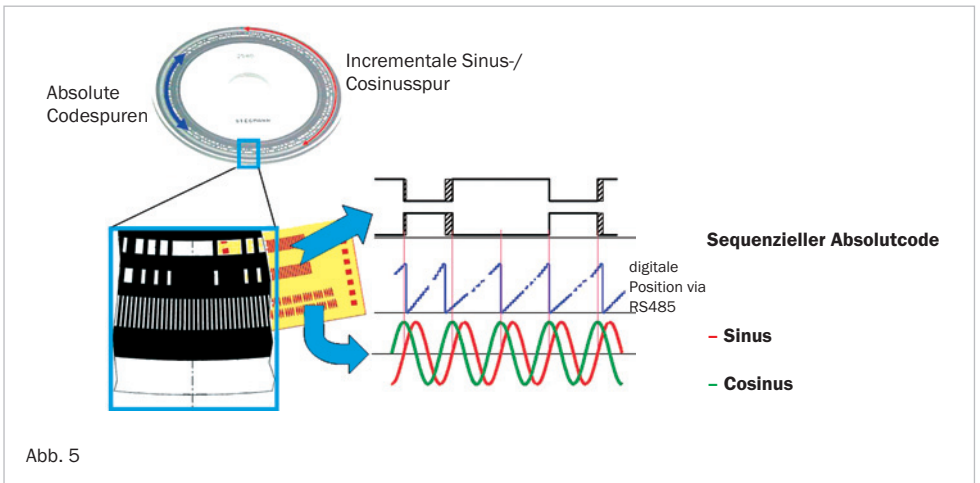
Abb. 4 Arctan-Interpolationsverfahren

Synchronisation

Der Überlauf, bzw. die Flanken auf den verschiedenen Übertragungswegen werden in Realität nie zum exakt identischen Zeitpunkt auftreten.

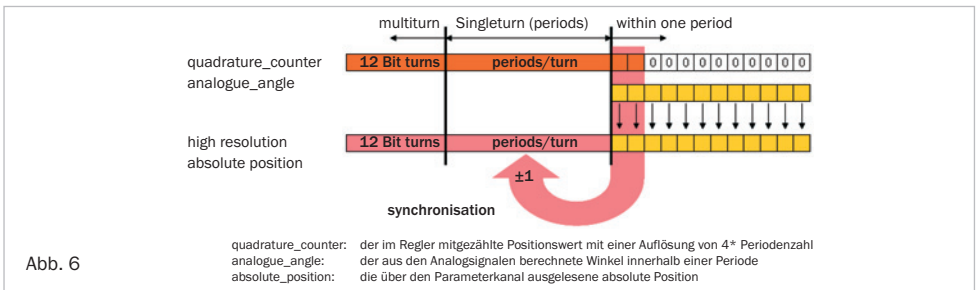
Dies lässt sich durch folgende Effekte erklären:

- die absolute Position wird über separate Codespuren ermittelt und weicht deshalb geringfügig von der aus den Analogsignalen interpolierten Position ab (s. Abb. 5) Diese Phasenlage kann sich deshalb innerhalb einer Umdrehung um bis zu $\pm 4/32$ einer Periode verändern.
- dynamische Effekte, wie Signallaufzeiten oder unterschiedliche Eingangsfiler
- Hysterese, z. B. am Komparator des Quadraturzählers

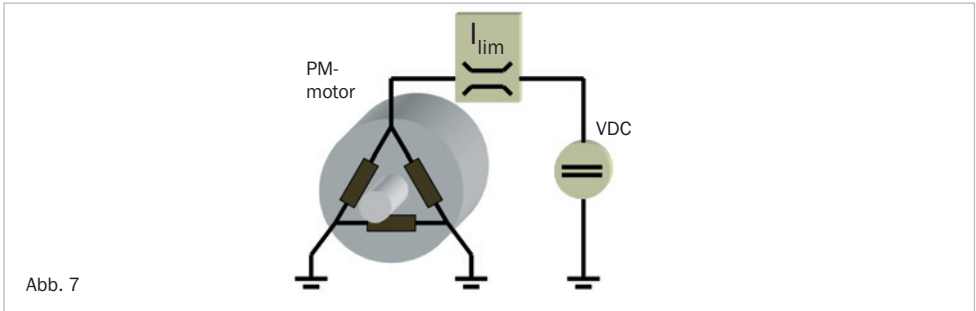


Die Synchronisation beschreibt nun ein Verfahren welches diese Effekte in weiten Grenzen ($\pm 1/2$ Periode) kompensieren kann. Damit ist es möglich, auch bei hohen Drehzahlen die analogen Signale einer bestimmten Periode zuzuordnen.

Dabei werden jeweils die überlappenden Bits zweier unterschiedlicher Positionsinformationen auf gleiche Wertigkeit ausgerichtet und sodann die Differenz der Positionswerte gebildet. Zeigen die beiden Positionswerte in unterschiedliche Perioden, so wird der niedriger aufgelöste Positionswert in Richtung der kleineren Differenz um ± 1 Periode korrigiert.



Inbetriebnahme/Kommutierungseinstellung mit Hilfe eines HIPERFACE®-Motorfeedback-Systems



Bei der Einstellung der Motorkommutierung kommt im Allgemeinen eines der folgenden Verfahren zur Anwendung:

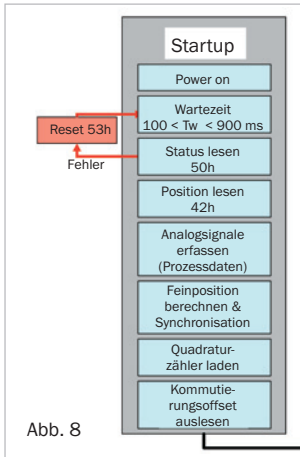
1. Unter Verwendung der Datenfelder

Bei statisch bestromtem Motor wird aus der digitalen Absolutposition und dem, aus den analogen Signalen interpolierten Winkelwert, eine hochaufgelöste Position errechnet. Dieser Positionswert wird sodann als Kommutierungsoffset im permanenten Speicher (EEPROM) des Motorfeedback abgelegt. Dieses Verfahren ist universell anwendbar und erlaubt auch bei hochpoligen Motoren eine genaue Einstellung der Kommutierung.

2. Verwendung der MFB-Funktion „Position setzen“ (43h)

Bei statisch bestromtem Motor erfasst die Steuerung zunächst die Analogsignale (Prozessdaten) und berechnet den Arcustangens. Danach wird die Funktion „Position setzen“ aufgerufen, wobei die 5 LSB des Positionswertes in Übereinstimmung mit den 5 MSB aus dieser Interpolation gesetzt werden müssen. Die Kommutierung kann hierbei nur noch mit der Auflösung ganzer Perioden des Motorfeedback gesetzt werden. Bei Motorfeedback mit geringer Periodenzahl, kann damit die Einstellung der Kommutierung zu ungenau sein und damit die erreichbare Performance des Antriebssystems einschränken.

Das HIPERFACE®-Feedback im Regelzyklus



Start Up

Nach dem Einschalten können neben dem Initialisierungsstatus des HIPERFACE®-Motorfeedback-Systems, auch alle Betriebsparameter der Motor-Regler-Kombination aus dem Eeprom ausgelesen und der Regler anhand des elektronischen Typenschildes auf die technischen Daten des Motorfeedback-Systems parametrisiert werden.

Sodann wird die absolute Position ausgelesen und damit die Kommutierungslage und ggf. die Position der Achse ermittelt.

Beim ersten Regelzyklus werden danach die Analogsignale erfasst, und mit dem daraus berechneten Arcus-Tangens, der Incrementalzähler initialisiert.

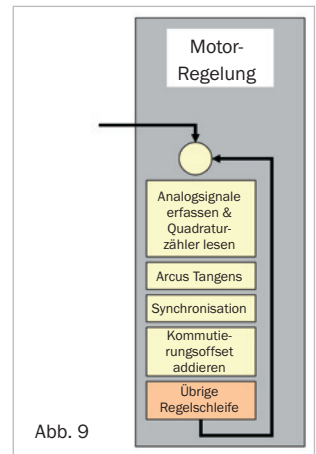
Geschwindigkeitsregler

Die hochlinearen, sinusförmigen Analogsignale erlauben einen hohen Interpolationsfaktor und damit eine hohe Auflösung der Geschwindigkeit. Hiermit ist beinahe mühelos ein weiterer Drehzahlstellbereich von 1:10.000 erreichbar.

Positionsregler

Was die Schnittstelle zum Positionsregler angeht, so wurden bereits unterschiedliche Konzepte realisiert. Es existieren Lösungen, bei denen der Positionsregler als zweiter RS485-Master direkt auf die Schnittstelle des HIPERFACE®-Motorfeedback-Systems zugreift um zyklisch die Position abzufragen. Häufiger jedoch, werden die vom Frequenzumrichter geförmten inkrementellen Zählsignale an die Positioniersteuerung durchgeschleust. Die Initialposition wird dabei durch den Controller ausgezählt. Ebenso häufig findet man eine zeitdiskrete Realisierung, bei der der Regler die in jedem Zyklus berechnete absolute Position in einem seriellen (z. B. SSI®) Ausgangsregler zur Verfügung stellt.

Ein weiterer, großer Vorteil der Realisierung mit einem HIPERFACE®-Motorfeedback-System ist dessen Verfügbarkeit als Multiturn. Dadurch ist es möglich in der Anwendung auf Endschalter zu verzichten.



Sicherheitskonzepte

Obwohl ein HIPERFACE®-Motorfeedback-System nur EIN Gerät ist, genügt es doch, aufgrund des redundanten Aufbaus, nicht nur der Schnittstelle, hohen Sicherheitsanforderungen.

Die Positionswerte werden auf zwei physikalisch getrennten Schnittstellen (analog & RS485) übertragen. Der analoge Übertragungskanal besitzt bereits eine hohe Eigensicherheit, da die Signale z. B. mittels Amplituden- & $\text{Sin}^2 + \text{Cos}^2$ -Überwachung im Regler in engen Grenzen auf ihre Gültigkeit überprüft werden können. Eine gegenseitige Überwachung der Informationen ist aufgrund der physikalischen Trennung ohne zusätzlichen Aufwand möglich. Im Innersten des Motorfeedback-Systems werden beide Informationen von einem System, bestehend aus Sender, Codescheibe und Empfänger abgetastet. Im Folgenden werden die Analogsignale, ohne Bearbeitung durch einen Mikroprozessor, nur noch verstärkt und übertragen, während der absolute Wert zuerst digital bearbeitet, überwacht und dann übertragen wird.

Wird die absolute Position auch zyklisch ausgelesen, so ist eine zuverlässige Überwachung des Incrementalzählers möglich. Außerdem kann so schon frühzeitig festgestellt werden, wenn beim HIPERFACE®-Motorfeedback-System kritische Betriebszustände erreicht werden.

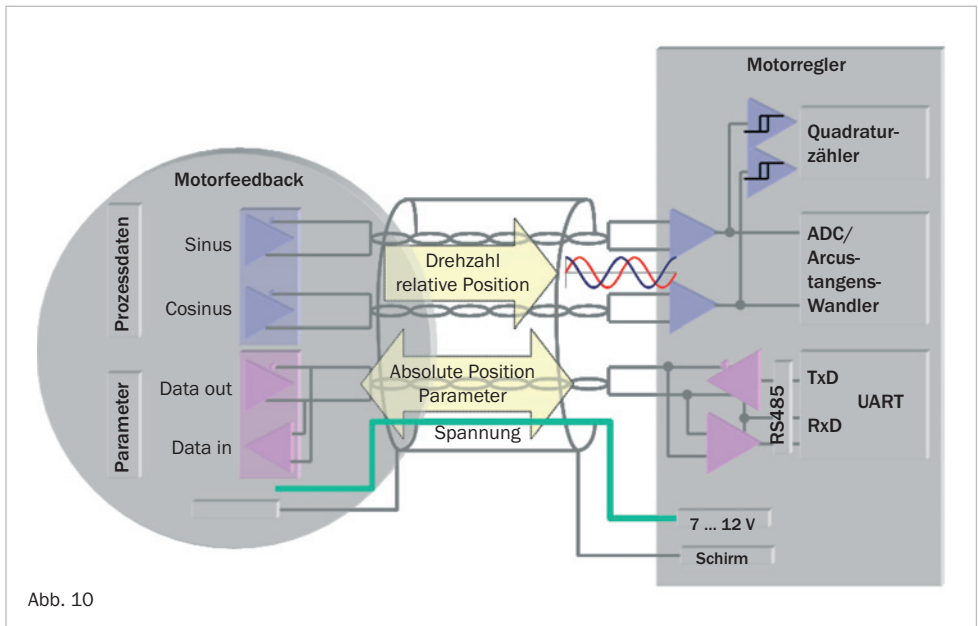


Abb. 10

HIPERFACE®

Allgemeine Spezifikationen zu HIPERFACE®

(sofern im Datenblatt nicht abweichend angegeben)

Versorgungsspannung	7 - 12 V
Stromaufnahme	≤ 250mA
Signalspezifikation für den Prozessdatenkanal	1 Vss, differentiell
Signalspezifikation für den Parameterkanal	gemäß RS 485
Schnittstellen-Timing	siehe Seite 16 + 17
Datenformat für Adressierung und Befehle	siehe Seite 16 + 17

Im Datenblatt eines Motorfeedback befinden sich unterschiedliche Angaben zu den Fehlergrenzen:

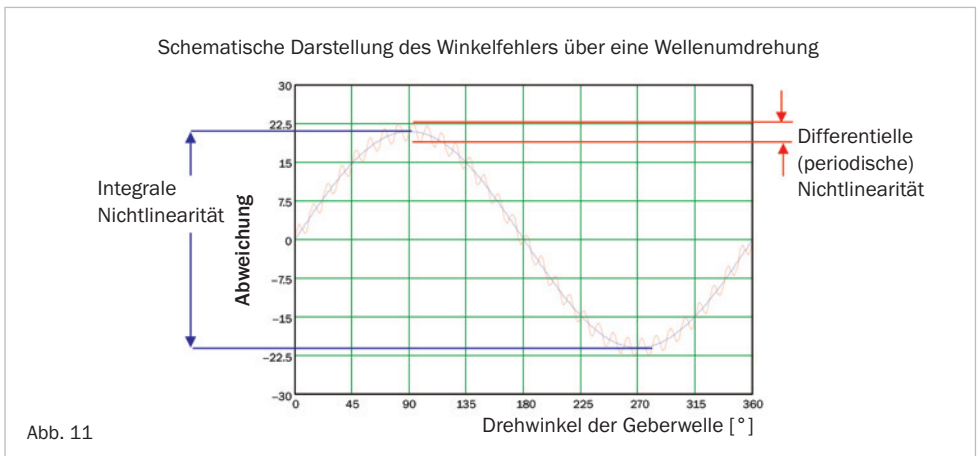
Fehlergrenzen bei Auswertung der Sinus-/Cosinussignale

- **Differentielle Nichtlinearität**
Dieser Wert spezifiziert die Abweichungen innerhalb einer elektrischen Periode des Prozessdatenkanals und ergibt sich daraus, dass die Sinus-/Cosinussignale von der idealen Form abweichen. Mögliche Fehlerquellen sind hier z. B. Signaloffsets, Amplitudenabweichungen, Phasenfehler oder Abbildungsfehler. Dieser Parameter ist vor allem für die Drehzahlregelung kleiner Drehzahlen relevant.
- **Integrale Nichtlinearität**
Dieser Wert beschreibt die Abweichungen über eine mechanische Umdrehung der Welle. Abweichungen über eine Wellenumdrehung erklären sich u. a. aus der Exzentrizität der Codescheibe, aus der Übertragungsfunktion der Statorkupplung oder aus Fehlern der Maßverkörperung.

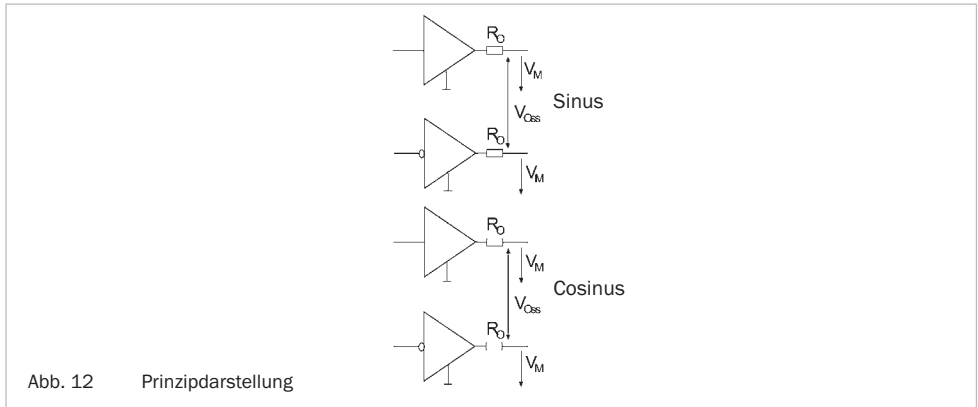
Fehlergrenzen des digitalen Absolutwertes

Zu den beiden vorstehend beschriebenen Fehlergrößen addieren sich in erster Linie interne Quantisierungseffekte bei der Berechnung der absoluten Position.

