



# HEIDENHAIN



## EIB 74x

Benutzerhandbuch für  
Applikationsentwicklung

Deutsch (de)  
12/2019

## Inhaltsverzeichnis

<b>1</b>	<b>Grundlegendes.....</b>	<b>7</b>
<b>2</b>	<b>Funktionsumfang.....</b>	<b>11</b>
<b>3</b>	<b>Treibersoftware.....</b>	<b>45</b>

<b>1</b>	<b>Grundlegendes</b>	<b>7</b>
1.1	Dokumentation	8
1.2	Zielgruppe und Qualifikation des Personals	9
1.3	Firmware-Version	9
1.4	Change History	9
<b>2</b>	<b>Funktionsumfang</b>	<b>11</b>
2.1	Allgemeine Funktionsbeschreibung	12
2.2	Konfiguration der Messgeräteingänge	13
2.2.1	Verarbeitung von Inkrementalsignalen	14
2.2.2	Behandlung von Referenzmarken	16
2.2.3	Überwachung von Referenzmarken	18
2.2.4	Verarbeitung von EnDat-Signalen	19
2.2.5	Hilfsachse	23
2.3	Verarbeitung von Trigger-Ereignissen	24
2.3.1	Trigger-Eingänge und -Ausgänge	24
2.3.2	Logische Eingänge und Ausgänge	26
2.3.3	Trigger-Modul	27
2.3.4	Interval Counter	28
2.3.5	Maximale Trigger-Rate	29
2.3.6	Zähler für akzeptierte Trigger-Ereignisse	29
2.4	Timestamp	30
2.5	Statuswort	31
2.6	Ethernet-Interface	34
2.7	Betriebsmodi	34
2.7.1	Konfiguration der Datenpakete	35
2.7.2	Betriebsmodus "Polling"	38
2.7.3	Betriebsmodus "Soft Realtime"	39
2.7.4	Betriebsmodus "Streaming"	40
2.7.5	Betriebsmodus "Recording"	41
2.8	Firmware-Update	42
2.8.1	Firmware-Update durchführen	42
2.8.2	TFTP-Client aktivieren	43
2.9	Reset	44

<b>3</b>	<b>Treibersoftware.....</b>	<b>45</b>
<b>3.1</b>	<b>Allgemeine Information.....</b>	<b>46</b>
<b>3.2</b>	<b>Installationsanleitung.....</b>	<b>46</b>
3.2.1	Installation unter Microsoft Windows.....	46
3.2.2	Installation unter Linux.....	47
<b>3.3</b>	<b>Überblick.....</b>	<b>47</b>
3.3.1	Kommunikationsaufbau.....	47
3.3.2	Betriebsmodus "Polling".....	47
3.3.3	Betriebsmodus "Soft Realtime".....	48
3.3.4	Betriebsmodus "Streaming".....	48
3.3.5	Betriebsmodus "Recording".....	49
3.3.6	Datentypen und Datenpaket.....	49
3.3.7	Parameter und Rückgabewerte.....	50
<b>3.4</b>	<b>Hilfsfunktionen.....</b>	<b>51</b>
3.4.1	IP-Adresse bestimmen.....	51
3.4.2	Positionsdatenformat ändern.....	51
<b>3.5</b>	<b>Gerätefunktionen.....</b>	<b>52</b>
3.5.1	Verbindung zur EIB 74x öffnen.....	52
3.5.2	Verbindung zur EIB 74x schließen.....	53
3.5.3	Status der Verbindung abfragen.....	53
3.5.4	Timeout einstellen.....	54
3.5.5	Anzahl der Achsen auslesen.....	54
3.5.6	Handle für Achse anfordern.....	55
3.5.7	IO-Port-Handle anfordern.....	56
3.5.8	Datenpaket erstellen.....	56
3.5.9	Datenpaket konfigurieren.....	58
3.5.10	Betriebsmodus wählen.....	59
3.5.11	Netzwerkparameter speichern.....	60
3.5.12	Netzwerkparameter auslesen.....	61
3.5.13	Hostname speichern.....	61
3.5.14	Hostname auslesen.....	62
3.5.15	Seriennummer auslesen.....	62
3.5.16	Geräte-Identnummer auslesen.....	63
3.5.17	MAC-Adresse auslesen.....	63
3.5.18	Firmware-Versionsnummer auslesen.....	64
3.5.19	Bootmodus auslesen.....	64
3.5.20	Update-Status auslesen.....	65
3.5.21	Anzahl der offenen Verbindungen auslesen.....	66
3.5.22	Verbindungsdaten auslesen.....	66
3.5.23	Verbindung abbrechen.....	67
3.5.24	Timestamp-Zeiteinheit lesen.....	67
3.5.25	Timestamp-Periodendauer einstellen.....	68

