

# HEIDENHAIN



Benutzer-Handbuch

## **EIB 741**

## **EIB 742**

Externe Interface-Box  
zum Anschluss von  
HEIDENHAIN-Messgeräten

Juli 2013

DOKUMENTATION .....	5
FIRMWAREVERSION .....	5
CHANGE HISTORY .....	5
TEIL 1: FUNKTIONSUMFANG .....	6
1 ALLGEMEINE FUNKTIONSBESCHREIBUNG .....	6
2 KONFIGURATION DER MESSGERÄT-EINGÄNGE .....	7
2.1 Verarbeitung von Inkremental-Signalen .....	7
2.2 Analogwerte der 1 VSS Inkrementalsignale A und B .....	9
2.3 Behandlung von Referenzmarken .....	9
2.4 Überwachung der Referenzmarken .....	10
2.5 Verarbeitung von EnDat-Signalen .....	11
2.6 Hilfsachse .....	14
3 VERARBEITUNG VON TRIGGER - EREIGNISSEN .....	15
3.1 Trigger Ein- und Ausgänge .....	15
3.2 Logische Ein- und Ausgänge .....	16
3.3 Triggermodul .....	16
3.4 Interval Counter .....	18
3.5 Maximale Triggerrate .....	18
3.6 Zähler für akzeptierte Trigger-Ereignisse .....	18
4 TIMESTAMP .....	19
5 STATUSWORT .....	19
6 ETHERNET INTERFACE .....	22
7 BETRIEBSMODI .....	22
7.1 Konfiguration der Datenpakete .....	22
7.2 Betriebsmodus „Polling“ .....	24
7.3 Betriebsmodus „Soft Realtime“ .....	25
7.4 Betriebsmodus „Streaming“ .....	26
7.5 Betriebsmodus „Recording“ .....	27
8 FIRMWARE UPDATE .....	28
9 RESET .....	28
TEIL 2: TREIBER-SOFTWARE .....	29
1 ALLGEMEINE INFORMATIONEN .....	29
2 INSTALLATIONSANLEITUNG .....	29
2.1 Windows .....	29
2.2 Linux .....	29
3 ÜBERBLICK .....	29
3.1 Kommunikationsaufbau .....	29
3.2 Konfiguration der Datenpakete .....	30
3.3 Polling Modus .....	30
3.4 Soft Realtime Modus .....	30
3.5 Streaming Modus .....	30
3.6 Recording Modus .....	30
4 DATENTYPEN .....	31
4.1 Einfache Datentypen .....	31
4.2 EnDat Zusatzinformation .....	31
4.3 Information für TCP-Verbindung .....	31
4.4 Konfiguration für Datenpaket .....	31
5 PARAMETER UND RÜCKGABEWERTE .....	31
6 HILFSFUNKTIONEN .....	32
6.1 IP-Adresse bestimmen .....	32
6.2 Positions-Datenformat ändern .....	32

<b>7 GERÄTEFUNKTIONEN</b>	<b>33</b>
7.1 Verbindung zur EIB 74x öffnen	33
7.2 Verbindung zur EIB 74x schließen	34
7.3 Status der Verbindung abfragen	34
7.4 Timeout einstellen	34
7.5 Anzahl der Achsen auslesen	35
7.6 Handle für Achse anfordern	35
7.7 IO-Port-Handle anfordern	35
7.8 Datenpaket erstellen	36
7.9 Datenpaket konfigurieren	36
7.10 Betriebsmodus wählen	37
7.11 Netzwerkparameter speichern	37
7.12 Netzwerkparameter auslesen	38
7.13 Hostnamen speichern	38
7.14 Hostnamen auslesen	39
7.15 Seriennummer auslesen	39
7.16 Geräte-Identnummer auslesen	39
7.17 MAC-Adresse auslesen	40
7.18 Firmware Versionsnummer auslesen	40
7.19 Bootmodus auslesen	41
7.20 Updatestatus auslesen	41
7.21 Anzahl der offenen Verbindungen lesen	41
7.22 Verbindungsdaten auslesen	42
7.23 Verbindung abbrechen	42
7.24 Timestamp Zeiteinheit lesen	42
7.25 Timestamp Periodendauer einstellen	43
7.26 Timestamp Zähler zurücksetzen	43
7.27 Timer Trigger Zeiteinheit lesen	43
7.28 Timer Trigger Periodendauer einstellen	43
7.29 Zeiteinheit für die Verzögerung an den Triggereingängen lesen	44
7.30 Trigger Counter löschen	44
7.31 Software Trigger	44
7.32 Master-Triggerquelle wählen	45
7.33 Triggerquellen aktivieren	46
7.34 Pulszähler konfigurieren	47
7.35 Interpolationsfaktor für den Interval Counter einstellen	48
7.36 Interval Counter konfigurieren	49
7.37 Abschlusswiderstände einstellen	49
7.38 Reset	50
7.39 EIB 74x identifizieren	50
7.40 Recording-Daten übertragen	51
7.41 Recording Status prüfen	51
7.42 Recording Speichergröße lesen	52
7.43 Streaming Status prüfen	52
7.44 Daten aus FIFO lesen	53
7.45 Größe eines FIFO-Elements lesen	53
7.46 Zugriff auf den Inhalt eines FIFO-Elements	54
7.47 Daten aus FIFO lesen und konvertieren	55
7.48 Größe eines FIFO-Elements nach der Konvertierung lesen	55
7.49 Zugriff auf den Inhalt eines FIFO-Elements mit konvertierten Daten	56
7.50 Anzahl der Elemente im FIFO lesen	56
7.51 FIFO löschen	57
7.52 FIFO-Größe einstellen	57
7.53 FIFO-Größe auslesen	57
7.54 Callback-Mechanismus aktivieren	58
7.55 Triggerquelle für Hilfsachse wählen	59
7.56 Position der Hilfsachse abfragen	59
7.57 Daten der Hilfsachse auslesen	60
7.58 Zähler der Hilfsachse löschen	60
7.59 Signalfehler der Hilfsachse quittieren	60
7.60 Triggerfehler der Hilfsachse quittieren	61
7.61 Statusbit für Referenzmarke der Hilfsachse löschen	61
7.62 Status der Referenzfahrt für die Hilfsachse prüfen	61

7.63	Referenzfahrt für die Hilfsachse starten .....	61
7.64	Referenzfahrt für die Hilfsachse stoppen.....	62
7.65	Timestamp für die Hilfsachse konfigurieren .....	62
7.66	Triggerflanke für Referenzimpuls der Hilfsachse einstellen.....	62
<b>8</b>	<b>ACHSFUNKTIONEN .....</b>	<b>63</b>
8.1	Achse initialisieren .....	63
8.2	Triggerquelle für Achse wählen .....	66
8.3	Triggerflanke für Referenzimpuls einstellen .....	66
8.4	Zähler löschen.....	67
8.5	Position abfragen.....	67
8.6	Daten für einen Kanal auslesen .....	68
8.7	Spannungsversorgungsfehler quittieren.....	68
8.8	Triggerfehler quittieren.....	69
8.9	Signalfehler quittieren.....	69
8.10	EnDat-Fehlerbits löschen .....	69
8.11	Statusbits für Referenzmarken löschen.....	70
8.12	Statusbits für abstandscodierte Referenzmarken löschen .....	70
8.13	Referenzfahrt starten.....	70
8.14	Referenzfahrt stoppen.....	71
8.15	Referenzfahrt Status prüfen .....	71
8.16	Überwachung der Referenzmarken einstellen .....	71
8.17	EnDat 2.1: Position lesen.....	72
8.18	EnDat 2.1: Speicherbereich wählen .....	72
8.19	EnDat 2.1: Daten senden .....	73
8.20	EnDat 2.1: Daten empfangen .....	73
8.21	EnDat 2.1: Messgerät Reset .....	74
8.22	EnDat 2.1: Testwert lesen .....	74
8.23	EnDat 2.1: Testbefehl zum Messgerät senden .....	75
8.24	EnDat 2.2: Position und Zusatzinformation lesen.....	75
8.25	EnDat 2.2: Position und Zusatzinformation lesen und Speicherbereich auswählen .....	76
8.26	EnDat 2.2: Position und Zusatzinformation lesen und Daten senden.....	76
8.27	EnDat 2.2: Position und Zusatzinformation lesen und Daten empfangen.....	77
8.28	EnDat 2.2: Position und Zusatzinformation lesen und Testkommando senden .....	77
8.29	EnDat 2.2: Position und Zusatzinformation lesen und Fehlerreset senden.....	78
8.30	EnDat 2.2: Zusatzinformation auswählen.....	79
8.31	EnDat 2.2: Sequenz für Zusatzinformationen auswählen .....	80
8.32	Absolute und inkrementale Positionswerte simultan auslesen .....	81
8.33	Spannungsversorgung für Messgeräte einstellen.....	81
8.34	Status der Spannungsversorgung für Messgeräte lesen.....	82
8.35	Timestamp konfigurieren.....	82
<b>9</b>	<b>IO-FUNKTIONEN .....</b>	<b>83</b>
9.1	Eingangsport konfigurieren .....	83
9.2	Ausgangsport konfigurieren .....	84
9.3	Triggerquelle für Triggereingang wählen .....	84
9.4	Verzögerungszeit für Triggereingang einstellen .....	85
9.5	Logischen Port auslesen .....	85
9.6	Logischen Ausgangsport setzen .....	86
9.7	Konfigurationsdaten für Eingang lesen .....	86
9.8	Konfigurationsdaten für Ausgang lesen .....	87
<b>10</b>	<b>ALLGEMEINE FUNKTIONEN.....</b>	<b>88</b>
10.1	Treiber ID-Nummer lesen.....	88
10.2	Fehlermeldung in Text umwandeln .....	88

## **Dokumentation**

Die Dokumentation zur EIB 741 bzw. EIB 742, im Folgenden als EIB 74x bezeichnet, umfasst folgende Unterlagen:

- Inbetriebnahme - Anleitung:
  - Unterlagen, die für die Inbetriebnahme erforderlich sind, sowie Technische Daten.
- Benutzer-Handbuch:
  - Beschreibung des Funktionsumfanges der EIB 74x.
  - Beschreibung der Installation und der Funktionsaufrufe der Treiber-Software.

## **Firmwareversion**

Die vorliegende Dokumentation beschreibt die Firmware-Version: 633281-10

## **Change History**

Änderungen zu vorhergehenden Versionen können aus der Change History entnommen werden.

Das Dokument zur Change History ist auf der CD im Unterverzeichnis EIB\_74x/doc zu finden. Bitte lesen Sie dieses Dokument, speziell die Hinweise zu neuen, geänderten und obsoleten Funktionsaufrufen.

# Teil 1: Funktionsumfang

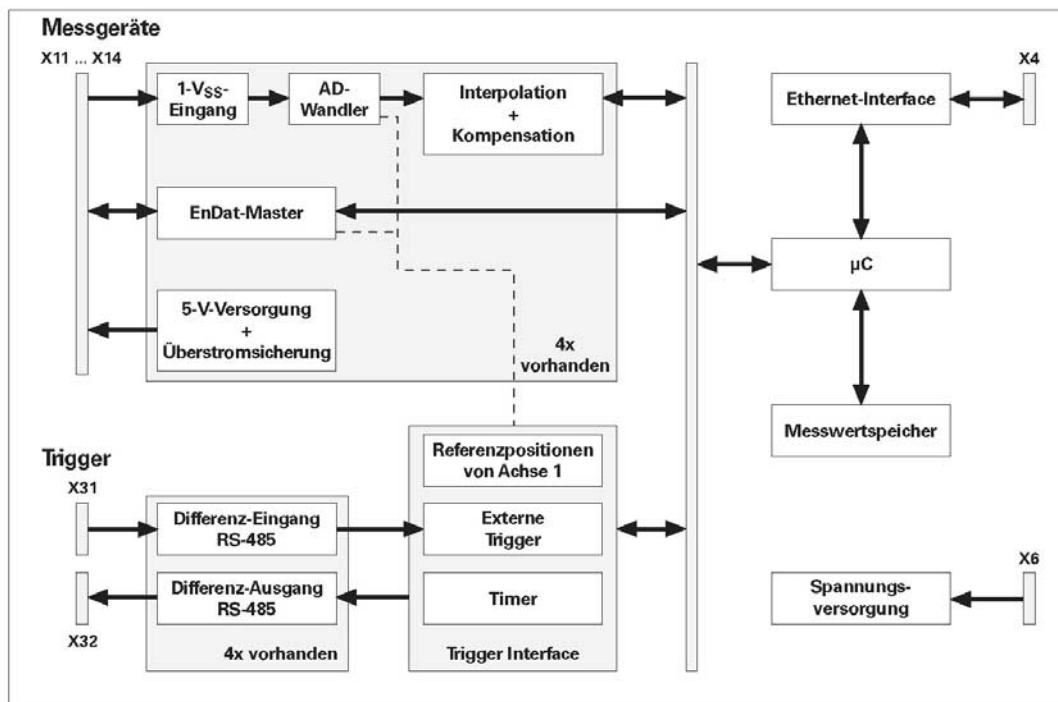
## 1 Allgemeine Funktionsbeschreibung

Die EIB 74x ist eine externe Interface-Box zur präzisen Positionsmessung speziell für Prüfplätze und Mehrstellen-Messplätze, sowie zur mobilen Datenerfassung, z.B. bei der Maschinenvermessung.

Die EIB 74x ist ideal für Anwendungen, die eine hohe Auflösung der Messgerätesignale und eine schnelle Messwerterfassung erfordern. Außerdem ermöglicht die Ethernet-Übertragung die Verwendung von Switches bzw. Hubs zur Verschaltung von mehreren EIB 74x.

An die EIB 74x können bis zu vier HEIDENHAIN-Messgeräte wahlweise mit sinusförmigen Inkrementalsignalen (1 V<sub>SS</sub>) oder mit EnDat-Schnittstellen (EnDat 2.1 und EnDat 2.2) angeschlossen werden. Zur Messwertbildung unterteilt die EIB 74x die Signalperioden der Inkrementalsignale 4096fach. Der automatische Abgleich der sinusförmigen Inkrementalsignale (Signalkompensation) reduziert die Abweichungen innerhalb einer Signalperiode. Durch den integrierten Messwertspeicher ermöglicht die EIB 74x im Betriebsmodus „Recording“ ein Abspeichern von bis zu 250 000 Messwerten pro Achse. Das Abspeichern der Messwerte erfolgt achsabhängig wahlweise über interne oder externe Trigger. Zur Datenausgabe steht eine Standard- Ethernet-Schnittstelle (Verwendung von TCP- bzw. UDP-Kommunikation) zur Verfügung. Damit ist eine direkte Anbindung an PC, Laptop oder Industrie-PC möglich. Die Art der Messwertübertragung kann über den Betriebsmodus eingestellt werden. Zur Verarbeitung der Messwerte im PC sind im Lieferumfang Treiber-Software für Windows, Linux und LabVIEW enthalten. Die Treiber-Software ermöglicht eine einfache Programmierung von Kundenapplikationen. Zusätzlich demonstrieren Beispielprogramme die Möglichkeiten der EIB 74x.

Prinzipschaltbild



### Messgerät-Eingänge:

An die EIB 74x können bis zu vier HEIDENHAIN-Messgeräte mit folgenden Schnittstellen angeschlossen werden (frei programmierbar):

- Inkrementalsignale 1 V<sub>SS</sub>
- EnDat 2.1
- EnDat 2.2
- Inkrementalsignale 11 µA<sub>SS</sub> (auf Anfrage)

Die Spannungsversorgung der Messgeräte erfolgt von der EIB 74x und ist durch eine rücksetzbare Überstromabschaltung abgesichert.

Technische Daten siehe „Inbetriebnahme - Anleitung“.

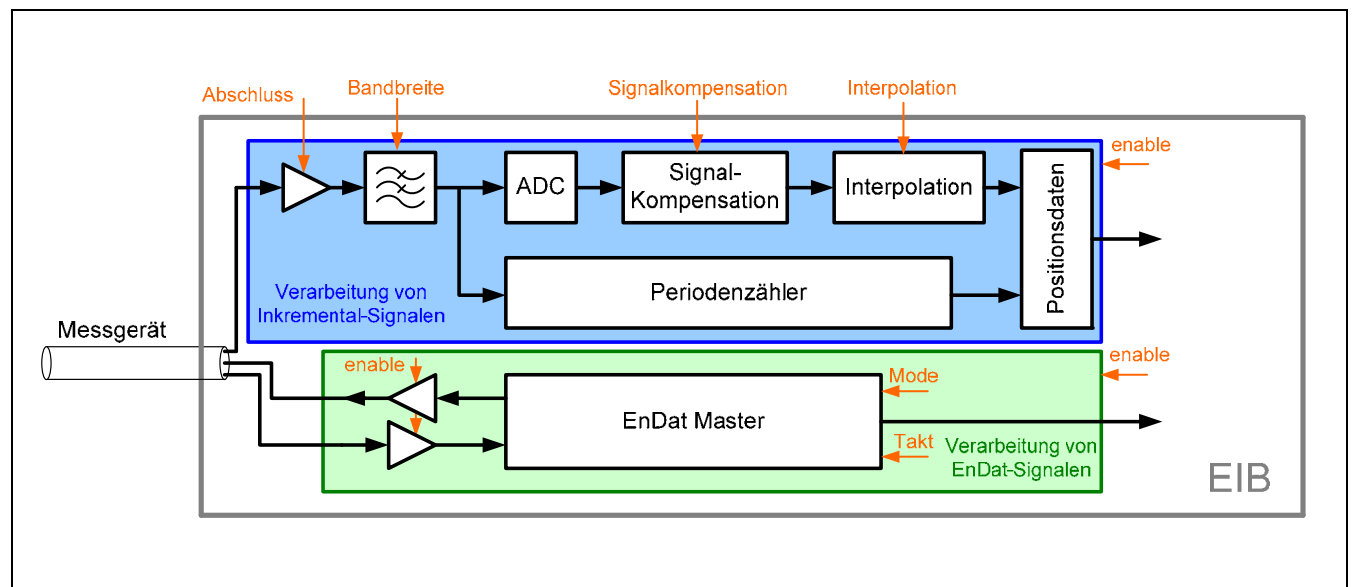
## 2 Konfiguration der Messgerät-Eingänge

Nach dem Power-up ist die Spannungsversorgung der Messgeräte eingeschaltet. Die weiteren Parameter zum Betrieb des Messgerät-Einganges müssen per Initialisierung gesetzt werden:

- Schnittstellentyp
- Bandbreite für die 1 V<sub>SS</sub> Eingangssignale
- Signalkompensation
- Verarbeitung der Referenzmarken
- Verarbeitung der Homing/Limit-Signale

Diese Einstellungen können per Software geändert werden.

Die Schnittstelle für den Messgerät-Eingang kann im Inkremental- oder EnDat Modus betrieben werden. Im EnDat Modus kann zusätzlich der Inkrementalblock mit betrieben werden, wenn vom Messgerät zusätzlich zur EnDat auch die 1 V<sub>SS</sub> Schnittstelle unterstützt wird.



### 2.1 Verarbeitung von Inkremental-Signalen

Zur Bildung des Positionswertes unterteilt die EIB 74x die Signalperioden der Inkrementalsignale 4096fach (12 Bit). Der Periodenzähler hat eine Breite von 32 Bit. Der Zählwert wird mit jeder Signalperiode des angeschlossenen Messgerätes um den Wert "1" erhöht oder erniedrigt.

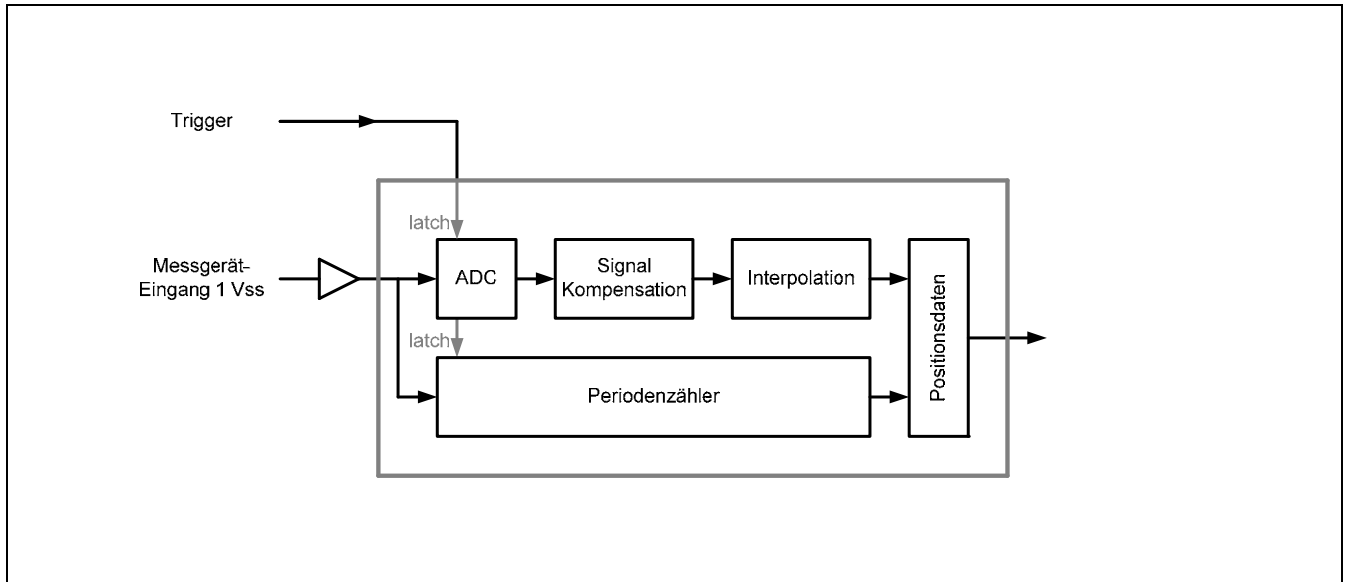
Der automatische Abgleich der sinusförmigen Inkrementalsignale (Signalkompensation) reduziert die Abweichungen innerhalb einer Signalperiode. Die Kompensation der Inkrementalsignale des Messgerätes und auch der Abschlusswiderstand kann per Software ein- bzw. ausgeschaltet werden.

Der Interpolationswert (12 Bit) bildet zusammen mit dem Wert des Periodenzählers (32 Bit) die 44 Bit breite Positionsinformation zum Zeitpunkt des Triggerereignisses. Die Positionsinformation wird in einem 48 Bit breiten Register gespeichert (siehe Tabelle). Der Periodenzähler wird dabei in der Zweierkomplement-Darstellung abgebildet; die Bit 43 .. 47 bilden das Vorzeichen ab.

Die übergeordnete Kunden-Softwareapplikation kann aus diesem Wert abhängig von der Art des Messgerätes (linear bzw. rotativ) den entsprechenden Winkel bzw. die Länge berechnen.

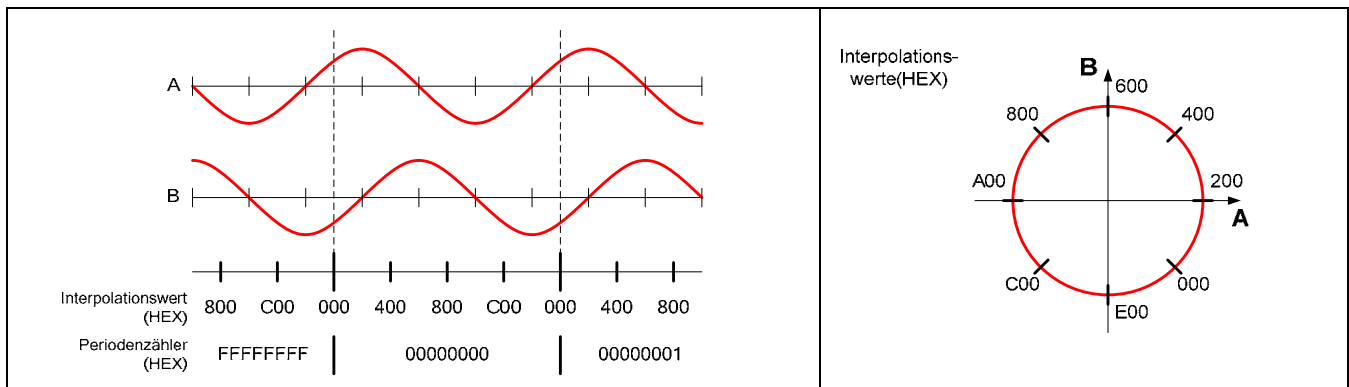
Der Überlauf des Periodenzählers erfolgt entsprechend der Zweierkomplement-Darstellung an der Stelle: 0x07FF FFFF FFFF (Maximum positiv) → 0xF800 0000 0000 (Maximum negativ). Dieser Überlauf hat keinen Einfluss auf die Funktionalität des Periodenzählers oder des Interpolators. Der Überlauf muss jedoch durch die übergeordnete Kunden-Softwareapplikation behandelt werden.

Bit Nr.	Breite (Bit)	Inhalt
0..11	12	Interpolationswert
12..43	32	Periodenzähler (Bit 43 = Vorzeichen)
44..47	4	Identischer Wert zu Bit 43



### Interpolationswert

Zum Zeitpunkt des Trigger-Ereignisses werden die Inkrementalsignale abgetastet und daraus ein 12 Bit breiter Interpolationswert berechnet (nicht bei der EnDat-Schnittstelle). Der Zusammenhang zwischen Interpolationswert und den Inkrementalsignalen ergibt sich dabei wie folgt:



### Einstellmöglichkeiten:

#### Abschlusswiderstand für die Inkrementalsignale

Der 120 Ohm Abschlusswiderstand für die 1 Vss Inkrementalsignale kann per Software zu- oder abgeschaltet werden (für alle Kanäle gleichzeitig; Default: Widerstände zugeschaltet).

#### Bandbreiten-Einstellung der Inkrementalsignale

Die Bandbreite der Inkrementalsignale des Messgerätes kann per Software umgeschaltet werden. Als Standard-Einstellung sollte die hohe Bandbreite (500 kHz) eingestellt werden.

Die Einstellung niedrige Bandbreite (33 kHz) sollte nur für spezielle Applikationen angewählt werden.

#### Signal-Kompensation

Die Kompensation der Inkrementalsignale des Messgerätes kann per Software ein- oder ausgeschaltet werden.



## 2.2 Analogwerte der 1 V<sub>SS</sub> Inkrementalsignale A und B

Die übertragenen Werte entsprechen den Werten des AD-Wandlers zum Zeitpunkt des Trigger-Ereignisses.

Bit Nr.	Breite (Bit)	Inhalt
0..11	12	12-Bit-AD-Wandler Wert
12..15	4	Reserviert

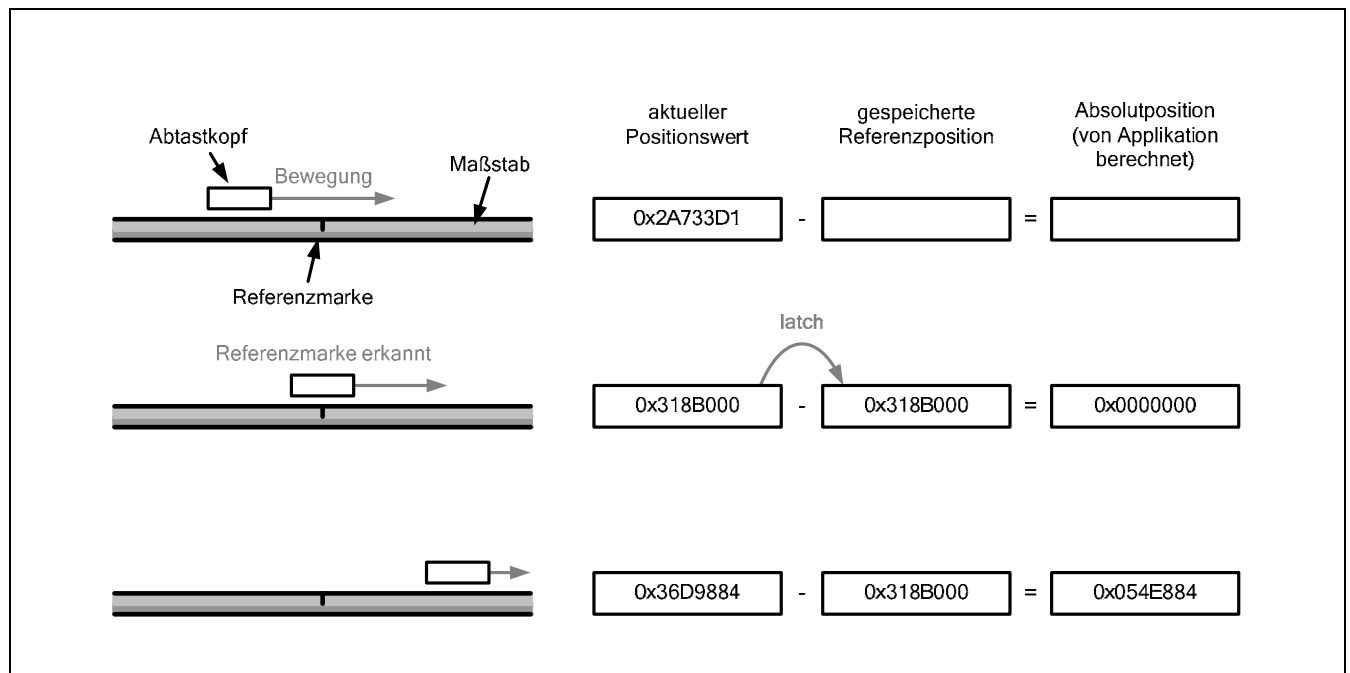
Wert (HEX)	Signalwert Inkrementalsignale
000	negatives Maximum
800	Null
FFF	positives Maximum

## 2.3 Behandlung von Referenzmarken

Bei inkrementellen Messgeräten wird die Referenzmarke bzw. werden die Referenzmarken dazu benutzt, einen absoluten Bezug für die Inkrementalsignale herzustellen.

Bei Messgeräten mit einer Referenzmarke hat diese einen eindeutigen Bezug zu einer bestimmten Signalperiode. Diese Signalperiode kann als Bezug zur Bildung von absoluten Positionswerten verwendet werden. Das Überfahren der Referenzmarke hat keinen Einfluss auf den Periodenzähler oder den Interpolationswert. Es wird lediglich der zum Zeitpunkt des Überfahrens gültige Periodenzählerwert in einem Register für die Referenzposition gespeichert. Mit diesem Wert können in der Kunden-Softwareapplikation absolute Positionswerte berechnet werden.

Das folgende Bild zeigt den prinzipiellen Ablauf bei der Ermittlung einer Referenzposition. Die angezeigten Werte sind nur als Beispiel zu verstehen und der Übersicht halber ist nur ein Ausschnitt des Positionswert-Registers gezeigt.



Registerinhalt Referenzposition

Bit Nr.	Breite (Bit)	Inhalt
0..11	12	immer 0
12..43	32	Referenzposition (Wert des Periodenzählers zum Zeitpunkt der Detektion der Referenzmarke; Bit 43 = Vorzeichen)
44..47	4	Identischer Wert zu Bit 43

Ein automatisches Speichern der Referenzposition muss per Software aktiviert werden. Nach diesem Kommando wartet die EIB 74x auf die nächste Referenzmarke und speichert dann die Referenzposition.

Ein erneutes Speichern muss anschließend wieder aktiviert werden.

Im Normalfall wird das Register für die Referenzposition zusammen mit dem Positionsregister und dem Statuswort in einem gemeinsamen Positions-Datenpaket nach dem nächsten Trigger-Ereignis übertragen. Die EIB 74x überträgt dabei zwei Referenzpositionen und ggf. den codierten Referenzwert:

- bei Messgeräten mit einer Referenzmarke wird gewöhnlich nur die Referenzposition 1 verwendet,
- bei Messgeräten mit abstandscodierten Referenzmarken werden je nach Auswertungsmethode beide Registerwerte oder der codierte Referenzwert verwendet.

### Abstandscodierte Referenzmarken

Bei abstandscodierten Messgeräten wird der Bezug zur Bildung von absoluten Positionswerten aus den Zählerwerten durch den Abstand zweier überfahrener (nebeneinander liegenden) Referenzmarken gewonnen.

Zu diesem Zweck erfolgt das Speichern des Periodenzählerwertes zweimal, jeweils bei Überfahren einer Referenzmarke. Aus dem Abstand der (nebeneinander liegenden) Referenzmarken wird der codierte Referenzwert gebildet und somit der Bezug zur Bildung von absoluten Positionswerten hergestellt.

Dieser Wert wird bei der Berechnung des absoluten Positionswertes durch die Kunden-Softwareapplikation genau so behandelt wie ein gespeicherter Referenz-Positionswert im Fall von Messgeräten mit einer Referenzmarke (siehe Zeichnung). Der codierte Referenzwert entspricht somit ebenfalls dem Offset zwischen absolutem Positionswert und ausgegebenem (inkrementellem) Positionswert.

Es gibt unterschiedliche Vorgehensweisen zur Bildung des codierten Referenzwertes:

#### Methode 1: (empfohlene Methode)

Die Achse wird als inkrementelles System mit abstandscodierten Referenzmarken initialisiert. Dabei werden weitere typ-abhängige Informationen über das Messsystem an die EIB 74x übergeben. Aus diesen Informationen berechnet die EIB 74x nach erfolgreichem Einspeichern der Referenzpositionen automatisch den codierten Referenzwert. Der Einspeicher-Vorgang wird per Software-Kommando gestartet (für zwei Referenzmarken). Nach Überfahren der zweiten Referenzmarke berechnet die EIB 74x automatisch den codierten Referenzwert und überträgt ihn an die Kunden-Softwareapplikation.