Außerdem wird die absolute Position je nach Gebertyp von unabhängigen Codespuren ermittelt, welche abweichende Fehlergrenzen aufweisen können.



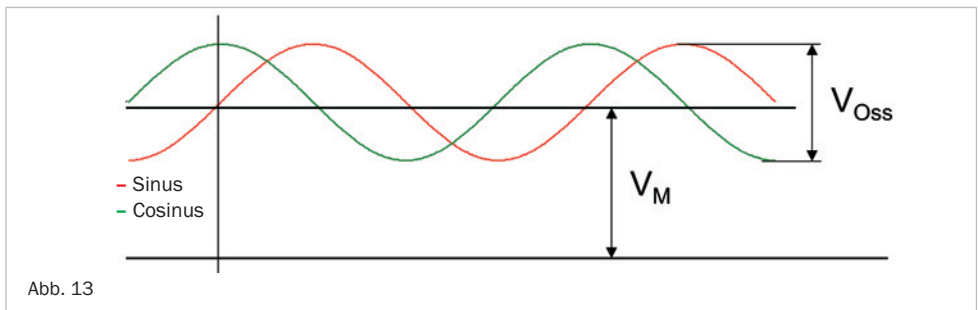
Prozessdatenkanal (Analogausgänge)



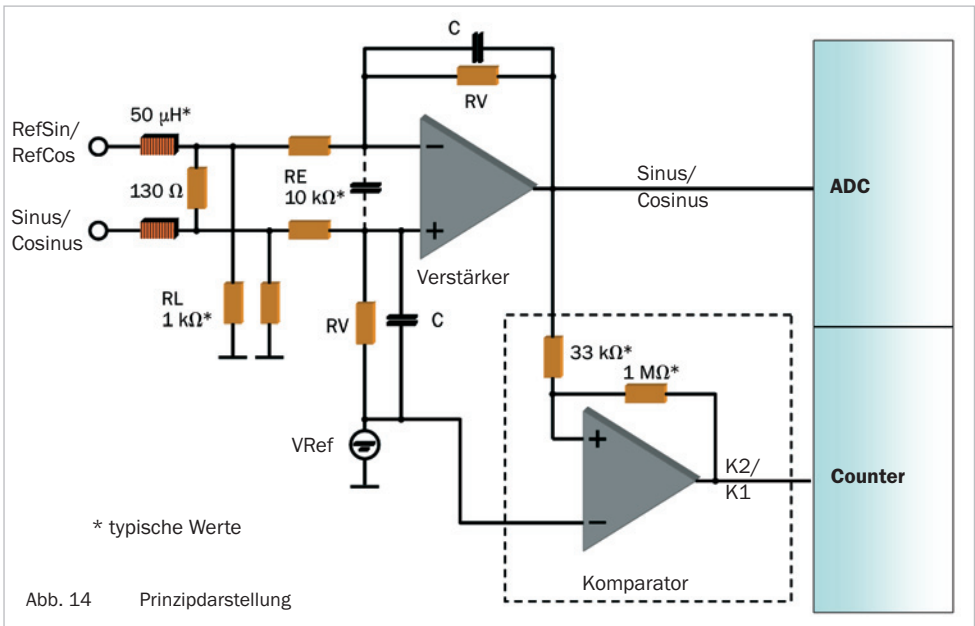
Elektrische Parameter des Prozessdatenkanals

(sofern im Datenblatt nicht abweichend angegeben)

Signalbandbreite	0 ... 150 kHz (3dB Signalamplitude)	f_{BW}
Signalmittenspannung der Einzelsignale	2,0 ... 3,0 V	V_M
Signalamplitude Spitze-Spitze, differenziell	0,9 ... 1,1 V	V_{OSS}
Ausgangsbelastbarkeit	$> = \pm 7$ mA	I_{Omin}



Typische Eingangsschaltung:



Die Filter müssen entsprechend der notwendigen Bandbreite dimensioniert werden. Um eine gute Unterdrückung von Gleichtaktstörungen zu erhalten, ist im Leiterplattenlayout auf die symmetrische Auslegung von SIN und REF-SIN zu achten.

Die Toleranz der verwendeten Widerstände muss $\leq 1\%$ sein.

Eine typische Dimensionierung für sinusförmige Eingangssignale bis zu einer Frequenz von 100 kHz ist:

L	= 50 μ H
RV	= bestimmt zusammen mit RE die Verstärkung (RV/RE)
C	= 22 ... 47 pF
RL	= ≥ 1 k Ω (optionale, zusätzliche Signalbelastung im Nulldurchgang der Differenzsignale)

Der aus RE und C gebildete Filter verringert das Signalrauschen auf den Analogsignalen, bestimmt aber auch die maximal verarbeitbare Signalfrequenz.

Phasenverschiebung

$$\varphi \cong \arctan(\omega * RV * C)$$

Ausgangsamplitude

$$U_a \cong \frac{1}{\sqrt{1 + (\omega * RV * C)^2}} * U_e$$

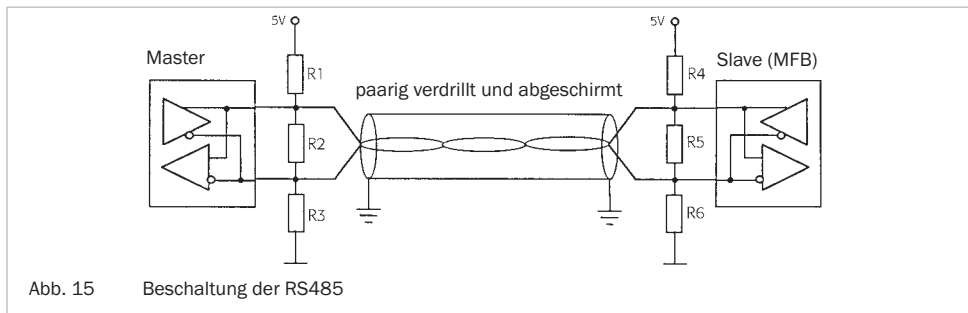
Die hier skizzierte Schaltung (Komparator hinter dem Differenzverstärker angeordnet) sorgt für ein sicheres Schaltverhalten des Komparators, da Signalrauschen und Gleichtaktstörungen bereits durch den Differenzverstärker wirkungsvoll unterdrückt werden. Außerdem stehen die Signale am AD-Converter und am Zähler phasengleich zur Verfügung, was die Synchronisation bei hohen Drehzahlen vereinfacht.

Der Parameterkanal

Der Parameterkanal ist eine asynchrone, halbduplex Schnittstelle, die physikalisch der EIA RS485 Spezifikation entspricht.

Um eine störungsfreie Übertragung sicherzustellen ist es notwendig die Übertragungsleitungen mit pull up/down Widerständen zu versehen.

Um auch die Unabhängigkeit von Line Transceiver Herstellern zu gewährleisten geben wir folgende Empfehlung für die Dimensionierung der Schnittstelle:



Dimensionierung für Bus & Single-ended

Betriebsart	Master			Slave (Motorfeedback-System)		
	R1	R2	R3	R4	R5	R6
Standard (single ended)	1K	130	1K	1K	130	1K
BUS (max. 8** Teilnehmer)	*	130	*	10K	∞ nicht bestückt	10K

* Da HIPERFACE®-Motorfeedback-Systeme immer Slave sind und die Anzahl der Teilnehmer im BUS variieren kann, muss die Dimensionierung der BUS-Schnittstelle kundenseitig vorgenommen werden.

** Bei mehr als 8 Teilnehmern setzen Sie sich bitte mit unserem Vertrieb in Verbindung.

Standardmäßig sind der 130 Ohm Abschlusswiderstand und die 1K Ohm pull up/down Widerstände eingebaut. Für Bus-Betrieb (mehr als 1 Geber) ist kein Abschlusswiderstand im Geber eingebaut, dieser muss kundenseitig an den am weitesten vom Master entfernten Teilnehmer angebracht werden.

Die Motorfeedback-Systeme müssen folglich in der Ausführung „BUS“ gesondert bestellt werden.

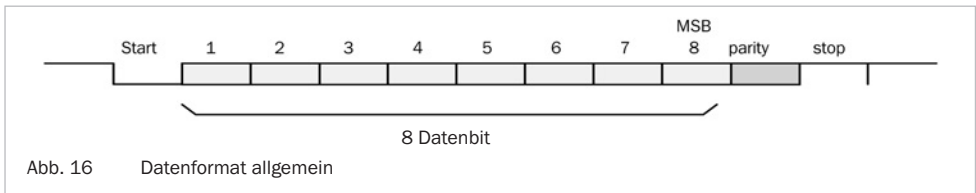
Das Übertragungsprotokoll

Datenformat

Die Daten auf der asynchronen Schnittstelle haben folgendes Grundformat:

- 1 Startbit
- 8 Datenbit (LSB first)
- 1 Stopbit
- Parity programmierbar: odd, even, none
- Baudrate programmierbar*: 600, 1200, 2400, 4800, 9600, 19200, 38400 Baud

* Hinweis: Die Baudrate ist nicht bei allen MFB-Typen frei programmierbar. Bitte die Datenblätter beachten.

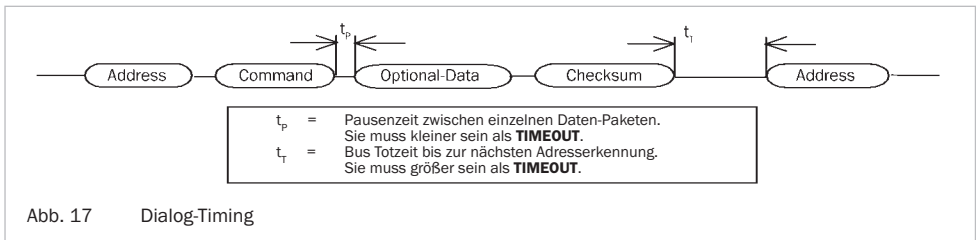


Die Parameter sind bei Auslieferung der Geräte in der Regel voreingestellt auf:

9600 Bd &

Parity = 1, wenn die Summe der Datenbits ungerade ist

Dialog Timing



- Checksum** - EXOR-Verknüpfung der übertragenen Bytes, einschließlich Address und Command
- TIMEOUT** - programmierbar, 11/Baudrate bzw. **44/Baudrate = Default**

Die Datenübertragung auf der RS485 wird über ein Timeout-Protokoll gesteuert, d. h. erst, wenn innerhalb einer bestimmten Zeit (**Timeout**), keine weiteren Informationen am MFB eintreffen, wird das aktuell empfangene Protokoll bearbeitet.

Unterbrechungen eines Datenpakets um mehr als Timeout führen zu entsprechenden Protokoll-Fehlermeldungen (s. auch COMMAND 50h).

Das 1. Byte nach einem Timeout wird als Adresse interpretiert.

Grundsätzlich belegt ein HIPERFACE®-Motorfeedback-System nicht von sich aus die Schnittstelle.

Das HIPERFACE®-Motorfeedback-System ist immer Slave.

Ein Informationsaustausch muss also immer vom Master, dem Motorregler, initiiert werden.

ADDRESS-Format

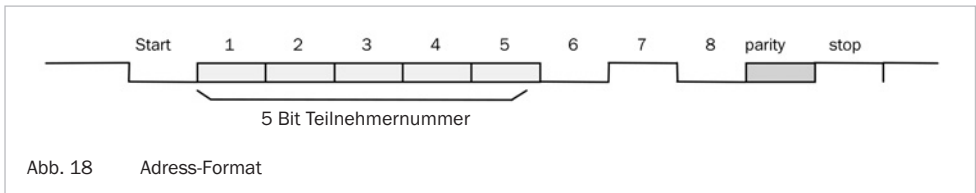


Abb. 18 Adress-Format

ADDRESS ist die Teilnehmernummer + 40h.

Broadcastadresse FFh

Statt der spezifischen Geberadresse kann das MFB auch mit einem BROADCAST, **FFh** angesprochen werden. Bei diesem BROADCAST führen **alle** am Bus angeschlossenen MFB's die selektierte Funktion aus und quittieren diese mit ihrer spezifischen Adresse.

Dabei ist zu beachten, dass die Befehle die mit einer Quittung abschließen, nur bei **einem** angeschlossenen MFB ausgeführt werden dürfen.

COMMAND-Format

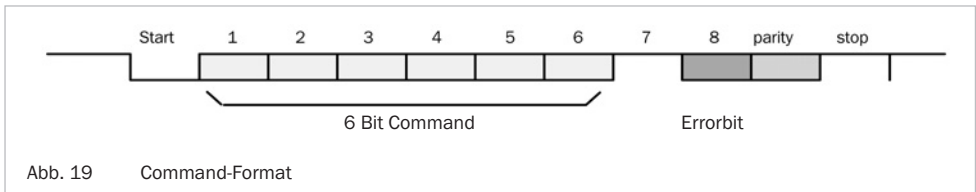


Abb. 19 Command-Format

Fehlerbehandlung

Fehlermeldung

- Befehle die nicht bearbeitet werden können (Protokollfehler, Befehlsargumente oder interne Fehler des MFB) führen zum Abbruch der Befehlsbearbeitung und das MFB antwortet mit einem Fehlerprotokoll (s. Command 50h)

Warnmeldung

- Während der Befehlsausführung wird der Betriebszustand des MFB überwacht (z. B. Temperatur, Senderstrom). Werden hierbei kritische Parameter festgestellt die jedoch auf den aktuell bearbeiteten Befehl ohne direkte Auswirkung sind, so generiert das MFB bei gesetztem Errorbit, das erwartete Antwortprotokoll.
- Das Fehlerbit wird mit dem Auslesen des Fehlercodes (Command 50h) oder durch einen MFB-Reset (Command 53h) gelöscht.
- Treten mehrere Fehler gleichzeitig auf, so werden vom MFB bis zu 4 Fehlercodes zwischengespeichert die mit dem Command 50h nacheinander ausgelesen werden können. Das Fehlerbit bleibt hierbei solange gesetzt, bis alle Fehlercodes ausgelesen sind.
- Werden hierbei kritische Parameter festgestellt die jedoch auf den aktuell bearbeiteten Befehl ohne direkte Auswirkung sind, so generiert das MFB bei gesetztem Errorbit das erwartete Antwortprotokoll.

Ein adressiertes MFB antwortet auf eine fehlerfrei bearbeitete Befehlssequenz stets mit der Wiederholung von ADDRESS und COMMAND, gefolgt von den angeforderten Daten und der CHECKSUM.

Der Befehlssatz

Der für den jeweiligen MFB-Typ verfügbare Befehlssatz kann variieren.

Der gültige Befehlssatz kann dem Datenblatt des jeweiligen Motorfeedback-Systems entnommen werden.

Übersicht der Commands

Commandbyte	Funktion	Code 0 ¹⁾	maximale Antwortzeit ²⁾ [ms]	Seite
42h	Position lesen		10	19
43h	Position setzen	•	40	20
44h	Analogwert lesen		5	21
46h	Zähler lesen		5	22
47h	Zähler erhöhen		30	22
49h	Zähler löschen	•	30	22
4Ah	Daten lesen		30	24
4Bh	Daten speichern		250	24
4Ch	Status eines Datenfeldes ermitteln		5	24
4Dh	Datenfeld anlegen		70	25
4Eh	Verfügbaren Speicherbereich ermitteln		5	25
4Fh	Zugriffsschlüssel ändern		40	25
50h	Geberstatus lesen		5	27
52h	Typenschild auslesen		5	27
53h	Geberreset		---	29
55h	Geberadresse vergeben	•	40	30
56h	Seriennummer & Programmversion lesen		5	30
57h	Serielle Schnittstelle konfigurieren	•	40	31
63h	Position setzen mit interner Spur-Synchronisation	•	40	32
67h	Serielle Schnittstelle temporär konfigurieren		5	32

¹⁾ Die entsprechend gekennzeichneten Befehle beinhalten den Parameter „Code 0“. Code 0 ist ein Byte, das zur zusätzlichen Absicherung wichtiger Systemparameter gegen versehentliches Überschreiben, ins Protokoll eingefügt ist.

Bei Auslieferung ist „Code 0“ = 55h.