3.5.26	Timestamp-Zähler zurücksetzen.....	69
3.5.27	Timer Trigger - Zeiteinheit lesen.....	69
3.5.28	Timer Trigger - Periodendauer einstellen.....	69
3.5.29	Zeiteinheit für Verzögerung an Trigger-Eingängen lesen.....	70
3.5.30	Trigger Counter löschen.....	70
3.5.31	Software Trigger.....	71
3.5.32	Master-Trigger-Quelle wählen.....	71
3.5.33	Trigger-Quellen aktivieren.....	72
3.5.34	Pulszähler konfigurieren.....	74
3.5.35	Interpolationsfaktor für Interval Counter einstellen.....	76
3.5.36	Interval Counter konfigurieren.....	77
3.5.37	Abschlusswiderstände einstellen.....	78
3.5.38	Reset.....	78
3.5.39	EIB 74x identifizieren.....	79
3.5.40	Recording - Daten übertragen.....	80
3.5.41	Recording - Status prüfen.....	81
3.5.42	Recording - Speichergröße lesen.....	82
3.5.43	Streaming - Status prüfen.....	83
3.5.44	Daten aus FIFO lesen.....	83
3.5.45	Größe eines FIFO-Elements lesen.....	84
3.5.46	Zugriff auf Inhalt eines FIFO-Elements.....	85
3.5.47	Daten aus FIFO lesen und konvertieren.....	87
3.5.48	Größe eines FIFO-Elements nach Konvertierung lesen.....	88
3.5.49	Zugriff auf Inhalt eines FIFO-Elements mit konvertierten Daten.....	88
3.5.50	Anzahl der Elemente im FIFO lesen.....	90
3.5.51	FIFO löschen.....	90
3.5.52	FIFO-Größe einstellen.....	90
3.5.53	FIFO-Größe auslesen.....	91
3.5.54	Callback-Mechanismus aktivieren.....	91
3.5.55	Trigger-Quelle für Hilfsachse wählen.....	92
3.5.56	Position der Hilfsachse abfragen.....	93
3.5.57	Daten der Hilfsachse auslesen.....	94
3.5.58	Zähler der Hilfsachse löschen.....	95
3.5.59	Signalfehler der Hilfsachse quittieren.....	95
3.5.60	Trigger-Fehler der Hilfsachse quittieren.....	95
3.5.61	Statusbit für Referenzmarke der Hilfsachse löschen.....	96
3.5.62	Status der Referenzfahrt für Hilfsachse prüfen.....	96
3.5.63	Referenzfahrt für Hilfsachse starten.....	97
3.5.64	Referenzfahrt für Hilfsachse stoppen.....	97
3.5.65	Timestamp für Hilfsachse konfigurieren.....	97
3.5.66	Trigger-Flanke für Referenzimpuls der Hilfsachse einstellen.....	98

**3.6 Achsfunktionen..... 99**

3.6.1	Achse initialisieren.....	99
3.6.2	Trigger-Quelle für Achse wählen.....	103
3.6.3	Trigger-Flanke für Referenzimpuls einstellen.....	103