#### Methode 2: (speziell bei Applikationen mit extrem niedriger Verfahrensgeschwindigkeit)

Die Achse wird als inkrementelles System mit einfacher Referenzmarke initialisiert. Die Kunden-Softwareapplikation sendet das entsprechende Software-Kommando zum Abspeichern der Referenzposition (eine Referenzmarke). Nach jeder erfolgreich gespeicherten Referenzposition wird der Vorgang zur Speicherung erneut ausgelöst. Dies muss solange wiederholt werden, bis zwei unterschiedliche Referenzpositionen erfasst wurden. Aus diesen beiden Werten kann dann der codierte Referenzwert und damit die Absolutposition von der Kunden-Softwareapplikation berechnet werden. Es muss sichergestellt sein, dass die Kunden-Softwareapplikation diese Prozedur schnell genug abarbeitet. Andernfalls könnten Referenzmarken „verloren gehen“, was zu einer falschen Berechnung der Absolutposition führt.

#### Methode 3:

Die Achse wird als inkrementelles System mit einfacher Referenzmarke initialisiert. Die Kunden-Softwareapplikation sendet das entsprechende Software-Kommando zum Abspeichern von zwei Referenzpositionen. Nach erfolgter Einspeicherung beider Referenzpositionen (beide Referenzpositionsregister werden genutzt) werden aus diesen beiden Werten der codierte Referenzwert und damit die Absolutposition von der Kunden-Softwareapplikation berechnet.

## 2.4 Überwachung der Referenzmarken

Die Referenzmarken eines Messgeräts können automatisch überwacht werden. Dazu wird fortlaufend die Referenzposition gespeichert und geprüft. Dies hat auch zur Folge, dass die ausgegebene Referenzposition mit jeder Referenzmarke aktualisiert wird und sich dadurch ändern kann. Abhängig vom Messgerät unterscheidet sich die Prüfung geringfügig, wie nachfolgend aufgeführt. Im Fehlerfall wird ein Bit im Statuswort für die Position gesetzt.

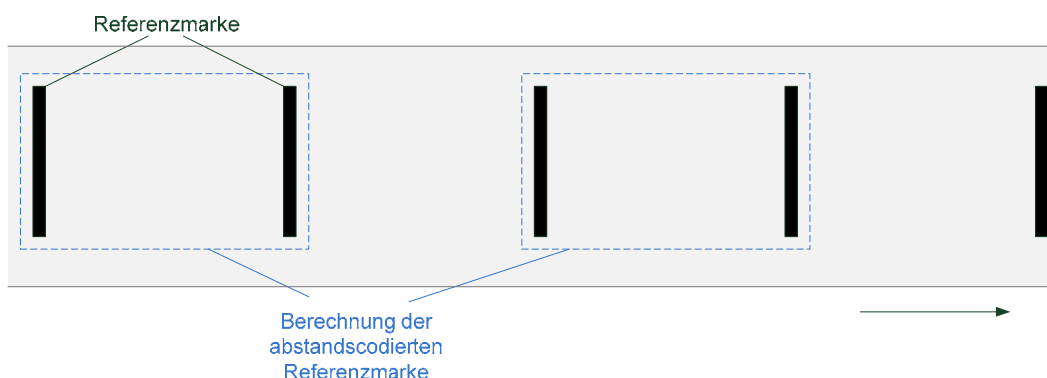
#### Messgeräte mit einer Referenzmarke:

Bei linearen Messgeräten mit einer Referenzmarke muss der Positionswert an der Referenzmarke immer gleich sein. Die Referenzposition wird fortlaufend gespeichert und mit dem alten Wert verglichen.

Bei rotativen Messgeräten mit einer Referenzmarke kann sich der Positionswert an der Referenzmarke ändern, wenn das Messgerät eine Umdrehung in der gleichen Richtung bewegt wird. Zwei nacheinander gespeicherte Referenzpositionen müssen gleich sein, oder dürfen sich um die Anzahl der Signalperioden pro Umdrehung unterscheiden. Im Datenpaket wird immer die aktuelle Referenzposition übertragen.

#### Messgeräte mit abstandscodierten Referenzmarken:

Bei Messgeräten mit abstandscodierten Referenzmarken wird die codierte Referenzposition fortlaufend neu berechnet. Die Berechnung erfolgt immer mit zwei benachbarten Referenzpositionen wie in der Abbildung nachfolgend dargestellt. Im Datenpaket wird immer die aktuelle berechnete Referenzposition übertragen.



Bei linearen Messgeräten muss die berechnete Referenzposition immer gleich sein. Bei rotativen Messgeräten kann sich die berechnete Referenzposition um die Anzahl der Signalperioden pro Umdrehung ändern. Wird die gleiche Referenzmarke vor und nach einem Richtungswechsel zweimal überfahren, kann keine Berechnung der abstandscodierten Referenzmarke erfolgen. In diesem Fall wird keine Prüfung durchgeführt. Insbesondere bei sehr kleinen Bewegungen um eine Referenzmarke muss dies beachtet werden.

Während die Überwachung der Referenzmarken aktiv ist, kann keine Referenzfahrt durchgeführt werden, da dies zu einer falschen Fehlermeldung in der Überwachung führen könnte. Es wird die folgende Reihenfolge empfohlen.

- Konfiguration der Achsen
- Referenzfahrt durchführen
- Überwachung der Referenzmarken aktivieren

## 2.5 Verarbeitung von EnDat-Signalen

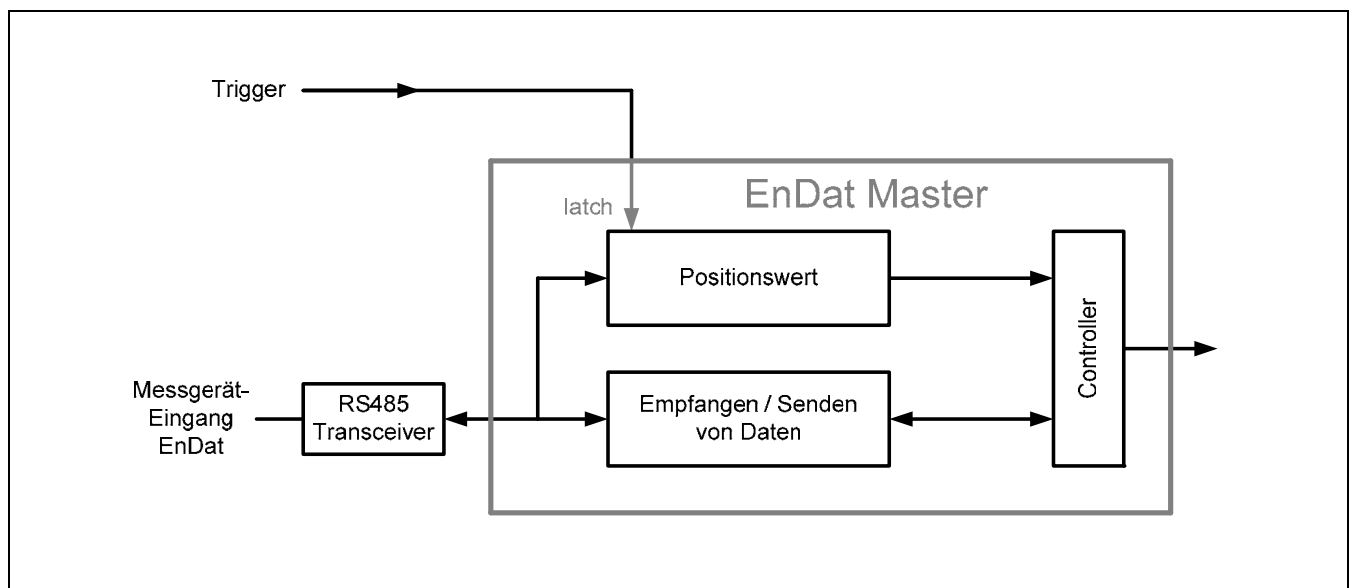
Absolute Messgeräte von HEIDENHAIN sind mit EnDat 2.1 oder EnDat 2.2 Schnittstelle erhältlich. Zusätzlich zu den EnDat Signalen werden, speziell bei EnDat 2.1 Messgeräten, 1 Vss Inkrementalsignale mit übertragen. Die EIB 74x kann alle EnDat Messgeräte mit EnDat 2.1 oder EnDat 2.2 Schnittstelle sowohl rein seriell, als auch mit 1 Vss Inkrementalsignalen verarbeiten.

Der EnDat Master wird bei der Initialisierung der Achse individuell eingestellt:

- EnDat 2.1 oder EnDat 2.2 Kommunikation kann eingestellt werden.
- Die Taktfrequenz für die EnDat Kommunikation ist einstellbar
- Laufzeitkompensation (EnDat 2.2) kann ein- oder ausgeschaltet werden.
- Die „Recovery time I“ kann eingestellt werden, wenn dies vom Messgerät unterstützt wird
- Die Überwachung der „Calculation time“ kann eingestellt werden

Anmerkungen zu EnDat 01:

- Bei gleichzeitiger Verwendung von EnDat-Positionsanfragen und 1 Vss Inkrementalsignalen können nur EnDat 2.1 Mode-Befehle an das Messgerät gesendet werden (Achse muss für EnDat 01 konfiguriert sein).
- Die EnDat Position kann nur per Software-Kommando eingelesen werden. Es muss also ein einmaliges Einlesen von EnDat Position und Inkrementalposition erfolgen (spezielles Kommando). Im Anschluss daran kann eine zyklische Übertragung der Inkrementalposition erfolgen.



### Register für den Positionswert

Das Positionsregister bildet die über die EnDat Schnittstelle übertragene Position zum Zeitpunkt des Trigger-Ereignisses ab. Das Positionsregister für die EnDat Position hat eine Breite von 48 Bit. Die Anzahl der benutzten Bits für den Positionswert hängt vom angeschlossenen EnDat Messgerät ab; die oberen ungenutzten Bits müssen ausmaskiert werden. Nähere Informationen siehe Technische Daten des Messgerätes.

Bit Nr.	Breite (Bit)	Inhalt
0..47	48	EnDat Positionswert

## EnDat Taktfrequenz

Die EnDat Taktfrequenz kann per Software Kommando eingestellt werden. Die Taktfrequenz kann in bestimmten Schritten zwischen 100 kHz und 6,66 MHz eingestellt werden. Die maximal zulässige Frequenz ist sowohl abhängig von der Kabellänge zwischen Messgerät und EIB 74x, als auch davon, ob eine Laufzeitkompensation aktiviert ist oder nicht.

Parameter Taktfrequenz	Taktfrequenz	Anmerkung
100000	100 kHz	
300000	300 kHz	Default bei EnDat 2.1
500000	500 kHz	
1000000	1 MHz	
2000000	2 MHz	Default bei EnDat 2.2
4000000	4 MHz	
5000000	5 MHz	
6666666	6,66 MHz	

## Laufzeitkompensation

Die Laufzeitkompensation für die EnDat Übertragung kann bei der Achskonfiguration ein- bzw. ausgeschaltet werden. Für EnDat 2.1 Messgeräte ist die Laufzeitkompensation von HEIDENHAIN nicht freigegeben (Ausnahme Messgeräte mit Bestellbezeichnung EnDat21). Für EnDat 2.2 Messgeräte ist die Laufzeitkompensation von HEIDENHAIN freigegeben. Damit ergibt sich folgende Abhängigkeit der maximal erlaubten EnDat-Taktfrequenz.

EnDat-Taktfrequenz	Kabellänge in Meter	
	Ohne Laufzeitkompensation	Mit Laufzeitkompensation
100 kHz	150	100
300 kHz	150	100
500 kHz	100	100
1 MHz	55	100
2 MHz	10	100
4 MHz	–	50
5 MHz	–	40
6,66 MHz	–	25

## Recovery time I

Für EnDat 2.2 Messgeräte (Bestellbezeichnung EnDat02 bzw. EnDat22) kann die „Recovery time I“ eingestellt werden. Hierbei gibt es die zwei Optionen „lang“ ( $10 \mu\text{s} < t_m < 30 \mu\text{s}$ ) und „kurz“ ( $1.25 \mu\text{s} < t_m < 3.75 \mu\text{s}$ ). Für EnDat 2.1 Messgeräte wird immer die lange „Recovery time I“ verwendet.

Anmerkungen zur Einstellung der „Recovery time I“ „kurz“:

- Default ist die Einstellung „lang“
- Die Einstellung „kurz“ wird gewählt um kürzere Zykluszeiten bei der EnDat Übertragung zu erreichen.
- Bei der Einstellung „kurz“ muss gleichzeitig die EnDat Taktfrequenz  $> 1 \text{ MHz}$  eingestellt werden.

## Calculation time

Die Calculation time gibt die Zeit für die Positionsbildung im Messgerät an und wirkt sich deshalb auf die Dauer der Positionsabfrage aus. Um die Kommunikation zu überwachen wird ein Timeout erzeugt, falls die Positionsabfrage eine bestimmte Zeit überschreitet. Dies wird als Fehler im Statuswort angezeigt. Falls die Calculation time zu kurz eingestellt ist, kann diese Fehlermeldung auftreten, obwohl das Messgerät die Daten ordnungsgemäß sendet. Im umgekehrten Fall kann eine zu lange Calculation time zu einer verspäteten Meldung des Fehlers führen. Insbesondere bei hohen Triggerraten kann sich die Fehlermeldung um mehrere Samples verschieben.

Die „Calculation time“ kann abhängig vom angeschlossenen Messgerät eingestellt werden. Es werden zwei Optionen unterstützt:

lang Die „Calculation time“ des Messgerätes ist  $< 1 \text{ ms}$   
kurz Die „Calculation time“ des Messgerätes ist  $< 15 \mu\text{s}$

## Anmerkungen:

- Default ist die Einstellung „lang“
- Bei der Einstellung „kurz“ muss gleichzeitig die EnDat Taktfrequenz  $> 1 \text{ MHz}$  eingestellt werden.

## EnDat 2.2 Zusatzinformationen

Die Übertragung der EnDat 2.2 Zusatzinformationen kann in den Betriebsmodi „Soft Realtime“, „Streaming“ und „Recording“ auf unterschiedliche Weisen erfolgen.

### 1) Keine Zusatzinformation

Mit jedem Triggerereignis wird eine Positionsabfrage gestartet. Dabei werden keine Zusatzinformationen übertragen.

### 2) Feste Zusatzinformationen

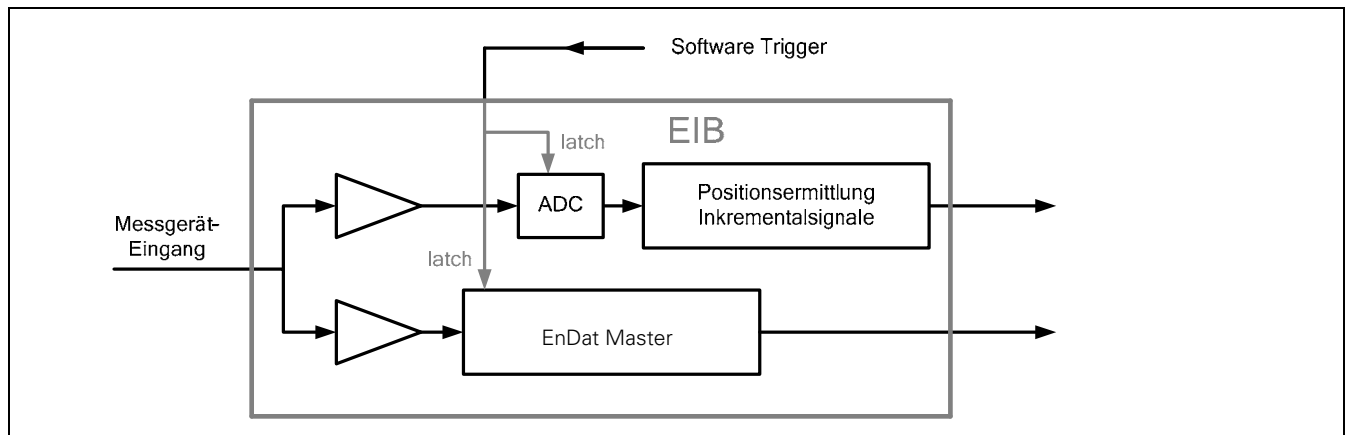
Mit jedem Triggerereignis wird neben dem Positionswert als Zusatzinformation 1 und Zusatzinformation 2 jeweils eine feste Information übertragen. Diese muss vor der Aktivierung des entsprechenden Betriebsmodus eingestellt werden. Sie kann nur im Betriebsmodus Polling verändert werden. Es ist ebenfalls möglich nur die Zusatzinformation 1 oder die Zusatzinformation 2 zu übertragen.

### 3) Variable Zusatzinformationen

Die Zusatzinformationen werden zyklisch umgeschaltet. Die EIB 74x besitzt einen Ringpuffer mit 10 Einträgen für die Einstellung der Zusatzinformationen, welcher zyklisch abgearbeitet wird. Mit jedem Triggerereignis wird der Positionswert und die Zusatzinformation 1 und 2 gesendet. Zusätzlich wird der EnDat 2.2 Sendezusatz übertragen, über den basierend auf den Daten im Ringpuffer eine neue Zusatzinformation ausgewählt wird. In dem Ringpuffer können Zusatzinformationen 1 und 2 gemischt werden. Es kann pro Positionsabfrage nur eine der beiden Zusatzinformationen umgeschaltet werden.

## Verarbeitung von zusätzlichen Inkrementalsignalen bei EnDat

Werden bei EnDat-Messgeräten die Inkrementalsignale zur Positionsbildung verwendet, kann zur Herstellung eines absoluten Bezugs ein gleichzeitiges Einspeichern der EnDat- und Inkrementalposition erfolgen. Dazu wird über die Kunden-Softwareapplikation ein spezielles Kommando an die EIB 74x gesendet, das dann ein internes Triggersignal generiert. Dieses Triggersignal löst die gleichzeitige Positionsermittlung über die EnDat Schnittstelle und über die Inkrementalsignale aus. Die beiden Positionen werden als Rückgabewert an die Kunden-Softwareapplikation übergeben.



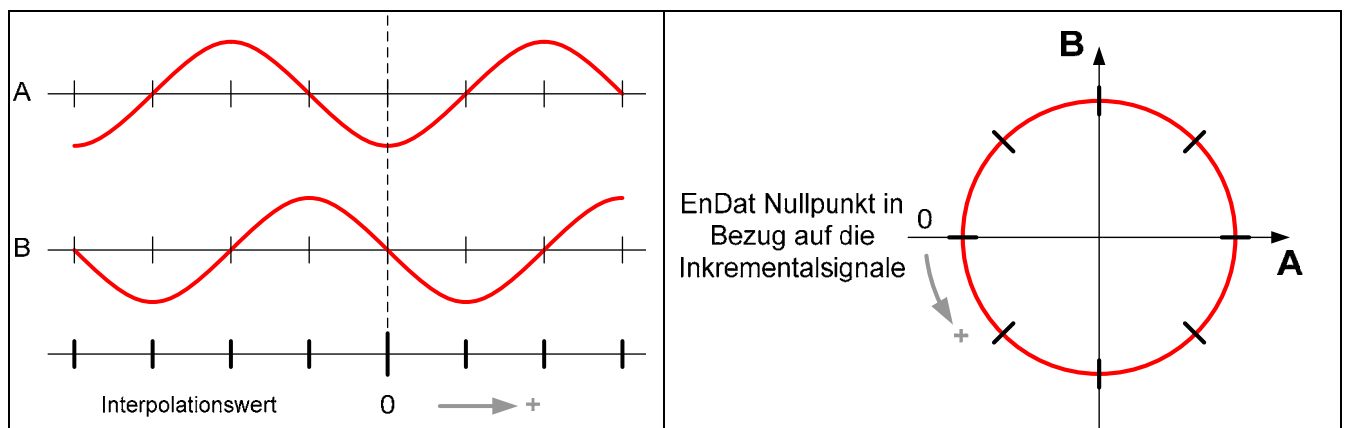
### Anmerkung:

Die Interpolations-Nullpunkte für die Inkrementalsignale und die EnDat Position sind unterschiedlich und müssen von der Kunden-Softwareapplikation mit berücksichtigt werden.

Außerdem ist auch die eventuell unterschiedliche Auflösung zwischen EnDat- und Inkrementalposition mit zu berücksichtigen.

Inkrementalsignale: Interpolationsnullpunkt siehe Abschnitt „Verarbeitung von Inkrementalsignalen“

EnDat Position: Interpolationsnullpunkt siehe Grafik



## 2.6 Hilfsachse

Die Hilfsachse ist gekoppelt an die Achse 1 und für Messgeräte mit 1Vss-Schnittstelle einsetzbar. Die Signale von Achse 1 werden interpoliert und einem Positionszähler zugeführt. Der Interpolationsfaktor ist in mehreren Stufen einstellbar. Zusätzlich kann die Flankenbewertung (1, 2 bzw. 4-fach) gewählt werden. Die maximal zulässige Eingangsfrequenz der Messgerätesignale für den Interpolator ist abhängig vom Interpolationsfaktor und in der nachfolgenden Tabelle dargestellt. Um die Eingangsfrequenz nicht unnötig zu begrenzen, sollte die Flankenbewertung auf 4x gestellt und dafür ein niedriger Interpolationsfaktor gewählt werden. Zum Beispiel führt ein Interpolationsfaktor von 5x mit einer Flankenbewertung von 4x zur gleichen Schrittweite wie ein Interpolationsfaktor von 20x mit einer Flankenbewertung von 1x, allerdings mit höherer maximal zulässiger Eingangsfrequenz.

Lineares Messgerät:

$$\frac{\text{Schrittweite}}{\mu\text{m}} = \frac{\text{Signalperiode des Messgeräts} / \mu\text{s}}{\text{Interpolationsfaktor} \cdot \text{Flankenbewertung}}$$

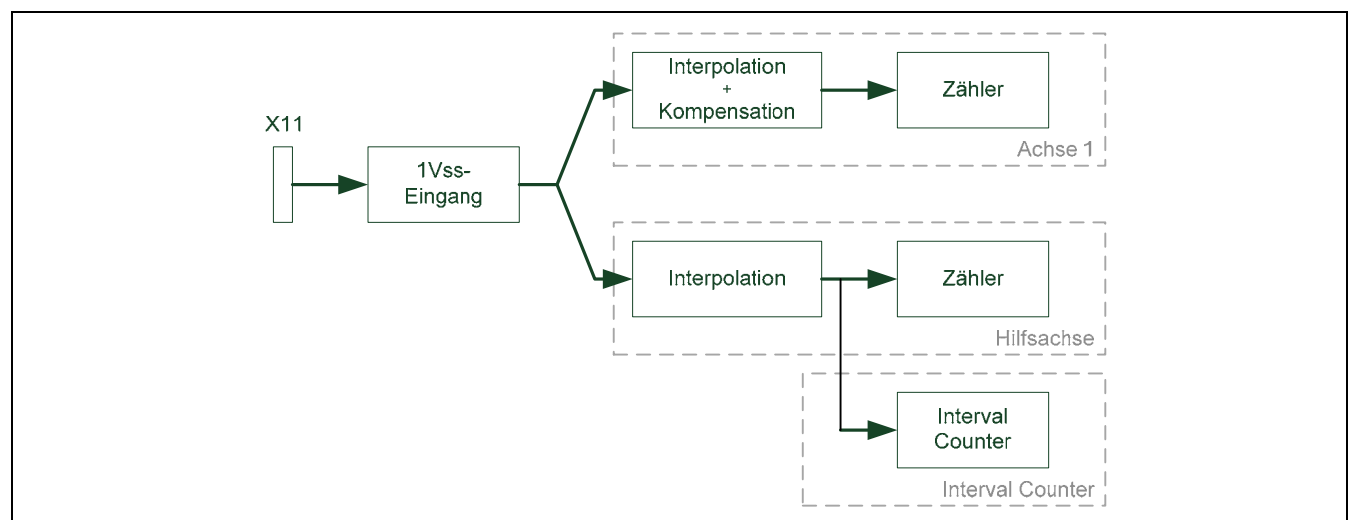
Rotatives Messgerät:

$$\frac{\text{Schrittweite}}{1^\circ} = \frac{\frac{360^\circ}{\text{Strichzahl des Messgeräts}}}{\text{Interpolationsfaktor} \cdot \text{Flankenbewertung}}$$

Interpolationsfaktor	max. Eingangsfrequenz in kHz
1x	500
2x	500
4x	500
5x	500
10x	400
20x	200
25x	160
50x	80
100x	40

Neben dem Positionswert verfügt die Hilfsachse über einen Timestamp und eine Referenzposition. Das Statuswort für die Hilfsachse enthält Status- und Fehlermeldungen. Sowohl der Positionswert als auch die Referenzposition ist ein 32-Bit-Wert. Die Anzahl der Zählschritte pro Signalperiode des Messgeräts ist abhängig vom Interpolationsfaktor für die Hilfsachse. Der Positionswert wird im Zweierkomplement dargestellt. Entsprechend erfolgt ein Überlauf an der Stelle 0x7FFF FFFF (Maximum positiv) → 0x8000 0000 (Maximum negativ). Der Überlauf muss gegebenenfalls durch die übergeordnete Kunden-Softwareapplikation behandelt werden.

Bit Nr.	Breite (Bit)	Inhalt
0..31	32	Positionswert der Hilfsachse (Bit 31 = Vorzeichen)



### 3 Verarbeitung von Trigger - Ereignissen

Die Positionswertermittlung innerhalb der EIB 74x wird über ein sogenanntes Triggerereignis angestoßen. Die EIB 74x unterstützt dabei folgende Triggerquellen:

- 4 externe Trigger-Eingänge
- Interne periodische Triggerquelle, timergesteuert
- Software-Kommando
- Referenzimpuls der Messgeräte
- Positionstrigger (Interval Counter)

Die Triggerquelle muss für jede Achse per Software-Kommando eingestellt werden, wobei zur gleichen Zeit nur eine Triggerquelle pro Achse wirken kann. Allerdings ist es möglich für verschiedene Achsen unterschiedliche Triggerquellen zu aktivieren. Zusätzlich muss eine Triggerquelle als Master-Triggerquelle definiert werden, die den Zeitpunkt der Datenübertragung bestimmt. Für alle Achsen, die ebenfalls mit der Master-Triggerquelle getriggert werden, wird in jedem Datenpaket eine neue Position übertragen. Für alle anderen Achsen wird nur dann eine gültige Position übertragen, wenn für diese Achse ebenfalls ein Triggerereignis auftrat. Andernfalls wird der Positionswert als ungültig markiert. Nicht in allen Betriebsarten werden alle Möglichkeiten des Trigger-Interfaces unterstützt; Details siehe Abschnitt „Betriebsmodi“.

#### 3.1 Trigger Ein- und Ausgänge

Es werden vier Trigger Ein- bzw. Ausgänge unterstützt. Technische Daten zum Trigger-Eingang können der „Inbetriebnahme - Anleitung“ entnommen werden.

##### Trigger Eingänge

Dienen zur Synchronisation der Positionsabfragen auf externe Ereignisse; bitte Kapitel 3.5 beachten.

Der 120 Ohm Abschlusswiderstand kann per Konfiguration zu- oder abgeschaltet werden

##### Trigger Ausgänge

Dienen zur Weiterleitung von Trigger-Ereignissen z.B. an weitere EIB 74x. Damit kann eine Trigger-Kette aufgebaut werden, die mehrere EIB 74x auf ein externes Trigger-Ereignis synchronisiert. Die verschiedenen EIB 74x sind dabei separat voneinander über Software-Kommandos zu konfigurieren. Die Positionsdaten werden über die jeweilige Ethernet Verbindung versendet. Zum Aufbau einer Trigger-Kette ist dabei folgende Verschaltung zwischen den EIB 74x zu verwenden:

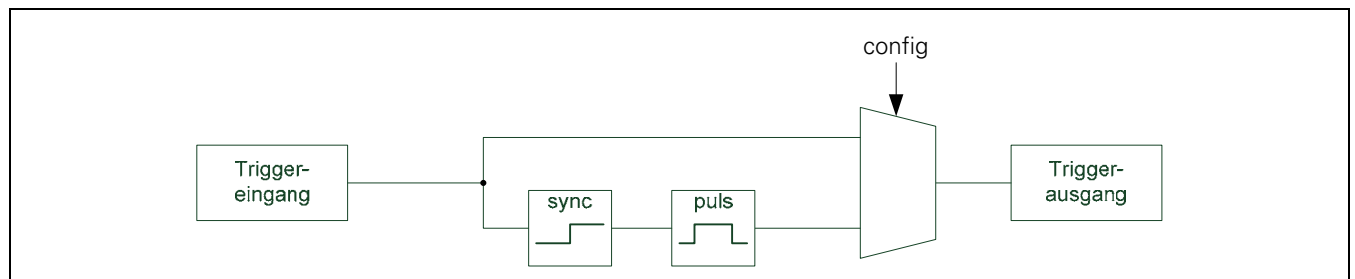
- Trigger Out + → Trigger In +
- Trigger Out - → Trigger In -
- GND to GND

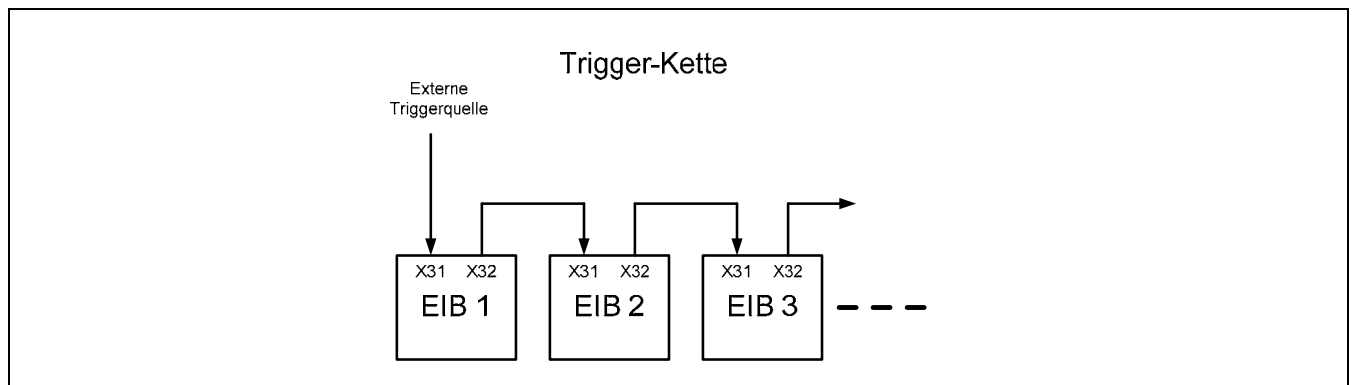
Ein Impuls am Triggereingang hat eine Länge von 2  $\mu$ s und wird synchron zum Systemtakt der EIB 74x erzeugt. Das Trigger-Ereignis entspricht der steigenden Flanke des Impulses. Wird ein Signal vom Triggereingang auf den Ausgang weitergeleitet, ist dieses durch die Synchronisation auf den Systemtakt mit einem Jitter behaftet. Um diesen Jitter zu vermeiden, ist es möglich das Signal am Triggereingang direkt auf den entsprechenden Triggereingang zu schalten,

Ein Durchschleifen der Triggersignale vom Eingang zum Ausgang kann für jeden Kanal separat erfolgen. Der Eingang 1 kann mit Ausgang 1 verbunden werden, usw. In der Abbildung unten ist ein Triggereingang und der entsprechende Ausgang dargestellt.

##### Anmerkung:

Um die Polarität des Signals am Triggereingang zu ändern, können die differenziellen Signale „Trigger Out +“ und „Trigger Out -“, getauscht werden. Für single-ended Signale ist entsprechend der Ausgang „Trigger Out -“, zu verwenden (siehe Anleitung zur Installation/Inbetriebnahme).





### Konfiguration der Trigger Ein- bzw. Ausgänge als logische Ein- bzw. Ausgänge

Die Trigger Ein- bzw. Ausgänge können auch als logische Ein- bzw. Ausgänge benutzt werden. Per Defaulteinstellung sind Trigger Ein- bzw. Ausgänge eingestellt. Über ein Software-Kommando können die Ports individuell als logische Ein- bzw. Ausgänge oder als Trigger Ein- bzw. Ausgänge konfiguriert werden. Eine gleichzeitige Benutzung als Trigger bzw. logischer Ein- oder Ausgang ist nicht möglich.