²⁾ Die angegebene Antwortzeiten verstehen sich OHNE Übertragungszeiten und Protokoll-Timeout.

Bei Defaulteinstellung der RS485 beträgt das Timeout ca. 4,7 ms.

Die hier aufgeführten Funktionen lassen sich auch alle mit dem als **Zubehör** erhältlichen **Programming-Tool** durchführen.

Es dient zur individuellen Parametrierung von HIPERFACE®-Motorfeedback-Systemen und besteht aus:

- [Programmieradapter](#)
- [Link-Kabel](#)
- [MFB-Kabel](#)
- [Steckernetzteil](#)
- [Programm CD-Rom](#)

Die Befehle

Position lesen → 42h

Um den Quadratur-Zähler im Motorregler auf den Absolutwert des MFB zu setzen, ist diese Funktion die wichtigste des gesamten Systems. Wesentliche Eigenschaft der MFB's ist es, dass diese Synchronisation auch bei voller Drehzahl ohne Einbuße von Information oder Regelgeschwindigkeit erfolgen kann.

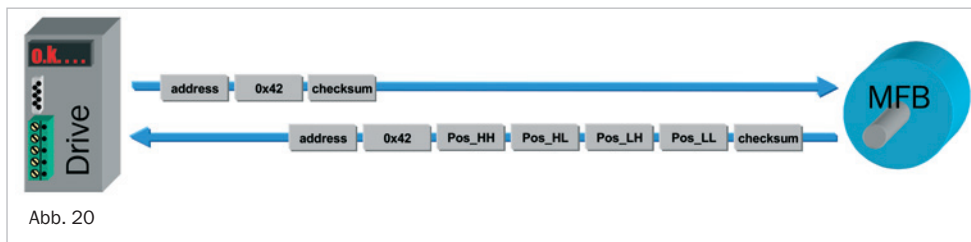


Abb. 20

Der Positionswert wird in Pos_HH ... Pos_LL als **unsigned long** mit rechtsbündigem LSB übertragen.

Zur Synchronisation der MFB-Position mit dem steuerungsseitigen mitgezählten Absolutwerts dient die erste Flanke des Startbits von „ADDRESS“ der MFB-Antwort

Um eine störreichere Synchronisation zu ermöglichen, wird der RS485-Treiber ca. 2 ms vor Beginn der Datenübertragung aktiv geschaltet.

Bei Ausführung dieser Funktion werden im MFB verschiedene Diagnosefunktionen ausgeführt die gegebenenfalls als Warnung oder Fehler signalisiert werden.

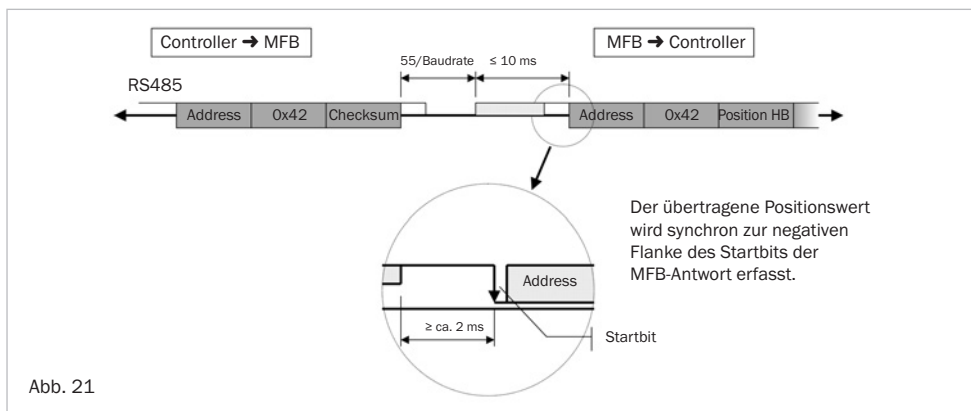


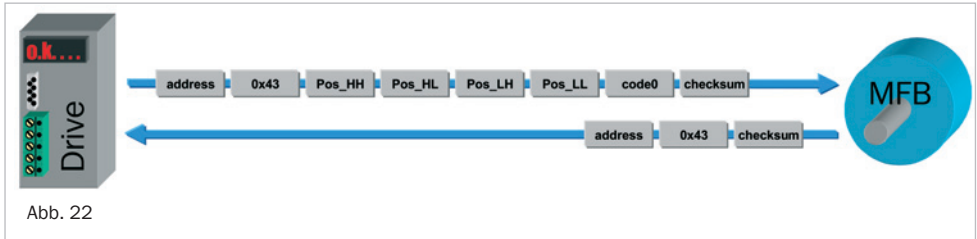
Abb. 21

Der absolute Positionswert dient zur eindeutigen Identifizierung einer der analogen Perioden und hat deshalb stets eine Auflösung von 5 Bit/Periode. Zur Überprüfung der im Regler mitgezählten Position, kann die Absolutposition des MFB zyklisch ausgelesen werden. Dabei empfiehlt es sich die ausgelesene Position nur zur Verifikation des Zählwertes heranzuziehen, nicht aber zu dessen Korrektur. **Ein sauber aufgebautes & geschirmtes System verzählt sich im Regelfall nicht.**

Soll die Überwachung von Incrementalzählung und Position auch bei hohen Drehzahlen mit hoher Genauigkeit erfolgen, so muss die Startflanke der Positionsantwort zur Speicherung des Zählerstandes verwendet werden (s. Abb.).

Position setzen → 43h

Die Absolutinformation pro Umdrehung wird bei Synchronmotoren auch für die Kommutierung des Motors verwendet. Da unsachgemäßer Gebrauch die Funktion des Motors beeinträchtigt, muss **Code 0** korrekt angegeben sein. Diese Funktion sollte nur vom Motorenhersteller aufgerufen werden.



Mit dieser Funktion kann der mechanischen Wellenstellung jeder beliebige Wert innerhalb des Messbereichs zugeordnet werden. Dazu erfolgt intern die Berechnung der momentanen Geberposition und das Abspeichern des daraus berechneten Positions-Offset.

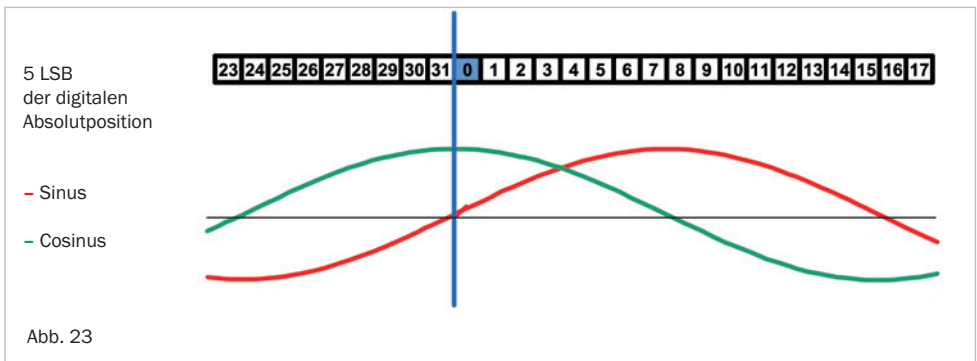
Achtung:

Da hier kein Synchronisationsmechanismus vorgesehen ist, muss diese Funktion **im Stillstand** aufgerufen werden.

ACHTUNG:

Diese Funktion verändert die Zuordnung zwischen der absoluten Position und den Analogsignalen.

Bei Auslieferung ist die Phasenlage gemäß unten stehender Abbildung definiert:



Um beim Einschalten des Motorreglers immer die größtmögliche Genauigkeit zu gewährleisten, ist es erforderlich, dass die Zuordnung der Analogsignale zur digitalen Position in der dargestellten Weise beibehalten wird, da nur dann die eindeutige Identifizierung einer Periode möglich ist.

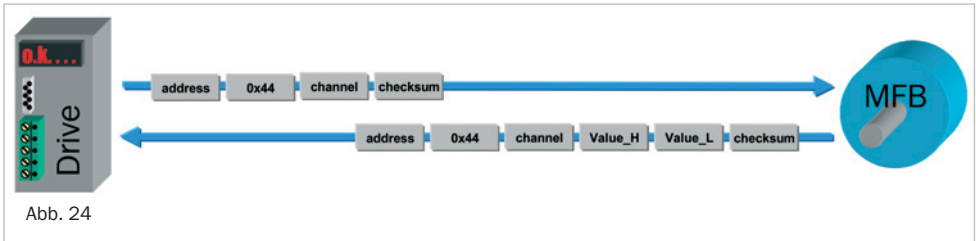
Analogwert lesen ⇒ 44h

Abb. 24

Das mit „CHANNEL“ selektierte Analogsignal wird digital gewandelt und im Regelfall mit einer Auflösung von 10 Bit/5 V vorzeichenbehaftet (SIGNED INT) in Value_xx rechtsbündig zurückgesandt.

Die verfügbaren Analogsignale, deren Verschlüsselung und Auflösung, sind jeweils im Datenblatt des einzelnen Geräts angegeben.

HIPERFACE®

Zählerfunktionen

In der Geberfirmware ist ein incrementeller 24Bit-Zähler realisiert.

Nachstehende Befehle erlauben beispielsweise die einfache Implementierung eines Betriebsstundenzählers. Ebenso einfach kann z. B. auch die Zählung der Einschaltvorgänge verwirklicht werden.

Dabei ist zu beachten, dass das Zählregister im EEPROM des MFB verwaltet wird, wodurch die Anzahl der Zählvorgänge auf maximal ca. 1 Million eingeschränkt ist.

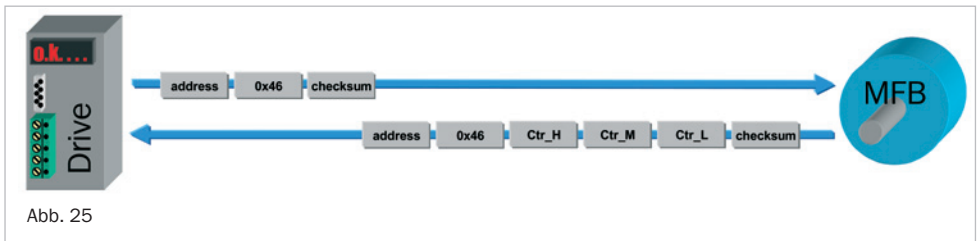
Zähler lesen ⇒ 46h

Abb. 25

Liest den aktuellen Zählerstand aus.

Zähler erhöhen ⇒ 47h

Abb. 26

Der interne 24-Bit Zähler wird um 1 erhöht. Ein eventueller Überlauf des Zählers wird je nach MFB-Typ durch das ERROR Bit oder durch eine Status-Antwort mit dem Status „08h“ angezeigt.

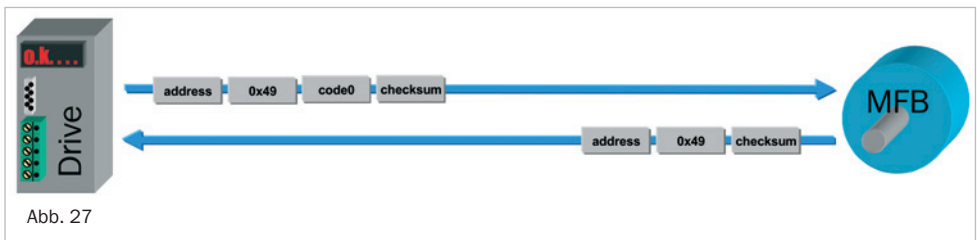
Zähler löschen ⇒ 49h

Abb. 27

Der Zähler wird auf „000“ gesetzt. Zum Schutz gegen versehentliches Löschen muss Code 0 angegeben werden.

Datenspeicherung und Verwaltung

In vielen Anwendungen ist es sinnvoll Parametrierdaten eines Motors im angebauten MFB hinterlegen zu können. Damit lässt sich beispielsweise der Austausch eines Motors erheblich vereinfachen.

Zur Abspeicherung von Daten im MFB, muss das EEPROM zuerst in einzelne Datenfelder mit Blockgrößen zwischen 16 und 128 Bytes eingeteilt werden (Command 4Dh). Dabei wird auch jeweils die Zuordnung zu einem der 4 im MFB hinterlegten Zugriffscodes festgelegt. Die Zugriffscodes (Default = 55h) ihrerseits, können mit der Funktion „4Fh“ modifiziert werden.

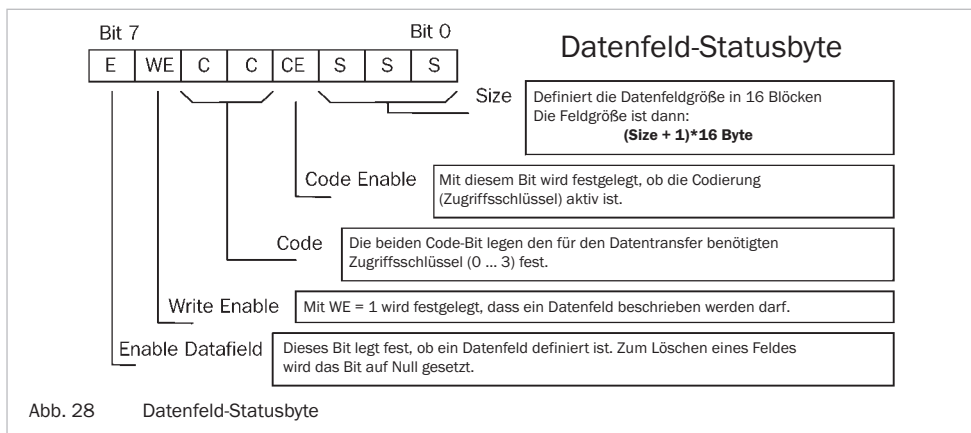
Weiterhin kann für jedes Datenfeld separat das Beschreiben gesperrt werden.

Der Zugriff auf ein Datenfeld wird im Wesentlichen durch drei Größen bestimmt:

1. Datenfeldnummer, 1 Byte, 0 ... x
2. Anfangsadresse, 1 Byte, innerhalb eines Datenfeldes jeweils von 0 ... n
3. Anzahl der zu schreibenden/lesenden Daten, 1 Byte, 1 ... 128

Dabei werden die folgenden Fehler erkannt und mit einer ERROR-Meldung quittiert:

- falscher Zugriffsschlüssel
- nicht initialisiertes Datenfeld (die angegebene Feldnummer ist zu groß)
- der Adressbereich des gewählten Datenfeldes ist überschritten



Daten lesen ⇒ 4Ah

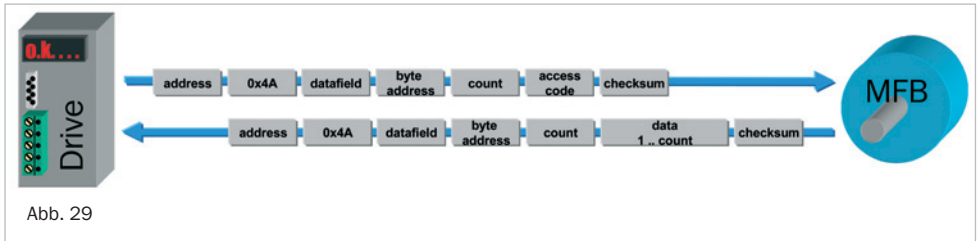


Abb. 29

Aus dem Datenfeld mit der Nummer **Datafield** und dem Zugriffsschlüssel **Access Code** werden ab der Adresse **byte address** die Anzahl **count** an Daten ausgelesen und übertragen. Der Code wird ignoriert, wenn das CE-Bit (siehe auch Datenfeldstatus lesen Commandbyte 4Ch) nicht gesetzt ist.

Daten speichern ⇒ 4Bh

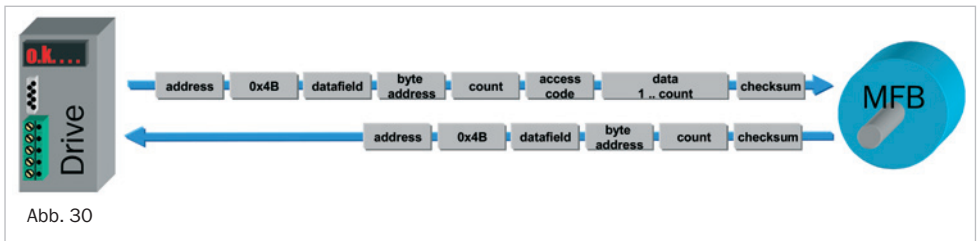


Abb. 30

Beim Beschreiben eines Datenfeldes wird außer der Zugriffsberechtigung und der Adressbereichsüberschreitung auch das WE-Bit (Write Enable, siehe auch Datenfeldstatus lesen Commandbyte 4Ch) im Statuswort der Datenfelddefinition überwacht. Die Antwortzeit des MFB ist von der Anzahl der zu programmierenden Daten abhängig und beträgt max. 250 ms.