3.6.4	Zähler löschen.....	104
3.6.5	Position abfragen.....	104
3.6.6	Daten für einen Kanal auslesen.....	105
3.6.7	Spannungsversorgungsfehler quittieren.....	106
3.6.8	Trigger-Fehler quittieren.....	107
3.6.9	Signalfehler quittieren.....	107
3.6.10	EnDat-Fehlerbits löschen.....	108
3.6.11	Statusbits für Referenzmarken löschen.....	108
3.6.12	Statusbits für abstandscodierte Referenzmarken löschen.....	109
3.6.13	Referenzfahrt starten.....	110
3.6.14	Referenzfahrt stoppen.....	111
3.6.15	Referenzfahrt Status prüfen.....	111
3.6.16	Überwachung der Referenzmarken einstellen.....	112
3.6.17	EnDat 2.1 - Position lesen.....	112
3.6.18	EnDat 2.1 - Speicherbereich wählen.....	113
3.6.19	EnDat 2.1 - Daten senden.....	114
3.6.20	EnDat 2.1 - Daten empfangen.....	114
3.6.21	EnDat 2.1 - Messgerät Reset.....	115
3.6.22	EnDat 2.1 - Testwert lesen.....	116
3.6.23	EnDat 2.1 - Testbefehl zum Messgerät senden.....	116
3.6.24	EnDat 2.2 - Position und Zusatzinformation lesen.....	117
3.6.25	EnDat 2.2 - Position und Zusatzinformation lesen und Speicherbereich wählen.....	118
3.6.26	EnDat 2.2 - Position und Zusatzinformation lesen und Daten senden.....	119
3.6.27	EnDat 2.2 - Position und Zusatzinformation lesen und Daten senden.....	120
3.6.28	EnDat 2.2 - Position und Zusatzinformation lesen und Testkommando senden.....	121
3.6.29	EnDat 2.2 - Position und Zusatzinformation lesen und Fehler-Reset senden.....	122
3.6.30	EnDat 2.2 - Zusatzinformation wählen.....	123
3.6.31	EnDat 2.2 - Sequenz für Zusatzinformation wählen.....	126
3.6.32	Absolute und inkrementale Positionswerte simultan auslesen.....	128
3.6.33	Spannungsversorgung für Messgeräte einstellen.....	129
3.6.34	Status der Spannungsversorgung für Messgeräte lesen.....	129
3.6.35	Timestamp konfigurieren.....	130

### **3.7 IO-Funktionen..... 131**

3.7.1	Eingangsport konfigurieren.....	131
3.7.2	Ausgangsport konfigurieren.....	132
3.7.3	Trigger-Quelle für Trigger-Ausgang wählen.....	132
3.7.4	Verzögerungszeit für Trigger-Eingang einstellen.....	133
3.7.5	Logischen Port auslesen.....	134
3.7.6	Logischen Ausgangsport setzen.....	134
3.7.7	Konfigurationsdaten für Eingang lesen.....	135
3.7.8	Konfigurationsdaten für Ausgang lesen.....	136

### **3.8 Allgemeine Funktionen..... 137**

3.8.1	Treiber ID-Nummer lesen.....	137
3.8.2	Fehlermeldung in Text umwandeln.....	138

# 1

**Grundlegendes**

## 1.1 Dokumentation

Die Dokumentation zur EIB 741, EIB 742 und EIB 749, im Folgenden als EIB 74x bezeichnet, umfasst folgende Unterlagen:

- Betriebsanleitung
  - Unterlagen, die für die Inbetriebnahme erforderlich sind, sowie Technische Daten
- Benutzerhandbuch für die Applikationsentwicklung
  - Beschreibung des Funktionsumfangs der EIB 74x
  - Beschreibung der Installation und der Funktionsaufrufe der Treiber-Software

### DHCP

Die EIB 74x kann mit statischen IP-Adressen oder alternativ mit dynamischen IP-Adressen, die von einem DHCP-Server bezogen werden, arbeiten. Per Default ist DHCP deaktiviert und die EIB 74x benutzt statische IP-Adressen. Diese Adresse kann durch den Benutzer gesetzt werden, um sich an die Gegebenheiten eines bestimmten Netzwerks anzupassen.