## 3.2 Logische Ein- und Ausgänge

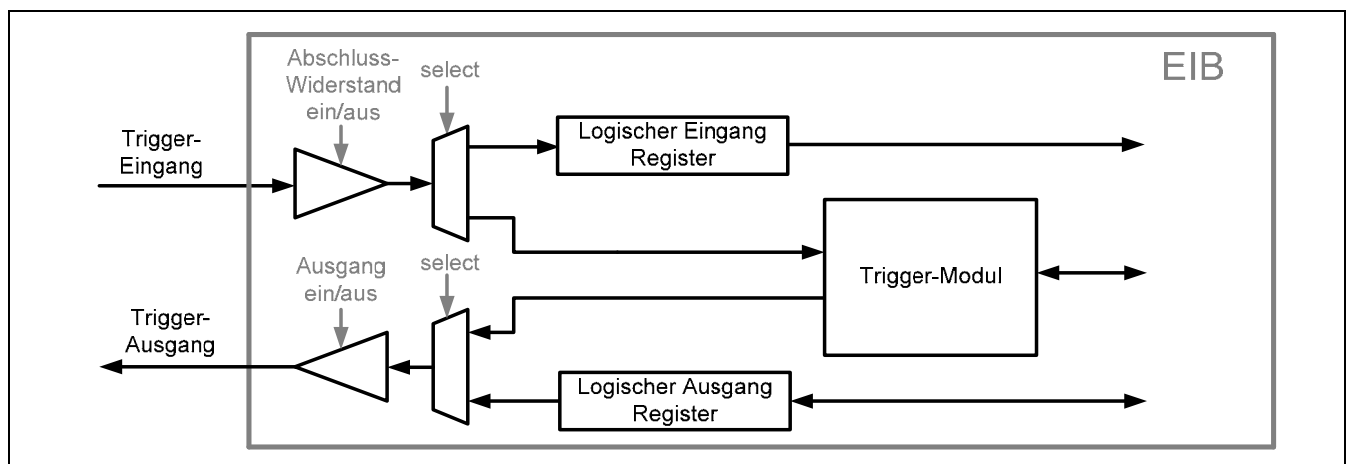
### Logische Eingänge

Jeder Trigger Eingang kann individuell auf einen logischen Eingang umgestellt werden. Der Pegel des entsprechenden Einganges kann per Software-Kommando ausgelesen werden. Der 120 Ohm Abschlusswiderstand ist auch in diesem Betriebsmodus per Konfiguration zu- oder abschaltbar.

### Logische Ausgänge

Jeder Trigger Ausgang kann individuell auf einen logischen Ausgang umgestellt werden. Zusätzlich ist es möglich den Ausgangspegel zurückzulesen. Die Ausgänge können unabhängig von der Konfiguration individuell aktiviert oder deaktiviert werden.

Die folgende Grafik zeigt die Möglichkeiten der Trigger Ein- bzw. Ausgänge im Überblick. Es ist nur ein Kanal dargestellt.



## 3.3 Triggermodul

Das Triggermodul ermöglicht die Auswahl und Kontrolle der Triggerquellen. Darüber hinaus generiert es interne Triggersignale. Externe Triggersignale können verzögert werden, wobei die Verzögerungszeit für jeden Eingang separat einstellbar ist. Der Referenzimpuls eines Messgeräts mit 1Vss-Schnittstelle kann jeweils für die zugehörige Achse als Triggerquelle verwendet werden. Der Referenzimpuls der Achse 1 wird zusätzlich mit den Signalen A und B der Achse 1 logisch UND verknüpft und kann als Triggersignal für beliebige Achsen dienen. Die aktive Flanke des Referenzsignals ist dabei jeweils einstellbar. Wird für mehrere Achsen als Triggerquelle der Referenzimpuls ausgewählt, so erfolgt die Triggerung für jede Achse mit dem eigenen Referenzimpuls. Wird in diesem Fall als Master-Triggerquelle ebenfalls der Referenzimpuls gewählt, muss auf jeder Achse ein Referenzimpuls auftreten, bevor das Datenpaket übertragen wird. Darüber hinaus existieren vier frei zuteilbare Kanäle für Software-Trigger.

Der Interval Counter erzeugt Triggersignale abhängig von der Position des Messgeräts an Achse 1. Eine Signalperiode des Messgeräts lässt sich über einen einstellbaren Interpolator in mehrere Zählschritte aufteilen. Die Triggerung erfolgt wahlweise an einer bestimmten Position oder in äquidistanten Abständen.

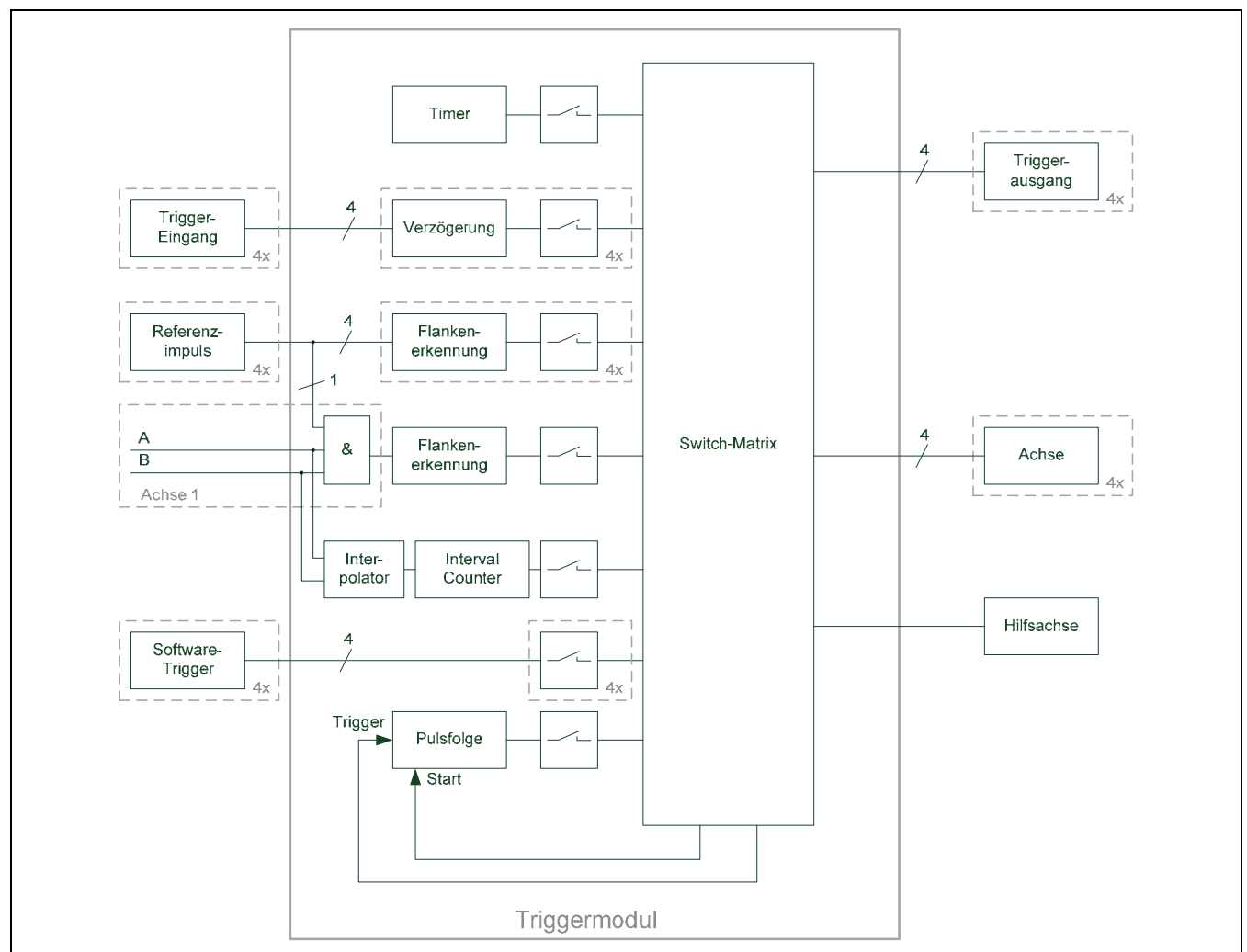


Der Pulszähler ist keine separate Triggerquelle, sondern erlaubt es die Anzahl der Triggerimpulse anderer Quellen zu begrenzen. Eine auswählbare Triggerquelle kann Impulse liefern, die so lange gesperrt werden, bis mit dem Startsignal das Tor geöffnet wird. Fortan werden alle Triggerimpulse gezählt und nach einer einstellbaren Anzahl das Tor wieder geschlossen. Außerdem ist es möglich den Zähler neu zu laden, während das Tor geöffnet ist. Die Anzahl der Triggerimpulse kann auf diese Weise erhöht werden.

Die Switch-Matrix erlaubt es die Triggerquellen individuell an die Senken, wie z.B. die Triggerausgänge oder die Achsen zu führen. Allerdings können nicht alle Quellen mit allen Senken verbunden werden. Die nachfolgende Tabelle gibt einen Überblick über die möglichen Kombinationen. Pro Senke ist immer nur eine Triggerquelle zulässig. Für den Pulszähler gibt es ein Triggersignal, dessen Triggerimpulse über das interne Tor gesteuert werden. Das Startsignal öffnet das Tor für die Triggerimpulse.

Triggerquelle	Triggerausgang	Achse	Hilfsachse	Pulszähler Trigger	Pulszähler Start
Triggereingang	x	x	x	x	x
Referenzimpuls	–	x	x	x	x
Referenzimpuls maskiert	x	x	x	x	x
Interval Counter	x	x	x	x	x
Pulszähler	x	x	x	–	–
Software-Trigger	x	x	x	–	x
Timer	x	x	x	x	–

Alle Triggerquellen können separat abgeschaltet werden. Dadurch ist es möglich die EIB 74x zu konfigurieren und zuletzt die Triggerquellen freizugeben. Dabei kann eine beliebige Kombination an Triggerquellen zeitgleich freigegeben oder gesperrt werden

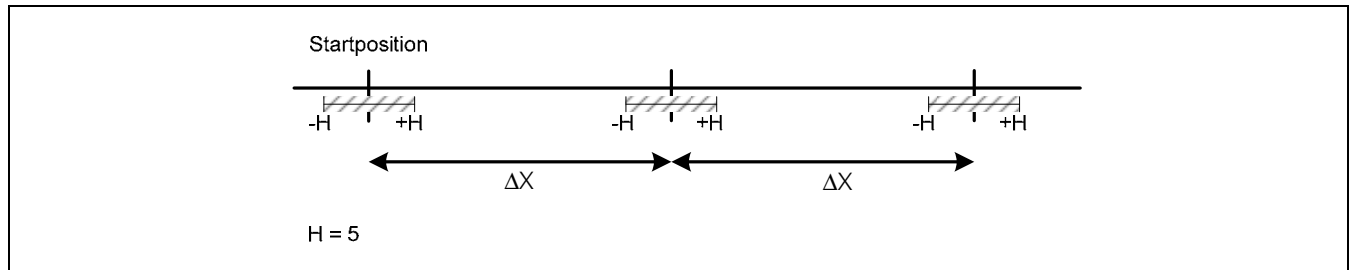


### 3.4 Interval Counter

Der Interval Counter ermöglicht eine positionsabhängige Triggerung in Verbindung mit einem inkrementalen Messerät an Achse 1. Das Messgerätesignal kann dabei interpoliert werden (siehe Kapitel „Hilfsachse“).

Die Triggerung erfolgt an einer bestimmten Position oder es werden äquidistante Triggerimpulse mit einem einstellbaren Positions-Abstand erzeugt. Die Ausgabe der Triggerimpulse erfolgt ab dem Überfahren einer einstellbaren Startposition und dann fortlaufend mit dem Positions-Abstand in beide Zählrichtungen. Der Positions-Abstand  $\Delta X$  muss in Zählsschritten angegeben werden (zur Berechnung der Schrittweite siehe Kapitel „Hilfsachse“).

Eine Hysterese verhindert ein Mehrfach-Triggeren vor allem bei hoher Interpolation des Messgerätesignals. Nachdem ein Triggerimpuls an einer Position ausgegeben wurde, muss sich der Wert des Positions Zählers um +H oder -H ändern, bevor an der gleichen Position erneut ein Triggerimpuls erzeugt wird.



### 3.5 Maximale Triggerrate

Die maximale Triggerrate der EIB 74x ist abhängig von der eingestellten Betriebsart (mit Ausnahme der Betriebsart „Polling“):

- |                       |             |
|-----------------------|-------------|
| • Soft Realtime Mode: | max. 10 kHz |
| • Recording Mode:     | max. 50 kHz |
| • Streaming Mode:     | max. 50 kHz |

#### Anmerkung:

Im Streaming Mode ist zusätzlich die Datenrate begrenzt auf 1.200.000 Byte/s. Die Datenrate ergibt sich aus der Größe eines Datenpakets und der Triggerrate.

$$\frac{\text{Datenpaketgröße}}{\text{Byte}} \cdot \frac{\text{Triggerrate}}{\text{Hz}} \leq 1.200.000$$

Hierbei muss sichergestellt sein, dass die Datenrate nicht durch den Host, auf dem die Daten weiter verarbeitet werden, begrenzt wird.

Zwischen zwei Trigger-Ereignissen muss ein bestimmter Zeitabstand eingehalten werden, den die EIB 74x für die Positionsberechnung benötigt. Wird dieser Zeitabstand nicht eingehalten, ist also die Triggerrate zu hoch, dann können Trigger-Ereignisse von der EIB 74x nicht akzeptiert werden, gehen also verloren (Lost Trigger). Dies wird von der EIB 74x detektiert und im Statuswort des Positions-Datenpakets mit dem Bit „Lost Trigger“ angezeigt. Dieses Bit ist solange auf „1“ gesetzt, bis es aktiv von der Kunden-Softwareapplikation mit einem Clear-Kommando rückgesetzt wird.

Obige Werte gelten bei Verwendung von Inkremental-Signalen. Bei Verwendung der EnDat-Schnittstelle ist die EnDat-Übertragungszeit zu beachten.

#### Achtung:

Wird die maximale Triggerrate massiv überschritten (z.B. Fehl-Parametrierung oder zu viele Ereignisse am externen Trigger-Eingang, dann kann dies dazu führen, daß die EIB nicht mehr auf externe Kommandos reagiert und nur nach einem Hard-Reset wieder ansprechbar ist.

### 3.6 Zähler für akzeptierte Trigger-Ereignisse

Neben der Überwachung auf Lost Trigger verfügt die EIB 74x zur weiteren Fehlerrückmeldung über einen Zähler, der mit jedem eintreffenden und akzeptierten Trigger-Ereignis der Master-Triggerquelle inkrementiert wird. Ein Triggerereignis wird dann akzeptiert, wenn oben erwähnter Zeitabstand eingehalten wird. Triggerereignisse, die zu Lost Trigger führen, werden nicht gezählt. Der Zählerwert wird im Positions-Datenpaket übertragen und kann auf Stetigkeit überwacht werden. Damit lassen sich verloren gegangene Positions-Datenpakete aufdecken.

## 4 Timestamp

Die Funktion "Timestamp" dient ebenfalls der Überwachung des Datenflusses. Der Timestamp Zähler ist ein frei laufender Timer mit einem frei programmierbaren Zeitintervall. Jedes Trigger-Ereignis, das zu einer Positionsvertermittlung führt, löst gleichzeitig eine Abspeicherung des aktuellen Timerwertes in das Timestamp-Register aus. Der Inhalt dieses Registers wird bei aktivierter Timestamp-Funktion mit dem Positions-Datenpaket übertragen. Damit kann die Kunden-Softwareapplikation überprüfen, ob der Latchzeitpunkt jedes einzelnen Positionswertes dem Erwartungswert entspricht. Bei Applikationen, die nicht über einen periodischen Trigger verfügen, kann mit diesem Register der Zeitpunkt des Trigger-Ereignisses übermittelt werden.

### Anmerkung:

Das Zeitintervall des Timestamp Zählers ist ein Vielfaches des internen Systemtaktes der EIB 74x. Bevor die Funktion Timestamp genutzt werden kann, muss das Zeitintervall per Software-Kommando eingestellt werden. Dazu muss zunächst der Wert „clock ticks per  $\mu$ s“ ausgelesen werden und davon abhängig das gewünschte Zeitintervall eingestellt werden. Dies ist notwendig um die Software-Kompatibilität unabhängig von verschiedenen Einstellungen für den Systemtakt zu halten.

### Hinweis:

Um den Wert für eine Zeitdauer (z.B. für den Parameter „period“ des Funktionsaufrufs „EIB7SetTimestampPeriod“) richtig zu berechnen muss an die Funktion übergeben werden:

$\text{period} = \text{Zeitintervall in } \mu\text{s} * \text{clock ticks per } \mu\text{s}$

Der Wert für „clock ticks per  $\mu$ s“ kann z.B. mit den Funktionen EIB7GetTimerTriggerTicks oder EIB7GetTimestampTicks ausgelesen werden.

Der Hinweis gilt für den Funktionsaufruf im Kapitel 7.25 und in entsprechender Weise auch für die Aufrufe im Kapitel 7.28 (bitte zusätzlich Kapitel 3.5 beachten) sowie in Kapitel 9.4.

## 5 Statuswort

Das Statuswort muss abhängig von der Art der Anfrage interpretiert werden:

- Inkrementale Positionsdaten
- EnDat Positionsdaten
- Abfrage von EnDat Zusatzinformationen

Das Statuswort wird für jeden Messgeräte-Kanal separat übermittelt und ist unabhängig vom eingestellten Betriebsmodus.

Bit Nr.	Inkrementelle Position	EnDat Position	EnDat Zusatzinformation	Hilfsachse
0	1 = Position gültig	1 = Position gültig	1 = Zusatzinformation gültig	1 = Position gültig
1	1 = Fehler Signalamplitude	1 = CRC Fehler	1 = CRC Fehler	1 = Fehler Signalamplitude
2	Reserviert	reserviert	reserviert	reserviert
3	1 = Frequenzüberschreitung	reserviert	reserviert	1 = Frequenzüberschreitung
4	1 = Fehler Spannungsversorgung Messgerät	1 = Fehler Spannungsversorgung Messgerät	reserviert	1 = Fehler Spannungsversorgung Messgerät
5	1 = Fehler Lüfter	1 = Fehler Lüfter	Inhalt I0	1 = Fehler Lüfter
6	Reserviert	reserviert	Inhalt I1	reserviert
7	1 = Lost Trigger	1 = Lost Trigger	Inhalt I2	1 = Lost Trigger
8	1 = Referenzposition 1 gespeichert	1 = EnDat Fehlermeldung 1	Inhalt I3	1 = Referenzposition gespeichert
9	1 = Referenzposition 2 gespeichert	1 = EnDat Fehlermeldung 2	Inhalt I4	reserviert
10	1 = codierter Referenzwert bei abstandscodierten Referenzmarken ist gültig	reserviert	EnDat Busy Bit	reserviert
11	1 = Fehler bei der Berechnung des codierten Referenzwertes bei abstandscodierten Referenzmarken. Fehler bei Überwachung der Referenzmarken	reserviert	EnDat RM Bit	reserviert
12	1 = Homing-Signal aktiv	reserviert	EnDat WRN Bit	reserviert
13	1 = Limit-Signal aktiv	reserviert	reserviert	reserviert
14	reserviert	reserviert	reserviert	reserviert
15	reserviert	reserviert	reserviert	reserviert

## Anmerkungen zu den Fehlerbits

Name	Bedeutung
Position gültig	1 → kein Fehler aufgetreten Dieses Bit gibt an, ob die übertragene Position gültig ist oder nicht
Zusatzinformation gültig	1 → Eine EnDat Zusatzinformation wurde empfangen Andernfalls ist keine Zusatzinformation angewählt oder wurde nicht empfangen
Fehler Signalamplitude	1 → Signalamplitude der 1 Vss Inkrementalsignale ist bzw. war zu niedrig (einmalig oder mehrmalig seit dem letzten Löschen dieser Fehlermeldung)
Frequenzüberschreitung	1 → zu hohe Eingangssignalfrequenz wurde detektiert (einmalig oder mehrmalig seit dem letzten Löschen dieser Fehlermeldung)
CRC Fehler	1 → CRC Fehler bei der EnDat Datenübertragung
Fehler Spannungsversorgung Messgerät	1 → Spannungsversorgung des Messgerätes wurde automatisch abgeschaltet. (Überstromabsicherung hat angesprochen)
Fehler Lüfter	1 → Der Lüfter der EIB 74x arbeitet fehlerhaft
Lost Trigger	Siehe Abschnitt "Maximale Triggerrate"
Referenzposition 1 gespeichert	1 → Referenzposition 1 wurde abgespeichert (seit dem letzten entsprechenden Software-Kommando)
Referenzposition 2 gespeichert	1 → Referenzposition 2 wurde abgespeichert (seit dem letzten entsprechenden Software-Kommando)
codierter Referenzwert bei abstandscodierten Referenzmarken ist gültig	1 → codierter Referenzwert für abstandscodierte Referenzmarken wurde erfolgreich berechnet (seit dem letzten entsprechenden Software-Kommando)
Fehler bei der Berechnung des codierten Referenzwertes bei abstandscodierten Referenzmarken	1 → Fehler bei Berechnung des codierten Referenzwertes; muss explizit rückgesetzt werden. Bei der automatischen Überwachung der Referenzmarken wurde ein Fehler erkannt. Der Fehler muss explizit gelöscht werden.
Homing-Signal	1 → Das Homing-Signal (L1) ist zum Zeitpunkt der Positionsabfrage aktiv
Limit-Signal	1 → Das Limit-Signal (L2) ist zum Zeitpunkt der Positionsabfrage aktiv
EnDat Fehlermeldung 1	1 → Fehlermeldung 1 aktiv
EnDat Fehlermeldung 2	1 → Fehlermeldung 2 aktiv
EnDat Busy bit	1 → Busy Bit ist gesetzt
EnDat RM bit	1 → RM (Referenzmarke) Bit ist gesetzt
EnDat WRN bit	1 → WRN (Warnung) Bit ist gesetzt
Inhalt I0..I4	Diese fünf Bit definieren den Inhalt der empfangenen Zusatzinformation. Diese Information wird benötigt, damit die Kunden-Softwareapplikation die Daten interpretieren kann.

Die Fehlerbits werden nicht automatisch rückgesetzt sondern müssen über ein Software-Kommando aktiv durch die Kunden-Softwareapplikation rückgesetzt werden. Wird ein Fehler nicht rückgesetzt, wird er mit jedem weiteren Positions-Datenpaket erneut übertragen.

Bei inkrementellen Messgeräten zeigt ein Fehler im Positions-Datenpaket an, dass die Position nicht mehr gültig ist und jeglichen Bezug zu Referenzmarke oder anderen Messkanälen verloren hat.

Das Auftreten eines Fehlers kann das Ansprechen anderer Fehler nach sich ziehen. Bei einem Fehler in der Versorgungsspannung des Messgerätes werden auch andere Fehler mit ansprechen. Aus diesem Grund sollte immer zuerst ein eventueller Fehler der Spannungsversorgung rückgesetzt werden. Nachdem die Spannungsversorgung stabil ist (Wartezeit ca. 1,5 Sekunden) sollten die anderen Fehler rückgesetzt werden.

**Lost Trigger**

Das „Lost Trigger“ Bit zeigt an, dass mindestens ein Triggerereignis aufgrund einer zu kurzen Zeitspanne zwischen zwei Triggerereignissen nicht korrekt verarbeitet wurde. Das „Lost Trigger“ Bit kann ebenfalls auftreten, wenn die Triggerleitung mit Störungen überlagert ist oder EMV Einflüsse die Übertragung negativ beeinflussen. Ein „Lost Trigger“ bedeutet nicht, dass die Positionswerte falsch sind, es wird lediglich angezeigt, dass Triggerereignisse nicht korrekt verarbeitet werden konnten. Das Rücksetzen muss ebenfalls aktiv über ein Software-Kommando erfolgen.

**Referenzposition gespeichert**

Die beiden Bits „Referenzposition 1 (2) gespeichert“ zeigen an, dass eine gültige Referenzmarke erkannt und abgespeichert wurde. Die entsprechende Referenzposition im Positions-Datenpaket ist damit gültig.

**Codierter Referenzwert bei abstandscodierten Referenzmarken ist gültig**

Dieses Bit wird rückgesetzt durch Senden des entsprechenden Software-Kommandos zur Einspeicherung von Referenzpositionen. Nach der erfolgreichen Berechnung des codierten Referenzwertes wird dieses Bit aktiv gesetzt. Dies bedeutet, dass der im Positions-Datenpaket übertragene Wert „codierter Referenzwert bei abstandscodierten Referenzmarken“ für die Berechnung der Absolutposition verwendet werden kann.

**Fehler bei Referenzposition bei abstandscodierten Referenzmarken**

Dieses Bit wird gesetzt, wenn während der Berechnung des codierten Referenzwertes für abstandscodierte Referenzmarken ein Fehler aufgetreten ist. Ein Grund dafür kann z.B. sein, dass während der Phase der Referenzierung ein Richtungswechsel stattgefunden hat und dadurch dieselbe Referenzmarke zweimal detektiert wurde. Der Fehler muss aktiv rückgesetzt werden und wird nicht automatisch durch ein erneutes Senden des Software-Kommandos zur Einspeicherung von Referenzpositionen rückgesetzt. Darüber hinaus wird dieses Bit gesetzt, wenn die automatische Überwachung der Referenzmarken aktiviert wurde und ein Fehler aufgetreten ist. Dies gilt auch für nicht abstandscodierte Messgeräte. Auch in diesem Fall muss der Fehler explizit gelöscht werden.

**Homing/Limit-Signale**

Das Homing/Limit-Signal zeigt an, ob das entsprechende Signal aktiv ist, sofern dies vom Messgerät unterstützt wird. Der Zustand des Signals wird nicht gespeichert und muss daher nicht gelöscht werden.

**Fehler Lüfter**

Dieses Bit zeigt an, ob der Lüfter der EIB 74x einwandfrei arbeitet oder nicht. Das Fehlerbit hat keinen Einfluss auf die Positionsdaten. Das Fehler-Bit wird nicht gespeichert und muss daher nicht gelöscht werden. Das Bit ist gesetzt, so lange der Lüfter fehlerhaft arbeitet.

**Anmerkung:**

Weitere Hinweise siehe „Inbetriebnahme - Anleitung“.

Wird die Lüfter-Überwachung nicht unterstützt, dann ist dieses Bit immer auf „0“.

## 6 Ethernet Interface

Das Ethernet (LAN) Interface wird für die Konfiguration der EIB 74x und für die Übertragung der Positions-Datenpakete genutzt. Die Konfiguration erfolgt über TCP-Kommandos und die Übertragung der Daten im Soft Realtime Modus mit UDP-Paketen. Die Einstellungen der PC-Firewall sind entsprechend zu wählen. Die Netzwerkeinstellungen der EIB 74x lassen sich mit Hilfe der Software-Kommandos ändern. Wahlweise ist die IP-Adresse fest einstellbar, oder kann dynamisch von einem DHCP-Server bezogen werden. Weitere Einzelheiten siehe „Inbetriebnahme – Anleitung“.

## 7 Betriebsmodi

Von der EIB 74x werden folgende Betriebsarten unterstützt:

- Polling
- Soft Realtime
- Streaming
- Recording

### 7.1 Konfiguration der Datenpakete

Für die Betriebsmodi „Soft Realtime“, „Streaming“ und „Recording“ ist es notwendig ein Datenpaket zu konfigurieren. Abhängig von dieser Konfiguration werden mit jedem Triggerereignis bestimmte Daten übertragen bzw. aufgezeichnet. Dadurch ist es möglich die Datenmenge auf die tatsächlich benötigten Elemente zu begrenzen. Dies reduziert die erforderliche Übertragungskapazität sowie den benötigten Speicherplatz im „Recording“ Modus.

Ein Datenpaket ist in mehrere Regionen aufgeteilt. Jede Region enthält die Daten für eine bestimmte Achse der EIB 74x bzw. globale Informationen. Die globalen Informationen müssen immer als erste Region im Datenpaket enthalten sein. Anschließend können eine oder mehrere Regionen für die Achsen folgen. Hierbei können Achsen ausgelassen werden, jedoch müssen sie in aufsteigender Reihenfolge im Datenpaket enthalten sein. Das Beispiel „InfoGlobal-Achse1-Achse3-Achse4“ stellt ein gültiges Datenpaket dar, nicht jedoch „InfoGlobal-Achse1-Achse4-Achse3“. Die Hilfsachse muss, sofern sie verwendet wird, die letzte Region im Datenpaket sein.

Innerhalb einer jeden Region können verschiedene Datenelemente enthalten sein. Alle Möglichkeiten sind nachfolgend in einer Tabelle aufgeführt. Die Länge gibt die Anzahl der Bytes für das Datenelement an. Die Summe aller Elemente aus allen Regionen ergibt die Größe des Datenpaketes. Die Länge eines Datenpakets muss allerdings immer ein Vielfaches von 4 Bytes betragen. Falls dies bei einer bestimmten Konfiguration nicht erfüllt ist, werden automatisch am Ende entsprechend viele „Füllbytes“ angehängt.

Globale Information

Datenelement	Beschreibung	Länge in Bytes
TriggerCounter	Zähler für Triggerereignisse	2

Achse

Datenelement	Beschreibung	Länge in Bytes
Statuswort	Status und Fehlermeldungen	2
Positionswert	Aktueller Positionswert des Messgeräts	6
Timestamp	Zeitstempel für Positionswert	4
Referenzposition	Positionswert bei Referenzmarken	12
Codierte Referenzposition bei abstandscodierten Referenzmarken	Berechnete Referenzposition bei abstandscodierten Referenzmarken	6
Amplitudenwert Inkrementalsignal	Byte 0..1: Signal A Byte 2..3: Signal B	4
EnDat Zusatzinformation 1	Byte 0..1: Statuswort Byte 2..3: Zusatzinformation	4
EnDat Zusatzinformation 2	Byte 0..1: Statuswort Byte 2..3: Zusatzinformation	4

Hilfsachse

Datenelement	Beschreibung	Länge in Bytes
Statuswort	Status und Fehlermeldungen	2
Positionswert	Aktueller Positionswert des Messgeräts	4
Timestamp	Zeitstempel für Positionswert	4
Referenzposition	Positionswert bei Referenzmarke	4

In dem folgenden Beispiel ist die Konfiguration eines Datenpaketes für zwei Achsen dargestellt. Zusätzlich wird eine Region für die globalen Informationen eingefügt.

Globale Information: Trigger Counter

Achse 1: Inkrementale Schnittstelle (1V<sub>SS</sub>)  
Eine Referenzmarke

Achse 2: Inkrementale Schnittstelle (1V<sub>SS</sub>)  
Eine Referenzmarke

Paketkonfiguration:

Region	Element	Länge in Bytes
Global	TriggerCounter	2
Achse1	Statuswort	2
	Positionswert	6
	Timestamp	4
	Referenzposition	12
Achse2	Statuswort	2
	Positionswert	6
	Timestamp	4
	Referenzposition	12
	Füllbytes	2

Daraus ergibt sich eine Gesamtlänge des Datenpaketes von 52 Bytes.

Nach dem Einschalten des Gerätes lädt die EIB 74x eine Default-Konfiguration für das Datenpaket. Diese Konfiguration umfasst die globalen Informationen und je eine Region für alle 4 Achsen. Die nachfolgende Tabelle gibt den Aufbau des Datenpaketes wieder.

Region	Element	Länge in Bytes
Global	TriggerCounter	2
Achse1	Statuswort	2
	Positionswert	6
	Timestamp	4
	Referenzposition 1	6
	Referenzposition 2	6
	Codierter Referenzwert bei abstandscodierten Referenzmarken	6
	Amplitudenwert Inkrementalsignal	4
Achse2	Statuswort	2
	Positionswert	6
	Timestamp	4
	Referenzposition 1	6
	Referenzposition 2	6
	Codierter Referenzwert bei abstandscodierten Referenzmarken	6
	Amplitudenwert Inkrementalsignal	4
Achse3	Statuswort	2
	Positionswert	6
	Timestamp	4
	Referenzposition 1	6
	Referenzposition 2	6
	Codierter Referenzwert bei abstandscodierten Referenzmarken	6
	Amplitudenwert Inkrementalsignal	4
Achse4	Statuswort	2
	Positionswert	6
	Timestamp	4
	Referenzposition 1	6
	Referenzposition 2	6
	Codierter Referenzwert bei abstandscodierten Referenzmarken	6
	Amplitudenwert Inkrementalsignal	4

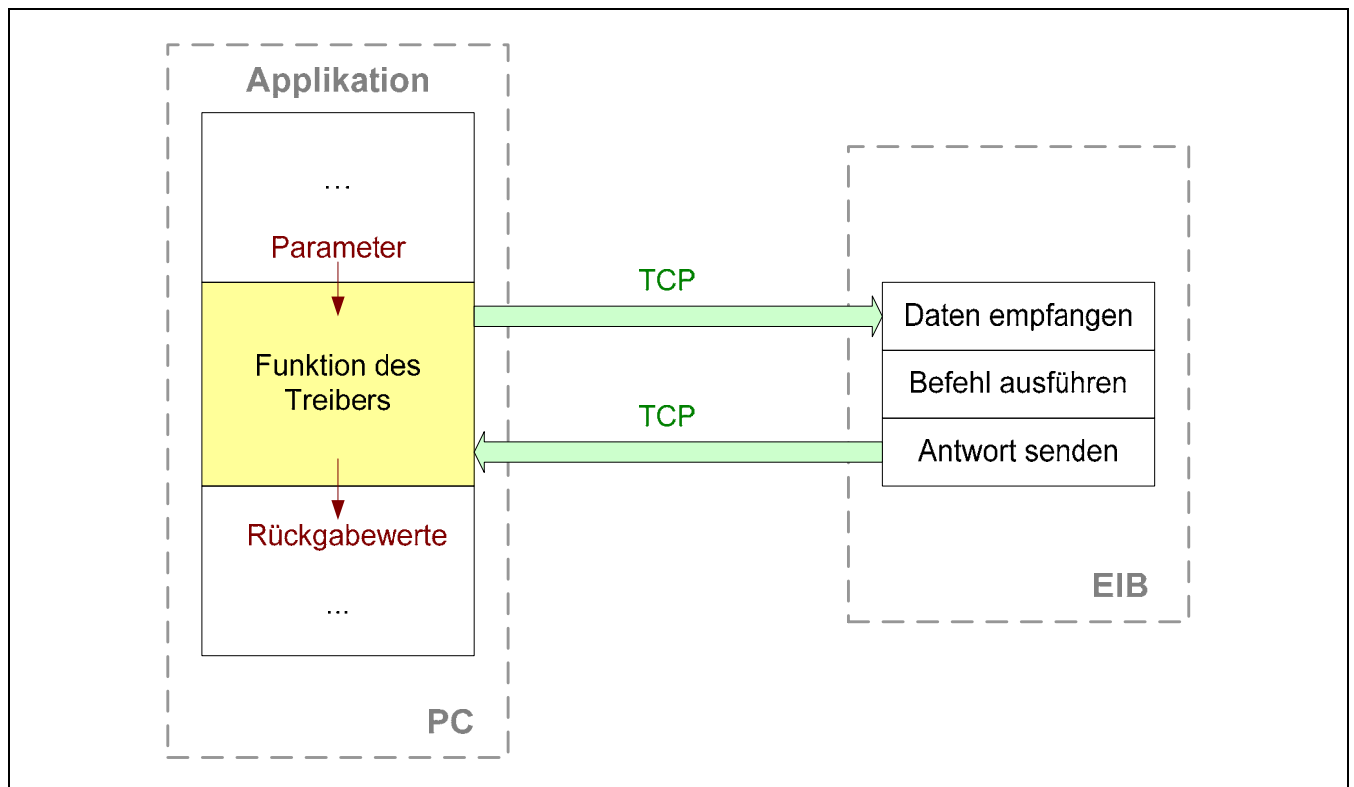
## 7.2 Betriebsmodus „Polling“

Diese Betriebsart ist per Default nach der Initialisierung der EIB 74x aktiviert. Die Positionsdaten werden in der EIB 74x ermittelt, sobald dort ein entsprechendes Kommando eintrifft. Die EIB 74x übermittelt die Daten innerhalb des Antwortpakets an die Kundenapplikation.