Status eines Datenfeldes ermitteln ⇒ 4Ch

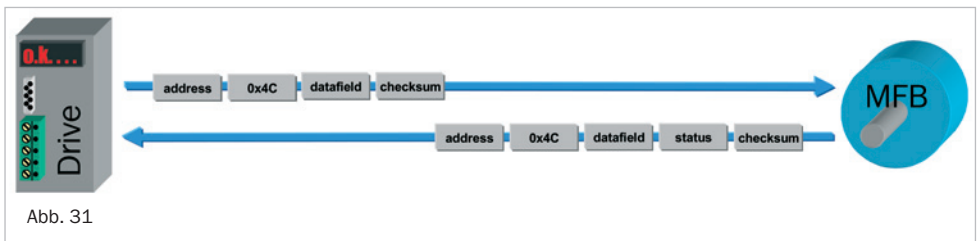
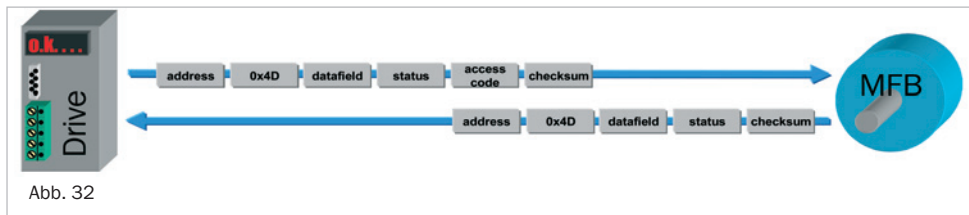


Abb. 31

Hiermit lassen sich die Kenngrößen und Zugriffsmodi eines bereits definierten Datenfeldes ermitteln.

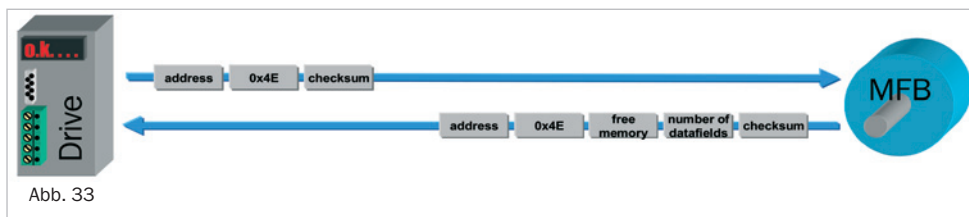
Datenfeld anlegen ⇒ 4Dh

Mit dieser Funktion werden die Datenfelder im EEPROM des MFB angelegt. Die Datenfelder müssen in aufsteigender Reihenfolge, beginnend mit der Datenfeldnummer (datafield) 0 angelegt werden.



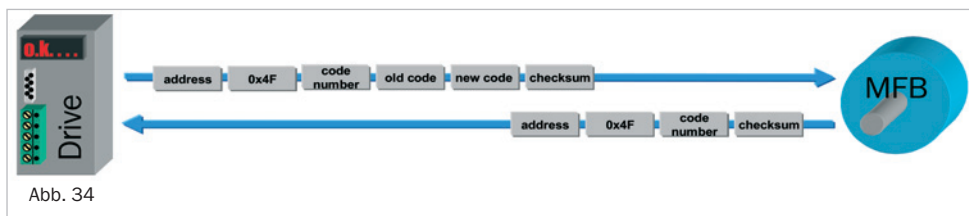
Neudefinitionen von Datenfeldern sind unabhängig vom **CE**-Bit nur unter Angabe des **Code**-Zugriffsschlüssels möglich. Dabei ist zu beachten, dass eine Änderung der Feldgröße nur beim zuletzt angelegten Feld (höchste Datenfeldnummer) möglich ist. Dasselbe gilt beim Löschen eines Datenfeldes (**E** = 0)

Verfügbaren Speicherbereich ermitteln ⇒ 4Eh



Mit Hilfe dieser Funktion lässt sich der für Datenfelddefinitionen zur Verfügung stehende Speicherplatz ermitteln. Der Parameter **free memory** gibt die Größe des noch nicht zugewiesenen Speichers an und wird entsprechend der Datenfelddefinition als Anzahl freier 16 Byte Blöcke angezeigt. Zusätzlich wird die Anzahl (**number of datafields**) bereits definierter Datenfelder ausgegeben.

Zugriffsschlüssel ändern ⇒ 4Fh



Der Motorhersteller kann für seine Kunden Zugriffswörter für bestimmte Speicherbereiche definieren. In diesen können dann Daten wie Kaufdatum, Serviceeinsätze, Projektnummern etc. hinterlegt werden.

Ein Zugriffsschlüssel kann nur unter Angabe seines aktuellen Wertes (**Old_Code**) neu bzw. undefiniert werden. Bei Auslieferung sind die vier möglichen Codes (**Code_Nr** = 0 ... 3) mit dem Wert **55h** vorbesetzt.

Beispiel zur Datenfelddefinition

Ausgangskonstellation

- Address 40h
- Codes 0 ... 3 55h
- freies EEPROM 128 Bytes

Kommunikation	Abbild des EEPROM	Erklärung															
(alle Zahlen in HEX!)																	
1. Zeile: Befehl 2. Zeile: MFB-Antwort		Verfügbaren Speicherbereich ermitteln <ul style="list-style-type: none"> • 8 x 16 Bytes freies EEPROM • 0 (keine) Datenfelder angelegt 															
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4E</td><td>0E</td></tr> </table> <table border="1"> <tr><td>40</td><td>4E</td><td>08</td><td>00</td><td>06</td></tr> </table>	40	4E	0E	40	4E	08	00	06		1. Datenfeld anlegen <ul style="list-style-type: none"> • Nummer 0 • Größe 32 Bytes • Code 0 verwenden • Schreibzugriff erlaubt • Codeangabe erforderlich ♦ Statusbyte = C9h							
40	4E	0E															
40	4E	08	00	06													
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4D</td><td>00</td><td>C9</td><td>55</td><td>91</td></tr> </table> <table border="1"> <tr><td>40</td><td>4D</td><td>00</td><td>C9</td><td>C4</td></tr> </table>	40	4D	00	C9	55	91	40	4D	00	C9	C4		2. Datenfeld anlegen <ul style="list-style-type: none"> • Nummer 1 • Größe 16 Bytes • Code 2 verwenden • Schreibzugriff erlaubt • Codeangabe erforderlich ♦ Statusbyte = E0h				
40	4D	00	C9	55	91												
40	4D	00	C9	C4													
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4D</td><td>01</td><td>E0</td><td>55</td><td>69</td></tr> </table> <table border="1"> <tr><td>40</td><td>4D</td><td>01</td><td>E0</td><td>EC</td></tr> </table>	40	4D	01	E0	55	69	40	4D	01	E0	EC		Zugriffsschlüssel ändern <ul style="list-style-type: none"> • Code Nr. 2 • bisher 55h • neu 66h 				
40	4D	01	E0	55	69												
40	4D	01	E0	EC													
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4F</td><td>02</td><td>55</td><td>66</td><td>3E</td></tr> </table> <table border="1"> <tr><td>40</td><td>4F</td><td>02</td><td>0D</td></tr> </table>	40	4F	02	55	66	3E	40	4F	02	0D		Daten speichern <ul style="list-style-type: none"> • ins 2. Datenfeld (Nr. 1) • an Adresse 5 • 2 Byte Daten • mit Code 2, (66h) • Datenwert (08h, 15h) 					
40	4F	02	55	66	3E												
40	4F	02	0D														
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4B</td><td>01</td><td>05</td><td>02</td><td>66</td></tr> </table> <table border="1" style="margin-left: 20px; margin-bottom: 5px;"> <tr><td>08</td><td>15</td><td>76</td></tr> </table> <table border="1"> <tr><td>40</td><td>4B</td><td>01</td><td>05</td><td>02</td><td>0D</td></tr> </table>	40	4B	01	05	02	66	08	15	76	40	4B	01	05	02	0D		
40	4B	01	05	02	66												
08	15	76															
40	4B	01	05	02	0D												

Geberstatus abfragen ⇒ 50h

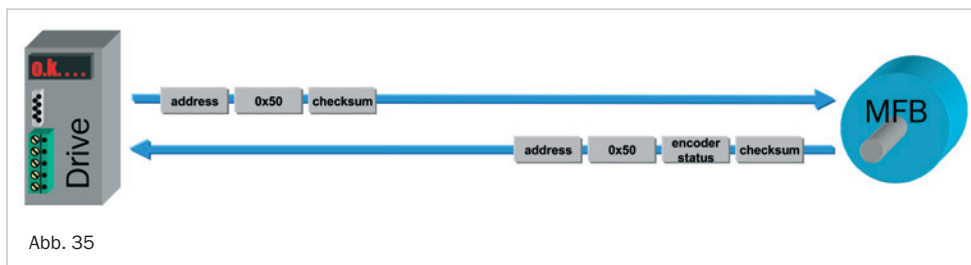


Abb. 35

Mit dieser Funktion kann der aktuelle MFB-Zustand abgefragt werden (z. B. nach dem Einschalten).

Diese Funktion wird auch benötigt, wenn bei der Ausführung eines Befehls das ERROR-Bit gesetzt wurde, der eigentliche Befehl jedoch fehlerfrei bearbeitet wurde (pending error). Dies ist beispielsweise dann der Fall, wenn die Temperaturüberwachung eine Bereichsüberschreitung erkannt hat.

Die tatsächlich verfügbaren Fehlercodes variieren je nach MFB-Typ und sind jeweils im Datenblatt angegeben. Nach dem Einschalten des MFBs und nach Ablauf der Initialisierungszeit wird empfohlen, zuerst eine fehlerfreie Initialisierung des MFBs mit Hilfe der Statusabfrage zu überprüfen. Es können während der Initialisierungsphase bis zu vier Fehlermeldungen zwischengespeichert werden.

Typenschild auslesen ⇒ 52h

Mit diesem Befehl werden die aktuelle Konfiguration der Schnittstelle, wie Baudrate und Parity, sowie die Hardwarekonfiguration des MFB angegeben.

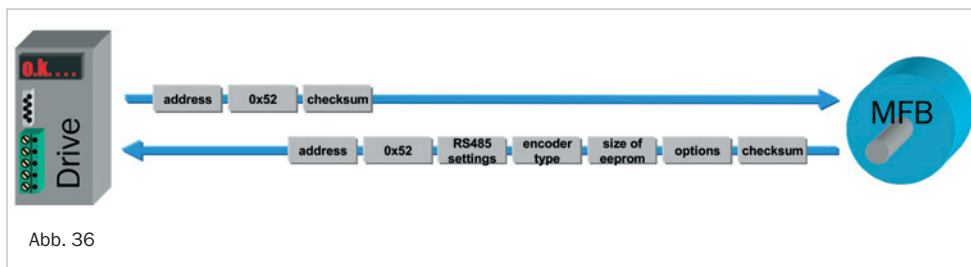
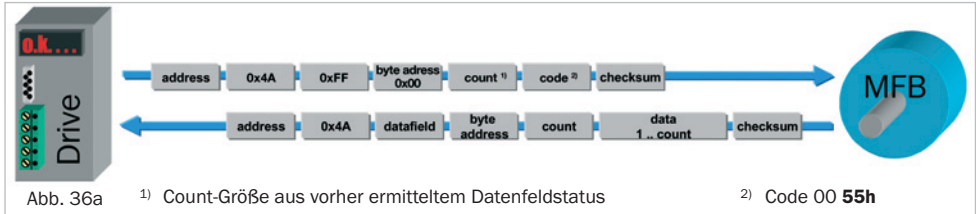


Abb. 36

- **RS485 settings**
siehe Command „57h“ (s. S. 31)
- **Encoder type**
„encoder type ungleich 0xFF“: siehe produktspezifisches Datenblatt
„encoder type 0xFF“: die Encoderbeschreibung kann direkt aus dem EEPROM des Encoders ausgelesen werden. (Siehe Ablauf für das Auslesen „Typenschild mit Encoder type“ 0xFF)
- **Size of eeprom:**
Die Größe des im MFB verwendeten EEPROMS = $EEPR_Size * 16$
- **Options**
spezifische Hardware-/Softwareerweiterungen
HIPERFACE®-Version

Ablauf zum Auslesen des Typenschild smit „Encoder type“ 0xFF

1. Status und Größe des Datenfeldes 0xFF für Encoderbeschreibung ermitteln. Siehe Befehl 4Ch (Seite 24).
2. Datenfeld 0xFF auslesen



Inhalt des Datenfeldes FFh (Typenschild)

Das Datenfeld FFh (Typenschild) besteht aus zwei Bereichen:

A. Geberbeschreibung (Adresse 00 ... 1B)

Adresse	Byte-Nr.	Beschreibung	Bytes
00	1	Checksum	1
01	2	Bit 1 1 = Zählweise bipolar; 0 = Zählweise unipolar	1
		Bit 0 1 = linear 0 = rotativ	
02 - 05	3 - 6	Periodenlänge in nm (1 ..n)	Anzahl der Perioden/Umdr. (1 ..n)
06 - 09	7 - 10	Codierter Messbereich in Anzahl Perioden (1 ..n)	Anzahl der codierten Umdr. (1 ..n)
0A - 1B	11 - 28	Bezeichnung, max. 18 Stellen linksbündig, ASCII	

B. Parameterselektor (Adresse 1C ... 1D für 1. Parameterselektor)

Durch einen zwei Byte breiten „Parameterselektor“ werden die zur Verfügung stehenden Grenzwerte angegeben. Jedes Bit indiziert einen 2-Byte-Parameter. Gesetzte Bits zeigen an, dass der betreffende Parameter für den jeweiligen Geber gültig ist. Sind mehr als 15 Parameter definiert, wird ein weiterer Parameterselektor benötigt. Dass in diesem Grenzwerte definiert sind, die für den betreffenden Geber gelten, wird dadurch angezeigt, dass im vorhergehenden Parameterselektor das Bit 15 gesetzt ist. Der für den jeweiligen Encoder gültige Parameterselektor ist im zugehörigen Datenblatt des Encoders aufgeführt. **Beispiel für den Aufbau eines Parameterselektors:**

Parameterselektor	H-Byte															L-Byte				
	Nr. Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Parameter-Bezeichnung																				
Enable next selector																				
tbd																				
tbd																				
tbd																				
tbd																				
tbd																				
tbd																				
tbd																				
tbd																				
tbd																				
a max. H-Byte																				
a max. H-Byte																				
v max. L-Byte																				
v max. H-Byte																				
Vektorl. max. L-Byte																				
Vektorl. max. H-Byte																				
Vektorl. min. L-Byte																				
Vektorl. min. H-Byte																				
Vektorl. Ch.-Nr. L-Byte																				
Vektorl. Ch.-Nr. H-Byte																				
Senderstr. max. L-Byte																				
Senderstr. max. H-Byte																				
Senderstr. min. L-Byte																				
Senderstr. min. H-Byte																				
Senderstr. Ch.-Nr. L-Byte																				
Senderstr. Ch.-Nr. H-Byte																				
Temp. max. L-Byte																				
Temp. max. H-Byte																				
Temp. min. L-Byte																				
Temp. min. H-Byte																				
Temp. Ch.-Nr. L-Byte																				
Temp. Ch.-Nr. H-Byte																				

C. Grenzwerte (Adresse 1E...3B für 1.Set Grenzwerte)

Alle Grenzwerte sind als Integer-Wert definiert. Einige Grenzwerte von Parametern, wie z. B. die interne Geber-temperatur beziehen sich auf Informationen, die über den Befehl 44h, „Analogwert lesen“ (falls verfügbar) abgefragt werden können. Für diese Grenzwerte ist als erstes Wort der Analogkanal angegeben, auf dem die jeweilige Information beim betreffenden Geber gelesen werden kann.