Wenn DHCP aktiviert wird, versucht die EIB 74x nach der Bootphase eine IP-Adresse von einem DHCP-Server zu beziehen. Diese Adresse wird so lange benutzt, wie in der Gültigkeitsdauer der "Lease" angezeigt wird. Wenn nötig erneuert die EIB 74x den "Lease" selbstständig. Wenn kein DHCP-Server gefunden wird, der eine Adresse zur Verfügung stellt, verwendet die EIB 74x nach Ablauf eines Timeouts die voreingestellte IP-Adresse. Die Bootphase verlängert sich in dem Fall, dass DHCP angewählt ist, aber kein DHCP-Server zur Verfügung steht.

Der DHCP-Client fordert eine IP-Adresse, die Subnetzmaske und den Standard-Gateway an. Zusätzlich wird der Hostname der EIB 74x an den DHCP-Server übermittelt. Wenn der DHCP-Server mit einem DNS-Server verbunden ist, dann kann der Hostname anstatt der IP-Adresse zur Kommunikation mit der EIB 74x verwendet werden.

Der Default-Hostname ist individuell für jede EIB 74x und enthält den Gerätenamen und die eindeutige Seriennummer. Anbei ein Beispiel für den Hostname:

EIB741-SN123456

Der Gerätename ist "EIB741" und die Seriennummer ist "SN123456". Die Seriennummer ist auf dem Typenschild auf der Rückseite der EIB 74x aufgedruckt. Der Hostname kann über ein Software-Kommando geändert werden.



Für die Änderung der Netzwerkeinstellung kann z. B. das auf CD mitgelieferte Programm "networksettings" verwendet werden (zu finden unter **...\windows\tools\networksettings\networksettings.exe**).



Um die störende Einflussnahme weiterer (für die Applikation nicht benötigter) Netzwerkteilnehmer auszuschließen, empfiehlt HEIDENHAIN für die Anbindung der EIB 74x ein separates Netzwerk aufzubauen.



## 1.2 Zielgruppe und Qualifikation des Personals

Die vorliegende Unterlage muss von jeder Person gelesen und beachtet werden, die mit der Applikationsentwicklung betraut ist.

Für die Applikationsentwicklung ist eine entsprechende Qualifikation erforderlich. Der Applikationsentwickler muss sich mit Hilfe der Dokumentation des Geräts und der angeschlossenen Peripherie ausreichend informiert haben.

Der Applikationsentwickler ist aufgrund seiner fachlichen Ausbildung, Kenntnisse und Erfahrung sowie Kenntnis der einschlägigen Bestimmungen in der Lage, die ihm übertragenen Arbeiten hinsichtlich der jeweiligen Applikation auszuführen und mögliche Gefahren selbstständig zu erkennen und zu vermeiden.

## 1.3 Firmware-Version

Die vorliegende Dokumentation beschreibt die Firmware-Version 633281-14.

## 1.4 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.



# 2

**Funktionsumfang**

## 2.1 Allgemeine Funktionsbeschreibung

Die EIB 74x ist eine Auswerte-Elektronik 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  $V_{SS}$ ) oder mit EnDat-Schnittstellen (EnDat 2.1 und EnDat 2.2) angeschlossen werden.

Zur Messwertbildung unterteilt die EIB 74x die Signalperioden der Inkrementalsignale 4096-fach. 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 ist 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.