Das nachfolgende Diagramm verdeutlicht den Ablauf einer Positionsabfrage. Aus einer Kunden-Softwareapplikation am PC wird ein Kommando an die EIB 74x gesendet. Die EIB 74x generiert die Positionsdaten und sendet sie in einem TCP-Paket zurück. Die Daten werden an die Applikation übergeben.

Verarbeitung von Trigger-Ereignissen:

- Der Zeitpunkt der Positionswertbildung wird von der Software beeinflusst und ist damit zeitlich nicht exakt bestimmbar.
- Die Triggerung erfolgt ausschließlich über Software-Trigger



### Datenpakete in der Betriebsart „Polling“:

Abhängig von der gewählten Funktion; siehe Kapitel Funktionsaufrufe



### 7.3 Betriebsmodus „Soft Realtime“

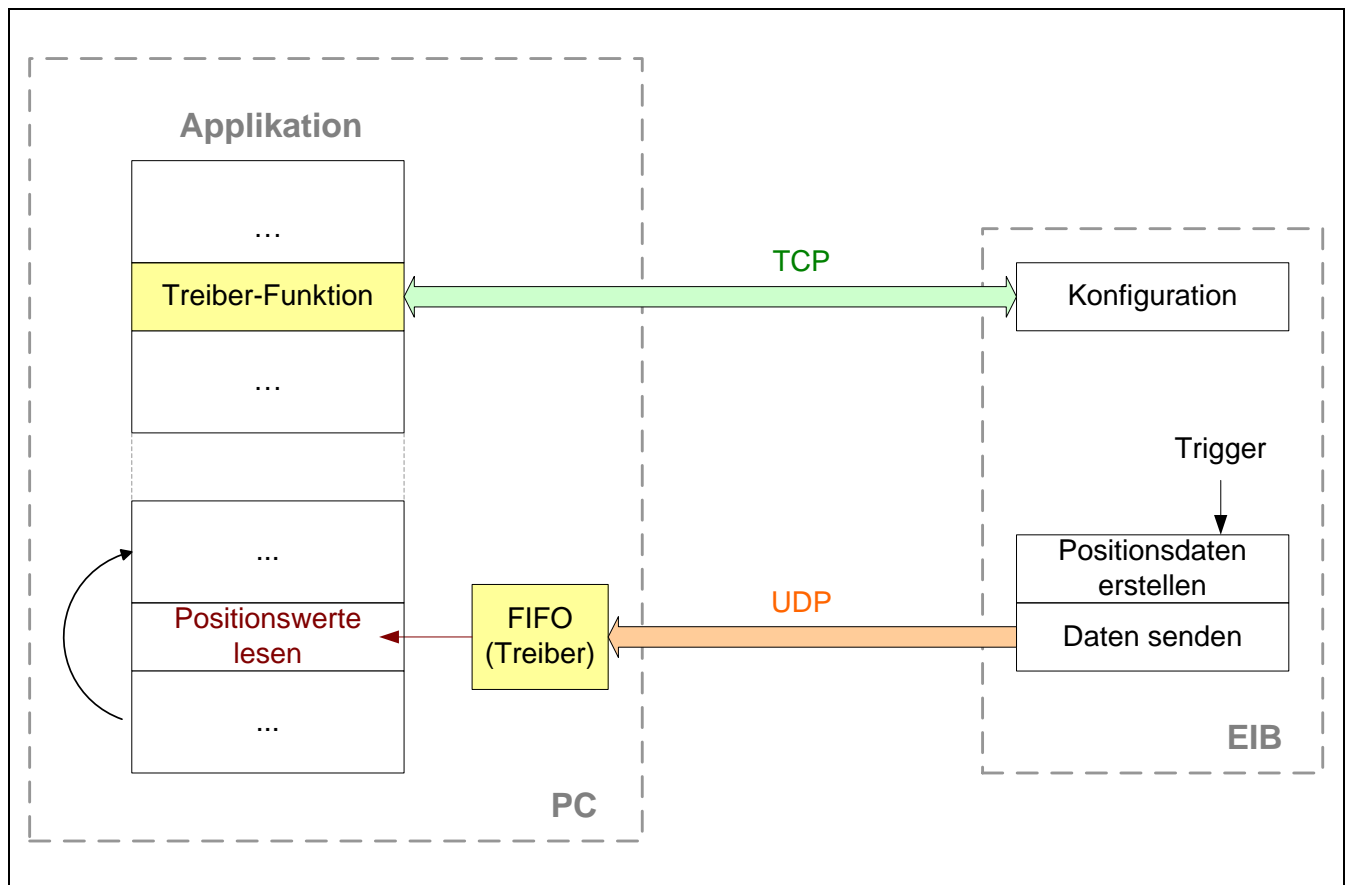
Die Positionsdaten werden mit UDP-Paketen von der EIB 74x zum PC transportiert. Dies erfolgt parallel zur TCP-Kommunikation über die Standard Ethernet-Schnittstelle. Die Positionsdaten werden generiert, wenn die EIB 74x ein Triggersignal erhält. Mit jedem Triggerereignis wird ein Datenpaket automatisch an den PC gesendet. Dort können die Pakete aus einem FIFO gelesen werden.

Für den Betrieb des Soft Realtime Modus muss die EIB 74x mit den nachfolgend aufgeführten Schritten konfiguriert werden.

- Initialisierung der EIB 74x
- Initialisierung und Konfiguration der Achsen
- Konfiguration des Datenpaketes
- Konfiguration der Trigger-Logik
- Auswahl des Betriebsmodus (Soft Realtime)
- Aktivierung der Triggerquelle

In dem nachfolgenden Diagramm ist die Kommunikation schematisch dargestellt. Die Kunden-Softwareapplikation muss die EIB 74x konfigurieren. Anschließend werden die Daten selbständig in den FIFO übertragen. Von dort kann sie die Applikation innerhalb einer Programmschleife auslesen.

Parallel zur Positionsabfrage kann der Status der EIB 74x abgerufen, oder Fehlermeldungen gelöscht werden.



Wird die Ethernet Verbindung im "Soft Realtime" Modus z.B. durch Abstecken des Ethernet Kabels getrennt, so deaktiviert die EIB 74x die Triggerquelle und sendet keine weiteren UDP-Pakete. Nachdem die Verbindung wieder hergestellt wurde muss die EIB 74x erneut für den Betriebsmodus konfiguriert werden.

Beim Beenden der Applikation müssen die oben genannten Schritte der Initialisierung in umgekehrter Reihenfolge durchlaufen werden. Zuerst ist die Triggerquelle zu deaktivieren. Anschließend kann der Betriebsmodus geändert oder die Verbindung zur EIB 74x geschlossen werden.

Verarbeitung von Trigger-Ereignissen:

- Externe Triggereingänge werden unterstützt
- Interne Triggerquellen werden unterstützt
- Software-Trigger wird unterstützt

Für eine korrekte Interpretation der Daten muss bei deren Auswertung der Aufbau des Datenpaketes berücksichtigt werden.

#### 7.4 Betriebsmodus „Streaming“

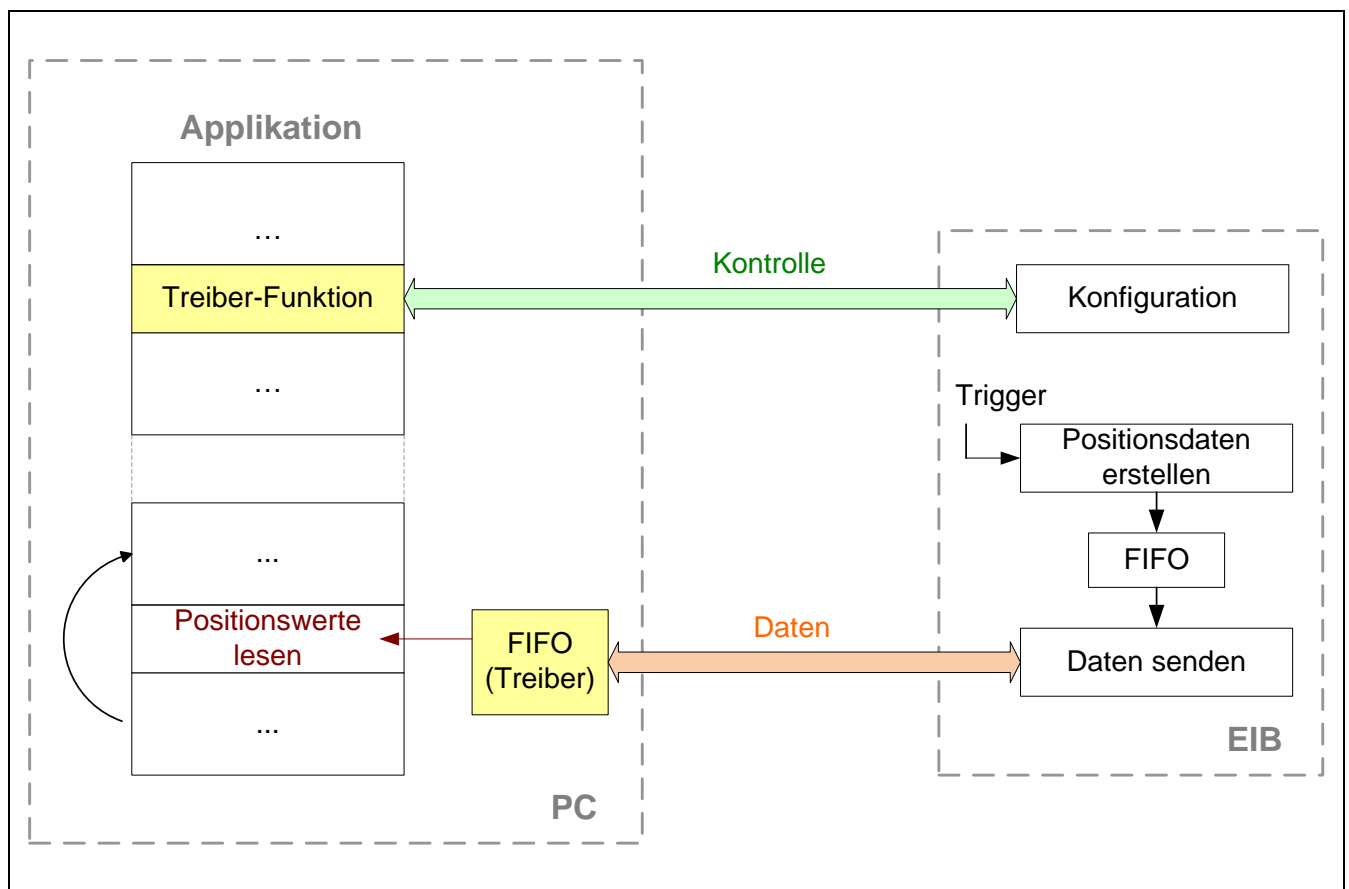
Die Positionsdaten werden von der EIB 74x gepuffert und zum PC transportiert. Dies erfolgt parallel zur TCP-Kommunikation über die Standard Ethernet-Schnittstelle. Die Positionsdaten werden generiert, wenn die EIB 74x ein Triggersignal erhält. Mit jedem Triggerereignis wird ein Datenpaket erzeugt. Je nach Triggerrate und Datenvolumen werden mehrere Datenpakete zusammengefasst und zum PC gesendet. Dort können die Pakete aus einem FIFO gelesen werden.

Für den Betrieb des Streaming Modus muss die EIB 74x mit den nachfolgend aufgeführten Schritten konfiguriert werden.

- Initialisierung der EIB 74x
- Initialisierung und Konfiguration der Achsen
- Konfiguration des Datenpaketes
- Konfiguration der Trigger-Logik
- Auswahl des Betriebsmodus (Streaming)
- Aktivierung der Triggerquelle

In dem nachfolgenden Diagramm ist die Kommunikation schematisch dargestellt. Die Kunden-Softwareapplikation muss die EIB 74x konfigurieren. Anschließend werden die Daten selbständig in den FIFO übertragen. Von dort kann sie die Applikation innerhalb einer Programmschleife auslesen.

Parallel zur Positionsabfrage kann der Status der EIB 74x abgerufen, oder Fehlermeldungen gelöscht werden. Insbesondere kann der Status des FIFO in der EIB 74x abgefragt werden, um einen Überlauf frühzeitig zu erkennen.



Sobald die Positionswerte aus dem FIFO am PC gelesen wurden, wird dies an die EIB bestätigt. Falls in der EIB 74x weitere Daten vorhanden sind, werden diese übertragen.

Beim Beenden der Applikation müssen die oben genannten Schritte der Initialisierung in umgekehrter Reihenfolge durchlaufen werden. Zuerst ist die Triggerquelle zu deaktivieren. Anschließend kann der Betriebsmodus geändert oder die Verbindung zur EIB 74x geschlossen werden.

Verarbeitung von Trigger-Ereignissen:

- Externe Triggereingänge werden unterstützt
- Interne Triggerquellen werden unterstützt
- Software-Trigger wird unterstützt

Für eine korrekte Interpretation der Daten muss bei deren Auswertung der Aufbau des Datenpaketes berücksichtigt werden.

## 7.5 Betriebsmodus „Recording“

Die Positionsdaten werden im Speicher der EIB 74x abgelegt. Mit jedem Triggerereignis wird ein Datenpaket erzeugt und gespeichert. Nach Abschluss der Aufzeichnungsphase können die Daten übertragen werden.

Der Betriebsmodus Recording unterstützt zwei Betriebsarten. Im „Single Shot“ Betrieb wird die Aufzeichnung der Daten automatisch beendet, sobald der gesamte Speicher gefüllt ist. Im „Rolling“ Betrieb werden die Daten in einem Ringspeicher abgelegt. Wenn der gesamte Speicher gefüllt ist, wird der älteste Eintrag überschrieben. Nach dem Beenden des Recording Betriebsmodus können die letzten n Samples aus dem Speicher gelesen werden.

Die Aufzeichnungstiefe ist abhängig von der Größe eines Datenpaketes und kann ausgelesen werden (siehe Teil 2, 7.42).

Für den Betrieb des Recording Modus muss die EIB 74x mit den nachfolgend aufgeführten Schritten konfiguriert werden.

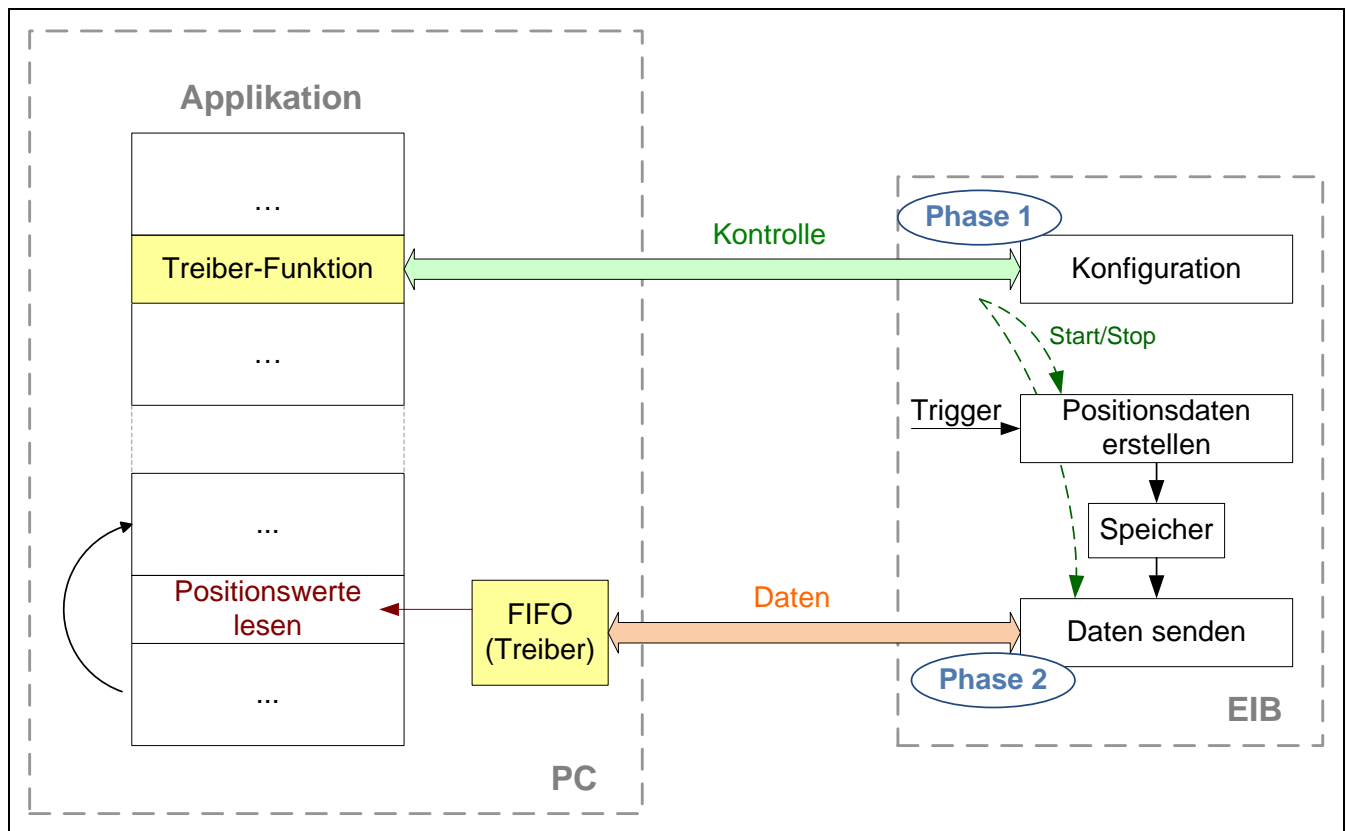
- Initialisierung der EIB 74x
- Initialisierung und Konfiguration der Achsen
- Konfiguration des Datenpaketes
- Konfiguration der Trigger-Logik
- Auswahl des Betriebsmodus (Recording)
- Aktivierung der Triggerquelle

Nach dem Abschluss der Aufzeichnungsphase müssen folgende Schritte ausgeführt werden

- Deaktivieren der Triggerquelle
- Auswahl des Betriebsmodus (Polling)
- Datenübertragung starten

In dem nachfolgenden Diagramm ist die Kommunikation schematisch dargestellt. Die Kunden-Softwareapplikation muss die EIB 74x konfigurieren. In der ersten Phase (Aufzeichnung) werden die Daten in der EIB 74x gespeichert. In der zweiten Phase (Datenübertragung) werden die Daten zum Host übertragen und in einem FIFO gespeichert. Von dort kann sie die Applikation innerhalb einer Programmschleife auslesen.

Während der Aufzeichnung kann der Status der EIB 74x abgerufen, oder Fehlermeldungen gelöscht werden. Insbesondere kann der Status des Speichers in der EIB 74x abgefragt werden.



Verarbeitung von Trigger-Ereignissen:

- Externe Triggereingänge werden unterstützt
- Interne Triggerquellen werden unterstützt
- Software-Trigger wird unterstützt

Für eine korrekte Interpretation der Daten muss bei deren Auswertung der Aufbau des Datenpaketes berücksichtigt werden.

## 8 Firmware Update

Ein Update der Firmware der EIB 74x kann durch den Benutzer mit einem TFTP client durchgeführt werden. Es dürfen allerdings nur spezielle Update-Files für die EIB 74x von HEIDENHAIN aufgespielt werden.

Das folgende Beispiel geht für das Firmwareupdate von einem Computer mit dem Betriebssystem „Windows“ aus. Die EIB 74x muss über Ethernet mit dem Computer verbunden sein. Der Dateiname für das Update ist in diesem Beispiel „update\_633281-10.flash“. Diese Datei ist gespeichert unter „C:\temp\EIB“.

- Ausführen der Windows Kommandozeile
- Speichern des Update-Files unter „C:\temp\EIB\update\_633281-10.flash“
- Start des TFTP File Transfers:  
    > tftp -i 192.168.1.2 put C:\temp\EIB\update\_633281-10.flash tmp\update.flash

Option -i:	Aktiviert den "binary file transfer"
IP Adresse:	"192.168.1.2" (Default-Einstellung) oder kundenspezifische Einstellung
Kommando "put":	Transfer vom Host zur EIB 74x
Quelldatei:	in diesem Beispiel "C:\temp\EIB\update_633281-10.flash"
Zieldatei:	immer "tmp\update.flash"

Wenn die Datei erfolgreich übertragen wurde, wird eine entsprechende Meldung vom TFTP client in der Kommandozeile ausgegeben. Die Status LED der EIB 74x wird ausgeschaltet. Nach der geräteinternen Datenübertragung in den Flash-Speicher wird die Status LED wieder eingeschaltet. Dieser Vorgang kann bis zu 60 Sekunden dauern. Während der Phase in der die Status LED abgeschaltet ist, darf die Spannungsversorgung nicht abgeschaltet werden und es dürfen auch keine Kommandos über die Ethernet Schnittstelle an die EIB 74x gesandt werden.

Nachdem die Status LED wieder aktiv ist, sollte über das entsprechende Software-Kommando abgefragt werden, ob das Update erfolgreich beendet wurde oder nicht. Der Status des Update Prozesses kann bis zum nächsten Booten der EIB 74x abgefragt werden.

Mit dem nächsten Reset bootet die EIB 74x die neue Version der Firmware.

Im Falle eines Fehlers während des Firmware-Updates werden die entsprechenden Einstellungen laut der Tabelle „Reset der EIB 74x“ (siehe „Inbetriebnahme – Anleitung“) gebootet.

## 9 Reset

Siehe „Inbetriebnahme – Anleitung“

# Teil 2: Treiber-Software

## 1 Allgemeine Informationen

Für den Zugriff auf die EIB 74x aus einer Softwareapplikation werden Funktionen zur Verfügung gestellt. Diese Funktionssammlung wird als DLL für Windows Systeme und als SO-Bibliothek für Linux geliefert. Folgende Betriebssysteme werden unterstützt:

- Windows 2000, Windows XP, Windows Vista, Windows 7
- Linux/Unix mit Kernel 2.6, (i386 Systeme)

Zusätzlich zu den Bibliotheken wird eine Header-Datei geliefert, die eine Integration der Funktionen in C/C++ Programme ermöglicht. Um ein Programm zu erstellen muss die Bibliothek in das Projekt eingebunden werden

Für LabView werden sog. „.vi“ zur Verfügung gestellt, die als Basis die Windows DLL haben. Die Benennung, als auch die Funktionalität und die Ein- bzw. Ausgabeparameter der „.vi“ orientieren sich an den entsprechenden Funktionsaufrufen, die im Anschluss dokumentiert sind. Speziell bei komplexeren Datentypen kann es erforderlich sein, dass die „.vi“ neben dem DLL Aufruf noch eine LabView spezifische Anpassung enthält. Die entsprechenden Anpassungen sind durch das Öffnen der „.vi“ ersichtlich.

## 2 Installationsanleitung

Die angegebenen Verzeichnisse und Dateien beziehen sich auf die Treiber-CD für die EIB 74x.

### 2.1 Windows

Damit eine Anwendung die DLL laden kann, muss die Datei „eib7.dll“ aus dem Verzeichnis „EIB\_74x\windows\bin“ ins Windows-Systemverzeichnis kopiert werden (z.B. „C:\Windows\system32“). Für 64 Bit Betriebssysteme muss die Datei eib7\_64.dll aus „EIB\_74x\windows\bin64“ in das Windows-Systemverzeichnis (z.B. „C:\Windows\system32“) kopiert und in „eib7.dll“ umbenannt werden. Für eine Kompatibilität mit 32 Bit Applikationen sollte zusätzlich die Datei eib7.dll in das Systemverzeichnis „SysWOW64“ im Windows-Ordner (z.B. „C:\Windows“) kopiert werden. Alternativ kann der Pfad für die DLL im System bekannt gegeben werden. Das Interface der DLL ist über die beiden Dateien „eib7.lib“ in „EIB\_74x\windows\lib“ und „eib7.h“ in „EIB\_74x\windows\include“ definiert. Diese müssen in das Softwareprojekt in der Entwicklungsumgebung eingebunden werden (für C/C++ Umgebungen). Die Datei „eib7.lib“ muss in das Library-Verzeichnis der Entwicklungsumgebung kopiert, oder deren Pfad eingetragen werden.

### 2.2 Linux

Damit eine Anwendung die SO-Bibliothek laden kann, sollte für 32 Bit Betriebssysteme die Datei „libeib7.so“ von der CD aus dem Verzeichnis „EIB\_74x/linux/lib“ ins Verzeichnis „usr/local/lib“ kopiert werden. Für 64 Bit Betriebssysteme muss die Datei „libeib7\_64.so“ im Verzeichnis „EIB\_74x/linux/lib64“ verwendet und in „libeib7.so“ umbenannt werden. Das Interface der Bibliothek ist über die Datei „eib7.h“ in „EIB\_74x/linux/include“ definiert. Diese sollte nach „usr/local/include“ kopiert werden und ist in das Softwareprojekt in der Entwicklungsumgebung einzubinden. Die angegebenen Verzeichnisse orientieren sich an dem „Filesystem Hierarchy Standard“ für Linux-Betriebssysteme. Die Bibliothek „libeib7.so“ wurde für i386 Systeme unter Kernel 2.6 kompiliert.

## 3 Überblick

### 3.1 Kommunikationsaufbau

Für die Kommunikation mit der EIB 74x muss zuerst eine Verbindung mit Hilfe der Funktion EIB7Open() aufgebaut werden. Unter Umständen ist es notwendig, zuvor mit EIB7GetHostIP() die IP-Adresse zu bestimmen. Anschließend kann über die Gerätefunktionen die EIB 74x konfiguriert werden.

Für den Zugriff auf die Achsen sind Handles nötig, die von der Funktion EIB7GetAxis() erzeugt werden. Analog gilt dies für die IO-Ports, deren Handles von der Funktion EIB7GetIO() erzeugt werden. Über die Handles kann die Konfiguration erfolgen oder der Status abgefragt werden.

Nach dem Ende der Kommunikation muss mit der Funktion EIB7Close() die Verbindung geschlossen werden.

### 3.2 Konfiguration der Datenpakete

Das Datenpaket für die Betriebsmodi „Soft Realtime“, „Streaming“ und „Recording“ müssen vor dem Aktivieren der Modi konfiguriert werden.

Die Konfigurationsdaten werden in einem Array vom Typ `EIB7_DataPacketSection` gespeichert. Für jede Region muss ein Element des Array konfiguriert werden. Dies kann über die Funktion `EIB7AddDataPacketSection()` erfolgen (siehe 7.8). Anschließend kann diese Konfiguration über die Funktion `EIB7ConfigDataPacket()` in die EIB 74x geladen werden (siehe 7.9).

### 3.3 Polling Modus

Mit Hilfe der Achsfunktionen kann auf die Messgeräte zugegriffen werden. Dazu muss zuerst über `EIB7InitAxis()` die Achse konfiguriert werden. Anschließend lassen sich Positionswerte auslesen oder Fehlermeldungen quittieren.

Es ist nicht notwendig, eine Triggerquelle auszuwählen. Die Triggerung erfolgt implizit mit dem Aufruf der Funktion `EIB7GetPosition()`.

### 3.4 Soft Realtime Modus

Zuerst müssen die Achsen mit `EIB7InitAxis()` initialisiert und das Datenpaket sowie die Trigger-Logik konfiguriert werden. Anschließend kann der Soft Realtime Modus aktiviert werden. Im Soft Realtime Modus lassen sich lediglich die Fehlermeldungen aus dem Statuswort für die Positionswerte zurücksetzen.

Nachdem in den Soft Realtime Modus gewechselt wurde, kann die Triggerquelle aktiviert werden. Die Kunden-Softwareapplikation am Host muss die Positionsdaten kontinuierlich aus dem Empfangspuffer auslesen, um einen Überlauf zu verhindern. Dies kann über die Funktionen `EIB7ReadFIFOData()` oder `EIB7ReadFIFODataRaw()` erfolgen (siehe 7.44, 7.47). Jede dieser Funktionen liest einen oder mehrere Einträge aus dem FIFO. Jeder Eintrag enthält ein Datenpaket der EIB 74x. Die Größe eines Eintrags kann im Vorfeld über die Funktionen `EIB7SizeOfFIFOEntry()` und `EIB7SizeOfFIFOEntryRaw()` bestimmt werden (siehe 7.45, 7.48). Auf die einzelnen Komponenten eines FIFO-Eintrags kann mit Hilfe der Funktion `EIB7GetDataFieldPtr()` oder `EIB7GetDataFieldPtrRaw()` zugegriffen werden.

Darüber hinaus ist es möglich, über den Callback-Mechanismus eine Funktion zu registrieren, die aufgerufen wird, sobald neue Daten im FIFO bereitstehen (siehe 7.54).

### 3.5 Streaming Modus

Zuerst müssen die Achsen mit `EIB7InitAxis()` initialisiert, und das Datenpaket sowie die Trigger-Logik konfiguriert werden. Anschließend kann der Streaming Modus aktiviert werden. Im Streaming Modus können lediglich die Fehlermeldungen aus dem Statuswort für die Positionswerte zurückgesetzt und der Status des Puffers ausgelesen werden (siehe 7.43).

Nachdem in den Streaming Modus gewechselt wurde, kann die Triggerquelle aktiviert werden. Die Kunden-Softwareapplikation am Host muss die Positionsdaten kontinuierlich aus dem Empfangspuffer auslesen, um einen Überlauf zu verhindern. Dies kann über die Funktionen `EIB7ReadFIFOData()` oder `EIB7ReadFIFODataRaw()` erfolgen (siehe 7.44, 7.47). Jede dieser Funktionen liest einen oder mehrere Einträge aus dem FIFO. Jeder Eintrag enthält ein Datenpaket der EIB 74x. Die Größe eines Eintrags kann im Vorfeld über die Funktionen `EIB7SizeOfFIFOEntry()` und `EIB7SizeOfFIFOEntryRaw()` bestimmt werden (siehe 7.45, 7.48). Auf die einzelnen Komponenten eines FIFO-Eintrags kann mit Hilfe der Funktion `EIB7GetDataFieldPtr()` oder `EIB7GetDataFieldPtrRaw()` zugegriffen werden.

Darüber hinaus ist es möglich, über den Callback-Mechanismus eine Funktion zu registrieren, die aufgerufen wird, sobald neue Daten im FIFO bereitstehen (siehe 7.54).

### 3.6 Recording Modus

Zuerst müssen die Achsen mit `EIB7InitAxis()` initialisiert, und das Datenpaket sowie die Trigger-Logik konfiguriert werden. Sobald der Recording Modus aktiv ist, werden mit jedem Triggerereignis die Daten in der EIB 74x gespeichert. Die Triggerquelle kann im Recording Modus ausgewählt werden. Der Status des Pufferspeichers ist auslesbar (siehe 7.41).

Nach dem Ende der Aufzeichnung können die Daten zum Host übertragen werden. Dies erfolgt im Polling Modus. Mit der Funktion `EIB7TransferRecordingData()` wird die Übertragung gestartet. Die Kunden-Softwareapplikation am Host kann die Positionsdaten aus dem Empfangspuffer über die Funktionen `EIB7ReadFIFOData()` oder `EIB7ReadFIFODataRaw()` auslesen (siehe 7.44, 7.47). Jede dieser Funktionen liest einen oder mehrere Einträge aus dem FIFO. Jeder Eintrag enthält ein Datenpaket der EIB 74x. Die Größe eines Eintrags kann im Vorfeld über die Funktionen `EIB7SizeOfFIFOEntry()` und `EIB7SizeOfFIFOEntryRaw()` bestimmt werden (siehe 7.45, 7.48). Auf die einzelnen Komponenten eines FIFO-Eintrags kann mit Hilfe der Funktion `EIB7GetDataFieldPtr()` oder `EIB7GetDataFieldPtrRaw()` zugegriffen werden.