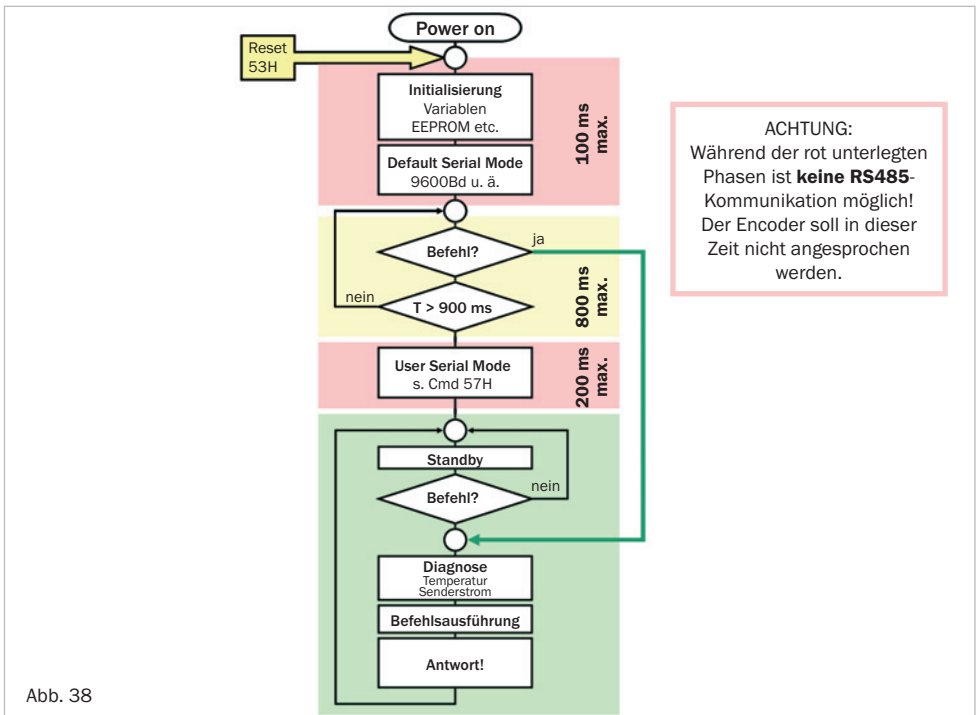
Beim Überschreiten der angegebenen Warn Grenzen wird vom Messsystem in der Befehlsantwort das Statusbit gesetzt.

Reset → 53h



Keine MFB-Quittung.

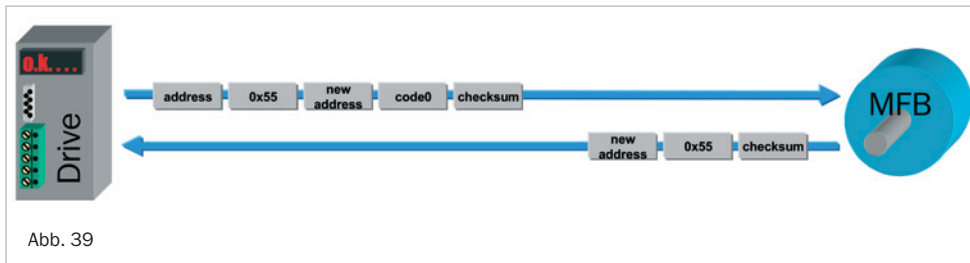
Das MFB durchläuft hierbei die gleiche Initialisierungsphase wie beim Einschalten der Versorgungsspannung. Die vom Benutzer einmal geänderten Parameter, wie z. B. die MFB-Adresse bleiben dabei erhalten!



Abweichende Sequenzen siehe Encoder-Datenblatt.

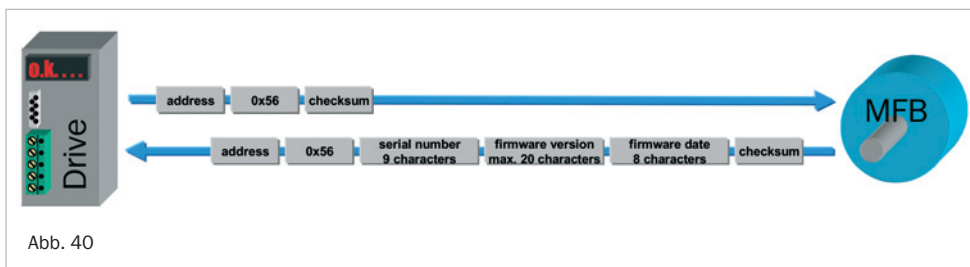
Adresse vergeben ⇒ 55h

Vergabe einer neuen Adresse an einen Teilnehmer. Nach Ausführung dieses Befehls muss das MFB mit seiner neuen Adresse **New_Address** angesprochen werden.



Seriennummer & Programmversion lesen ⇒ 56h

Mit dieser Funktion wird die 9stellige Seriennummer des MFB, sowie Version und Datum des internen Programms ausgelesen.

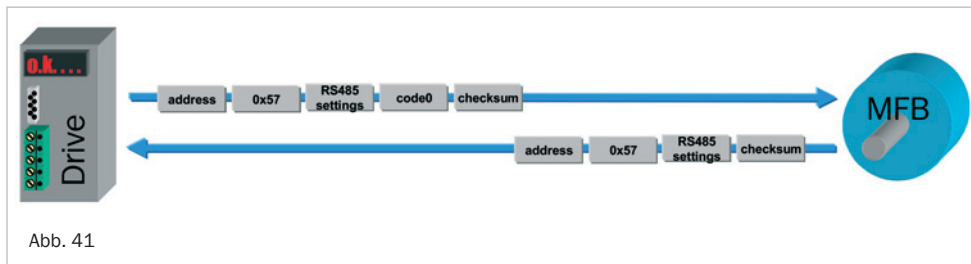


Die Programmversion „**firmware version**“ besteht aus **bis zu** 20 Zeichen.

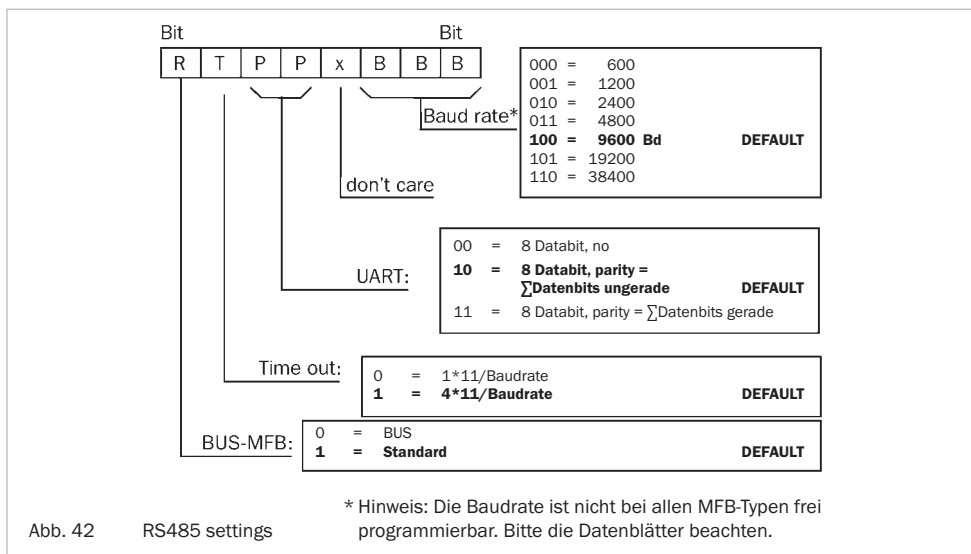
Das 8-stellige „**firmware date**“ spezifiziert den Zeitpunkt der Programmerstellung und ist im Format „TT.MM.JJ“ angegeben.

Serielle Schnittstelle konfigurieren ⇒ 57h

Mit dieser Funktion lassen sich die Übertragungsparameter des asynchronen Parameterkanals in weiten Grenzen an die Erfordernisse der Applikation anpassen.



Aufbau von RS485 settings



Grundsätzlich ist jedes HIPERFACE®-MFB nach dem Einschalten für die Zeit von ca. einer Sekunde auf die Default-Werte eingestellt. Trifft während dieser Sekunde eine plausible Information ein, so bleibt diese Einstellung erhalten. Ist dies nicht der Fall, so wird die über diese Funktion festgelegte Einstellung übernommen. Dieses Verfahren ist notwendig, um mit Geräten, deren Einstellung unbekannt ist, kommunizieren zu können.

Die eingestellte Baudrate sowie die Definition des Parity werden erst durch einen GEBERRESET (53h) bzw. Aus- und Einschalten des MFB aktiv.

Position setzen mit interner Synchronisation ⇒ 63h

Für besondere Anwendungsfälle gibt es auch MFB's ohne eigene Lagerung, so genannte KITs, im Produkt-Portfolio von SICK-STEGMANN. Hierbei werden die Codescheibe (Maßverkörperung), die Auswerte-Elektronik, sowie gegebenenfalls die Multiturn-Einheit, getrennt geliefert und vom Anwender selbst angebaut. Neben der korrekten mechanischen Ausrichtung der Teile, welche durch die konstruktive Auslegung der Gebersysteme einfach gewährleistet ist, ist es auch notwendig, die verschiedenen Codespuren elektronisch aufeinander auszurichten, die Spuren zueinander zu synchronisieren. Dies erfolgt durch Aufruf dieses Befehls.

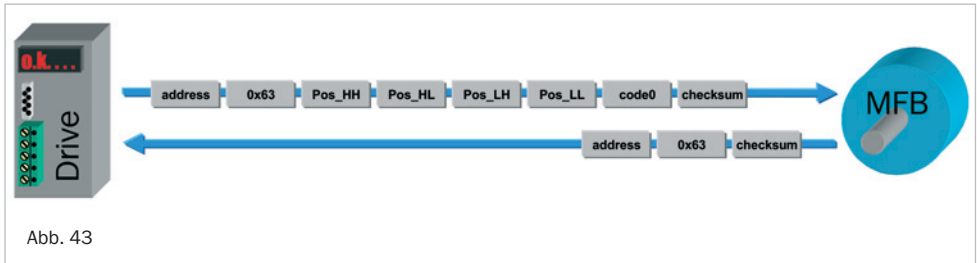


Abb. 43

Das MFB ermittelt die Werte der internen Codespuren und berechnet daraus entsprechende interne Winkeloffsets, es erzeugt sich eine virtuelle 0.

Zusammen mit diesen Winkeloffsets wird der im Protokoll angegebene Positionsoffset (Pos_HH .. Pos_LL) unverlierbar im EEPROM des MFB abgespeichert.

Der Positionsoffsetwert wird im Format unsigned long mit dem LSB rechtsbündig in Pos_LL übertragen.

Achtung: Dieser Befehl darf nur **bei stehender Geberwelle** aufgerufen werden, da sonst fehlerhafte Winkeloffsetwerte ermittelt werden können.

Serielle Schnittstelle temporär konfigurieren ⇒ 67h

Mit dieser Funktion lassen sich die Übertragungsparameter wie bei Befehl 57H konfigurieren. Die Änderung ist im flüchtigen Speicher abgelegt und nach einem Neustart oder Reset nicht gespeichert. Die Antwort des Encoders wird noch in der ursprünglichen Schnittstelleneinstellung geschickt. Unmittelbar nach dem Senden der Antwort wird vor Beginn des Timeout auf die gewählte Einstellung umgeschaltet.

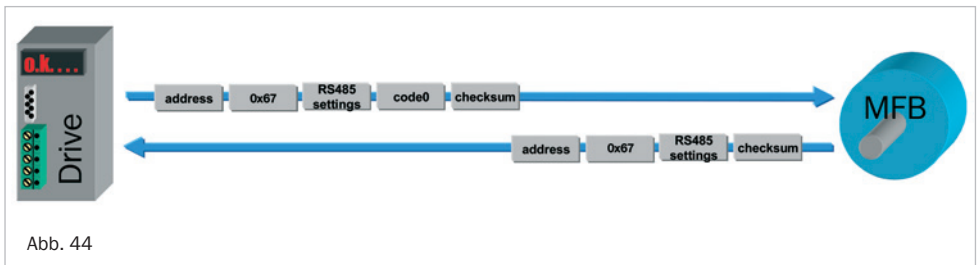


Abb. 44

Die eingestellte Schnittstellenkonfiguration ist nur bis zum nächsten GEBERRESET (53h) bzw. Aus- und Einschalten des MFB aktiv.

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.

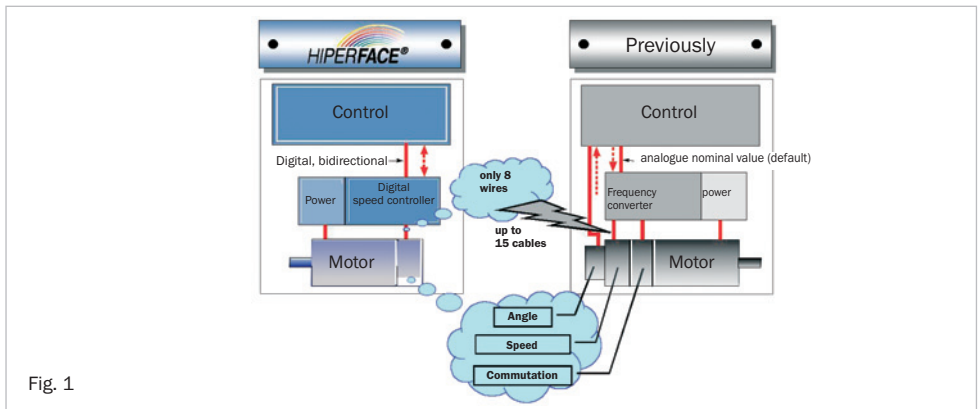


Fig. 1

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 **High Performance Interface** 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®

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.

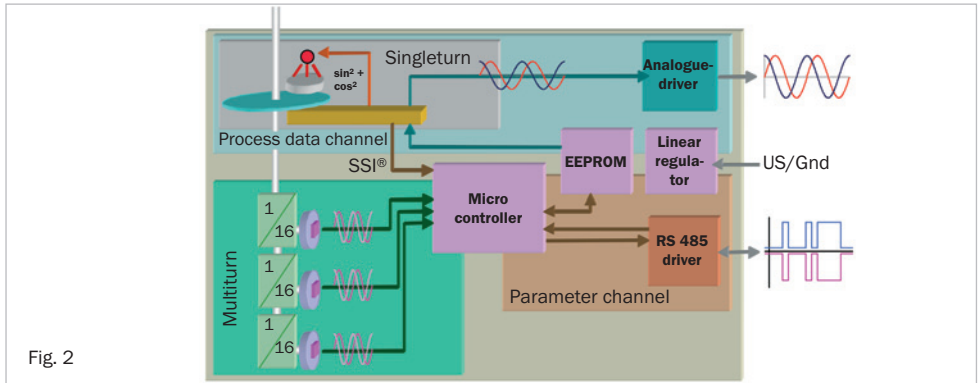
HIPERFACE® motor feedback systems are a mix of incremental encoders and absolute encoders and combine the advantages of both encoder types.

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 formed 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.

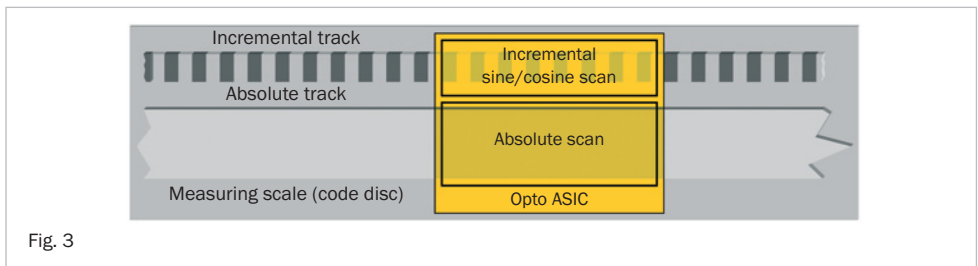
Internal arrangement

Typical arrangement of a HIPERFACE® motor feedback system



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.



Usually, in a motor feedback system, **2ⁿ 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 2ⁿ periods.

HIPERFACE®**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.

The screen should be connected with the protective earth. A separate equipotential bonding conductor must be used, if large equalising currents are expected in systems.

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 check-sum. 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.