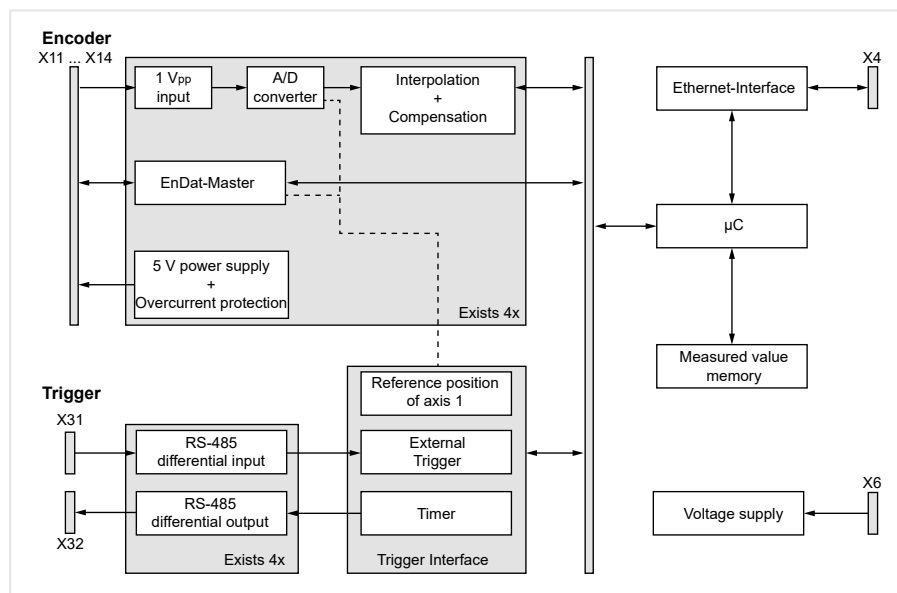


Abbildung 1: Prinzipschaltbild

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

- Inkrementalsignale 1  $V_{SS}$
- EnDat 2.1
- EnDat 2.2
- Inkrementalsignale 11  $\mu A_{SS}$  (auf Anfrage)

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

 Technische Daten siehe "Betriebsanleitung".

## 2.2 Konfiguration der Messgeräteingänge

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

- Schnittstellentyp
- Bandbreite für die 1  $V_{SS}$ -Eingangssignale
- Signalkompensation
- Verarbeitung der Referenzmarken
- Verarbeitung der Homing-/Limit-Signale

Diese Einstellungen können per Software geändert werden.

Die Schnittstelle für den Messgeräteingang 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  $V_{SS}$ -Schnittstelle unterstützt wird.

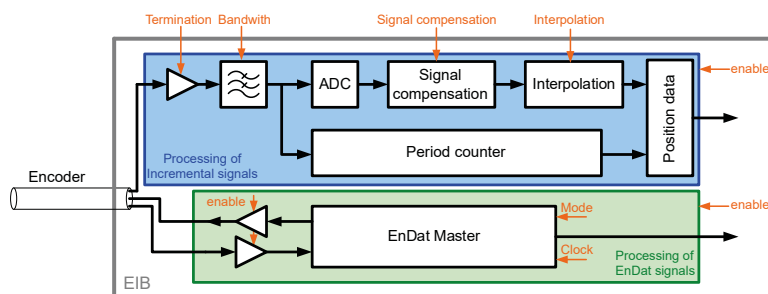


Abbildung 2: Messgeräteingang

### 2.2.1 Verarbeitung von Inkrementalsignalen

Zur Bildung des Positionswerts unterteilt die EIB 74x die Signalperioden der Inkrementalsignale 4096-fach (12 Bit). Der Periodenzähler hat eine Breite von 32 Bit. Der Zählwert wird mit jeder Signalperiode des angeschlossenen Messgeräts 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 Messgeräts 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 Trigger-Ereignisses. Die Positionsinformation wird in einem 48 Bit breiten Register gespeichert (siehe Tabelle). Der Periodenzähler wird dabei in der Zweierkomplementdarstellung abgebildet; die Bits 43 ... 47 bilden das Vorzeichen ab.

Die übergeordnete Kunden-Softwareapplikation kann aus diesem Wert abhängig von der Art des Messgeräts (linear oder rotativ) den entsprechenden Winkel oder die Länge berechnen. Der Überlauf des Periodenzählers erfolgt entsprechend der Zweierkomplementdarstellung 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.