## 4 Datentypen

### 4.1 Einfache Datentypen

EIB7_HANDLE	Handle für eine EIB 74x
EIB7_AXIS	Handle für eine Achse der EIB 74x
EIB7_IO	Handle für einen Eingangs- oder Ausgangsport der EIB 74x
EIB7_ERR	Fehlermeldung
ENCODER_POSITION	Positionswert (64 Bit Integer)

### 4.2 EnDat Zusatzinformation

struct ENDAT\_ADDINFO

Komponente	Beschreibung
status	Statuswort für die Zusatzinformation
info	Daten der Zusatzinformation

### 4.3 Information für TCP-Verbindung

struct EIB7\_CONN\_INFO

Komponente	Beschreibung
id	Identifikationsnummer für die Verbindung
local_ip	lokale IP-Adresse für diese Verbindung
local_port	lokale Port-Nummer für diese Verbindung
remote_ip	IP-Adresse der EIB 74x für diese Verbindung
remote_port	Port-Nummer der EIB 74x für diese Verbindung

### 4.4 Konfiguration für Datenpaket

struct EIB7\_DataPacketSection

Komponente	Beschreibung
region	Globale Information, oder Achse der EIB 74x
items	Datenelemente innerhalb der Region

## 5 Parameter und Rückgabewerte

Alle Funktionen liefern einen Rückgabewert vom Typ EIB7\_ERR. Dieser kennzeichnet einen Funktionsaufruf als erfolgreich oder meldet einen Fehler, der während der Ausführung auftrat.

Eingabewerte für die Funktionen werden als Variable übergeben (transfer by value). Für Rückgabewerte wird ein Pointer auf eine Variable übergeben, in der sich nach einer erfolgreichen Ausführung der Funktion das Ergebnis befindet (transfer by reference).

## 6 Hilfsfunktionen

### 6.1 IP-Adresse bestimmen

Der Hostname der EIB 74x oder die IP-Adresse (als C-String) wird in eine IP-Adresse in „Host Byte Order“ umgewandelt. Der Name muss als C-String übergeben werden. Dieser kann zum Beispiel „192.168.1.2“ oder „EIB74x-SN1234567“ lauten.

#### Funktion

```
EIB7_ERR EIB7GetHostIP      ( const char*      hostname,  
                             unsigned long*    ip  
                             )
```

#### Parameter

hostname	Pointer auf einen C-String, der die IP-Adresse oder den Hostnamen der EIB 74x enthält.
ip	<i>[Rückgabewert]</i> Pointer auf eine Variable in der die IP-Adresse der EIB 74x gespeichert wird

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Mögliche Werte sind nachfolgend aufgelistet.

EIB7_NoError	Funktionsaufruf erfolgreich
EIB7_HostNotFound	IP-Adresse konnte nicht ermittelt werden

### 6.2 Positions-Datenformat ändern

Das Datenformat eines Positionswertes wird von 64-Bit Integer in Double konvertiert. Die Funktion kann nur für inkrementale Messsysteme benutzt werden. Der konvertierte Wert hat die Einheit „1 Signalperiode“. Der Periodenzählerwert entspricht dem Vorkomma-Teil des Ergebnisses, aus dem Interpolationswert werden die Nachkommastellen gebildet.

#### Funktion

```
EIB7_ERR EIB7IncrPosToDouble ( ENCODER_POSITION src,  
                               double*          dest  
                               )
```

#### Parameter

src	Positionswert eines inkrementalen Messsystems
ip	<i>[Rückgabewert]</i> Pointer auf eine Variable in der die konvertierte Position gespeichert wird

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Mögliche Werte sind nachfolgend aufgelistet.

EIB7_NoError	Funktionsaufruf erfolgreich
EIB7_ParamInvalid	übergebener Positionswert ist ungültig



## 7 Gerätefunktionen

Die Gerätefunktionen beziehen sich immer auf die gesamte EIB 74x. Eine Unterscheidung zwischen den Achsen ist nicht möglich. Bei einigen Funktionen werden Parameter aller Achsen beeinflusst.

Alle Gerätefunktionen können als Rückgabewert die nachfolgend aufgeführten Fehlermeldungen liefern. Zusätzlich dazu können sie individuell weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

### Standard Rückgabewerte

EIB7_NoError	Funktionsaufruf erfolgreich
EIB7_InvalidHandle	Das Handle auf die EIB 74x ist ungültig
EIB7_FuncNotSupp	Funktion wird von der EIB 74x nicht unterstützt
EIB7_InvalidResponse	Fehler bei der Datenübertragung
EIB7_AccNotAllowed	Funktion kann nicht ausgeführt werden, da die EIB 74x den Zugriff nicht erlaubt
EIB7_ConnReset	Verbindung wurde von der EIB 74x beendet
EIB7_ConnTimeout	Timeout bei der Datenübertragung zur EIB 74x
EIB7_ReceiveError	Fehler beim Empfangen der Daten
EIB7_SendError	Fehler beim Senden der Daten
EIB7_OutOfMemory	Vom System kann nicht genügend Speicher allokiert werden

### 7.1 Verbindung zur EIB 74x öffnen

Zur EIB 74x wird eine TCP-Verbindung aufgebaut. Dabei werden keine Einstellungen in der EIB 74x verändert. Falls die Verbindung nicht hergestellt werden kann, wird eine Fehlermeldung zurückgegeben. Für eine korrekte Funktion muss der Treiber kompatibel zur Firmware der EIB 74x sein. Dies wird nach dem Verbindungsaufbau geprüft. Gegebenenfalls kann die Firmwareversion der EIB 74x mit dieser Funktion ausgelesen werden. Dazu muss über den Parameter „ident“ die Adresse eines Speicherbereichs übergeben werden, in den die Versionsnummer als C-String geschrieben wird.

### Funktion

```
EIB7_ERR EIB7Open( unsigned long ip,
                   EIB7_HANDLE* eib,
                   long timeout,
                   char* ident,
                   unsigned long len
                 )
```

### Parameter

ip	IP-Adresse in „Host Byte Order“
eib	[Rückgabewert] Handle für die EIB 74x falls die Funktion erfolgreich beendet wurde
timeout	Timeout für folgende Kommandos in Millisekunden (nicht gültig für EIB7Open())
ident	[Rückgabewert] Pointer auf den Zielspeicher, in dem die Firmware Version der EIB 74x als C-String gespeichert wird. Dieser Speicher muss mindestens 9 Bytes groß sein. Ist dieser Parameter ein NULL-Pointer, wird die Firmwareversion der EIB 74x nicht ausgelesen.
len	Größe des Zielspeichers in Bytes (0, wenn ident = NULL)

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_CantInitWinSock	Socketlayer des Betriebssystems kann nicht initialisiert werden (nur für Windows)
EIB7_CantOpenSocket	Systemressourcen für Verbindung nicht verfügbar
EIB7_OutOfMemory	Nicht genügend Speicher vorhanden
EIB7_IFVersionInv	Firmware der EIB 74x ist inkompatibel zum Treiber
EIB7_CantConnect	Verbindung kann nicht hergestellt werden (EIB 74x ist eventuell ausgeschaltet oder nicht erreichbar)

## 7.2 Verbindung zur EIB 74x schließen

Die Verbindung zur EIB 74x wird geschlossen. Das EIB-Handle darf anschließend nicht weiter verwendet werden. Ebenso sind alle Handles auf die Achsen ungültig, welche aus diesem EIB-Handle erzeugt wurden. Falls über dieses Handle ein spezieller Betriebsmodus der EIB 74x aktiviert wurde, wird beim Schließen der Verbindung der Polling Modus aktiviert. Alle weiteren Einstellungen in der EIB 74x bleiben erhalten.

### Funktion

```
EIB7_ERR EIB7Close          ( EIB7_HANDLE      eib
                             )
```

### Parameter

eib                            EIB-Handle

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.3 Status der Verbindung abfragen

Der Status der Verbindung zur EIB 74x wird abgefragt. Dadurch kann bestimmt werden, ob eine Verbindung bereits geschlossen wurde, oder ob ein Kommunikationsfehler auftrat. Diese Funktion sendet keine Daten zur EIB 74x. Der Status bezieht sich auf die vorhergehenden Kommandos.

### Funktion

```
EIB7_ERR EIB7GetConnectionStatus ( EIB7_HANDLE      eib,
                                   EIB7_CONN_STATUS*    status
                                   )
```

### Parameter

eib                            EIB-Handle  
status                        [Rückgabewert] Pointer auf die Zielvariable für den Status

status	Beschreibung
EIB7_CS_Connected	Verbindung zur EIB 74x aufgebaut
EIB7_CS_Closed	keine Verbindung zur EIB 74x
EIB7_CS_Timeout	Zeitüberschreitung bei der Datenübertragung aufgetreten
EIB7_CS_ConnectionReset	Die Verbindung wurde von der EIB 74x geschlossen
EIB7_CS_TransmissionError	Übertragungsfehler aufgetreten

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.4 Timeout einstellen

Der Timeout für die TCP-Verbindung zur EIB 74x wird neu gesetzt. Dieser Wert ist für alle folgenden Funktionsaufrufe gültig. Der Timeout muss mindestens 100 ms sein. Kleinere Werte werden automatisch auf 100 vergrößert.

### Funktion

```
EIB7_ERR EIB7SetTimeout      ( EIB7_HANDLE      eib,
                               long              timeout
                               )
```

### Parameter

eib                            EIB-Handle  
timeout                        Timeout in Millisekunden (>= 100)

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_IllegalParameter        Der Timeout kann nicht eingestellt werden

## 7.5 Anzahl der Achsen auslesen

Die Anzahl der Achsen mit Sub-D Eingang in der EIB 74x wird ausgelesen.

### Funktion

```
EIB7_ERR EIB7GetNumOfAxes ( EIB7_HANDLE    eib,
                           unsigned long*    dsub,
                           unsigned long*    res1,
                           unsigned long*    res2,
                           unsigned long*    res3
                           )
```

### Parameter

eib	EIB-Handle
dsub	<i>[Rückgabewert]</i> Pointer auf Zielvariable für die Anzahl der Achsen mit Sub-D Eingang
res1	<i>[Rückgabewert]</i> reserviert
res2	<i>[Rückgabewert]</i> reserviert
res3	<i>[Rückgabewert]</i> reserviert

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.6 Handle für Achse anfordern

Die Handles für den Zugriff auf die Achsen der EIB 74x werden erzeugt. Sie werden in einem Array gespeichert, dessen Größe als Parameter mit übergeben werden muss. Als Rückgabewert wird die Anzahl der gültigen Handles geliefert. Die Funktion liefert für jede Achse der EIB 74x ein Handle, maximal aber so viele, wie im Array Platz finden (Parameter „size“). Die Handles werden in aufsteigender Reihenfolge, beginnend mit Achse 1, im Array abgelegt.

### Funktion

```
EIB7_ERR EIB7GetAxis ( EIB7_HANDLE    eib,
                      EIB7_AXIS*      set,
                      unsigned long    size,
                      unsigned long*    len
                      )
```

### Parameter

eib	EIB-Handle
set	<i>[Rückgabewert]</i> Pointer auf das erste Element des Handle-Array
size	Maximale Anzahl der Einträge im Array
len	<i>[Rückgabewert]</i> Zahl der gültigen Einträge im Array

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.7 IO-Port-Handle anfordern

Für den Zugriff auf die IO-Ports der EIB 74x werden Handles erzeugt. Die Handles für die Eingänge und die Ausgänge werden in je einem Array gespeichert, dessen Größe als Parameter mit übergeben werden muss. Die Anzahl der gültigen Handles im Array wird in „ilen“, bzw. „olen“ ausgegeben. Die Funktion liefert für jeden IO-Port der EIB 74x ein Handle, maximal aber so viele, wie im Array Platz finden (Parameter „isize“, „osize“).

### Funktion

```
EIB7_ERR EIB7GetIO ( EIB7_HANDLE    eib,
                    EIB7_IO*         iset,
                    unsigned long     isize,
                    unsigned long*    ilen,
                    EIB7_IO*         oset,
                    unsigned long     osize,
                    unsigned long*    olen
                    )
```

### Parameter

eib	EIB-Handle
iset	<i>[Rückgabewert]</i> Pointer auf das erste Element des Arrays mit den Input-Handles
isize	Maximale Anzahl der Einträge im Array „iset“
ilen	<i>[Rückgabewert]</i> Zahl der gültigen Einträge im Array „iset“
oset	<i>[Rückgabewert]</i> Pointer auf das erste Element des Arrays mit den Output-Handles
osize	Maximale Anzahl der Einträge im Array „oset“
olen	<i>[Rückgabewert]</i> Zahl der gültigen Einträge im Array „oset“

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.8 Datenpaket erstellen

Mit Hilfe dieser Funktion kann die Konfiguration für das Datenpaket erstellt werden. Pro Funktionsaufruf wird ein Element aus dem Array für die Konfigurationsdaten initialisiert. Der Index gibt das Element an, wobei das erste Element den Index 0 besitzt. Jedes Element besteht aus einer Region und den Datenelementen. Die Region spezifiziert die Achse oder die globalen Informationen. Für jede Region können verschiedene Datenelemente hinzugefügt werden. Alle Datenelemente für eine Region müssen ODER-Verknüpft und als Parameter „items“ übergeben werden. Die Datenelemente sind als EIB7\_DataPacketItem spezifiziert.

### Funktion

```
EIB7_ERR EIB7AddDataPacketSection ( EIB_DataPacketSection* packet,
                                     unsigned long          index,
                                     EIB7_DataRegion        region,
                                     unsigned long          items
                                   )
```

### Parameter

packet      Pointer auf das Array für die Konfigurationsdaten  
index      Index des Array-Elements  
region      Region des Datenpakets

region	Beschreibung
EIB7_DR_Global	Globale Informationen
EIB7_DR_Encoder1	Daten für Achse 1
EIB7_DR_Encoder2	Daten für Achse 2
EIB7_DR_Encoder3	Daten für Achse 3
EIB7_DR_Encoder4	Daten für Achse 4
EIB7_DR_AUX	Daten für Hilfsachse

items      Datenelemente innerhalb der Region (ODER-Verknüpfung mehrere Elemente möglich)

items	Beschreibung
EIB7_PDF_TriggerCounter	Trigger Counter (nur in EIB7_DR_Global)
EIB7_PDF_StatusWord	Statuswort für Position
EIB7_PDF_PositionData	Positionswert
EIB7_PDF_Timestamp	Zeitstempel für Position
EIB7_PDF_Analog	ADC-Werte für Signal A und B
EIB7_PDF_ReferencePos	Referenzposition 1 und Referenzposition 2
EIB7_PDF_DistCodedRef	Codierter Referenzwert
EIB7_PDF_EnDat_AI1	EnDat 2.2 Zusatzinformation 1
EIB7_PDF_EnDat_AI2	EnDat 2.2 Zusatzinformation 2

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid      Parameter ungültig

## 7.9 Datenpaket konfigurieren

Das Datenpaket für die Modi „Soft Realtime“, „Streaming“ und „Recording“ kann konfiguriert werden. Die Konfiguration ist nur im Modus „Polling“ möglich. Die Konfiguration wird übernommen, sobald ein anderer Betriebsmodus (außer „Polling“) aktiviert wird.

### Funktion

```
EIB7_ERR EIB7ConfigDataPacket ( EIB7_HANDLE          eib,
                                 EIB_DataPacketSection* packet,
                                 unsigned long          size
                               )
```

### Parameter

eib      EIB-Handle  
packet      Pointer auf ein Array mit den Konfigurationsdaten für das Datenpaket  
size      Anzahl der Einträge im Array „packet“

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid      Parameter ungültig  
EIB7\_PacketTooLong      Konfigurationsdaten beschreiben ein zu langes Datenpaket  
EIB7\_InvalidPacket      Konfigurationsdaten beschreiben ein ungültiges Datenpaket

## 7.10 Betriebsmodus wählen

Der Betriebsmodus der EIB 74x kann eingestellt werden. Es werden die Modi „Polling“, „Soft Realtime“, „Streaming“ und „Recording“ unterstützt. Im Betriebsmodus „Recording“ kann noch zwischen „Single Shot“ und „Rolling“ Betrieb unterschieden werden.

### Funktion

```
EIB7_ERR EIB7SelectMode ( EIB7_HANDLE eib,
                          EIB7_OPERATING_MODE mode
                        )
```

### Parameter

eib                    EIB-Handle  
mode                  Betriebsmodus

mode	Betriebsmodus
EIB7_OM_Polling	Polling Modus
EIB7_OM_SoftRealtime	Soft Realtime Modus
EIB7_OM_Streaming	Streaming Modus
EIB7_OM_RecordingSingle	Recording Modus Single Shot
EIB7_OM_RecordingRoll	Recording Modus Rolling

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantOpenSocket      interner Fehler (Socket-Fehler)  
EIB7\_CantStartThread    interner Fehler (Thread-Fehler)  
EIB7\_InvalidOpMode      der gewählte Betriebsmodus wird nicht unterstützt  
EIB7\_OpModeActive       der gewählte Betriebsmodus ist bereits aktiv  
EIB7\_OpModeBlocked      der gewählte Betriebsmodus kann nicht aktiviert werden  
EIB7\_InvalidIPAddr      interner Fehler (IP-Adress Fehler)

## 7.11 Netzwerkparameter speichern

Die Parameter für die Ethernet Schnittstelle der EIB 74x können eingestellt werden. Dadurch ist die EIB 74x an das Netzwerk anpassbar. Die Einstellungen werden erst nach dem nächsten Bootvorgang wirksam. Falls der DHCP-Client aktiv ist, versucht die EIB 74x vom DHCP-Server eine IP-Adresse zu erhalten. Antwortet der Server innerhalb des eingestellten Timeouts nicht, wird die konfigurierte IP-Adresse verwendet.

### Funktion

```
EIB7_ERR EIB7SetNetwork ( EIB7_HANDLE eib,
                          unsigned long ip,
                          unsigned long netmask,
                          unsigned long gateway,
                          EIB7_MODE dhcp,
                          unsigned long timeout
                        )
```

### Parameter

eib                    EIB-Handle  
ip                    IP-Adresse der EIB 74x in „Host Byte Order“  
netmask               Netzwerkmaske für das Netzwerk in „Host Byte Order“  
gateway               IP-Adresse des Standard Gateway in „Host Byte Order“  
dhcp                  Flag für den DHCP-Client in der EIB 74x

dhcp	Beschreibung
EIB7_MD_Disable	DHCP-Client deaktivieren
EIB7_MD_Enable	DHCP-Client aktivieren

timeout               Timeout für den DHCP-Client in Sekunden

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantSaveCustNW      Netzwerkeinstellungen können nicht gespeichert werden  
EIB7\_CantSaveDHCP       DHCP Timeout kann nicht gespeichert werden  
EIB7\_DHCPTimeoutInv      DHCP Timeout ungültig  
EIB7\_ParamInvalid       Parameter sind keine gültige Netzwerkkonfiguration

## 7.12 Netzwerkparameter auslesen

Die Parameter für die Ethernet Schnittstelle können ausgelesen werden. Es werden immer die benutzerdefinierten Einstellungen ausgegeben, auch wenn mit den Standard Einstellungen gebootet wurde.

### Funktion

```
EIB7_ERR EIB7GetNetwork      ( EIB7_HANDLE      eib,
                               unsigned long*    ip,
                               unsigned long*    netmask,
                               unsigned long*    gateway,
                               EIB7_MODE*        dhcp
                               )
```

### Parameter

eib	EIB-Handle
ip	[Rückgabewert] Pointer auf die Variable für die IP-Adresse in „Host Byte Order“
netmask	[Rückgabewert] Pointer auf die Variable für die Netzwerkmaske in „Host Byte Order“
gateway	[Rückgabewert] Pointer auf die Variable für die IP-Adresse des Standard Gateway in „Host Byte Order“
dhcp	[Rückgabewert] Pointer auf die Variable für das Flag für den DHCP-Client

dhcp	Beschreibung
EIB7_MD_Disable	DHCP-Client inaktiv
EIB7_MD_Enable	DHCP-Client aktiv

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_NoCustNetwork	Keine Kundenspezifischen Einstellungen vorhanden
--------------------	--

## 7.13 Hostnamen speichern

Der Hostname der EIB 74x wird gespeichert. Der Name muss als C-String übergeben werden, der einschließlich des Null-Bytes maximal 32 Zeichen lang sein darf. Falls er länger ist, wird der Rest abgeschnitten. Wird ein String mit der Länge Null oder ein NULL-Pointer übergeben, setzt die EIB 74x den Hostnamen auf den Standardwert bei der Auslieferung.

### Funktion

```
EIB7_ERR EIB7SetHostname    ( EIB7_HANDLE      eib,
                               const char*      hostname
                               )
```

### Parameter

eib	EIB-Handle
hostname	Pointer auf den neuen Hostnamen

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_HostnameTooLong	Hostname ist zu lang
EIB7_HostnameInvalid	Hostname ist ungültig
EIB7_CantSaveHostn	Hostname kann nicht gespeichert werden
EIB7_CantRestDefHn	Standard Hostname kann nicht geladen werden

### 7.14 Hostnamen auslesen

Der Hostname der EIB 74x wird ausgelesen und als C-String im Zielspeicher abgelegt. Der String ist maximal 32 Zeichen lang (incl. Null-Byte). Falls der Zielspeicher nicht groß genug ist, um den ganzen String aufzunehmen, wird nur der erste Teil kopiert.

#### Funktion

```
EIB7_ERR EIB7GetHostname      ( EIB7_HANDLE      eib,  
                                char*              hostname,  
                                unsigned long       len  
                                )
```

#### Parameter

eib	EIB-Handle
hostname	<i>[Rückgabewert]</i> Pointer auf den Zielspeicher für den Hostnamen
len	Größe des Zielspeichers in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantRdHostname      Hostname kann nicht gelesen werden

### 7.15 Seriennummer auslesen

Die Seriennummer der EIB 74x wird als C-String ausgegeben. Der String wird in den Zielspeicher geschrieben. Falls der Zielstring nicht genügend Platz für die Seriennummer bereitstellt, wird ein Fehler ausgegeben. Die Seriennummer kann maximal 24 Zeichen lang sein (incl. Null-Byte).

#### Funktion

```
EIB7_ERR EIB7GetSerialNumber  ( EIB7_HANDLE      eib,  
                                char*              serial,  
                                unsigned long       len  
                                )
```

#### Parameter

eib	EIB-Handle
serial	<i>[Rückgabewert]</i> Pointer auf den Zielspeicher für die Seriennummer
len	Größe des Zielspeichers in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantRdSer            Seriennummer kann nicht gelesen werden  
EIB7\_BufferTooSmall      Zielspeicher ist zu klein

### 7.16 Geräte-Identnummer auslesen

Die Geräte-Identnummer der EIB 74x wird als C-String ausgegeben. Der String wird in den Zielspeicher geschrieben. Falls der Zielstring nicht genügend Platz für die Nummer bereitstellt, wird ein Fehler ausgegeben. Die Nummer kann maximal 16 Zeichen lang sein (incl. Null-Byte).

#### Funktion

```
EIB7_ERR EIB7GetIdentNumber   ( EIB7_HANDLE      eib,  
                                char*              ident,  
                                unsigned long       len  
                                )
```

#### Parameter

eib	EIB-Handle
ident	<i>[Rückgabewert]</i> Pointer auf den Zielspeicher für die Gerätenummer
len	Größe des Zielspeichers in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantRdIdent          Gerätenummer kann nicht gelesen werden  
EIB7\_BufferTooSmall      Zielspeicher ist zu klein

### 7.17 MAC-Adresse auslesen

Die MAC-Adresse der EIB 74x wird ausgegeben. Die Adresse wird im Binärformat ausgegeben. Der Zielspeicher muss mindestens 6 Bytes groß sein. Es werden immer die ersten sechs Bytes verwendet. Das niederwertigste Byte der MAC-Adresse wird in das erste Byte des Zielspeichers kopiert. Zum Beispiel für „00:A0:CD:85:00:01“.

Offset	Speicherinhalt
0	0x01
1	0x00
2	0x85
3	0xCD
4	0xA0
5	0x00

#### Funktion

```
EIB7_ERR EIB7GetMAC          ( EIB7_HANDLE      eib,  
                               unsigned char*    mac  
                               )
```

#### Parameter

eib	EIB-Handle
mac	[Rückgabewert] Pointer auf den Zielspeicher für die MAC-Adresse

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.18 Firmware Versionsnummer auslesen

Die Versionsnummer der EIB 74x-Firmware wird ausgelesen. Der Parameter „select“ bestimmt, von welcher Firmware die Versionsnummer als C-String ausgegeben wird. Für den String inklusive dem Null-Byte sollte der Zielspeicher mindestens 9 Bytes groß sein. Ist der Zielspeicher zu klein, um den ganzen String aufzunehmen, wird nur der erste Teil kopiert.

#### Funktion

```
EIB7_ERR EIB7GetVersion      ( EIB7_HANDLE      eib,  
                               char*            ident,  
                               unsigned long     len,  
                               EIB7_FIRMWARE    select  
                               )
```

#### Parameter

eib	EIB-Handle
ident	[Rückgabewert] Pointer auf den Zielspeicher für die Versionsnummer der Firmware
len	Größe des Zielspeichers in Bytes
select	Auswahl der Firmware, deren Versionsnummer ausgelesen wird

select	Beschreibung
EIB7_FW_CurrentlyBooted	Aktuell geladene Firmware
EIB7_FW_Factory	Firmware des Auslieferungszustandes
EIB7_FW_User	Firmware des letzten Updates

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.



## 7.19 Bootmodus auslesen

Der Bootmodus, in dem die EIB 74x beim letzten Bootvorgang gestartet wurde, wird ausgelesen.

### Funktion

```
EIB7_ERR EIB7GetBootMode      ( EIB7_HANDLE      eib,  
                                EIB7_BOOT_MODE*    mode  
                                )
```

### Parameter

eib                                      EIB-Handle  
mode                                    [Rückgabewert] Pointer auf die Variable für den Bootmodus

mode	Beschreibung
EIB7_BM_User	Firmware des letzten Updates mit Netzwerk-Benutzereinstellungen
EIB7_BM_FactoryUser	Firmware des Auslieferungszustands (Werkseinstellung) mit Netzwerk-Benutzereinstellungen
EIB7_BM_FactoryDefault	Firmware des Auslieferungszustands (Werkseinstellung) mit Standard-Netzwerkeinstellungen

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.20 Updatestatus auslesen

Um zu prüfen, ob ein Update erfolgreich durchgeführt wurde, kann der Status ausgelesen werden. Der Funktionsaufruf setzt den Status nach dem Lesevorgang automatisch zurück in den Grundzustand (EIB7\_US\_NoUpdate). Bei jedem Bootvorgang wird die Statusinformation gelöscht.

### Funktion

```
EIB7_ERR EIB7UpdateState      ( EIB7_HANDLE      eib,  
                                EIB7_UPDATE_STATUS* status  
                                )
```

### Parameter

eib                                      EIB-Handle  
status                                  [Rückgabewert] Pointer auf die Variable für den Updatestatus

status	Beschreibung
EIB7_US_NoUpdate	Es wurde kein Update eingespielt
EIB7_US_UpdateFailed	Update konnte nicht durchgeführt werden
EIB7_US_UpdateSuccessful	Update erfolgreich durchgeführt
EIB7_US_VersionIncompatible	Firmware ist nicht kompatibel mit der Hardware der EIB 74x

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.21 Anzahl der offenen Verbindungen lesen

Die Anzahl der aktuell geöffneten Verbindungen zur EIB 74x wird ausgegeben. Hierzu zählen auch halboffene Verbindungen, welche die Gegenstelle bereits geschlossen hat, aber auf der EIB 74x noch geöffnet sind.

### Funktion

```
EIB7_ERR EIB7GetNumberOfOpenConnections      ( EIB7_HANDLE      eib,  
                                                unsigned long*    cnt  
                                                )
```

### Parameter

eib                                      EIB-Handle  
cnt                                      [Rückgabewert] Pointer auf die Variable für die Anzahl der offenen Verbindungen

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.22 Verbindungsdaten auslesen

Die Verbindungsdaten aller derzeit geöffneten Verbindungen zur EIB 74x können ausgelesen werden. Für jede Verbindung wird ein Eintrag im Array belegt. Maximal aber nur so viele, wie durch den Parameter „size“ vorgegeben. Die Zahl der gültigen Elemente im Array, wird über den Parameter „cnt“ zurückgegeben. Der Inhalt der Verbindungsdaten ist im Kapitel „Datentypen“ aufgeführt.

### Funktion

```
EIB7_ERR EIB7ConnectionInfo ( EIB7_HANDLE    eib,
                              EIB7_CONN_INFO*  info,
                              unsigned long    size,
                              unsigned long*    cnt
                              )
```

### Parameter

eib	EIB-Handle
info	<i>[Rückgabewert]</i> Pointer auf das erste Element im Array für die Verbindungsdaten
size	Größe des Array „info“
cnt	<i>[Rückgabewert]</i> Pointer auf die Variable für die Anzahl der gültigen Elemente im Array

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.23 Verbindung abbrechen

Eine offene Verbindung zur EIB 74x kann beendet werden. Es ist nicht möglich die Verbindung zu schließen, welche für den Aufruf der Funktion verwendet wird. Diese Funktion sollte hauptsächlich benutzt werden, um halboffene Verbindungen zu schließen, die zum Beispiel aufgrund eines Fehlers am Host nicht ordnungsgemäß beendet wurden. Die ID kann aus den Verbindungsdaten EIB7\_CONN\_INFO entnommen werden (siehe „Verbindungsdaten auslesen“).

### Funktion

```
EIB7_ERR EIB7TerminateConnection ( EIB7_HANDLE    eib,
                                   unsigned long    id
                                   )
```

### Parameter

eib	EIB-Handle
id	ID der Verbindung, die abgebrochen wird

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_CantTermConn	Die Verbindung kann nicht beendet werden
EIB7_CantTermSelf	Die Verbindung kann sich nicht selbst beenden
EIB7_ParamInvalid	Der Parameter ist kein gültiger Index für eine Verbindung

## 7.24 Timestamp Zeiteinheit lesen

Der Timestamp Zähler wird von einer Taktquelle gespeist. Die Timestamp Ticks geben an, wie viele Takte pro Mikrosekunde von der Taktquelle ausgegeben werden.

### Funktion

```
EIB7_ERR EIB7GetTimestampTicks ( EIB7_HANDLE    eib,
                                 unsigned long*    ticks
                                 )
```

### Parameter

eib	EIB-Handle
ticks	<i>[Rückgabewert]</i> Pointer auf die Variable für die Takte pro Mikrosekunde

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.25 Timestamp Periodendauer einstellen

Die Periodendauer des frei laufenden Timestamp Zählers kann eingestellt werden. Dazu muss die Länge der Timestamp Periode in Timestamp Ticks angegeben werden. Dieser Wert muss eine natürliche Zahl größer Null sein.

#### Funktion

```
EIB7_ERR EIB7SetTimestampPeriod      ( EIB7_HANDLE      eib,  
                                       unsigned long      period  
                                       )
```

#### Parameter

eib	EIB-Handle
period	Ticks pro Timestamp Periode (>0); siehe auch Hinweis im Kapitel 4

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Timestamp Periode ungültig
-------------------	----------------------------

### 7.26 Timestamp Zähler zurücksetzen

Der Timestamp Zähler wird auf Null gesetzt und zählt von diesem Wert weiter.

#### Funktion

```
EIB7_ERR EIB7ResetTimestamp          ( EIB7_HANDLE      eib  
                                       )
```

#### Parameter

eib	EIB-Handle
-----	------------

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.27 Timer Trigger Zeiteinheit lesen

Der Timer Trigger wird von einer Taktquelle gespeist. Die Timer Trigger Ticks geben an, wie viele Takte pro Mikrosekunde von der Taktquelle ausgegeben werden.

#### Funktion

```
EIB7_ERR EIB7GetTimerTriggerTicks    ( EIB7_HANDLE      eib,  
                                       unsigned long*      ticks  
                                       )
```

#### Parameter

eib	EIB-Handle
ticks	[Rückgabewert] Pointer auf die Variable für die Takte pro Mikrosekunde

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.28 Timer Trigger Periodendauer einstellen

Die Periodendauer des Timer Triggers kann eingestellt werden. Dazu muss angegeben werden, wie viele Timer Trigger Ticks eine Periode lang ist. Dieser Wert muss eine natürliche Zahl größer Null sein. Falls der Timer Trigger aktiviert ist, löst er nach jeder Periode ein Triggerereignis aus.

#### Funktion

```
EIB7_ERR EIB7SetTimerTriggerPeriod   ( EIB7_HANDLE      eib,  
                                       unsigned long      period  
                                       )
```

#### Parameter

eib	EIB-Handle
period	Ticks pro Timer Trigger Periode (>0); siehe auch Hinweis im Kapitel 4

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Trigger Periode ungültig
-------------------	--------------------------

### 7.29 Zeiteinheit für die Verzögerung an den Triggereingängen lesen

Die Trigger Input Delay Ticks geben an, wie viele Takte pro Mikrosekunde von der Taktquelle für die Verzögerung der Signale am Triggereingang ausgegeben werden. Die Verzögerung der Triggersignale kann als Vielfaches der internen Taktperiode eingestellt werden.

#### Funktion

```
EIB7_ERR EIB7GetTriggerDelayTicks ( EIB7_HANDLE eib,
                                   unsigned long* ticks
                                   )
```

#### Parameter

eib                                      EIB-Handle  
ticks                                    *[Rückgabewert]* Pointer auf Zielvariable für die Takte pro Mikrosekunde

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.30 Trigger Counter löschen

Der Trigger Counter wird auf Null gesetzt.