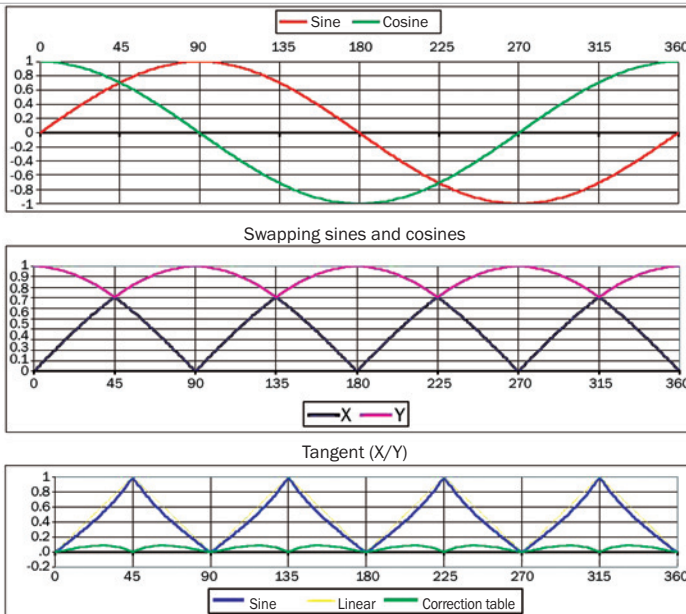


Fig. 4 Arc tangent interpolation method

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 $\pm 4/32$ 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

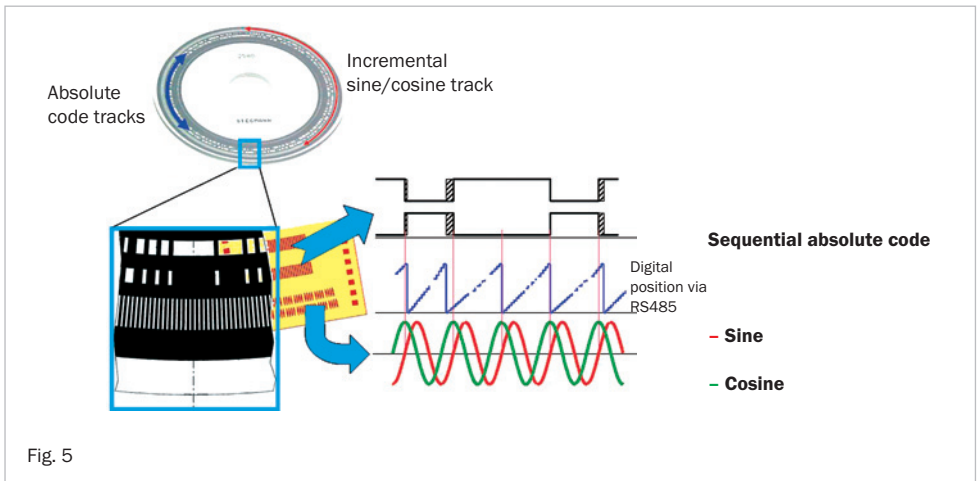


Fig. 5

The synchronisation now describes a method which can compensate these effects within wide limits ($\pm 1/2$ 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.

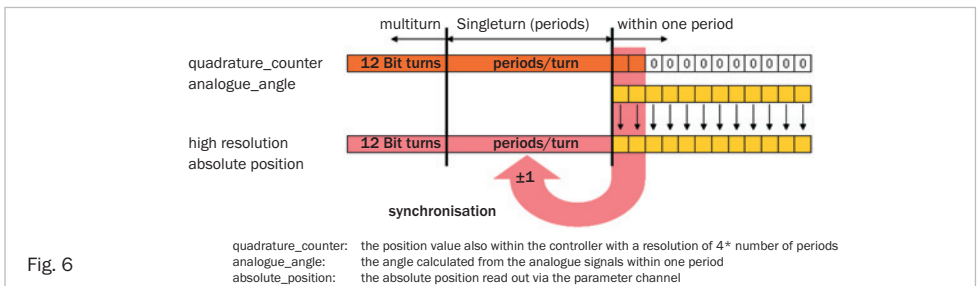
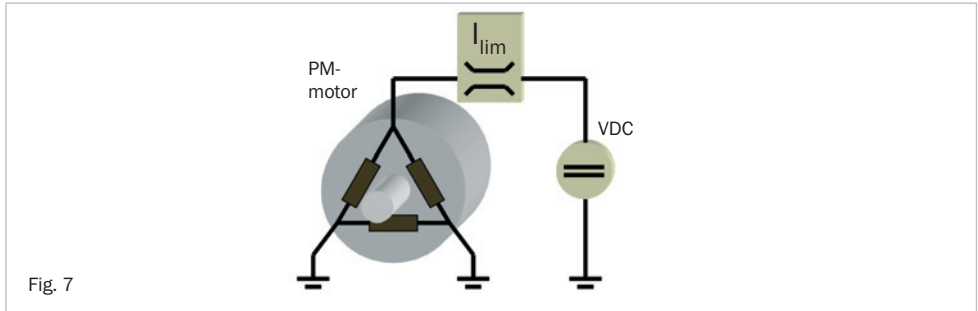


Fig. 6

Commissioning/Commutation Adjustment Using a HIPERFACE® Motor Feedback System



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

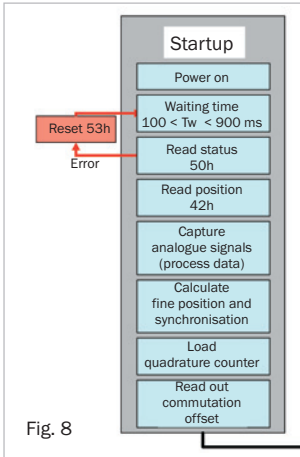
1. Using the data fields

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



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.

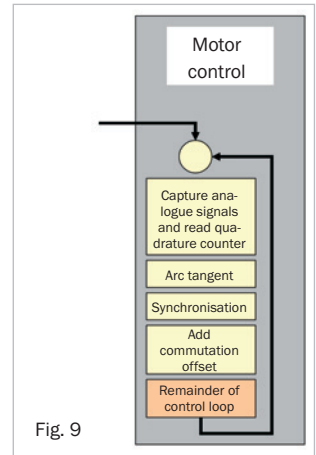
Speed controller

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.



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 $\text{Sin}^2 + \text{Cos}^2$ 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.

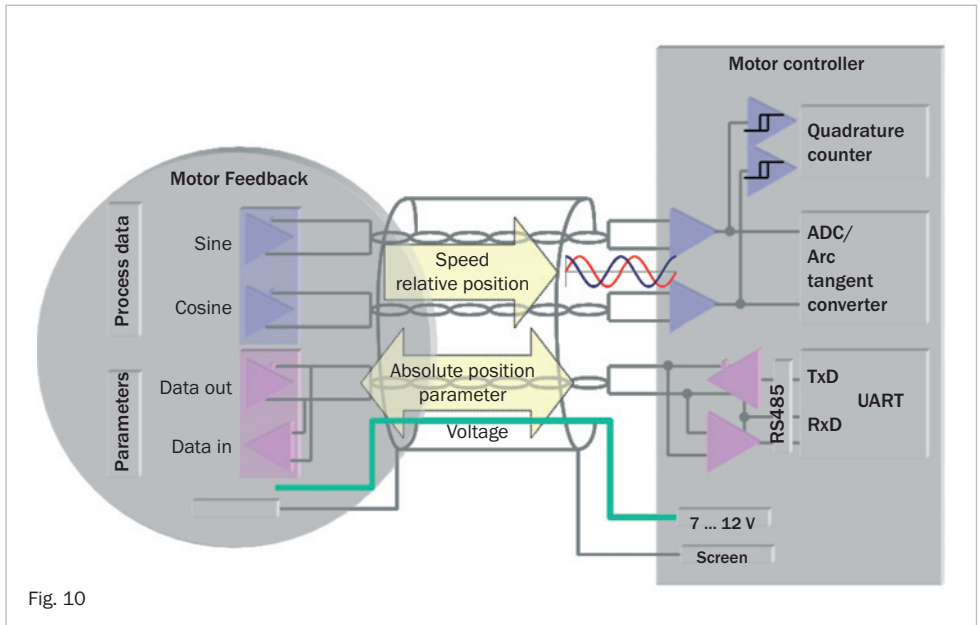


Fig. 10

HIPERFACE®

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

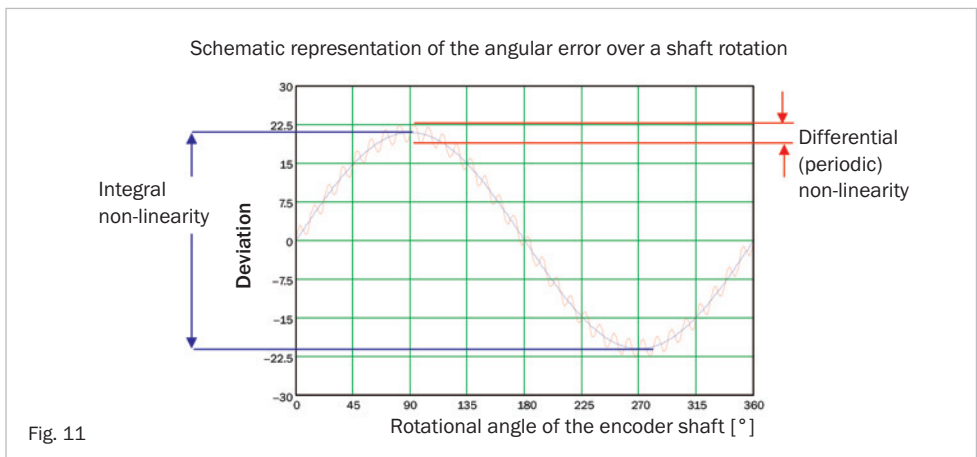
This value describes the deviations over a mechanical rotation of the shaft.

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.

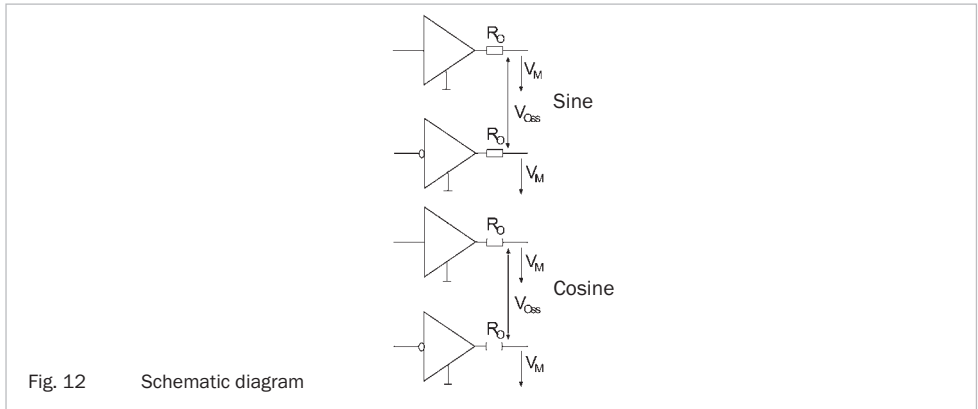
Error limits for the digital absolute value

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.



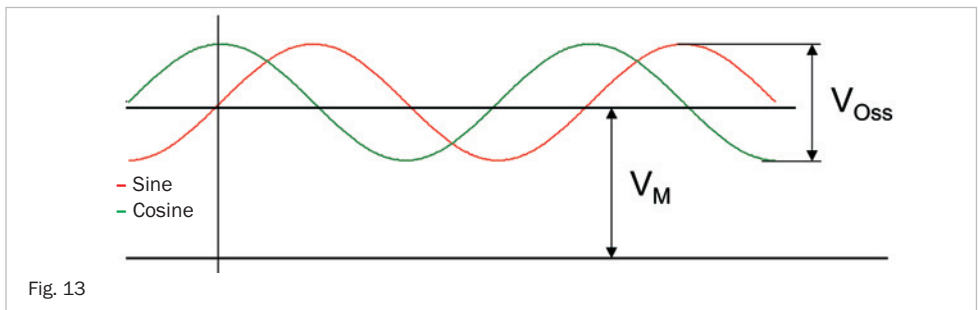
Process data channel (analogue outputs)



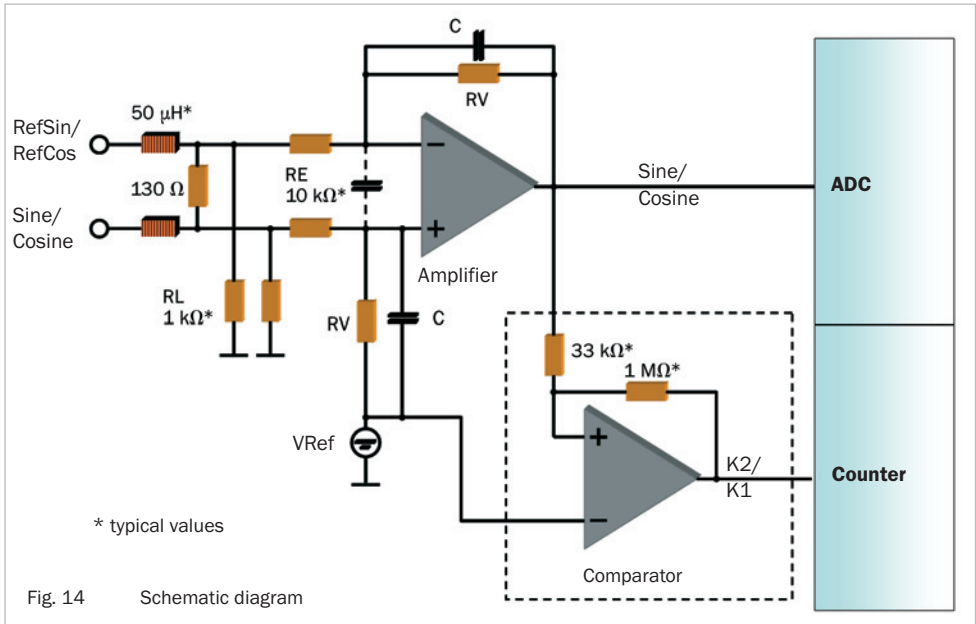
Electrical parameters of the process data channels

(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	$> = \pm 7$ mA	I_{Omin}



Typical Input Circuit:



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 $\leq 1\%$.

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

$$\varphi \cong \arctan(\omega * RV * C)$$

Output amplitude

$$U_a \cong \frac{1}{\sqrt{1 + (\omega * RV * C)^2}} * U_e$$

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:

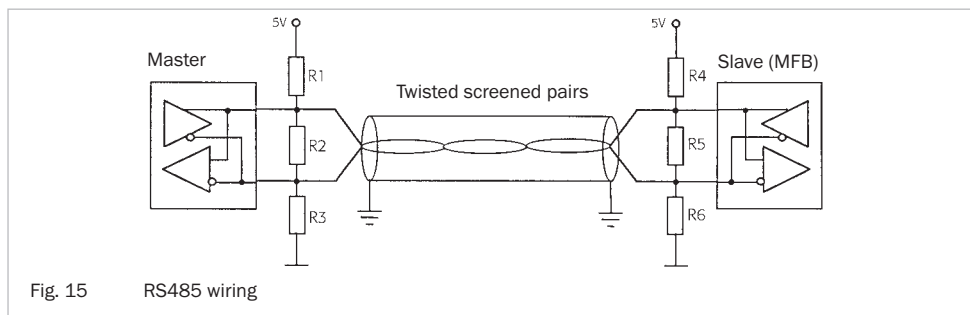


Fig. 15 RS485 wiring

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.

HIPERFACE®

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.

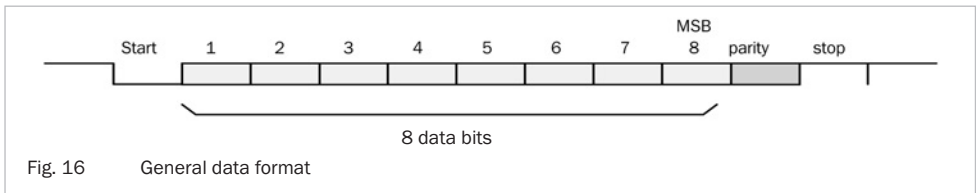


Fig. 16 General data format

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

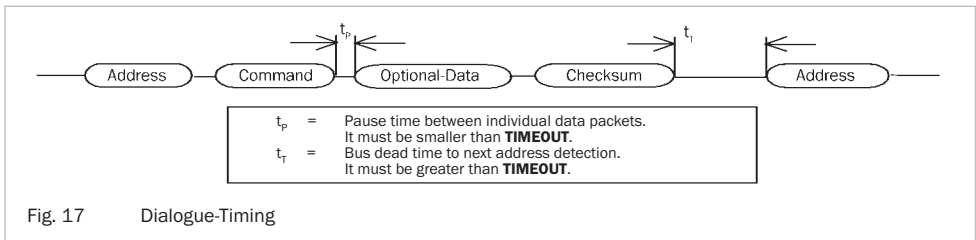


Fig. 17 Dialogue-Timing

- 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.

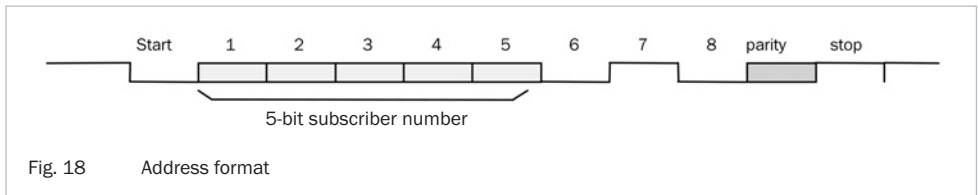
ADDRESS Format

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.