#### Registerinhalt Positionswert bei Inkrementalsignalen

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

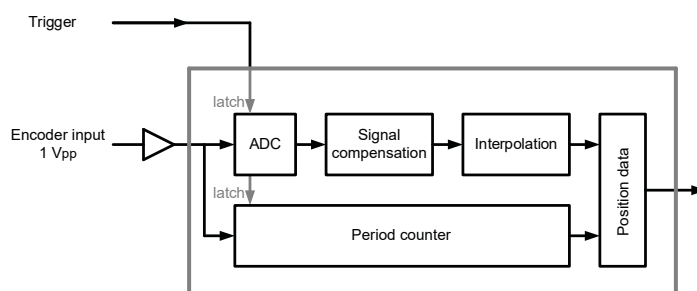


Abbildung 3: Blockschaltbild Messgeräteingang

**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:

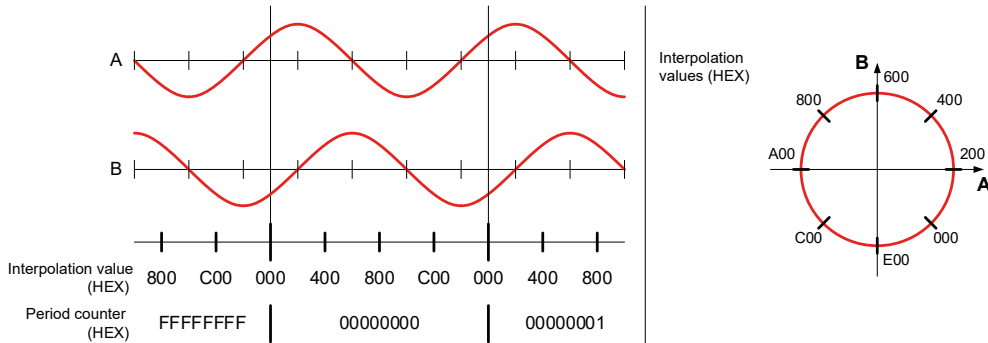


Abbildung 4: Interpolationswert bei Inkrementalsignalen

Einstellmöglichkeiten:

- Abschlusswiderstand für die Inkrementalsignale  
Der Abschlusswiderstand 120 Ω für die 1 V<sub>SS</sub>-Inkrementalsignale kann per Software zu- oder abgeschaltet werden (für alle Kanäle gleichzeitig; Default: Widerstände zugeschaltet)
- Bandbreiten-Einstellung der Inkrementalsignale  
Die Bandbreite der Inkrementalsignale des Messgeräts kann per Software umgeschaltet werden. Als Standardeinstellung 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 Messgeräts kann per Software ein- oder ausgeschaltet werden

**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.

**Registerinhalt AD-Wandler bei Inkrementalsignalen**

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

**Wertetabelle AD-Wandler bei Inkrementalsignalen**

Wert (HEX)	Signalwert Inkrementalsignale
000	Negatives Maximum
800	Null
FFF	Positives Maximum

## 2.2.2 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.

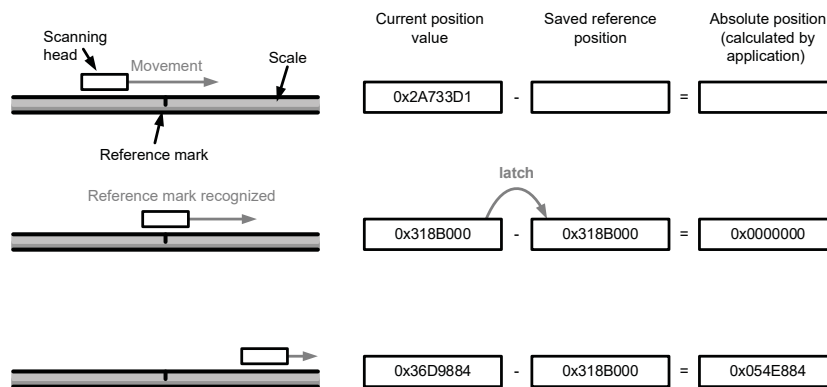


Abbildung 5: Ermittlung einer Referenzposition

### 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 Positionsdatenpaket 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 Periodenzählerwerts 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 Positionswerts durch die Kunden-Softwareapplikation genauso behandelt wie ein gespeicherter Referenzpositionswert 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 Referenzwerts:

- Methode 1 (empfohlene Methode):  
Die Achse wird als inkrementelles System mit abstandscodierten Referenzmarken initialisiert. Dabei werden weitere typabhängige 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 "verlorengehen", 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.

### 2.2.3 Überwachung von 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.

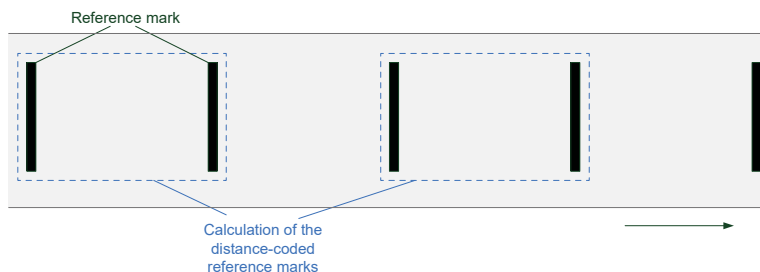


Abbildung 6: Fortlaufende Berechnung der Referenzposition mit abstandscodierten Referenzmarken

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. Wenn die gleiche Referenzmarke vor und nach einem Richtungswechsel zweimal überfahren wird, 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. Folgende Reihenfolge wird empfohlen:

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

### 2.2.4 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  $V_{SS}$ -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  $V_{SS}$ -Inkrementalsignalen verarbeiten.

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

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

Anmerkungen zu EnDat 01:

- Bei gleichzeitiger Verwendung von EnDat-Positionsanfragen und 1  $V_{SS}$ -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

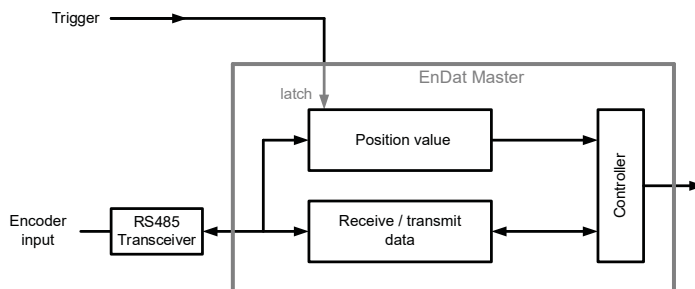


Abbildung 7: Blockschaltbild EnDat Master

#### 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 Messgeräts.

#### Registerinhalt Positionswert

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

### 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.

#### Taktfrequenz EnDat Master

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 je nach Kabellänge

Taktfrequenz	Kabellänge in Meter (mit HEIDENHAIN-Kabeln)	
	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 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 Recovery time I "lang" verwendet.

Anmerkungen:

- 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 Trigger-Raten kann sich die Fehlermeldung um mehrere Samples verschieben.

Die Calculation time kann abhängig vom angeschlossenen Messgerät eingestellt werden. Hierbei gibt es die Optionen "lang" ( $< 1 \text{ ms}$ ) und "kurz" ( $< 15 \mu\text{s}$ ).

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:

- Keine Zusatzinformation  
Mit jedem Trigger-Ereignis wird eine Positionsabfrage gestartet. Dabei werden keine Zusatzinformationen übertragen.
- Feste Zusatzinformationen  
Mit jedem Trigger-Ereignis 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.
- 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 Trigger-Ereignis 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 Trigger-Signal generiert. Dieses Trigger-Signal löst die gleichzeitige Positionsermittlung über die EnDat-Schnittstelle und über die Inkrementalsignale aus. Die beiden Positionen werden als Rückgabewert an die Kunden-Softwareapplikation übergeben.