#### Funktion

```
EIB7_ERR EIB7ResetTriggerCounter ( EIB7_HANDLE eib
                                   )
```

#### Parameter

eib                                      EIB-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.31 Software Trigger

Der Software Trigger erzeugt ein Triggerereignis und veranlasst die EIB 74x die Daten an die Gegenstelle zu senden. Über den Parameter „source“ lässt sich einer der Software-Trigger-Kanäle auswählen. Diese Funktion kann nicht im „Polling“ Modus ausgeführt werden.

#### Funktion

```
EIB7_ERR EIB7SoftwareTrigger ( EIB7_HANDLE eib,
                               unsigned long source
                               )
```

#### Parameter

eib                                      EIB-Handle  
source                                  Software-Trigger-Kanal

source	Beschreibung
EIB7_ST_SWtrigger1	Software-Trigger Kanal 1
EIB7_ST_SWtrigger2	Software-Trigger Kanal 2
EIB7_ST_SWtrigger3	Software-Trigger Kanal 3
EIB7_ST_SWtrigger4	Software-Trigger Kanal 4

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten kann die nachfolgend aufgelistete Fehlermeldung auftreten.

EIB7\_ParamInvalid                      Parameter ungültig

### 7.32 Master-Triggerquelle wählen

Das Master-Triggersignal kann aus verschiedenen Quellen gewählt werden. Diese Funktion muss nach der Konfiguration der Triggermatrix für die Achsen ausgeführt werden und ist nur im Betriebsmodus „Polling“ zulässig.

#### Funktion

```
EIB7_ERR EIB7MasterTriggerSource ( EIB7_HANDLE eib,  
                                   EIB7_AxisTriggerSrc src  
                                   )
```

#### Parameter

eib                    EIB-Handle  
src                   Triggerquelle

src	Beschreibung
EIB7_AT_TrgInput1	Triggereingang Kanal 1
EIB7_AT_TrgInput2	Triggereingang Kanal 2
EIB7_AT_TrgInput3	Triggereingang Kanal 3
EIB7_AT_TrgInput4	Triggereingang Kanal 4
EIB7_AT_TrgSW1	Software-Trigger Kanal 1
EIB7_AT_TrgSW2	Software-Trigger Kanal 2
EIB7_AT_TrgSW3	Software-Trigger Kanal 3
EIB7_AT_TrgSW4	Software-Trigger Kanal 4
EIB7_AT_TrgRI	Referenzimpuls der entsprechenden Achse
EIB7_AT_TrgRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_AT_TrgIC	Interval Counter
EIB7_AT_TrgPuls	Trigger Puls Zähler
EIB7_AT_TrgTimer	Timer Trigger

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid            Parameter ungültig

### 7.33 Triggerquellen aktivieren

Die Triggerquellen der EIB 74x lassen sich einzeln oder gemeinsam aktivieren bzw. deaktivieren. Über den Parameter „src“ können mehrere Triggerquellen durch eine ODER-Verknüpfung der entsprechenden Konstanten ausgewählt werden. Für den Timer Trigger sollte vor der Aktivierung die Periodendauer konfiguriert werden. Wenn mehrere Triggerquellen mit einem Funktionsaufruf aktiviert werden, so erfolgt dies zeitgleich.

#### Funktion

```
EIB7_ERR EIB7GlobalTriggerEnable ( EIB7_HANDLE eib,  
                                   EIB7_MODE mode,  
                                   unsigned long src  
                                   )
```

#### Parameter

eib                    EIB-Handle  
mode                  Triggerquellen aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Enable	Triggerquelle aktivieren
EIB7_MD_Disable	Triggerquelle deaktivieren

src                    Triggerquelle

src	Beschreibung
EIB7_TS_TrgInput1	Triggereingang Kanal 1
EIB7_TS_TrgInput2	Triggereingang Kanal 2
EIB7_TS_TrgInput3	Triggereingang Kanal 3
EIB7_TS_TrgInput4	Triggereingang Kanal 4
EIB7_TS_TrgRI1	Referenzimpuls Achse 1
EIB7_TS_TrgRI2	Referenzimpuls Achse 2
EIB7_TS_TrgRI3	Referenzimpuls Achse 3
EIB7_TS_TrgRI4	Referenzimpuls Achse 4
EIB7_TS_TrgRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_TS_TrgIC	Interval Counter
EIB7_TS_TrgPuls	Trigger Pulszähler
EIB7_TS_TrgTimer	Timer Trigger
EIB7_TS_All	Alle Triggerquellen

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid            Parameter ungültig

### 7.34 Pulszähler konfigurieren

Für den Pulszähler kann ein Triggersignal und ein Startsignal gewählt werden. Das Startsignal gibt den Pulszähler frei. Dieser wird fortan mit jedem Impuls am Triggersignal dekrementiert, bis der Wert Null erreicht ist. Anschließend werden alle weiteren Triggerimpulse gesperrt. Wird die Funktion erneut ausgeführt, bevor der Zähler den Wert Null erreicht hat, wird dieser auf den Ausgangswert zurückgesetzt.

#### Funktion

```
EIB7_ERR EIB7ConfigPulsCounter ( EIB7_HANDLE eib,  
                                EIB7_PulsCounterStart start,  
                                EIB7_PulsCounterTrigger trigger,  
                                unsigned long count  
                                )
```

#### Parameter

eib EIB-Handle  
start Startsignal für den Pulszähler

start	Beschreibung
EIB7_PS_TrgInput1	Triggereingang Kanal 1
EIB7_PS_TrgInput2	Triggereingang Kanal 2
EIB7_PS_TrgInput3	Triggereingang Kanal 3
EIB7_PS_TrgInput4	Triggereingang Kanal 4
EIB7_PS_TrgRI1	Referenzimpuls Achse 1
EIB7_PS_TrgRI2	Referenzimpuls Achse 2
EIB7_PS_TrgRI3	Referenzimpuls Achse 3
EIB7_PS_TrgRI4	Referenzimpuls Achse 4
EIB7_PS_TrgSW1	Software-Trigger Kanal 1
EIB7_PS_TrgSW2	Software-Trigger Kanal 2
EIB7_PS_TrgSW3	Software-Trigger Kanal 3
EIB7_PS_TrgSW4	Software-Trigger Kanal 4
EIB7_PS_TrgRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_PS_TrgIC	Interval Counter

trigger Triggersignal für den Pulszähler

trigger	Beschreibung
EIB7_PT_TrgInput1	Triggereingang Kanal 1
EIB7_PT_TrgInput2	Triggereingang Kanal 2
EIB7_PT_TrgInput3	Triggereingang Kanal 3
EIB7_PT_TrgInput4	Triggereingang Kanal 4
EIB7_PT_TrgRI1	Referenzimpuls Achse 1
EIB7_PT_TrgRI2	Referenzimpuls Achse 2
EIB7_PT_TrgRI3	Referenzimpuls Achse 3
EIB7_PT_TrgRI4	Referenzimpuls Achse 4
EIB7_PT_TrgRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_PT_TrgIC	Interval Counter
EIB7_PT_TrgTimer	Timer Trigger

count Startwert für den Pulszähler (0x00000 ... 0xFFFFF)

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid Parameter ungültig

### 7.35 Interpolationsfaktor für den Interval Counter einstellen

Der Interpolationsfaktor für den Interval Counter ist einstellbar und bestimmt die Anzahl der Zählschritte pro Signalperiode. Diese Einstellung wirkt sich gleichermaßen auf den Interval Counter und die Hilfsachse aus. Die Anzahl der Zählschritte pro Signalperiode des angeschlossenen Messgeräts ergibt sich aus dem Interpolationsfaktor multipliziert mit der Flankenauswertung.

#### Funktion

```
EIB7_ERR EIB7SetIntervalCounterInterpolation ( EIB7_HANDLE      eib,  
                                              EIB7_IntervalCounterIPF ipf,  
                                              EIB7_IntervalCounterEdge edge  
                                              )
```

#### Parameter

eib                    EIB-Handle  
ipf                    Interpolationsfaktor

ipf	Beschreibung
EIB7_ICF_1x	Interpolation 1x
EIB7_ICF_2x	Interpolation 2x
EIB7_ICF_4x	Interpolation 4x
EIB7_ICF_5x	Interpolation 5x
EIB7_ICF_10x	Interpolation 10x
EIB7_ICF_20x	Interpolation 20x
EIB7_ICF_25x	Interpolation 25x
EIB7_ICF_50x	Interpolation 50x
EIB7_ICF_100x	Interpolation 100x

edge                    Flankenauswertung

ipf	Beschreibung
EIB7_ICE_1x	Flankenauswertung 1x
EIB7_ICE_2x	Flankenauswertung 2x
EIB7_ICE_4x	Flankenauswertung 4x

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid            Parameter ungültig



### 7.36 Interval Counter konfigurieren

Die Triggerung durch den Interval Counter kann mit dieser Funktion konfiguriert werden. Der Interval Counter bietet zwei Modi. Im ersten Modus wird nur an einer festen Position ein Triggerimpuls ausgegeben. Diese Position ist einstellbar. Der zweite Modus ermöglicht eine Triggerung in festen Abständen. Der erste Triggerimpuls wird an der Startposition ausgegeben. Anschließend wird in festen, über den Parameter „interval“ einstellbaren Abständen ein Triggerimpuls erzeugt. Alternativ lässt sich auch die aktuelle Position als Startposition verwenden.

#### Funktion

```
EIB7_ERR EIB7SetIntervalCounterTrigger ( EIB7_HANDLE          eib,  
                                          EIB7_IntervalCounterMode mode,  
                                          EIB7_IntervalCounterStart start,  
                                          unsigned long          startpos,  
                                          unsigned long          interval  
                                          )
```

#### Parameter

eib                      EIB-Handle  
mode                    Triggermodus

mode	Beschreibung
EIB7_ICM_Disable	Keine Triggerung
EIB7_ICM_Single	Triggerung nur an einer festen Position
EIB7_ICM_Periodic	Periodische Triggerung in festen Abständen

start                    Start der Triggerung

start	Beschreibung
EIB7_ICS_Current	Triggerung startet bei aktueller Position
EIB7_ICS_StartPos	Triggerung startet bei Startposition

startpos                Positionswert für den ersten Triggerimpuls  
interval                Abstand zwischen zwei Triggerimpulsen in Zählschritten (≥ 6)

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid      Parameter ungültig

### 7.37 Abschlusswiderstände einstellen

Die Abschlusswiderstände für die Inkrementalsignale der Messgeräteeingänge können deaktiviert werden. Diese Einstellung gilt immer für alle 1V<sub>SS</sub> Eingänge der EIB 74x. Nach jedem Bootvorgang der EIB 74x sind die Widerstände aktiviert.

#### Funktion

```
EIB7_ERR EIB7EnableIncrementalTermination ( EIB7_HANDLE          eib,  
                                              EIB7_MODE              mode  
                                              )
```

#### Parameter

eib                      EIB-Handle  
mode                    Abschlusswiderstände aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Disable	Abschlusswiderstände deaktivieren
EIB7_MD_Enable	Abschlusswiderstände aktivieren

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantChInclnTrm    Modus kann nicht verändert werden

### 7.38 Reset

Die EIB 74x führt einen Reset durch und bootet neu. Die Funktion hat dieselbe Wirkung wie das Betätigen des Reset-Tasters. Es wird der Standard Bootmodus verwendet (Firmware des letzten Updates mit Netzwerk-Benutzereinstellungen). Die Verbindung zur EIB 74x wird automatisch geschlossen (wie bei EIB7Close).

#### Funktion

```
EIB7_ERR EIB7Reset          ( EIB7_HANDLE      eib
                               )
```

#### Parameter

eib                            EIB-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.39 EIB 74x identifizieren

Die LAN LED auf der Frontplatte der EIB 74x kann in einen Blink-Modus versetzt werden. Wenn mehrere Geräte nebeneinander stehen, ist so eine EIB 74x mit einer bestimmten IP-Adresse leicht auffindbar. Die LED blinkt, bis mit Hilfe der Funktion der Modus beendet wird.

#### Funktion

```
EIB7_ERR EIB7Identify      ( EIB7_HANDLE      eib,
                               EIB7_MODE       mode
                               )
```

#### Parameter

eib                            EIB-Handle  
mode                           Blinken der LED aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Disable	Blink-Modus deaktivieren
EIB7_MD_Enable	Blink-Modus aktivieren

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_IllegalParameter        LED-Status kann nicht verändert werden (Parameter ist ungültig)

## 7.40 Recording-Daten übertragen

Die Übertragung der Daten aus dem internen Recording-Puffer der EIB 74x kann aktiviert oder deaktiviert werden. Beim aktivieren der Datenübertragung ist es möglich nur einen Bereich der aufgezeichneten Daten für die Übertragung auszuwählen. Über den Offset wird das erste zu übertragende Byte angegeben und die Länge spezifiziert die Anzahl der Bytes.

Um alle Daten zu übertragen sollte die Parameter offset = 0 und length = 0xFFFFFFFF gesetzt werden.

### Funktion

```
EIB7_ERR EIB7TransferRecordingData ( EIB7_HANDLE eib,  
                                     EIB7_MODE mode,  
                                     unsigned long offset,  
                                     unsigned long length  
                                     )
```

### Parameter

eib                      EIB-Handle  
mode                    aktivieren oder deaktivieren der Datenübertragung

mode	Beschreibung
EIB7_MD_Disable	Datenübertragung stoppen
EIB7_MD_Enable	Datenübertragung starten

offset                      Offset für das erste Byte, das übertragen wird  
length                    Anzahl der Bytes, die übertragen werden (0xFFFFFFFF = bis Ende der Aufzeichnung)

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantOpenSocket      interner Fehler (Socket-Fehler)  
EIB7\_CantStartThread     interner Fehler (Thread-Fehler)  
EIB7\_OpModeBlocked      die EIB 74x befindet sich nicht im Modus „Polling“  
EIB7\_RecDataReadErr     Daten können nicht aus der EIB 74x gelesen werden  
EIB7\_ParamInvalid        Parameter ungültig

## 7.41 Recording Status prüfen

Der Status im „Recording“ Modus kann ausgelesen werden. Ferner kann der aktuelle Inhalt im Pufferspeicher bestimmt werden. Dies ist auch während der Aufzeichnung im „Recording“ Modus möglich. Der Fortschritt der Datenübertragung aus dem Puffer der EIB 74x zum Host kann ebenfalls ausgelesen werden.

### Funktion

```
EIB7_ERR EIB7GetRecordingStatus ( EIB7_HANDLE eib,  
                                   unsigned long* length,  
                                   unsigned long* status,  
                                   unsigned long* progress  
                                   )
```

### Parameter

eib                      EIB-Handle  
length                    [Rückgabewert] Pointer auf die Zielvariable für die Anzahl der Datenpakete im Puffer  
status                    [Rückgabewert] Pointer auf die Zielvariable für den Status

status	Beschreibung
0	„Recording“ Modus deaktiviert
1	„Recording“ Modus aktiviert
2	Daten werden übertragen
3	Warten auf Verbindungsaufbau für Datenübertragung

progress                      [Rückgabewert] Pointer auf die Zielvariable für den Fortschritt der Datenübertragung  
Fortschritt in Prozent (0..100)

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.42 Recording Speichergröße lesen

Die Größe des Speichers für die Recording-Daten in der EIB 74x kann ausgelesen werden. Die Größe wird als Anzahl der Datenpakete geliefert, die im Speicher Platz finden. Diese Anzahl ist abhängig von der Größe eines Datenpaketes. Aus diesem Grund muss zuerst das Datenpaket konfiguriert werden.

### Funktion

```
EIB7_ERR EIB7GetRecordingMemSize      ( EIB7_HANDLE      eib,  
                                         unsigned long*    size  
                                         )
```

### Parameter

eib	EIB-Handle
size	<i>[Rückgabewert]</i> Pointer auf die Zielvariable für die Größe des Speichers (als Anzahl von Datenpaketen)

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_InvalidPacket      Aktuelle Konfiguration für das Datenpaket ist ungültig

## 7.43 Streaming Status prüfen

Während der „Streaming“ Modus aktiv ist, kann der Zustand des Puffers in der EIB 74x ausgelesen werden. Neben der Größe des Puffers in Bytes wird die aktuell gespeicherte Datenmenge in Bytes angegeben. Außerdem wird die maximal im Puffer gespeicherte Datenmenge seit dem Aktivieren des „Streaming“ Modus in Bytes angegeben.

### Funktion

```
EIB7_ERR EIB7GetStreamingStatus      ( EIB7_HANDLE      eib,  
                                         unsigned long*    length,  
                                         unsigned long*    max,  
                                         unsigned long*    size  
                                         )
```

### Parameter

eib	EIB-Handle
length	<i>[Rückgabewert]</i> Pointer auf die Zielvariable für die Anzahl der Bytes im Puffer
max	<i>[Rückgabewert]</i> Pointer auf die Zielvariable für die maximale Anzahl der Bytes im Puffer seit dem Start des Streaming Modus
size	<i>[Rückgabewert]</i> Pointer auf die Zielvariable für die Größe des Puffers in Bytes

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

#### 7.44 Daten aus FIFO lesen

Datenpakete werden aus dem FIFO in den Zielspeicher kopiert (im Rohdatenformat). Der Parameter „cnt“ gibt die Zahl der zu kopierenden Einträge aus dem FIFO an. Falls der FIFO weniger Datensätze enthält, wird der gesamte Inhalt des FIFOs kopiert. Über den Parameter „entries“ wird die Zahl der tatsächlich kopierten Einträge zurückgegeben. Die Funktion wartet, bis mindestens ein Datensatz aus dem FIFO kopiert wurde, aber maximal bis der Timeout abgelaufen ist. In diesem Fall wird in „entries“ Null zurückgegeben. Aus dem FIFO werden immer ganze Datenpakete kopiert. Der Zielspeicher muss mindestens so groß sein, dass er die angegebene Anzahl an FIFO-Einträge aufnehmen kann. Der Inhalt eines Eintrags im FIFO entspricht dem aktuell konfigurierten Datenpaket ohne den angehängten „Füllbytes“. Alle Datenworte werden im Format „Little Endian“ gespeichert.

##### Funktion

```
EIB7_ERR EIB7ReadFIFODataRaw ( EIB7_HANDLE    eib,
                               void*          data,
                               unsigned long   cnt,
                               unsigned long*  entries,
                               long            timeout
                               )
```

##### Parameter

eib	EIB-Handle
data	[Rückgabewert] Pointer auf Zielspeicher
cnt	Anzahl der zu lesenden Einträge (>=0)
entries	[Rückgabewert] Anzahl der kopierten Einträge
timeout	Timeout in Millisekunden

timeout	Beschreibung
0	Funktion kehrt sofort zurück, wenn keine Daten vorhanden sind
>0	Funktion wartet für x Millisekunden auf Daten
-1	Funktion wartet unendlich

##### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_FIFOEmpty	Keine Daten im FIFO
EIB7_ElementSizeInv	Interner Fehler
EIB7_FIFOOverflow	FIFO-Überlauf seit dem letzten Aufruf der Funktion (Daten gingen verloren)

#### 7.45 Größe eines FIFO-Elements lesen

Die Größe eines FIFO-Elements im Rohdatenformat wird ausgegeben. Dieser Wert entspricht der Größe eines FIFO-Eintrags, der mit der Funktion EIB7ReadFIFODataRaw() ausgelesen wird. Ein FIFO-Element enthält ein Datenpaket, dessen Größe abhängig von der aktuellen Konfiguration ist. Die Größe wird ohne „Füllbytes“ angegeben.

##### Funktion

```
EIB7_ERR EIB7SizeOfFIFOEntryRaw ( EIB7_HANDLE    eib,
                                   unsigned long*  size
                                   )
```

##### Parameter

eib	EIB-Handle
size	[Rückgabewert] Pointer auf Variable für die Größe eines FIFO-Elements in Bytes

##### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.46 Zugriff auf den Inhalt eines FIFO-Elements

Mit dieser Funktion kann auf einzelne Felder eines FIFO-Elements (in Rohdaten) zugegriffen werden. Ein Eintrag des FIFO kann zum Beispiel den Trigger Counter, Positionsdaten, das Statuswort und weitere Daten enthalten. Abhängig von der Konfiguration des Datenpaketes kann dessen Inhalt variieren. Dies muss beim Zugriff auf die Elemente beachtet werden, um die Daten richtig zu interpretieren. Diese Funktion liefert einen Pointer auf das jeweilige Feld innerhalb der Datenstruktur und zusätzlich die Größe des Feldes in Bytes. Über den Parameter „region“ wird eine Grobauswahl getroffen. Hiermit lässt sich die Achse auswählen, von der das Feld bezogen wird. Die Feinselektion kann über den Parameter „type“ erfolgen. Er gibt an, auf welches Datenfeld einer Achse zugegriffen werden soll.

### Funktion

```
EIB7_ERR EIB7GetDataFieldPtrRaw      ( EIB7_HANDLE      eib,
                                       void*              data,
                                       EIB7_DataRegion     region,
                                       EIB7_PositionDataField type,
                                       void**              field,
                                       unsigned long*      size
                                       )
```

### Parameter

eib                    EIB-Handle  
data                   Pointer auf die Datenstruktur (FIFO-Element)  
region                  Achse der EIB 74x

region	Beschreibung
EIB7_DR_Global	Globales Datenfeld für Trigger Counter
EIB7_DR_Encoder1	Daten für Achse 1
EIB7_DR_Encoder2	Daten für Achse 2
EIB7_DR_Encoder3	Daten für Achse 3
EIB7_DR_Encoder4	Daten für Achse 4
EIB7_DR_AUX	Daten für Hilfsachse

type                    Datenelement für eine Achse

type	Beschreibung
EIB7_PDF_TriggerCounter	Trigger Counter (nur in EIB7_DR_Global)
EIB7_PDF_StatusWord	Statuswort für Position
EIB7_PDF_PositionData	Positionswert
EIB7_PDF_AUXPosition	Positionswert für Hilfsachse
EIB7_PDF_Timestamp	Zeitstempel für Position
EIB7_PDF_Analog	ADC-Wert für Signal A und B
EIB7_PDF_ReferencePos	Referenzposition 1 und Referenzposition 2
EIB7_PDF_DistCodedRef	Codierter Referenzwert
EIB7_PDF_EnDat_AI1	EnDat 2.2 Zusatzinformation 1
EIB7_PDF_EnDat_AI2	EnDat 2.2 Zusatzinformation 2

field                    [Rückgabewert] Pointer auf Speicheradresse des Elements aus der Datenstruktur  
size                    [Rückgabewert] Größe des Elements in Bytes

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_FieldNotAvail                    Das angegebene Feld kann nicht gefunden werden

### 7.47 Daten aus FIFO lesen und konvertieren

Datenpakete werden aus dem FIFO in den Zielspeicher kopiert und konvertiert. Der Parameter „cnt“ gibt die Zahl der zu kopierenden Einträge aus dem FIFO an. Falls der FIFO weniger Datensätze enthält, wird der gesamte Inhalt des FIFOs kopiert. Über den Parameter „entries“ wird die Zahl der tatsächlich kopierten Einträge zurückgegeben. Die Funktion wartet, bis mindestens ein Datensatz aus dem FIFO kopiert wurde, aber maximal bis der Timeout abgelaufen ist. In diesem Fall wird in „entries“ Null zurückgegeben. Aus dem FIFO werden immer ganze Datenpakete kopiert. Der Zielspeicher muss mindestens so groß sein, dass er die angegebene Anzahl an FIFO-Einträge aufnehmen kann. Der Inhalt eines Eintrags entspricht dem aktuell konfigurierten Datenpaket ohne den angehängten „Füllbytes“. Alle Datenworte werden im Standard-Format für 16 Bit oder 32 Bit Integer gespeichert, und die Positionswerte werden in das Format ENCODER\_POSITION konvertiert.

#### Funktion

```
EIB7_ERR EIB7ReadFIFOData ( EIB7_HANDLE eib,
                           void* data,
                           unsigned long cnt,
                           unsigned long* entries,
                           long timeout
                           )
```

#### Parameter

eib	EIB-Handle
data	[Rückgabewert] Pointer auf Zielspeicher
cnt	Anzahl der zu lesenden Einträge (>= 0)
entries	[Rückgabewert] Anzahl der kopierten Einträge
timeout	Timeout in Millisekunden

timeout	Beschreibung
0	Funktion kehrt sofort zurück, wenn keine Daten vorhanden sind
>0	Funktion wartet für x Millisekunden auf Daten
-1	Funktion wartet unendlich

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_FIFOEmpty	Keine Daten im FIFO
EIB7_ElementSizeInv	Interner Fehler
EIB7_FIFOOverflow	FIFO-Überlauf seit dem letzten Aufruf der Funktion (Daten gingen verloren)

### 7.48 Größe eines FIFO-Elements nach der Konvertierung lesen

Die Größe eines FIFO-Elements nach der Konvertierung wird ausgegeben. Dieser Wert entspricht der Größe eines FIFO-Eintrags, der mit der Funktion EIB7ReadFIFOData() ausgelesen wird. Ein FIFO-Element enthält ein Datenpaket, dessen Größe abhängig von der aktuellen Konfiguration ist. Die Größe wird ohne „Füllbytes“ angegeben.

#### Funktion

```
EIB7_ERR EIB7SizeOfFIFOEntry ( EIB7_HANDLE eib,
                               unsigned long* size
                               )
```

#### Parameter

eib	EIB-Handle
size	[Rückgabewert] Pointer auf Variable für die Größe eines FIFO-Elements in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.49 Zugriff auf den Inhalt eines FIFO-Elements mit konvertierten Daten

Mit dieser Funktion kann auf einzelne Felder eines FIFO-Elements mit konvertierten Positionsdaten (Positionsformat ENCODER\_POSITION) zugegriffen werden. Ein Eintrag des FIFO kann zum Beispiel den Trigger Counter, Positionsdaten, das Statuswort und weitere Daten enthalten. Abhängig von der Konfiguration des Datenpaketes kann dessen Inhalt variieren. Dies muss beim Zugriff auf die Elemente beachtet werden, um die Daten richtig zu interpretieren. Die Funktion liefert einen Pointer auf das jeweilige Feld innerhalb der Datenstruktur und zusätzlich die Größe des Feldes in Bytes. Über den Parameter „region“ wird eine Grobauswahl getroffen. Hiermit lässt sich die Achse auswählen, von der das Feld bezogen wird. Die Feinselektion kann über den Parameter „type“ erfolgen. Er gibt an, auf welches Datenfeld einer Achse zugegriffen werden soll.

### Funktion

```
EIB7_ERR EIB7GetDataFieldPtr ( EIB7_HANDLE      eib,
                               void*            data,
                               EIB7_DataRegion   region,
                               EIB7_PositionDataField type,
                               void**           field,
                               unsigned long*    size
                               )
```

### Parameter

eib                    EIB-Handle  
data                   Pointer auf die Datenstruktur (FIFO-Element)  
region                  Achse der EIB 74x

region	Beschreibung
EIB7_DR_Global	Globales Datenfeld für Trigger Counter
EIB7_DR_Encoder1	Daten für Achse 1
EIB7_DR_Encoder2	Daten für Achse 2
EIB7_DR_Encoder3	Daten für Achse 3
EIB7_DR_Encoder4	Daten für Achse 4
EIB7_DR_AUX	Daten für Hilfsachse

type                   Datenelement für eine Achse

type	Beschreibung
EIB7_PDF_TriggerCounter	Trigger Counter (nur in EIB7_DR_Global)
EIB7_PDF_StatusWord	Statuswort für Position
EIB7_PDF_PositionData	Positionswert
EIB7_PDF_AUXPosition	Positionswert für Hilfsachse
EIB7_PDF_Timestamp	Zeitstempel für Position
EIB7_PDF_Analog	ADC-Wert für Signal A und B
EIB7_PDF_ReferencePos	Referenzposition 1 und Referenzposition 2
EIB7_PDF_DistCodedRef	Codierter Referenzwert
EIB7_PDF_EnDat_AI1	EnDat 2.2 Zusatzinformation 1
EIB7_PDF_EnDat_AI2	EnDat 2.2 Zusatzinformation 2

field                   *[Rückgabewert]* Pointer auf Speicheradresse des Elements aus der Datenstruktur  
size                   *[Rückgabewert]* Größe des Elements in Bytes

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_FieldNotAvail                   Das angegebene Feld kann nicht gefunden werden

## 7.50 Anzahl der Elemente im FIFO lesen

Die Anzahl der aktuell im FIFO gespeicherten Elemente wird ausgegeben.

### Funktion

```
EIB7_ERR EIB7FIFOEntryCount ( EIB7_HANDLE      eib,
                               unsigned long*    cnt
                               )
```

### Parameter

eib                    EIB-Handle  
cnt                   *[Rückgabewert]* Pointer auf Variable für die Anzahl der FIFO-Elemente

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.



### 7.51 FIFO löschen

Der Inhalt des FIFOs wird gelöscht. Dieses Kommando hat keine Auswirkung, wenn der Polling Modus aktiv ist.

#### Funktion

```
EIB7_ERR EIB7ClearFIFO          ( EIB7_HANDLE      eib
                                )
```

#### Parameter

eib	EIB-Handle
-----	------------

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.52 FIFO-Größe einstellen

Die Größe des FIFOs wird neu festgelegt. Alle Daten im FIFO werden gelöscht. Die Größe kann nur im Polling Modus eingestellt werden. Der FIFO muss mindestens 2000 Bytes groß sein. Falls der Wert kleiner ist, wird intern der Wert 2000 Bytes verwendet.

#### Funktion

```
EIB7_ERR EIB7SetFIFOSize        ( EIB7_HANDLE      eib,
                                unsigned long      size
                                )
```

#### Parameter

eib	EIB-Handle
size	FIFO-Größe in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_SoftRTEn	Soft Realtime Modus ist aktiviert
---------------	-----------------------------------

### 7.53 FIFO-Größe auslesen

Die Größe des FIFOs in Bytes wird ausgegeben.

#### Funktion

```
EIB7_ERR EIB7GetFIFOSize        ( EIB7_HANDLE      eib,
                                unsigned long*      size
                                )
```

#### Parameter

eib	EIB-Handle
size	[Rückgabewert] Pointer auf die Variable für die FIFO-Größe in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 7.54 Callback-Mechanismus aktivieren

Der Callback-Mechanismus wird aktiviert, bzw. deaktiviert und gegebenenfalls der Funktionspointer gespeichert. Die Callback-Funktion wird aufgerufen, wenn mindestens so viele Elemente im FIFO gespeichert sind, wie im Parameter „threshold“ angegeben. Anschließend wird die Funktion erst wieder aufgerufen, wenn neue Daten in den FIFO geschrieben wurden, und danach mindestens „threshold“ Elemente im FIFO gespeichert sind.

### Funktion

```
EIB7_ERR EIB7SetDataCallback ( EIB7_HANDLE      eib,  
                               void*            data,  
                               EIB7_MODE        activate,  
                               unsigned long    threshold,  
                               EIB7OnDataAvailable handler  
                               )
```

### Parameter

eib	EIB-Handle
data	Pointer auf Benutzerdaten, dieser Pointer wird als Parameter an die Callback-Funktion übergeben
activate	Callback aktivieren oder deaktivieren

activate	Beschreibung
EIB7_MD_Disable	Callback Mechanismus deaktivieren
EIB7_MD_Enable	Callback Mechanismus aktivieren

threshold	Anzahl der Elemente im FIFO, ab dem der Callback-Mechanismus auslöst (>0)
handler	Pointer auf die Callback-Funktion (NULL ist erlaubt, falls activate = 0)

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### Callback-Funktion

Die Callback-Funktion wird vom Treiber ausgeführt und läuft in einem separaten Thread. Der Benutzer muss sich selbst um die eventuell notwendige Synchronisation mit dem Hauptprogramm kümmern. Der Parameter „eib“ enthält das Handle auf die EIB 74x, welche den Callback ausgelöst hat. In „cnt“ steht die Anzahl der aktuell im FIFO gespeicherten Elemente. Der Parameter „data“ enthält den Pointer, welcher bei der Registrierung der Callback-Funktion angegeben wurde.

### Prototyp

```
typedef void (*EIB7OnDataAvailable) ( EIB7_HANDLE      eib,  
                                       unsigned long    cnt,  
                                       void*            data  
                                       )
```

### Parameter

eib	EIB-Handle
cnt	Anzahl der Elemente im FIFO
data	Pointer auf Benutzerdaten

### 7.55 Triggerquelle für Hilfsachse wählen

Das Triggersignal für die Hilfsachse kann aus verschiedenen Quellen gewählt werden. Diese Einstellung ist nur im Betriebsmodus „Polling“ möglich.

#### Funktion

```
EIB7_ERR EIB7AuxAxisTriggerSource ( EIB7_HANDLE eib,  
                                     EIB7_AxisTriggerSrc src  
                                     )
```

#### Parameter

eib                    EIB-Handle  
src                    Triggerquelle

src	Beschreibung
EIB7_AT_TrgInput1	Triggereingang Kanal 1
EIB7_AT_TrgInput2	Triggereingang Kanal 2
EIB7_AT_TrgInput3	Triggereingang Kanal 3
EIB7_AT_TrgInput4	Triggereingang Kanal 4
EIB7_AT_TrgSW1	Software-Trigger Kanal 1
EIB7_AT_TrgSW2	Software-Trigger Kanal 2
EIB7_AT_TrgSW3	Software-Trigger Kanal 3
EIB7_AT_TrgSW4	Software-Trigger Kanal 4
EIB7_AT_TrgRI	Referenzimpuls der entsprechenden Achse
EIB7_AT_TrgRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_AT_TrgIC	Interval Counter
EIB7_AT_TrgPuls	Trigger Puls Zähler
EIB7_AT_TrgTimer	Timer Trigger

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid                    Parameter ungültig

### 7.56 Position der Hilfsachse abfragen

Der aktuelle Positionswert wird ausgelesen. Zusätzlich wird ein Statuswort übertragen, aus dem mögliche Positionsfehler hervorgehen. Die Positionsabfrage kann nur im Polling Modus erfolgen. Die Einstellung des Interpolationsfaktors und der Flankenauswertung für den Interval Counter bestimmen das Verhältnis eines LSB im Positionswert zur Signalperiode des Messgeräts.