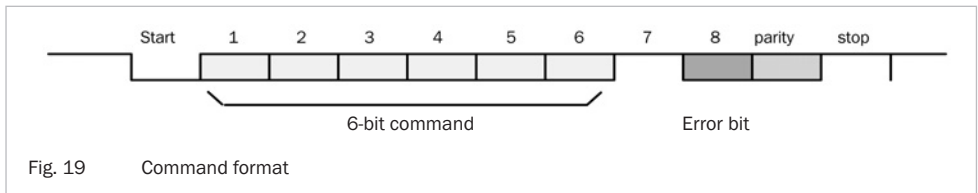
COMMAND-Format

Fig. 19 Command format

Error Handling**Error message**

- 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)

Warning message

- 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 direct 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.
- If this detects critical parameters which, however, have no direct effect on the command currently processed, then the MFB generates, with the error bit set, the expected response protocol.

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
67h	Temporarily configure the serial interface		5	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.

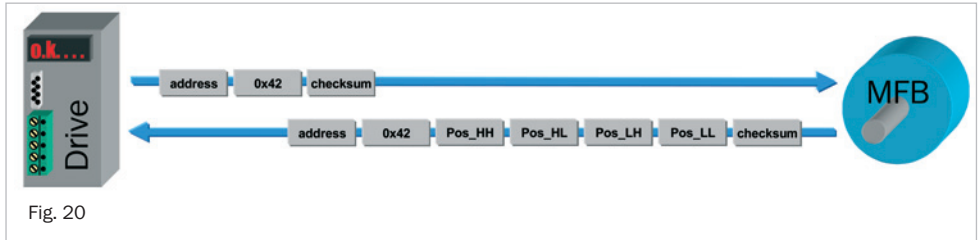


Fig. 20

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.

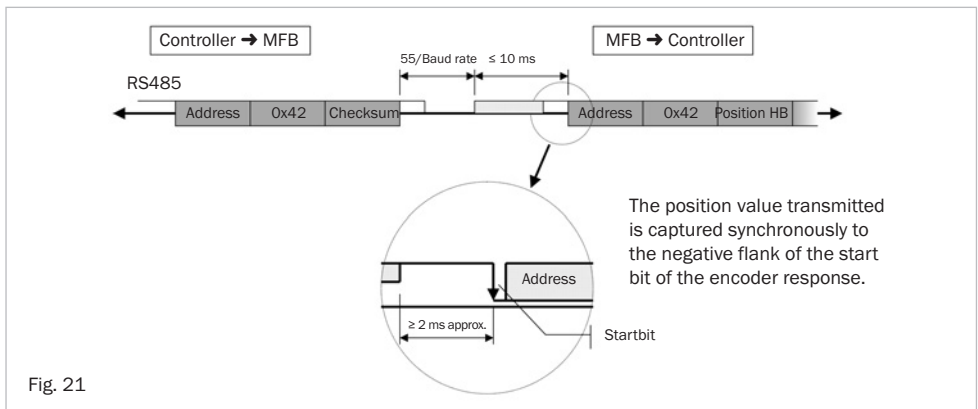


Fig. 21

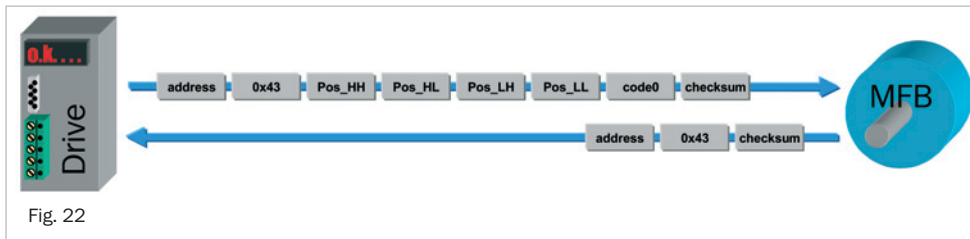
The absolute position value serves to uniquely identify one of the analogue periods and, therefore, always has a resolution of 5 bits/period. To check the position counted within the controller, the absolute position of the MFB can be cyclically read out. Consequently, it is recommended to use only the 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).

HIPERFACE®

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.



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.

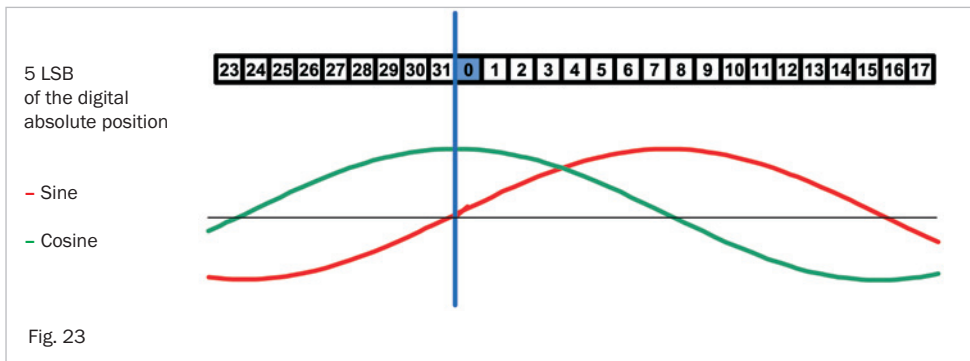
Caution:

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:



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

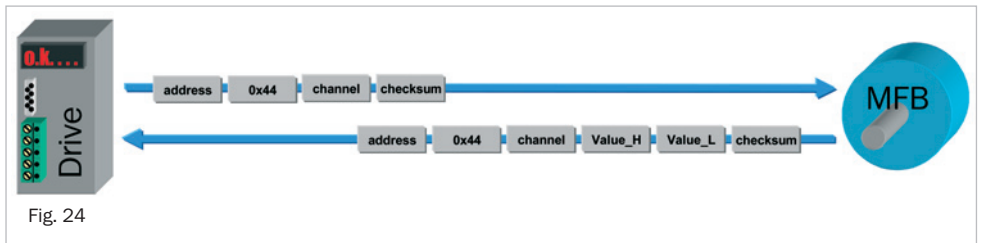


Fig. 24

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.

HIPERFACE®

Counter Functions

An incremental 24-bit counter is implemented in the encoder firmware.

The commands listed below permit simple implementation of a pseudo hour meter. 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

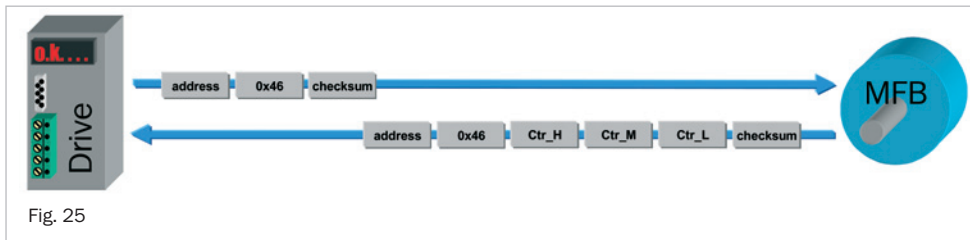


Fig. 25

Reads out the current count.

Increase counter ⇒ 47h

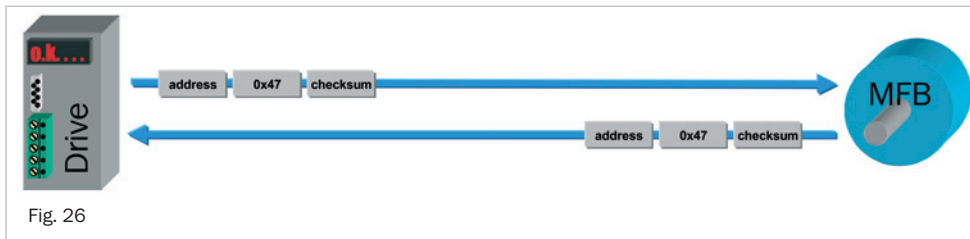


Fig. 26

The internal 24-bit counter is increased by 1. Depending upon the MFB type, a counter overflow will be indicated either by the MFB setting the ERROR Bit or an answer of "08h" to the Status request command.

Delete counter ⇒ 49h

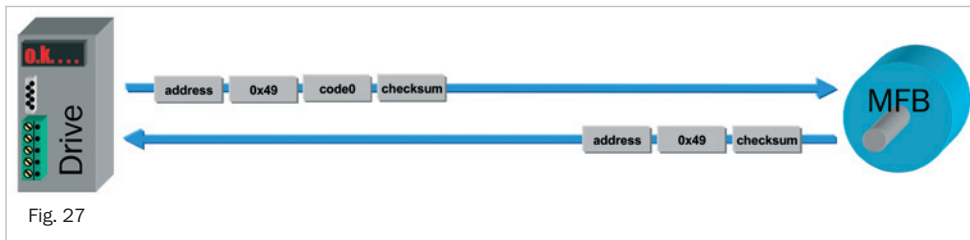


Fig. 27

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.

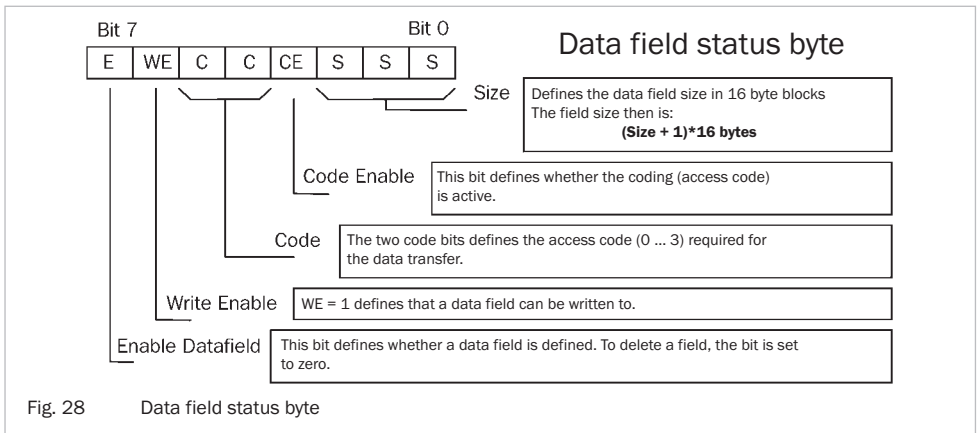
To store data in the MFB, the EEPROM must first be divided into individual data fields with block sizes between 16 and 128 bytes (command 4Dh). This also defines the assignment to one of the 4 access codes deposited in the MFB. As regards the access codes (default = 55h), they can be modified with the 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



HIPERFACE®

Read data ⇒ 4Ah

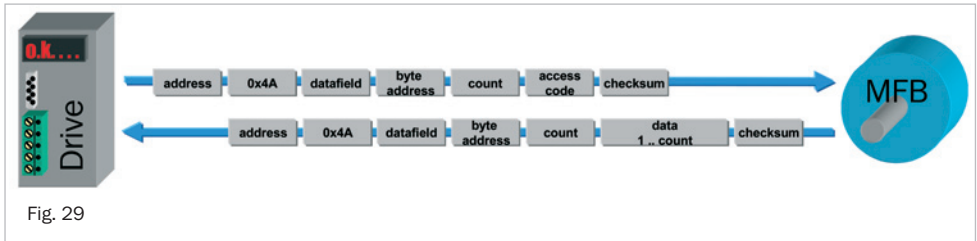


Fig. 29

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

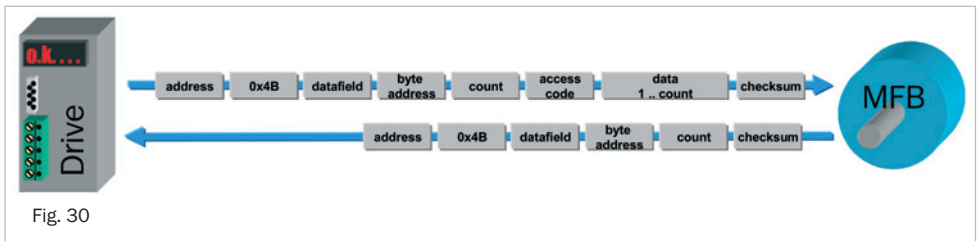


Fig. 30

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

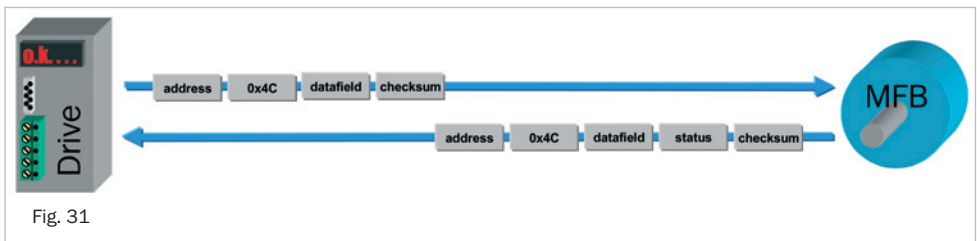


Fig. 31

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.

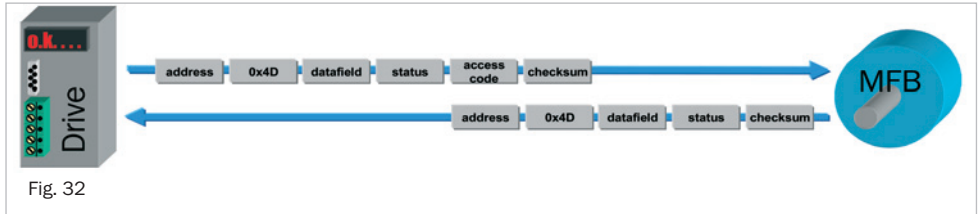


Fig. 32

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

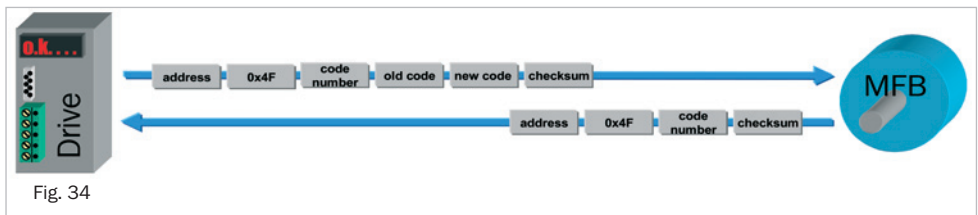


Fig. 34

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!)																	
1 st line: command 2 nd line: MFB response		Determine available memory area • 8 x 16 bytes free EEPROM • 0 (no) data fields created															
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4E</td><td>0E</td></tr> </table> <table border="1"> <tr><td>40</td><td>4E</td><td>08</td><td>00</td><td>06</td></tr> </table>	40	4E	0E	40	4E	08	00	06		Create 1 st data field • Number 0 • Size 32 bytes • Use code 0 • Write access allowed • Code must be given ♦ Status byte = C9h							
40	4E	0E															
40	4E	08	00	06													
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4D</td><td>00</td><td>C9</td><td>55</td><td>91</td></tr> </table> <table border="1"> <tr><td>40</td><td>4D</td><td>00</td><td>C9</td><td>C4</td></tr> </table>	40	4D	00	C9	55	91	40	4D	00	C9	C4		Create 2 nd data field • Number 1 • Size 16 bytes • Use code 2 • Write access allowed • Code must be given ♦ Status byte = E0h				
40	4D	00	C9	55	91												
40	4D	00	C9	C4													
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4D</td><td>01</td><td>E0</td><td>55</td><td>69</td></tr> </table> <table border="1"> <tr><td>40</td><td>4D</td><td>01</td><td>E0</td><td>EC</td></tr> </table>	40	4D	01	E0	55	69	40	4D	01	E0	EC		Change access code • Code no. 2 • previously 55h • new 66h				
40	4D	01	E0	55	69												
40	4D	01	E0	EC													
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4F</td><td>02</td><td>55</td><td>66</td><td>3E</td></tr> </table> <table border="1"> <tr><td>40</td><td>4F</td><td>02</td><td>0D</td></tr> </table>	40	4F	02	55	66	3E	40	4F	02	0D	Save data • in the 2 nd data field (no. 1) • at address 5 • 2 bytes Data • with Code 2, (66h) • data value (08h, 15h)						
40	4F	02	55	66	3E												
40	4F	02	0D														
<table border="1" style="margin-bottom: 5px;"> <tr><td>40</td><td>4B</td><td>01</td><td>05</td><td>02</td><td>66</td></tr> </table> <table border="1" style="margin-left: 20px; margin-bottom: 5px;"> <tr><td>08</td><td>15</td><td>76</td></tr> </table> <table border="1"> <tr><td>40</td><td>4B</td><td>01</td><td>05</td><td>02</td><td>0D</td></tr> </table>	40	4B	01	05	02	66	08	15	76	40	4B	01	05	02	0D		Save data • in the 2 nd data field (no. 1) • at address 5 • 2 bytes Data • with Code 2, (66h) • data value (08h, 15h)
40	4B	01	05	02	66												
08	15	76															
40	4B	01	05	02	0D												

Read encoder status ⇒ 50h

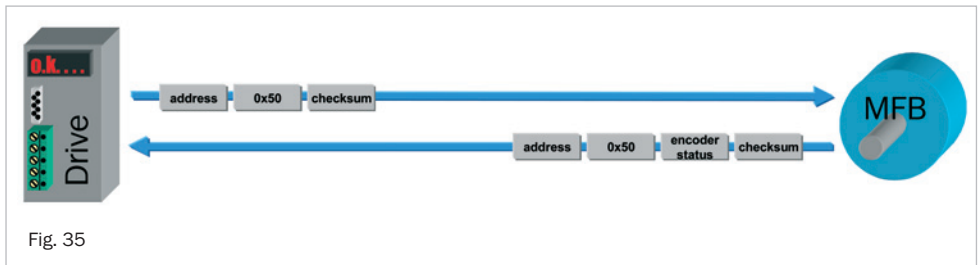


Fig. 35

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.

Once the initialisation time has elapsed, following power up of the MFB, it is recommended to check that initialisation proceeded without errors using the Status request command. In the event of errors occurring during the initialisation phase, up to 4 error messages will be stored in the MFB.

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.

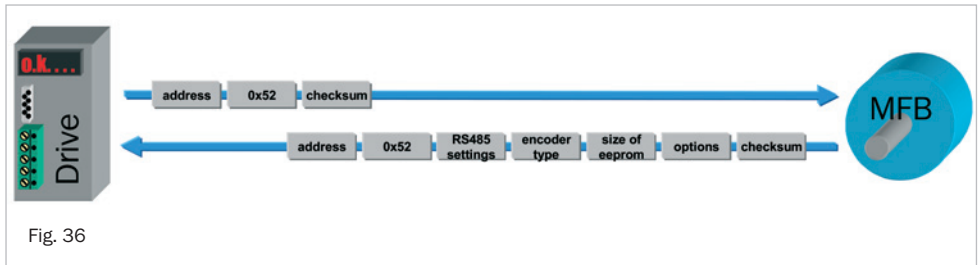


Fig. 36

- **RS485 settings**

see command “57h” (see p. 59)

- **Encoder type**

Encoder type” unequal 0xFF: see product data sheet

“Encoder type” = 0xFF : the encoder description can be directly read out from the encoder’s EEPROM.
(See sequence for reading out “Type label with encoder type” 0xFF)

- **Size of EEPROM:**

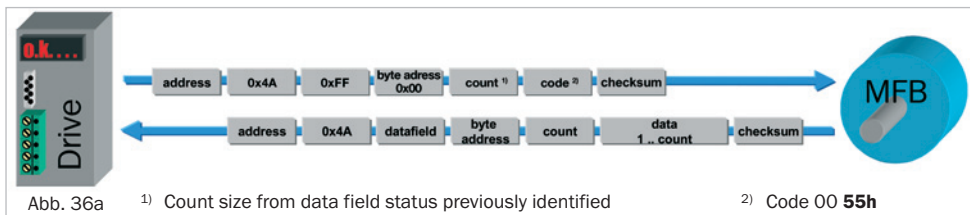
The size of EEPROM used in the MFB = EEPR_Size * 16

- **Options**

specific hardware/software upgrades
HIPERFACE® version

Sequence for reading out “Type label with encoder type” 0xFF

1. Determine status and size of the data field 0xFF for encoder description. See figure 4Ch (page 54)
2. Read out data field 0xFF



Contents of the data field FFh (type label)

The data field FFh (type label) consists of two areas:

A. Encoder description (address 00...1B)

Address	Byte no.	Description	Bytes
00	1	Checksum	1
01	2	Bit 1 1 = bipolar counting; 0 = unipolar counting	1
		Bit 0 1 = linear 0 = rotary	
02 - 05	3 - 6	Period length in nm (1 ..n) No. of periods/rev. (1 ..n)	4
06 - 09	7 - 10	Coded measuring range in no. of periods (1 ..n) No. of coded revolutions (1 ..n)	4
0A - 1B	11 - 28	Designation, 18 digits max. left-justified, ASCII	4

B. Parameter selector (address 1C ... 1D for 1st parameter selector)

A “parameter selector” two bytes wide specifies the limit values available. Each bit indicates a 2-byte parameter. Set bits show that the parameter concerned is valid for the particular encoder. If more than 15 parameters are defined, a further parameter selector will be required. The fact that it defines limit values, which apply to the particular encoder, is shown by the bit 15 in the previous parameter selector. The parameter selector valid for the particular encoder is listed in the data sheet for the encoder. **Example of the structure of a parameter selector:**

Parameter designation	H-Byte															L-Byte															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Enable next selector	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd
a max. H-Byte																															
a max. H-Byte																															
v max. L-Byte																															
v max. H-Byte																															
Vektorl. max. L-Byte																															
Vektorl. max. H-Byte																															
Vektorl. min. L-Byte																															
Vektorl. min. H-Byte																															
Vektorl. Ch.-Nr. L-Byte																															
Vektorl. Ch.-Nr. H-Byte																															
Senderstr. max. L-Byte																															
Senderstr. max. H-Byte																															
Senderstr. min. L-Byte																															
Senderstr. min. H-Byte																															
Senderstr. Ch.-Nr. L-Byte																															
Senderstr. Ch.-Nr. H-Byte																															
Temp. max. L-Byte																															
Temp. max. H-Byte																															
Temp. min. L-Byte																															
Temp. min. H-Byte																															
Temp. Ch.-Nr. L-Byte																															
Temp. Ch.-Nr. H-Byte																															

C. limit values (address 1E ... 3B for 1st set of limit values)

All limit values are defined as integer values. Some limit values of parameters, such as the internal encoder temperature, relate to information which can be queried via the command 44h “Read analogue value” (if available). For these limit values, the first word specified is the analogue channel on which the respective information can be read for the encoder concerned.

When the specified warning limits are exceeded, the measuring system sets the status bit in the command response.

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!

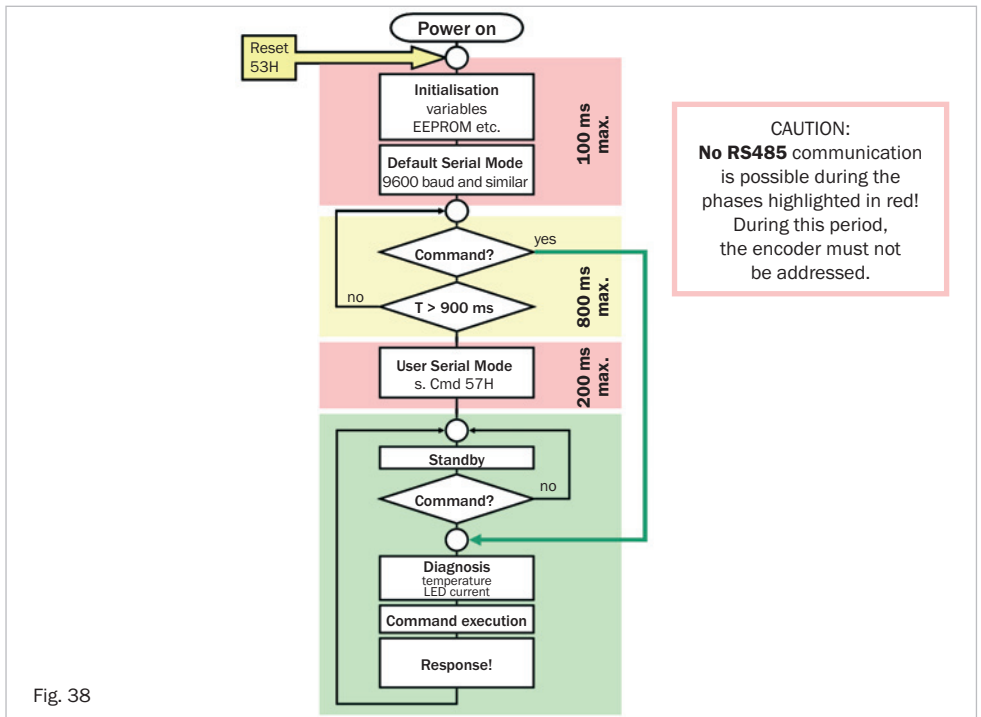


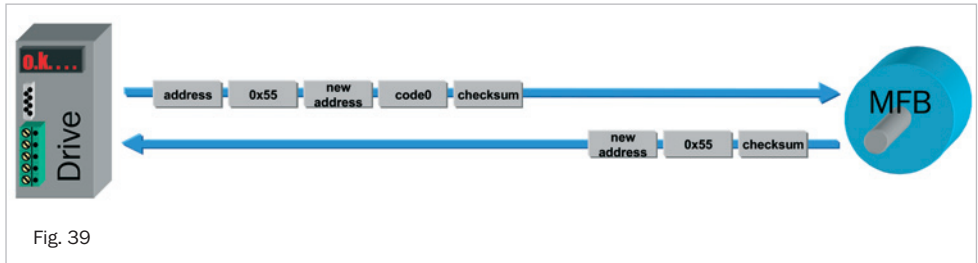
Fig. 38

See encoder data sheet for deviating sequences.

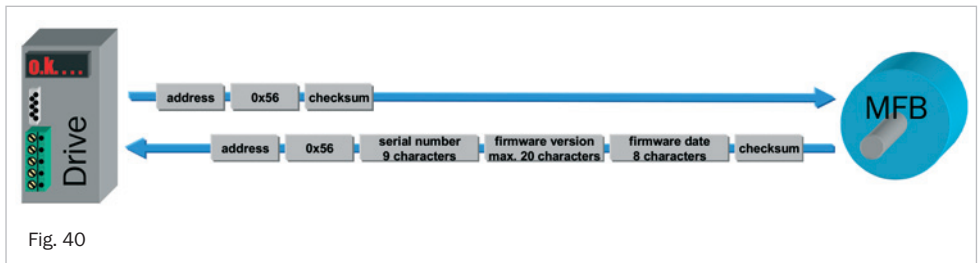
HIPERFACE®

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**.

**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.



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.

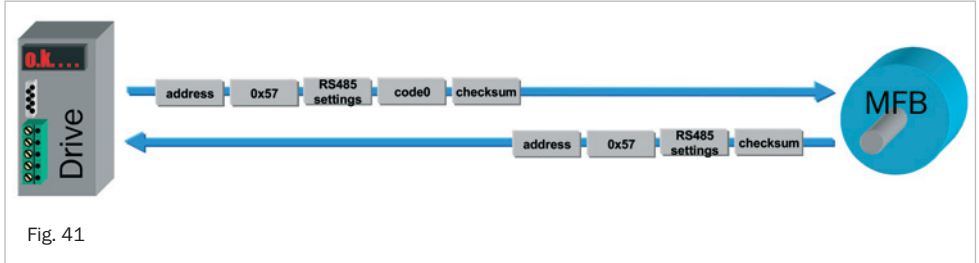


Fig. 41

Structure of RS485 settings

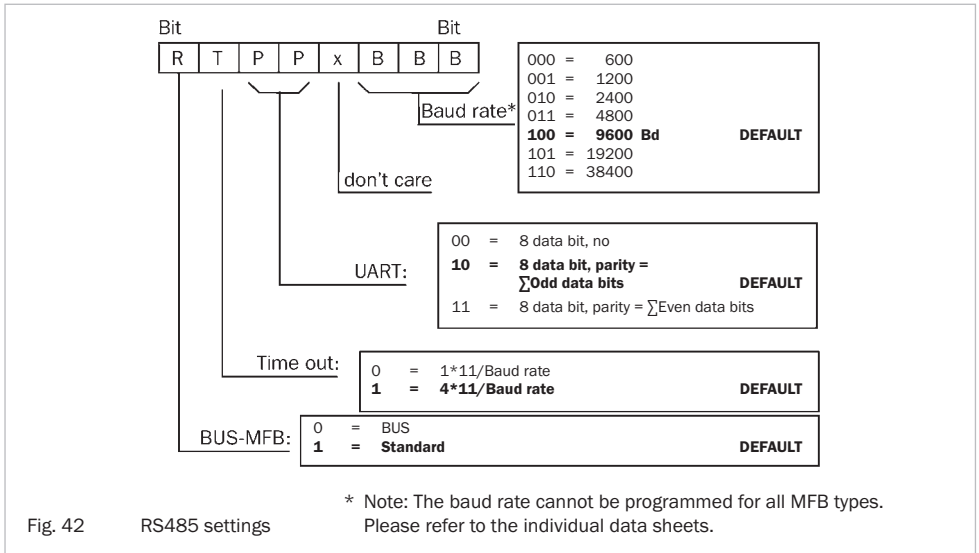


Fig. 42 RS485 settings

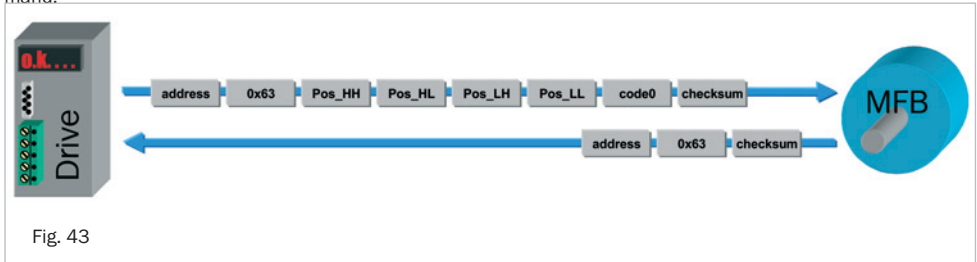
Basically, each HIPERFACE® MFB, after power-up, is set to the default values for a time of 1 second approx. If, during this second, a plausible piece of information arrives, this setting is retained. If this is not the case, the setting defined via this function is adopted. This procedure is necessary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

HIPERFACE®

Set position with internal synchronisation ⇒ 63h

For particular applications there are also encoders without their own bearings, so-called KITS, in the SICK-STEG-MANN product portfolio. For this, the code disc (measuring scale), the evaluation electronics as well as (if required) the multturn 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.



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.

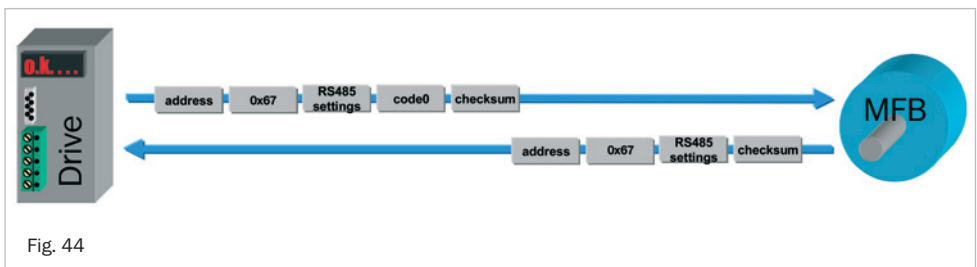
The position offset value is transmitted as unsigned long with the LSB right-justified in Pos_LL.

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

Temporarily configure the serial interface ⇒ 67h

With this function, the transmission parameters can be configured, in the same way as for command 57H.

The change is stored in the volatile memory and not saved after a restart or reset. Moreover, the response of the encoder is sent in the original interface setting. Immediately after sending the response, before start of the timeout, the new setting selected is switched to.



The set interface configuration is only active until the next ENCODER RESET (53h) resp. switching the MFB off and on.

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 3215-4900
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 +49 211 5301-250
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

India

Phone +91-22-4033 8333
E-Mail info@sick-india.com

Israel

Phone +972-4-999-0590
E-Mail info@sick-sensors.com

Italia

Phone +39 02 27 43 41
E-Mail info@sick.it

Japan

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

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

Republic of Korea

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

Republika Slovenija

Phone +386 (0)1-47 69 990
E-Mail office@sick.si

România

Phone +40 356 171 120
E-Mail office@sick.ro

Russia

Phone +7 495 775 05 34
E-Mail info@sick-automation.ru

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 10 110 10 00
E-Mail info@sick.se

Taiwan

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

Türkiye

Phone +90 216 587 74 00
E-Mail info@sick.com.tr

USA/Canada/México

Phone +1(952) 941-6780
1 800-325-7425 - tollfree
E-Mail info@sickusa.com

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