#### Funktion

```
EIB7_ERR EIB7AuxGetPosition ( EIB7_HANDLE eib,  
                               unsigned short* status,  
                               ENCODER_POSITION* pos  
                               )
```

#### Parameter

eib                    EIB-Handle  
status                *[Rückgabewert]* Pointer auf Zielvariable für das Statuswort  
pos                    *[Rückgabewert]* Pointer auf Zielvariable für die Position

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_CantLatchPos                    Position kann nicht bestimmt werden  
EIB7\_EncPwrSuppErr                   Fehler in der Spannungsversorgung des Messgeräts  
EIB7\_NotInitialized                   Hilfsachse ist nicht konfiguriert

### 7.57 Daten der Hilfsachse auslesen

Die aktuelle Position und einige zusätzliche Parameter werden bestimmt und ausgelesen. Die Funktion darf nur im Polling Modus ausgeführt werden. Das Statuswort gibt an, ob die Position und die Referenzposition gültig sind. Der Positionswert und der Timestamp werden zeitgleich gespeichert. Um dies zu erreichen wird intern der Software-Trigger Kanal 1 verwendet. Die Triggerquelle für die Hilfsachse wird entsprechend konfiguriert und dabei die aktuelle Einstellung überschrieben.

#### Funktion

```
EIB7_ERR EIB7AuxGetEncoderData ( EIB7_HANDLE          eib,  
                                unsigned short*       status,  
                                ENCODER_POSITION*     pos,  
                                ENCODER_POSITION*     ref,  
                                unsigned long*        timestamp,  
                                unsigned short*       counter  
                                )
```

#### Parameter

eib	EIB-Handle
status	<i>[Rückgabewert]</i> Pointer auf Zielvariable für das Statuswort
pos	<i>[Rückgabewert]</i> Pointer auf Zielvariable für die Position
ref	<i>[Rückgabewert]</i> Pointer auf Zielvariable für die Referenzposition
timestamp	<i>[Rückgabewert]</i> Pointer auf Zielvariable für den Timestamp Wert
counter	<i>[Rückgabewert]</i> Pointer auf Zielvariable für den Trigger Counter

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_CantLatchPos	Position kann nicht bestimmt werden
EIB7_EncPwrSuppErr	Fehler in der Spannungsversorgung des Messgeräts
EIB7_NotInitialized	Hilfsachse ist nicht konfiguriert

### 7.58 Zähler der Hilfsachse löschen

Der Positionszähler der Hilfsachse wird gelöscht.

#### Funktion

```
EIB7_ERR EIB7AuxClearCounter ( EIB7_HANDLE          eib  
                                )
```

#### Parameter

eib	EIB-Handle
-----	------------

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.59 Signalfehler der Hilfsachse quittieren

Die Fehlermeldungen für die Hilfsachse werden gelöscht. Dies betrifft den Fehler für die Signalamplitude und für die Frequenzüberschreitung.

#### Funktion

```
EIB7_ERR EIB7AuxClearSignalErrors ( EIB7_HANDLE          eib  
                                )
```

#### Parameter

eib	EIB-Handle
-----	------------

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.60 Triggerfehler der Hilfsachse quittieren

Die Fehlermeldungen für nicht erkannte Triggerereignisse in der Trigger-Logik für die Hilfsachse werden gelöscht.

#### Funktion

```
EIB7_ERR EIB7AuxClearLostTriggerError ( EIB7_HANDLE eib
)
```

#### Parameter

eib EIB-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.61 Statusbit für Referenzmarke der Hilfsachse löschen

Das Flag „Referenzposition gespeichert“ im Statuswort für die Hilfsachse wird zurückgesetzt.

#### Funktion

```
EIB7_ERR EIB7AuxClearRefStatus ( EIB7_HANDLE eib
)
```

#### Parameter

eib EIB-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.62 Status der Referenzfahrt für die Hilfsachse prüfen

Der Status der Referenzfahrt wird ausgegeben. Er gibt an, ob die Referenzfahrt noch aktiv ist, oder die Referenzpositionen bereits gespeichert wurden.

#### Funktion

```
EIB7_ERR EIB7AuxGetRefActive ( EIB7_HANDLE eib,
                               EIB7_MODE* active
)
```

#### Parameter

eib EIB-Handle  
active *[Rückgabewert]* Pointer auf Variable für den Status der Referenzfahrt

active	Beschreibung
EIB7_MD_Enable	Referenzfahrt aktiv
EIB7_MD_Disable	Referenzfahrt nicht aktiv

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.63 Referenzfahrt für die Hilfsachse starten

Nach Aufruf dieses Kommandos wird die Referenzposition beim Überfahren des nächsten Referenzimpulses gespeichert. Der gespeicherte Wert entspricht dem Stand des Periodenzählers bei der Referenzmarke. Sobald die Referenzfahrt gestartet wurde, wird der Status für eine zuvor gespeicherte Referenzposition auf „ungültig“ gesetzt.

#### Funktion

```
EIB7_ERR EIB7AuxStartRef ( EIB7_HANDLE eib
)
```

#### Parameter

eib EIB-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.64 Referenzfahrt für die Hilfsachse stoppen

Die Referenzfahrt (Modus zur automatischen Speicherung der Referenzposition) wird beendet. Bereits gespeicherte Referenzpositionen werden dadurch nicht gelöscht.

#### Funktion

```
EIB7_ERR EIB7AuxStopRef      ( EIB7_HANDLE      eib
                               )
```

#### Parameter

eib                            EIB-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.65 Timestamp für die Hilfsachse konfigurieren

Der Timestamp kann für die Hilfsachse aktiviert oder deaktiviert werden. Als Periodendauer kommt die globale Einstellung der EIB 74x zur Anwendung. Der Timestamp-Wert wird bei einer Positionsabfrage für die Hilfsachse kopiert, wenn diese Funktion zuvor aktiviert wurde.

#### Funktion

```
EIB7_ERR EIB7AuxSetTimestamp ( EIB7_HANDLE      eib,
                               EIB7_MODE        mode
                               )
```

#### Parameter

eib                            EIB-Handle  
mode                          Timestamp aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Enable	Timestamp aktivieren
EIB7_MD_Disable	Timestamp deaktivieren

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 7.66 Triggerflanke für Referenzimpuls der Hilfsachse einstellen

Der Zeitpunkt für die Triggerung mit einem Referenzimpuls für die Hilfsachse kann auf die steigende, fallende oder beide Flanke des RI-Signals eingestellt werden. Falls beide Flanken als Triggerereignis dienen, muss beachtet werden, dass die maximale Triggerrate nicht überschritten wird.

#### Funktion

```
EIB7_ERR EIB7AuxSetRITriggerEdge ( EIB7_HANDLE      eib,
                                   EIB7_RITriggerEdge edge
                                   )
```

#### Parameter

eib                            EIB-Handle  
edge                          Aktive Triggerflanke für den Referenzimpuls

edge	Beschreibung
EIB7_RI_Rising	Steigende Flanke des Referenzimpuls
EIB7_RI_Falling	Fallende Flanke des Referenzimpuls
EIB7_RI_Both	Beide Flanken des Referenzimpuls

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid            Parameter ungültig

## 8 Achsfunktionen

Die Achsfunktionen beziehen sich immer nur auf eine Achse der EIB 74x. Alle anderen Achsen werden nicht beeinflusst.

Alle Achsfunktionen können als Rückgabewert die nachfolgend aufgeführten Fehlermeldungen liefern. Zusätzlich dazu können sie individuell weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

### Standard Rückgabewerte

EIB7_NoError	Funktionsaufruf erfolgreich
EIB7_InvalidHandle	Das Handle auf die Achse der EIB 74x ist ungültig
EIB7_FuncNotSupp	Funktion wird von der EIB 74x nicht unterstützt
EIB7_InvalidResponse	Fehler bei der Datenübertragung
EIB7_AccNotAllowed	Funktion kann nicht ausgeführt werden, da die EIB 74x den Zugriff nicht erlaubt
EIB7_ConnReset	Verbindung wurde von der EIB 74x beendet
EIB7_ConnTimeout	Timeout bei der Datenübertragung zur EIB 74x
EIB7_ReceiveError	Fehler beim Empfangen der Daten
EIB7_SendError	Fehler beim Senden der Daten
EIB7_OutOfMemory	Vom System kann nicht genügend Speicher allokiert werden

### 8.1 Achse initialisieren

Eine Achse der EIB 74x wird für das angeschlossene Messgerät konfiguriert. Die Nummer der Achse einer EIB 74x wird über das Axis-Handle bestimmt. Als grundlegende Option muss die Art der Schnittstelle des Messgeräts ausgewählt werden. Einige Parameter werden nur für inkrementale und andere nur für EnDat-Schnittstellen benötigt. Für ein EnDat 2.2 Interface kann über den Parameter „iface“ zusätzlich noch die Laufzeitkompensation aktiviert werden. Dazu müssen die Konstanten „EIB7\_IT\_EnDat22“ und „EIB7\_IT\_EnDatDelayMeasurement“ mit „Oder“ verknüpft werden.

Die Parameter „EnDatclock“, „recovery“ und „calculation“ werden nur für Messgeräte mit EnDat-Schnittstelle verwendet. Der Takt der EnDat-Schnittstelle kann eingestellt werden. Hierfür sollten die vordefinierten Konstanten verwendet werden. Bei der Initialisierung einer Achse für EnDat-Betrieb wird an das angeschlossene Messgerät ein EnDat Reset-Befehl gesendet. Zusätzlich kann für EnDat 2.2 Messgeräte (Bestellbezeichnung EnDat02 bzw. EnDat22) die „Recovery time I“ über den Parameter „recovery“ eingestellt werden. Die „EnDat Calculation time“ muss über den Parameter „calculation“ für das Messgerät eingestellt werden.

Der Parameter „bandwidth“ und „comp“ wirkt sich nur für inkrementale Messgeräte aus. Die Bandbreite kann auf die zwei Zustände hoch und niedrig konfiguriert werden. Die Online-Kompensation kann aktiviert und deaktiviert werden. Die Angaben für „linecounts“ und „increment“ werden nur im Zusammenhang mit abstandscodierten Referenzmarken benötigt.

Nach Aufruf dieser Funktion sind folgende Flags gelöscht:

- Fehler Signalamplitude
- Frequenzüberschreitung
- Referenzposition 1 gespeichert
- Referenzposition 2 gespeichert
- codierter Referenzwert bei abstandscodierten Referenzmarken gültig
- Fehler bei Berechnung des codierten Referenzwertes bei abstandscodierten Referenzmarken
- CRC-Fehler

Die Einstellungen für die Abschlusswiderstände der Inkrementalsignale sowie der Wert des Periodenzählers werden durch die Funktion nicht beeinflusst.

## Funktion

```

EIB7_ERR EIB7InitAxis      (  EIB7_AXIS      axis,
                             unsigned long   iface,
                             EIB7_EncoderType type,
                             EIB7_Refmarks   refmarks,
                             unsigned long   linecounts,
                             unsigned long   increment,
                             EIB7_Homing     homing,
                             EIB7_Limit      limit,
                             EIB7_Compensation comp,
                             EIB7_Bandwidth   bandwidth,
                             unsigned long   EnDatclock,
                             EIB7_EnDatRecoveryTime recovery,
                             EIB7_EnDatCalcTime calculation,
                             )

```

## Parameter

axis                    AXIS-Handle  
iface                    Art der Schnittstelle des Messgeräts

iface	Beschreibung
EIB7_IT_Disabled	Achse deaktiviert
EIB7_IT_Incremental	Messgerät mit Inkrementalsignale (1V <sub>SS</sub> )
EIB7_IT_Incremental_11u	Messgerät mit Inkrementalsignale (11µA)
EIB7_IT_EnDat21	Messgerät mit EnDat 2.1 Interface
EIB7_IT_EnDat01	Messgerät mit EnDat 2.1 Interface und Inkrementalsignale (1V <sub>SS</sub> )
EIB7_IT_EnDat22	Messgerät mit EnDat 2.2 Interface
EIB7_IT_EnDatDelayMeasurement	Laufzeitkompensation für EnDat 2.2

type                    Typ des Messgeräts

type	Beschreibung
EIB7_EC_Linear	Längenmessgerät
EIB7_EC_Rotary	Winkelmessgerät / Drehgeber

refmarks                Art der Referenzmarken

refmarks	Beschreibung
EIB7_RM_None	Keine Referenzmarke
EIB7_RM_One	Eine Referenzmarke (EIB 74x berechnet keinen codierten Referenzwert)
EIB7_RM_DistanceCoded	Abstandscodierte Referenzmarken (EIB 74x berechnet codierten Referenzwert automatisch)

linecounts              Anzahl der Signalperioden pro Umdrehung (nur für rotative Messgeräte)  
increment                Nominaler Abstand in Signalperioden zwischen zwei festen Referenzmarken (nur für abstandscodierte Referenzmarken)

homing                  Homing-Signalauswertung aktivieren oder deaktivieren

homing	Beschreibung
EIB7_HS_None	Homing-Signal wird nicht ausgewertet
EIB7_HS_Available	Homing-Signal wird ausgewertet

limit                    Limit-Signalauswertung aktivieren oder deaktivieren

limit	Beschreibung
EIB7_LS_None	Limit -Signal wird nicht ausgewertet
EIB7_LS_Available	Limit -Signal wird ausgewertet

compensation            Online-Kompensation aktivieren oder deaktivieren

compensation	Beschreibung
EIB7_CS_None	Signalkompensation deaktiviert
EIB7_CS_CompActive	Signalkompensation aktiviert



bandwidth                      Eingangsbandbreite für Inkrementalsignale (high/low)

<b>bandwidth</b>	<b>Beschreibung</b>
EIB7_BW_High	Hohe Eingangsbandbreite für 1V <sub>SS</sub> Signale
EIB7_BW_Low	Niedrige Eingangsbandbreite für 1V <sub>SS</sub> Signale

EnDatclock                      EnDat-Taktrate

<b>EnDatclock</b>	<b>Beschreibung</b>
EIB7_CLK_Default	Default EnDat 2.1 / EnDat 2.2 Takt
EIB7_CLK_100KHz	EnDat Takt 100 kHz
EIB7_CLK_300KHz	EnDat Takt 300 kHz (Default für EnDat 2.1)
EIB7_CLK_500KHz	EnDat Takt 500 kHz
EIB7_CLK_1MHz	EnDat Takt 1 MHz
EIB7_CLK_2MHz	EnDat Takt 2 MHz (Default für EnDat 2.2)
EIB7_CLK_4MHz	EnDat Takt 4 MHz
EIB7_CLK_5MHz	EnDat Takt 5 MHz
EIB7_CLK_6_66MHz	EnDat Takt 6,66 MHz

recovery                      "Recovery Time I" für EnDat 2.2

<b>recovery</b>	<b>Beschreibung</b>
EIB7_RT_Long	Lange Recovery time I nach EnDat Spezifikation
EIB7_RT_Short	Kurze Recovery time I nach EnDat Spezifikation

calculation                      "Calculation time" für EnDat

<b>calculation</b>	<b>Beschreibung</b>
EIB7_CT_Long	Lange Calculation time
EIB7_CT_Short	Kurze Calculation time

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Parameter ungültig
EIB7_InvInterface	Schnittstellentyp ungültig
EIB7_InvRefMarkOpt	Referenzmarke ungültig
EIB7_InvDistCodeRef	Parameter für abstandscodierte Referenzmarken ungültig (linecount, increment)
EIB7_ConfOptIncons	Parameter sind in dieser Form nicht kombinierbar
EIB7_AccNotAllowed	Zugriff verweigert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)

### Anmerkung:

Wird die Achse für EnDat konfiguriert (EnDat01, EnDat21, EnDat22), dann wurden mit Ausführung von „EIB7InitAxis“ bis zur Firmware-Version 9 automatisch die EnDat Fehler- und Warnmeldungen rückgesetzt. Ab Firmware-Version 10 werden die Fehler- und Warnmeldungen nicht mehr rückgesetzt. Hintergrund ist die Fehlerbehandlung für batteriegepufferte Messgeräte (siehe auch EnDat Application Notes).

## 8.2 Triggerquelle für Achse wählen

Das Triggersignal für die Achse kann aus verschiedenen Quellen gewählt werden. Diese Einstellung ist nur im Betriebsmodus „Polling“ möglich.

### Funktion

```
EIB7_ERR EIB7AxisTriggerSource ( EIB7_AXIS          axis,  
                                EIB7_AxisTriggerSrc  src  
                                )
```

### Parameter

axis	AXIS-Handle
src	Triggerquelle

src	Beschreibung
EIB7_AT_TrgInput1	Triggereingang Kanal 1
EIB7_AT_TrgInput2	Triggereingang Kanal 2
EIB7_AT_TrgInput3	Triggereingang Kanal 3
EIB7_AT_TrgInput4	Triggereingang Kanal 4
EIB7_AT_TrgSW1	Software-Trigger Kanal 1
EIB7_AT_TrgSW2	Software-Trigger Kanal 2
EIB7_AT_TrgSW3	Software-Trigger Kanal 3
EIB7_AT_TrgSW4	Software-Trigger Kanal 4
EIB7_AT_TrgRI	Referenzimpuls der entsprechenden Achse
EIB7_AT_TrgRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_AT_TrgIC	Interval Counter
EIB7_AT_TrgPuls	Trigger Puls Zähler
EIB7_AT_TrgTimer	Timer Trigger

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid          Parameter ungültig

## 8.3 Triggerflanke für Referenzimpuls einstellen

Der Zeitpunkt für die Triggerung mit einem Referenzimpuls kann auf die steigende, fallende oder beide Flanke des RI-Signals eingestellt werden. Falls beide Flanken als Triggerereignis dienen, muss beachtet werden, dass die maximale Triggerrate nicht überschritten wird.

### Funktion

```
EIB7_ERR EIB7SetRITriggerEdge ( EIB7_AXIS          axis,  
                                EIB7_RITriggerEdge  edge  
                                )
```

### Parameter

axis	AXIS-Handle
edge	Aktive Triggerflanke für den Referenzimpuls

edge	Beschreibung
EIB7_RI_Rising	Steigende Flanke des Referenzimpuls
EIB7_RI_Falling	Fallende Flanke des Referenzimpuls
EIB7_RI_Both	Beide Flanken des Referenzimpuls

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid          Parameter ungültig

## 8.4 Zähler löschen

Der Periodenzähler einer Achse wird gelöscht. Diese Funktion ist nur zulässig, wenn die Achse für inkrementale Messgeräte konfiguriert ist. Andernfalls wird eine Fehlermeldung generiert. Es wird nur der Periodenzähler zurück gesetzt. Der Interpolationswert wird nicht verändert.

### Funktion

```
EIB7_ERR EIB7ClearCounter      ( EIB7_AXIS      axis
                                )
```

### Parameter

axis	AXIS-Handle
------	-------------

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Schnittstellentyp ungültig
-------------------	----------------------------

## 8.5 Position abfragen

Der aktuelle Positionswert des Messgeräts wird ermittelt und ausgelesen. Zusätzlich wird ein Statuswort übertragen, aus dem mögliche Positionsfehler hervorgehen. Die Positionsabfrage kann nur im Polling Modus erfolgen. Abhängig davon, ob die Achse für inkrementale oder für EnDat-Messgeräte konfiguriert ist, wird der interpolierte Wert der Inkrementalsignale geliefert, oder eine EnDat-Abfrage an das Messgerät gesendet. Bei EnDat 01 Konfiguration wird der interpolierte Wert der Inkrementalsignale gelesen.

### Funktion

```
EIB7_ERR EIB7GetPosition      ( EIB7_AXIS      axis,
                                unsigned short* status,
                                ENCODER_POSITION* pos
                                )
```

### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für die Position

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_CantLatchPos	Position kann nicht bestimmt werden
EIB7_EncPwrSuppErr	Fehler in der Spannungsversorgung des Messgeräts (nur EnDat-Messgeräte)
EIB7_NotInitialized	Achse ist nicht konfiguriert

## 8.6 Daten für einen Kanal auslesen

Die aktuelle Position und einige zusätzliche Parameter werden bestimmt und ausgelesen. Der Parameter „refc“ ist nur gültig, wenn die Achse entsprechend für Messgeräte mit abstandscodierten Referenzmarken konfiguriert ist. Die Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für inkrementale Messgeräte konfiguriert sein. Der Positionswert und der Timestamp werden zeitgleich gespeichert. Um dies zu erreichen wird intern der Software-Trigger Kanal 1 verwendet. Die Triggerquelle für die Hilfsachse wird entsprechend konfiguriert und dabei die aktuelle Einstellung überschrieben.

### Funktion

```
EIB7_ERR EIB7GetEncoderData ( EIB7_AXIS          axis,
                              unsigned short*      status,
                              ENCODER_POSITION*    pos,
                              ENCODER_POSITION*    ref1,
                              ENCODER_POSITION*    ref2,
                              ENCODER_POSITION*    refc,
                              unsigned long*       timestamp,
                              unsigned short*      counter,
                              unsigned short*      adc00,
                              unsigned short*      adc90
                              )
```

### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für die Position
ref1	[Rückgabewert] Pointer auf Variable für die Referenzposition 1
ref2	[Rückgabewert] Pointer auf Variable für die Referenzposition 2
refc	[Rückgabewert] Pointer auf Variable für den codierten Referenzwert
timestamp	[Rückgabewert] Pointer auf Variable für den Timestamp Wert
counter	[Rückgabewert] Pointer auf Variable für den Trigger Counter
adc00	[Rückgabewert] Pointer auf Variable für den ADC-Wert für Signal A
adc90	[Rückgabewert] Pointer auf Variable für den ADC-Wert für Signal B

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Parameter ungültig (evtl. ist Achse nicht für inkrementale Messgeräte konfiguriert)
EIB7_EncPwrSuppErr	Fehler in der Spannungsversorgung des Messgeräts (nur für EnDat-Messgeräte)
EIB7_NotInitialized	Achse ist nicht konfiguriert

## 8.7 Spannungsversorgungsfehler quittieren

Die Fehlermeldung für die Spannungsversorgung des Messgeräts wird quittiert. Falls für diese Achse kein Fehler aufgetreten ist, wird die Funktion mit einer Fehlermeldung beendet. Die Spannungsversorgung für das Messgerät wird nach der Quittierung des Fehlers wieder eingeschaltet.

### Funktion

```
EIB7_ERR EIB7ClearPowerSupplyError ( EIB7_AXIS          axis
                                      )
```

### Parameter

axis	AXIS-Handle
------	-------------

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_CantClearEnc	Fehler kann nicht gelöscht werden
-------------------	-----------------------------------

## 8.8 Triggerfehler quittieren

Die Fehlermeldung für das Triggerinterface wird quittiert. Der Triggerfehler wird für alle Achsen einer EIB 74x gleichzeitig gelöscht, unabhängig davon welches AXIS-Handle übergeben wird.

### Funktion

```
EIB7_ERR EIB7ClearLostTriggerError      ( EIB7_AXIS      axis
                                         )
```

### Parameter

axis                      AXIS-Handle

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 8.9 Signalfehler quittieren

Die Fehlermeldungen für die Signale des Messgeräts werden gelöscht. Bei inkrementalen Messgeräten wird der Fehler für die Signalamplitude und für die Frequenzüberschreitung quittiert. Bei EnDat Messgeräten wird der CRC-Fehler für die Datenübertragung und die EnDat Fehlermeldungen gelöscht. Der Fehlerspeicher im Messgerät wird nicht beeinflusst. Es wird kein EnDat-Kommando gesendet.

### Funktion

```
EIB7_ERR EIB7ClearEncoderErrors        ( EIB7_AXIS      axis
                                         )
```

### Parameter

axis                      AXIS-Handle

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 8.10 EnDat-Fehlerbits löschen

Die EnDat Fehlerflags werden gelöscht. Diese Funktion ist nur für Achsen zulässig, die für EnDat-Messgeräte konfiguriert sind. An das EnDat Messgerät wird ein EnDat Reset-Befehl gesendet, um den Fehlerspeicher zu löschen. Die Funktion wartet nach dem Reset-Befehl die laut EnDat-Spezifikation erforderliche Zeit von 50 ms. Diese Funktion darf nur im Polling Modus ausgeführt werden.

### Funktion

```
EIB7_ERR EIB7ClearEnDatErrorMsg        ( EIB7_AXIS      axis
                                         )
```

### Parameter

axis                      AXIS-Handle

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat konfiguriert
EIB7_EncPwrSuppErr	Fehler in der Spannungsversorgung des Messgeräts
EIB7_NotInitialized	Achse ist nicht konfiguriert

### 8.11 Statusbits für Referenzmarken löschen

Die Flags für die Referenzposition im Statuswort werden zurückgesetzt. Folgende Flags werden zurückgesetzt: „Referenzposition 1 gespeichert“, „Referenzposition 2 gespeichert“. Dieses Kommando ist nur für Achsen zulässig, die für inkrementale Messgeräte mit Referenzmarken konfiguriert sind.

#### Funktion

```
EIB7_ERR EIB7ClearRefLatched ( EIB7_AXIS axis
                               )
```

#### Parameter

axis                      AXIS-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_InvInterface              Achse ist nicht für Inkrementale Messgeräte konfiguriert  
EIB7\_NotInitialized            Achse ist nicht konfiguriert

### 8.12 Statusbits für abstandscodierte Referenzmarken löschen

Die Flags für die Referenzposition im Statuswort werden zurückgesetzt. Folgende Flags werden zurückgesetzt: „Referenzposition 1 gespeichert“, „Referenzposition 2 gespeichert“, „Codierter Referenzwert gültig“, „Fehler bei Berechnung des codierten Referenzwertes“. Dieses Kommando ist nur für Achsen zulässig, die für Messgeräte mit abstandskodierten Referenzmarken konfiguriert sind.

#### Funktion

```
EIB7_ERR EIB7ClearRefStatus ( EIB7_AXIS axis
                               )
```

#### Parameter

axis                      AXIS-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_InvInterface              Achse ist nicht für Inkrementale Messgeräte konfiguriert  
EIB7\_NotInitialized            Achse ist nicht konfiguriert

### 8.13 Referenzfahrt starten

Nach Aufruf dieses Kommandos wird die Referenzposition beim Überfahren des nächsten Referenzimpulses gespeichert. Über den Parameter „ref“ kann festgelegt werden, ob nur eine oder zwei Referenzpositionen gespeichert werden. Falls zwei Referenzpositionen aktiviert sind, wird mit jedem der zwei folgenden Referenzimpulse ein Positionswert gespeichert. Die gespeicherten Werte entsprechen dem Stand des Periodenzählers bei der jeweiligen Referenzmarke. Dieses Kommando ist nur für Achsen zulässig, die für inkrementale Messgeräte konfiguriert sind.

Wird diese Funktion erneut aufgerufen bevor alle Referenzpositionen aus dem ersten Aufruf gespeichert wurden, werden die alten Referenzpositionswerte ungültig, was durch die Flags im Statuswort gekennzeichnet wird. Die Referenzfahrt wird neu gestartet.

#### Funktion

```
EIB7_ERR EIB7StartRef ( EIB7_AXIS axis,
                        EIB7_ReferencePosition ref
                        )
```

#### Parameter

axis                      AXIS-Handle  
ref                        Option für die zu speichernde Referenzposition

ref	Beschreibung
EIB7_RP_RefPos1	Eine Referenzposition speichern
EIB7_RP_RefPos2	Zwei Referenzpositionen speichern

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_InvInterface              Achse ist nicht für inkrementale Messgeräte konfiguriert  
EIB7\_NotInitialized            Achse ist nicht konfiguriert  
EIB7\_ParamInvalid              Parameter ist keine gültige Option für die Referenzmarken

### 8.14 Referenzfahrt stoppen

Die Referenzfahrt (Modus zur automatischen Speicherung der Referenzposition) wird beendet. Wurden bereits Referenzimpulse überfahren, so bleiben die zugehörigen Positionswerte erhalten. Dieses Kommando ist nur für Achsen zulässig, die für inkrementale Messgeräte konfiguriert sind.

#### Funktion

```
EIB7_ERR EIB7StopRef ( EIB7_AXIS axis )
```

#### Parameter

axis                      AXIS-Handle

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_InvInterface              Achse ist nicht für Inkrementale Messgeräte konfiguriert  
EIB7\_NotInitialized           Achse ist nicht konfiguriert  
EIB7\_ParamInvalid              Ungültiger Parameter für Achse

### 8.15 Referenzfahrt Status prüfen

Der Status der Referenzfahrt wird ausgegeben. Hiermit lässt sich prüfen, ob die Referenzfahrt noch aktiv ist, oder bereits alle Referenzpositionen gespeichert wurden.

#### Funktion

```
EIB7_ERR EIB7GetRefActive ( EIB7_AXIS axis,  
                             EIB7_MODE* active  
                             )
```

#### Parameter

axis                      AXIS-Handle  
active                    [Rückgabewert] Pointer auf Variable für den Status

mode	Beschreibung
EIB7_MD_Disable	Referenzfahrt abgeschlossen
EIB7_MD_Enable	Referenzfahrt aktiv

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 8.16 Überwachung der Referenzmarken einstellen

Die Überwachung der Referenzmarke kann für inkrementale Messgeräte aktiviert werden. Zusätzlich ist es möglich einen Toleranzwert anzugeben. Dieser Wert spezifiziert die maximal zulässige Abweichung vom Sollwert der Referenzposition.

#### Funktion

```
EIB7_ERR EIB7SetReferenceCheck ( EIB7_AXIS axis,  
                                     EIB7_MODE mode,  
                                     unsigned long limit  
                                     )
```

#### Parameter

axis                      AXIS-Handle  
mode                      Prüfung der Referenzmarken aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Enable	Prüfung aktivieren
EIB7_MD_Disable	Prüfung deaktivieren

limit                      Maximale Abweichung zwischen zwei Referenzpositionen in Signalperioden

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_InvalidInterface           Achse ist nicht für inkrementale Messgeräte konfiguriert  
EIB7\_ParamInvalid              Parameter ungültig

### 8.17 EnDat 2.1: Position lesen

Die Position eines EnDat Messgeräts wird gelesen. Dies erfolgt über ein EnDat 2.1 Kommando. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat01, EnDat21 oder EnDat22 Messgeräte konfiguriert sein. Die Positionsabfrage wird immer über ein EnDat 2.1 Kommando durchgeführt, auch wenn die Achse für EnDat 2.2 konfiguriert ist.

#### Funktion

```
EIB7_ERR EIB7EnDat21GetPosition ( EIB7_AXIS      axis,
                                  unsigned short * status,
                                  ENCODER_POSITION* pos
                                )
```

#### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für den Positionswert

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)

### 8.18 EnDat 2.1: Speicherbereich wählen

Der Speicherbereich im EnDat Messgerät wird ausgewählt. Dazu wird ein EnDat 2.1 Kommando gesendet. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat01, EnDat21 oder EnDat22 Messgeräte konfiguriert sein. Die Auswahl des Speicherbereichs erfolgt immer über ein EnDat 2.1 Kommando, auch wenn die Achse für EnDat 2.2 konfiguriert ist.

#### Funktion

```
EIB7_ERR EIB7EnDat21SelectMemRange ( EIB7_AXIS      axis,
                                       unsigned char   mrs
                                     )
```

#### Parameter

axis	AXIS-Handle
mrs	MRS Code für den Speicherbereich

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)



### 8.19 EnDat 2.1: Daten senden

Ein Datenwort wird in den Speicher des EnDat Messgeräts geschrieben. Es werden immer 16 Bit Wörter gespeichert. Die Adresse gibt die Speicherzelle innerhalb des aktiven Speicherblocks an. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat01, EnDat21 oder EnDat22 Messgeräte konfiguriert sein. Es wird immer ein EnDat 2.1 Kommando gesendet, auch wenn die Achse für EnDat 2.2 konfiguriert ist.

#### Funktion

```
EIB7_ERR EIB7EnDat21WriteMem ( EIB7_AXIS axis,
                                unsigned char addr,
                                unsigned short data
                                )
```

#### Parameter

axis	AXIS-Handle
addr	Speicheradresse innerhalb des aktiven Speicherblocks
data	Datenwort, das in den Speicher geschrieben wird

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrll	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)

### 8.20 EnDat 2.1: Daten empfangen

Ein Datenwort wird aus dem Speicher des EnDat Messgeräts gelesen. Es wird immer ein 16 Bit Wort gelesen. Der Parameter „addr“ gibt die Speicherzelle innerhalb des aktiven Speicherblocks an, von der die Daten gelesen werden. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat01, EnDat21 oder EnDat22 Messgeräte konfiguriert sein. Es wird immer ein EnDat 2.1 Kommando gesendet, auch wenn die Achse für EnDat 2.2 konfiguriert ist.

#### Funktion

```
EIB7_ERR EIB7EnDat21ReadMem ( EIB7_AXIS axis,
                                unsigned char addr,
                                unsigned short* data
                                )
```

#### Parameter

axis	AXIS-Handle
addr	Speicheradresse innerhalb des aktiven Speicherblocks
data	[Rückgabewert] Pointer auf Variable für das empfangene Datenwort

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrll	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)

## 8.21 EnDat 2.1: Messgerät Reset

Das EnDat Reset-Kommando wird zum Messgerät gesendet. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat01, EnDat21 oder EnDat22 Messgeräte konfiguriert sein. Es wird immer ein EnDat 2.1 Reset gesendet, auch wenn die Achse für EnDat 2.2 konfiguriert ist. Das Messgerät führt einen Reset durch und ist für eine bestimmte Zeit nicht ansprechbar. Weitere Angaben dazu können dem Datenblatt des Messgeräts entnommen werden.

### Funktion

```
EIB7_ERR EIB7EnDat21ResetEncoder ( EIB7_AXIS axis
                                   )
```

### Parameter

axis                      AXIS-Handle

### Rückgabewert

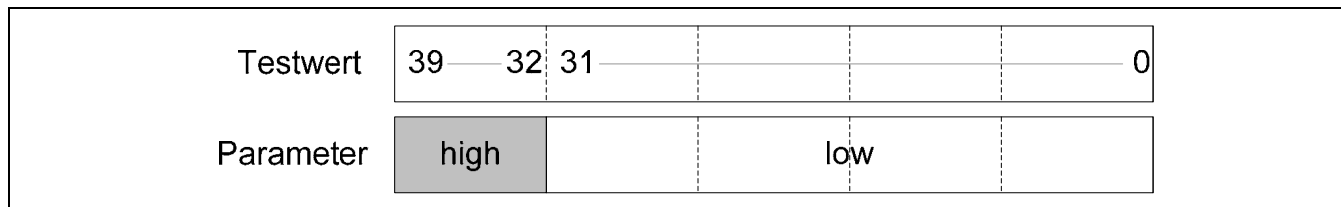
Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrll	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)

## 8.22 EnDat 2.1: Testwert lesen

Ein Testwert wird vom EnDat Messgerät gelesen. Der Testwert ist 40 Bits lang und wird über zwei Parameter zurückgegeben. Der Inhalt der Parameter ist in nachfolgender Tabelle aufgeführt. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat01, EnDat21 oder EnDat22 Messgeräte konfiguriert sein. Es wird immer ein EnDat 2.1 Kommando gesendet, auch wenn die Achse für EnDat 2.2 konfiguriert ist.

Parameter	Parameter-Datenbits	Testwert-Datenbits
low	D0..D31	D0..D31
high	D0..D7 D8..D31	D32..D39 reserviert



**Bild 1: Testwert Datenbits**

### Funktion

```
EIB7_ERR EIB7EnDat21ReadTestValue ( EIB7_AXIS axis,
                                     unsigned long* high,
                                     unsigned long* low
                                     )
```

### Parameter

axis                      AXIS-Handle  
high                      [Rückgabewert] Pointer auf Variable für den Testwert (höchstwertiger Teil)  
low                        [Rückgabewert] Pointer auf Variable für den Testwert (niederwertiger Teil)

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrll	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)

### 8.23 EnDat 2.1: Testbefehl zum Messgerät senden

Ein Testkommando wird zum EnDat Messgerät gesendet. Über den Parameter „port“ kann die Port-Adresse für das Testkommando angegeben werden. Die Achse muss für EnDat21, EnDat21 oder EnDat22 Messgeräte konfiguriert sein. Es wird immer ein EnDat 2.1 Kommando gesendet, auch wenn die Achse für EnDat 2.2 konfiguriert ist.

#### Funktion

```
EIB7_ERR EIB7EnDat21WriteTestCommand ( EIB7_AXIS      axis,
                                       unsigned char    port
                                       )
```

#### Parameter

axis	AXIS-Handle
port	Port-Adresse für das Testkommando

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrll	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)

### 8.24 EnDat 2.2: Position und Zusatzinformation lesen

Die Position eines EnDat22 Messgeräts wird gelesen. Zusätzlich werden die EnDat Zusatzinformationen übertragen, falls diese aktiviert sind. Jede Zusatzinformation besteht aus einem Statuswort und dem Datenwort. Das Statuswort kennzeichnet die Daten als gültig oder ungültig und spezifiziert den Inhalt der Zusatzinformation. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat22 Messgeräte konfiguriert sein.

#### Funktion

```
EIB7_ERR EIB7EnDat22GetPosition ( EIB7_AXIS      axis,
                                  unsigned short*   status,
                                  ENCODER_POSITION* pos,
                                  ENDAT_ADDINFO*    ai1,
                                  ENDAT_ADDINFO*    ai2
                                  )
```

#### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für den Positionswert
ai1	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 1
ai2	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 2

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrll	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert

## 8.25 EnDat 2.2: Position und Zusatzinformation lesen und Speicherbereich auswählen

Die Position und Zusatzinformation eines EnDat22 Messgeräts wird übertragen wie in Kapitel 8.24 beschrieben. Das Messgerät aktiviert den Speicherbereich, der über den MRS Code und die Block-Adresse bestimmt wird. Falls kein Block aus dem Speicherbereich „Sektion 2“ ausgewählt wird, sollte der Parameter „block“ auf Null gesetzt werden. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat22 Messgeräte konfiguriert sein.

### Funktion

```
EIB7_ERR EIB7EnDat22SelectMemRange ( EIB7_AXIS      axis,  
                                     unsigned short*  status,  
                                     ENCODER_POSITION* pos,  
                                     ENDAT_ADDINFO*    ai1,  
                                     ENDAT_ADDINFO*    ai2,  
                                     unsigned char     mrs,  
                                     unsigned char     block  
                                     )
```

### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für den Positionswert
ai1	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 1
ai2	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 2
mrs	MRS Code für den Speicherbereich
block	Block-Adresse für Speicherbereiche der „Sektion 2“

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert

## 8.26 EnDat 2.2: Position und Zusatzinformation lesen und Daten senden

Die Position und Zusatzinformation eines EnDat22 Messgeräts wird übertragen wie in Kapitel 8.24 beschrieben. Das 16 Bit Datenwort wird in den Speicher des Messgeräts geschrieben. Die Adresse (8 Bit) gibt die Speicherzelle innerhalb des ausgewählten Speicherbereichs an. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat22 Messgeräte konfiguriert sein.

### Funktion

```
EIB7_ERR EIB7EnDat22WriteMem ( EIB7_AXIS      axis,  
                               unsigned short*  status,  
                               ENCODER_POSITION* pos,  
                               ENDAT_ADDINFO*    ai1,  
                               ENDAT_ADDINFO*    ai2,  
                               unsigned char     addr,  
                               unsigned short    data  
                               )
```

### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für den Positionswert
ai1	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 1
ai2	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 2
addr	Speicheradresse innerhalb des aktiven Speicherblocks
data	Datenwort, das in den Speicher geschrieben wird

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert

## 8.27 EnDat 2.2: Position und Zusatzinformation lesen und Daten empfangen

Die Position und Zusatzinformation eines EnDat22 Messgeräts wird übertragen wie in Kapitel 8.24 beschrieben. Das Messgerät liest ein Datenwort aus dessen Speicher, wobei die Adresse der Speicherzelle innerhalb des ausgewählten Speicherbereichs über den Parameter „addr“ vorgegeben wird. Die Daten werden über die Zusatzinformation übertragen und können erst mit dem nächsten EnDat Kommando ausgelesen werden. Dazu muss die passende Zusatzinformation ausgewählt werden (siehe EnDat Spezifikation). Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat22 Messgeräte konfiguriert sein.

### Funktion

```
EIB7_ERR EIB7EnDat22ReadMem ( EIB7_AXIS      axis,
                              unsigned short*  status,
                              ENCODER_POSITION* pos,
                              ENDAT_ADDINFO*   ai1,
                              ENDAT_ADDINFO*   ai2,
                              unsigned char     addr
                              )
```

### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für den Positionswert
ai1	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 1
ai2	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 2
addr	Speicheradresse innerhalb des aktiven Speicherblocks

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert

## 8.28 EnDat 2.2: Position und Zusatzinformation lesen und Testkommando senden

Die Position und Zusatzinformation eines EnDat22 Messgeräts wird übertragen wie in Kapitel 8.24 beschrieben. Der Parameter „port“ beinhaltet die Port-Adresse für das Testkommando. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat22 Messgeräte konfiguriert sein.

### Funktion

```
EIB7_ERR EIB7EnDat22WriteTestCommand ( EIB7_AXIS      axis,
                                        unsigned short*  status,
                                        ENCODER_POSITION* pos,
                                        ENDAT_ADDINFO*   ai1,
                                        ENDAT_ADDINFO*   ai2,
                                        unsigned char     port
                                        )
```

### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für das Statuswort
pos	[Rückgabewert] Pointer auf Variable für den Positionswert
ai1	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 1
ai2	[Rückgabewert] Pointer auf Struktur für die EnDat Zusatzinformation 2
port	Port-Adresse für das Testkommando

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert

## 8.29 EnDat 2.2: Position und Zusatzinformation lesen und Fehlerreset senden

Die Position und Zusatzinformation eines EnDat22 Messgeräts wird übertragen wie in Kapitel 8.24 beschrieben. Außerdem wird der Fehlerspeicher des EnDat22 Messgeräts gelöscht. Diese Funktion darf nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat22 Messgeräte konfiguriert sein.

### Funktion

```
EIB7_ERR EIB7EnDat22ErrorReset ( EIB7_AXIS      axis,
                                unsigned short* status,
                                ENCODER_POSITION* pos,
                                ENDAT_ADDINFO*   ai1,
                                ENDAT_ADDINFO*   ai2,
                                )
```

### Parameter

axis	AXIS-Handle
status	<i>[Rückgabewert]</i> Pointer auf Variable für das Statuswort
pos	<i>[Rückgabewert]</i> Pointer auf Variable für den Positionswert
ai1	<i>[Rückgabewert]</i> Pointer auf Struktur für die EnDat Zusatzinformation 1
ai2	<i>[Rückgabewert]</i> Pointer auf Struktur für die EnDat Zusatzinformation 2

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert

### 8.30 EnDat 2.2: Zusatzinformation auswählen

Die Zusatzinformationen für ein EnDat 2.2 Messgerät können konfiguriert werden. Die Konfiguration muss im „Polling“ Modus erfolgen. Die Zusatzinformationen werden in den Modi „Soft Realtime“, „Streaming“ und „Recording“ übertragen.

Die entsprechenden Zusatzinformationen werden im Messgerät ausgewählt, wenn aus dem Modus Polling in einen anderen Modus gewechselt wird. Es ist weiterhin möglich nur die Zusatzinformation 1 oder die Zusatzinformation 2 zu übertragen. Um eine Zusatzinformation zu deaktivieren muss als Parameter EIB7\_AI1\_Stop oder EIB7\_AI2\_Stop übergeben werden.

#### Funktion

```
EIB7_ERR EIB7EnDat22SetAddInfo ( EIB7_AXIS axis,
                                unsigned long addinfo1,
                                unsigned long addinfo2
                                )
```

#### Parameter

axis                      AXIS-Handle  
addinfo1                EnDat 2.2 Zusatzinformation 1

Addinfo1	Wert
EIB7_AI1_NOP	0x00
EIB7_AI1_Diagnostic	0x01
EIB7_AI1_Position2_word1	0x02
EIB7_AI1_Position2_word2	0x03
EIB7_AI1_Position2_word3	0x04
EIB7_AI1_MemoryLSB	0x05
EIB7_AI1_MemoryMSB	0x06
EIB7_AI1_MRS	0x07
EIB7_AI1_TestCommand	0x08
EIB7_AI1_TestValue_word1	0x09
EIB7_AI1_TestValue_word2	0x0A
EIB7_AI1_TestValue_word3	0x0B
EIB7_AI1_Temperature1	0x0C
EIB7_AI1_Temperature2	0x0D
EIB7_AI1_AddSensor	0x0E
EIB7_AI1_Stop	0x0F

addinfo2                EnDat 2.2 Zusatzinformation 2

Addinfo2	Wert
EIB7_AI2_NOP	0x10
EIB7_AI2_Commutation	0x11
EIB7_AI2_Acceleration	0x12
EIB7_AI2_CommAndAccel	0x13
EIB7_AI2_LimitSignal	0x14
EIB7_AI2_LimitAndAccel	0x15
EIB7_AI2_AsyncPos_word1	0x16
EIB7_AI2_AsyncPos_word2	0x17
EIB7_AI2_AsyncPos_word3	0x18
EIB7_AI2_OPSErrorSource	0x19
EIB7_AI2_ReservedA	0x1A
EIB7_AI2_ReservedB	0x1B
EIB7_AI2_ReservedC	0x1C
EIB7_AI2_ReservedD	0x1D
EIB7_AI2_ReservedE	0x1E
EIB7_AI2_Stop	0x1F

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Parameter ungültig
EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert

### 8.31 EnDat 2.2: Sequenz für Zusatzinformationen auswählen

Die Zusatzinformationen für ein EnDat 2.2 Messgerät können konfiguriert werden. Die Konfiguration muss im „Polling“ Modus erfolgen. Die Zusatzinformationen werden in den Modi „Soft Realtime“, „Streaming“ und „Recording“ übertragen.

Die Sequenz der Zusatzinformationen wird mit jedem Trigger weiter geschaltet und beginnt nach dem letzten Eintrag wieder mit dem ersten. Die Sequenz darf maximal 10 Einträge umfassen. Es können Zusatzinformationen 1 und 2 ausgewählt werden.

#### Funktion

```
EIB7_ERR EIB7EnDat22SetAddInfoCycle ( EIB7_AXIS      axis,  
                                       EIB7_MODE      mode,  
                                       unsigned char*  data,  
                                       unsigned long   len  
                                       )
```

#### Parameter

axis                    AXIS-Handle  
mode                    FIFO aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Disable	FIFO für Zusatzinformation deaktivieren
EIB7_MD_Enable	FIFO für Zusatzinformation aktivieren

data                    Pointer auf ein Array mit den Konfigurationsdaten. Jedes Byte enthält eine Zusatzinformation 1 oder 2.

Array-Element	Wert
EIB7_AI1_NOP	0x00
EIB7_AI1_Diagnostic	0x01
EIB7_AI1_Position2_word1	0x02
EIB7_AI1_Position2_word2	0x03
EIB7_AI1_Position2_word3	0x04
EIB7_AI1_MemoryLSB	0x05
EIB7_AI1_MemoryMSB	0x06
EIB7_AI1_MRS	0x07
EIB7_AI1_TestCommand	0x08
EIB7_AI1_TestValue_word1	0x09
EIB7_AI1_TestValue_word2	0x0A
EIB7_AI1_TestValue_word3	0x0B
EIB7_AI1_Temperature1	0x0C
EIB7_AI1_Temperature2	0x0D
EIB7_AI1_AddSensor	0x0E
EIB7_AI1_Stop	0x0F
EIB7_AI2_NOP	0x10
EIB7_AI2_Commutation	0x11
EIB7_AI2_Acceleration	0x12
EIB7_AI2_CommAndAccel	0x13
EIB7_AI2_LimitSignal	0x14
EIB7_AI2_LimitAndAccel	0x15
EIB7_AI2_AsyncPos_word1	0x16
EIB7_AI2_AsyncPos_word2	0x17
EIB7_AI2_AsyncPos_word3	0x18
EIB7_AI2_OPSErrorSource	0x19
EIB7_AI2_ReservedA	0x1A
EIB7_AI2_ReservedB	0x1B
EIB7_AI2_ReservedC	0x1C
EIB7_AI2_ReservedD	0x1D
EIB7_AI2_ReservedE	0x1E
EIB7_AI2_Stop	0x1F

len                    Größe des Array in Bytes (≤ 9)

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Parameter ungültig
EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EnDat22NotSupp	Das Messgerät unterstützt keine EnDat 2.2 Befehle oder die Achse ist nicht für den EnDat 2.2 Betrieb konfiguriert



### 8.32 Absolute und inkrementale Positionswerte simultan auslesen

Die Position eines EnDat Messgeräts wird gelesen. Dafür wird ein EnDat Kommando an das Messgerät gesendet. Gleichzeitig wird der Positionswert aus den Inkrementalsignalen gebildet. Beide Positionswerte werden zusammen mit den Statuswörtern zurückgegeben. Diese Funktion kann nur im Polling Modus ausgeführt werden. Die Achse muss für EnDat 01 Messgeräte konfiguriert sein.

#### Funktion

```
EIB7_ERR EIB7ReadEnDatIncrPos ( EIB7_AXIS      axis,
                                unsigned short*   statusEnDat,
                                ENCODER_POSITION*  posEnDat,
                                unsigned short*   statusIncr,
                                ENCODER_POSITION*  posIncr
                                )
```

#### Parameter

axis	AXIS-Handle
statusEnDat	[Rückgabewert] Pointer auf Variable für das Statuswort der EnDat-Position
posEnDat	[Rückgabewert] Pointer auf Variable für den EnDat-Positionswert
statusIncr	[Rückgabewert] Pointer auf Variable für das Statuswort der Inkrementalposition
posEnDat	[Rückgabewert] Pointer auf Variable für den inkrementalen Positionswert

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_InvInterface	Achse ist nicht für EnDat Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung für Messgerät (Messgerät ist nicht betriebsbereit)
EIB7_EnDatErrII	EnDat Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät ist eventuell nicht angeschlossen)
EIB7_CantLatchPos	Position kann nicht bestimmt werden

### 8.33 Spannungsversorgung für Messgeräte einstellen

Die Spannungsversorgung für das Messgerät kann aktiviert oder deaktiviert werden. Über den Parameter „mode“ wird bestimmt, ob die Spannung ein- oder ausgeschaltet wird.

#### Funktion

```
EIB7_ERR EIB7SetPowerSupply ( EIB7_AXIS      axis,
                               EIB7_MODE      mode
                               )
```

#### Parameter

axis	AXIS-Handle
mode	Spannungsversorgung aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Disable	Spannungsversorgung ausschalten
EIB7_MD_Enable	Spannungsversorgung einschalten

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 8.34 Status der Spannungsversorgung für Messgeräte lesen

Der Status der Spannungsversorgung für das Messgerät kann ausgelesen werden. Über den Parameter „power“ kann bestimmt werden, ob die Spannungsversorgung für diese Achse ein- oder abgeschaltet ist. Der Parameter „err“ gibt an, ob ein Fehler aufgetreten ist, und die Spannungsversorgung aufgrund einer zu hohen Strombelastung abgeschaltet wurde.

#### Funktion

```
EIB7_ERR EIB7GetPowerSupplyStatus ( EIB7_AXIS      axis,  
                                     EIB7_MODE*      power,  
                                     EIB7_POWER_FAILURE* err  
                                     )
```

#### Parameter

axis                      AXIS-Handle  
power                    [Rückgabewert] Pointer auf Variable für den Status der Spannungsversorgung

power	Beschreibung
EIB7_MD_Disable	Spannungsversorgung ausgeschaltet
EIB7_MD_Enable	Spannungsversorgung eingeschaltet

err                      [Rückgabewert] Pointer auf Variable für den Überstromfehler

err	Beschreibung
EIB7_PF_None	kein Fehler
EIB7_PF_Overcurrent	Spannungsversorgung wurde wegen Überstrom deaktiviert

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

### 8.35 Timestamp konfigurieren

Der Timestamp kann für jede Achse aktiviert oder deaktiviert werden. Die Periodendauer wird global für alle Achsen einer EIB 74x eingestellt. Der Timestamp-Wert wird bei einer Positionsabfrage für eine Achse kopiert, wenn diese Funktion zuvor aktiviert wurde.

#### Funktion

```
EIB7_ERR EIB7SetTimestamp ( EIB7_AXIS      axis,  
                             EIB7_MODE      mode  
                             )
```

#### Parameter

axis                      AXIS-Handle  
mode                      Timestamp aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Disable	Timestamp deaktivieren
EIB7_MD_Enable	Timestamp aktivieren

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard Rückgabewerten aufgelistet.

## 9 IO-Funktionen

Die IO-Funktionen beziehen sich immer nur auf einen einzelnen Ausgangs- oder Eingangsport der EIB 74x. Alle anderen Ports werden nicht beeinflusst.

Alle IO-Funktionen können als Rückgabewert die nachfolgend aufgeführten Fehlermeldungen liefern. Zusätzlich dazu können sie individuell weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

### Standard Rückgabewerte

EIB7_NoError	Funktionsaufruf erfolgreich
EIB7_InvalidHandle	Das Handle auf die Achse der EIB 74x ist ungültig
EIB7_FuncNotSupp	Funktion wird von der EIB 74x nicht unterstützt
EIB7_InvalidResponse	Fehler bei der Datenübertragung
EIB7_AccNotAllowed	Funktion kann nicht ausgeführt werden, da die EIB 74x den Zugriff nicht erlaubt
EIB7_ConnReset	Verbindung wurde von der EIB 74x beendet
EIB7_ConnTimeout	Timeout bei der Datenübertragung zur EIB 74x
EIB7_ReceiveError	Fehler beim Empfangen der Daten
EIB7_SendError	Fehler beim Senden der Daten
EIB7_OutOfMemory	Vom System kann nicht genügend Speicher allokiert werden
EIB7_PortDirInv	Signalrichtung des Ports falsch

### 9.1 Eingangsport konfigurieren

Mit dieser Funktion kann der Modus eines Eingangsports konfiguriert werden. Der Port kann als Triggereingang oder als logischer Eingang verwendet werden. Außerdem lässt sich der Abschlusswiderstand des differenziellen Eingangs zu- oder abschalten. Die Funktion ist nur in Verbindung mit Handles auf Eingangsports zulässig.

#### Funktion

```
EIB7_ERR EIB7InitInput      (  EIB7_IO      io,  
                               EIB7_IOMODE   mode,  
                               EIB7_MODE     termination  
                               )
```

#### Parameter

io	IO-Handle
mode	Triggereingang oder logischer Eingangsport

mode	Beschreibung
EIB7_IOM_Trigger	Triggereingang
EIB7_IOM_Logical	Logik-Eingang

termination	Abschlusswiderstand aktivieren oder deaktivieren
-------------	--

termination	Beschreibung
EIB7_MD_Disable	Abschlusswiderstand deaktivieren
EIB7_MD_Enable	Abschlusswiderstand aktivieren

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Parameter ungültig
-------------------	--------------------

## 9.2 Ausgangsport konfigurieren

Mit dieser Funktion kann der Modus eines Ausgangsports konfiguriert werden. Der Port kann als Triggerausgang oder als logischer Ausgang verwendet werden. Zudem lässt sich der Ausgangstreiber deaktivieren. In diesem Fall befindet sich der Ausgang in einem hochohmigen Zustand. Die Funktion ist nur in Verbindung mit Handles auf Ausgangsports zulässig.

### Funktion

```
EIB7_ERR EIB7InitOutput      ( EIB7_IO      io,
                               EIB7_IOMODE   mode,
                               EIB7_MODE     enable
                               )
```

### Parameter

io                    IO-Handle  
mode                Triggerausgang oder logischer Ausgangsport

mode	Beschreibung
EIB7_IOM_Trigger	Triggerausgang
EIB7_IOM_Logical	Logik-Ausgang

enable              Ausgangstreiber aktivieren oder deaktivieren

enable	Beschreibung
EIB7_MD_Disable	Ausgangstreiber deaktivieren
EIB7_MD_Enable	Ausgangstreiber aktivieren

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid      Parameter ungültig

## 9.3 Triggerquelle für Triggerausgang wählen

Das Triggersignal für den Triggerausgang kann aus verschiedenen Quellen gewählt werden. Diese Einstellung ist nur im Betriebsmodus „Polling“ und für Triggerausgänge möglich.

### Funktion

```
EIB7_ERR EIB7OutputTriggerSource ( EIB7_IO      io,
                                   EIB7_OutputTriggerSrc src
                                   )
```

### Parameter

io                    IO-Handle  
src                  Triggerquelle

src	Beschreibung
EIB7_OT_TrgInSync	Triggereingang synchronisiert
EIB7_OT_TrgInAsync	Triggereingang nicht synchronisiert
EIB7_OT_TrgSW1	Software-Trigger Kanal 1
EIB7_OT_TrgSW2	Software-Trigger Kanal 2
EIB7_OT_TrgSW3	Software-Trigger Kanal 3
EIB7_OT_TrgSW4	Software-Trigger Kanal 4
EIB7_OT_TrgRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_OT_TrgIC	Interval Counter
EIB7_OT_TrgPuls	Trigger Puls Zähler
EIB7_OT_TrgTimer	Timer Trigger

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid      Parameter ungültig

## 9.4 Verzögerungszeit für Triggereingang einstellen

Für jeden Triggereingang kann separat eine Zeit eingestellt werden, um die ein Triggerimpuls verzögert wird. Die Verzögerungszeit muss als ganzzahliges Vielfaches der Taktperiode angegeben werden, wobei die Anzahl der Taktperioden pro Mikrosekunde auslesbar ist (EIB7GetTriggerDelayTicks()). Mit dem Parameterwert Null kann die Verzögerung deaktiviert werden. Diese Funktion ist nur für Trigger-Eingänge gültig.

### Funktion

```
EIB7_ERR EIB7SetTriggerInputDelay ( EIB7_IO      io,
                                   unsigned long  dly
                                   )
```

### Parameter

io	IO-Handle (nur für Eingänge)
dly	Verzögerungszeit in Taktzyklen (<= 256); siehe auch Hinweis im Kapitel 4

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Parameter ungültig
-------------------	--------------------

## 9.5 Logischen Port auslesen

Der Pegel an einem logischen Eingang oder Ausgang wird ausgelesen (Parameter „level“). Zusätzlich wird der Betriebsmodus des Ports bestimmt. Falls der Port als Triggereingang oder Triggerausgang betrieben wird, ist der Wert von „level“ ungültig. Bei einem Logik-Ausgang wird der eingestellte Pegel zurückgelesen.

### Funktion

```
EIB7_ERR EIB7ReadIO ( EIB7_IO      io,
                     EIB7_IOMODE*  mode,
                     unsigned long* level
                     )
```

### Parameter

io	IO-Handle
mode	[Rückgabewert] Pointer auf Variable für den Betriebsmodus

mode	Beschreibung
EIB7_IOM_Trigger	Triggerport
EIB7_IOM_Logical	Logik-Port

level	[Rückgabewert] Pointer auf Variable für den logischen Pegel des Ports
-------	---

level	Beschreibung
0	low Pegel
1	high Pegel

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_ParamInvalid	Parameter ungültig
-------------------	--------------------

## 9.6 Logischen Ausgangsport setzen

Der Pegel eines logischen Ausgangsports wird eingestellt. Der Parameter „level“ gibt an, ob der Ausgang auf high oder low gesetzt wird. Diese Funktion kann nur auf Ausgänge angewendet werden, die für den Logik-Modus konfiguriert wurden. Wird der Port als Triggerausgang eingesetzt, erzeugt die Funktion eine Fehlermeldung.

### Funktion

```
EIB7_ERR EIB7WriteIO          ( EIB7_IO          io,  
                                unsigned long    level  
                                )
```

### Parameter

io                    IO-Handle  
level                Logischen Pegel des Ausgangs

level	Beschreibung
0	low Pegel
1	high Pegel

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid        Parameter ungültig  
EIB7\_TrgNotConf        Ausgang ist kein Logik-Port

## 9.7 Konfigurationsdaten für Eingang lesen

Die Konfigurationsdaten für einen Eingangsport werden ausgelesen. Der Parameter „mode“ liefert den Betriebsmodus des Eingangs. In „termination“ wird der Zustand des Abschlusswiderstands zurückgegeben. Die Funktion darf nur für Eingangsports verwendet werden.

### Funktion

```
EIB7_ERR EIB7GetInputConfig   ( EIB7_IO          io,  
                                EIB7_IOMODE*      mode,  
                                EIB7_MODE*        termination  
                                )
```

### Parameter

io                    IO-Handle  
mode                *[Rückgabewert]* Pointer auf Variable für den Betriebsmodus

mode	Beschreibung
EIB7_IOM_Trigger	Triggereingang
EIB7_IOM_Logical	Logik-Eingang

termination        *[Rückgabewert]* Pointer auf Variable für den Abschlusswiderstand

termination	Beschreibung
EIB7_MD_Disable	Abschlusswiderstand deaktiviert
EIB7_MD_Enable	Abschlusswiderstand aktiviert

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid        Parameter ungültig

## 9.8 Konfigurationsdaten für Ausgang lesen

Die Konfigurationsdaten für einen Ausgangsport werden gelesen. Der Parameter „mode“ liefert den Betriebsmodus des Ausgangs. In „enable“ wird der Zustand des Ausgangstreibers zurückgegeben. Die Funktion darf nur für Ausgangsports verwendet werden.

### Funktion

```
EIB7_ERR EIB7GetOutputConfig      (  EIB7_IO          io,  
                                     EIB7_IOMODE*       mode,  
                                     EIB7_MODE*        enable  
                                     )
```

### Parameter

io                                    IO-Handle  
mode                                *[Rückgabewert]* Pointer auf Variable für den Betriebsmodus

mode	Beschreibung
EIB7_IOM_Trigger	Triggerausgang
EIB7_IOM_Logical	Logik-Ausgang

enable                              *[Rückgabewert]* Pointer auf Variable für den Status des Ausgangstreibers

enable	Beschreibung
EIB7_MD_Disable	Ausgangstreiber deaktiviert
EIB7_MD_Enable	Ausgangstreiber aktiviert

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7\_ParamInvalid                Parameter ungültig

## 10 Allgemeine Funktionen

Alle allgemeinen Funktionen können als Rückgabewert die nachfolgend aufgeführten Fehlermeldungen liefern. Zusätzlich dazu können sie individuell weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

### Standard Rückgabewerte

EIB7_NoError	Funktionsaufruf erfolgreich
EIB7_OutOfMemory	Vom System kann nicht genügend Speicher allokiert werden

### 10.1 Treiber ID-Nummer lesen

Die Produktnummer (ID) des Treibers wird als C-String ausgegeben. Der String wird auf den Pointer „ident“ gespeichert. Über den Parameter „len“ muss die Größe des Speichers für den String in Bytes angegeben werden. Falls der String inklusive dem abschließenden Nullbyte länger als der Speicherbereich ist, wird eine Fehlermeldung generiert. Der Zielspeicher sollte mindestens 9 Bytes groß sein.

#### Funktion

```
EIB7_ERR EIB7GetDriverID      ( char*      ident,
                               unsigned long  len
                               )
```

#### Parameter

ident	<i>[Rückgabewert]</i> Zielspeicher für den C-String
len	Größe des Zielspeichers in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_BufferTooSmall	Zielspeicher zu klein
---------------------	-----------------------

### 10.2 Fehlermeldung in Text umwandeln

Ein Fehlercode wird in eine Textmeldung umgewandelt und als C-String zurückgegeben. Im System sind für alle bekannten Fehlercodes ein beschreibender Text und eine Kurzbezeichnung definiert. Über den Parameter „mnemonic“ wird eine Kurzbeschreibung der Fehlermeldung in Textform zurückgegeben (ca. 30-40 Zeichen). Der Parameter „message“ enthält eine ausführlichere Beschreibung (ca. 100-150 Zeichen). Wird für einen der Parameter „mnemonic“ oder „message“ ein NULL-Pointer übergeben, so kopiert die Funktion den betreffenden Text nicht. Falls der Zielspeicher zu klein ist, um den gesamten Text aufzunehmen, wird nur der erste Teil kopiert. Der String wird immer mit einem Null-Byte abgeschlossen.

#### Funktion

```
EIB7_ERR EIB7GetErrorInfo    ( EIB7_ERR    code,
                               char*        mnemonic,
                               unsigned long mnemlen,
                               char*        message,
                               unsigned long msglen
                               )
```

#### Parameter

code	Fehlercode, der in Text umgewandelt wird
mnemonic	<i>[Rückgabewert]</i> Pointer auf den Zielspeicher für die Kurzbeschreibung
mnemlen	Größe des Zielspeichers „mnemonic“ in Bytes
message	<i>[Rückgabewert]</i> Pointer auf den Zielspeicher für den Fehlertext
msglen	Größe des Zielspeichers „message“ in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard Rückgabewerten können die nachfolgend aufgelisteten Fehlermeldungen auftreten.

EIB7_IllegalParameter	Ungültiger Errorcode
-----------------------	----------------------



# HEIDENHAIN

---

## **DR. JOHANNES HEIDENHAIN GmbH**

Dr.-Johannes-Heidenhain-Straße 5

**83301 Traunreut, Germany**

☎ +49 8669 31-0

**FAX** +49 8669 5061

E-mail: [info@heidenhain.de](mailto:info@heidenhain.de)

---

**Technical support** **FAX** +49 8669 32-1000

**Measuring systems** ☎ +49 8669 31-3104

E-mail: [service.ms-support@heidenhain.de](mailto:service.ms-support@heidenhain.de)

**TNC support** ☎ +49 8669 31-3101

E-mail: [service.nc-support@heidenhain.de](mailto:service.nc-support@heidenhain.de)

**NC programming** ☎ +49 8669 31-3103

E-mail: [service.nc-pgm@heidenhain.de](mailto:service.nc-pgm@heidenhain.de)

**PLC programming** ☎ +49 8669 31-3102

E-mail: [service.plc@heidenhain.de](mailto:service.plc@heidenhain.de)

**Lathe controls** ☎ +49 8669 31-3105

E-mail: [service.lathe-support@heidenhain.de](mailto:service.lathe-support@heidenhain.de)

---

**[www.heidenhain.de](http://www.heidenhain.de)**