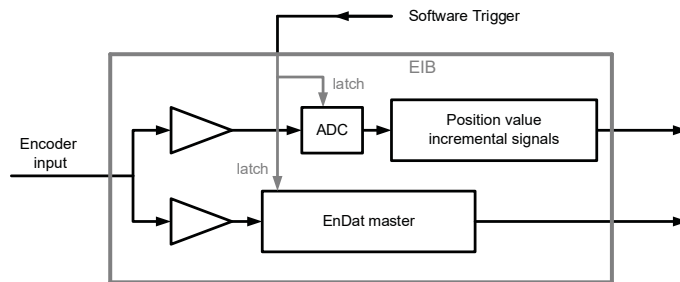


Abbildung 8: Blockschaltbild Verarbeitung zusätzlicher Inkrementalsignale bei EnDat-Messgeräten

Anmerkungen:

- 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 "Verarbeitung von Inkrementalsignalen", Seite 14  
 EnDat-Position: Interpolationsnullpunkt siehe Grafik

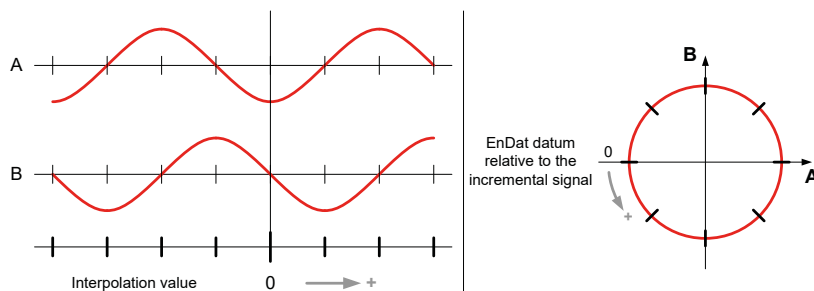


Abbildung 9: Interpolationswert bei EnDat mit Inkrementalsignalen

### 2.2.5 Hilfsachse

Die Hilfsachse ist gekoppelt an die Achse 1 und für Messgeräte mit 1 V<sub>SS</sub>-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. So 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.

Signalperiode für lineares Messgerät:

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

Signalperiode für rotatives Messgerät:

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

#### Maximale Eingangsfrequenz der Hilfsachse

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

1) ab Teilenummer xxx-02: Maximale Eingangsfrequenz 70 kHz bei Referenzierung

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 ggf. durch die übergeordnete Kunden-Softwareapplikation behandelt werden.

**Registerinhalt Positionswert der Hilfsachse**

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

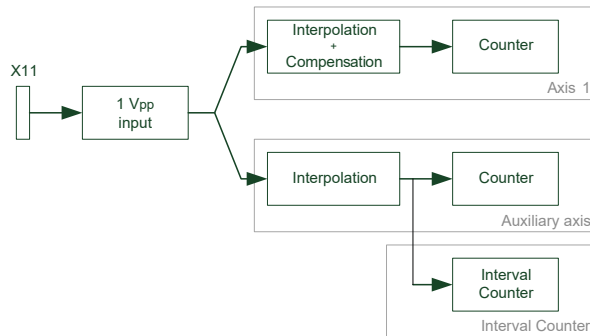


Abbildung 10: Blockschaltbild Hilfsachse und Interval Counter

## 2.3 Verarbeitung von Trigger-Ereignissen

Die Positionswertermittlung innerhalb der EIB 74x wird über ein sogenanntes Trigger-Ereignis angestoßen. Die EIB 74x unterstützt dabei folgende Trigger-Quellen:

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

Die Trigger-Quelle muss für jede Achse per Software-Kommando eingestellt werden, wobei zur gleichen Zeit nur eine Trigger-Quelle pro Achse wirken kann. Allerdings ist es möglich, für verschiedene Achsen unterschiedliche Trigger-Quellen zu aktivieren.

Zusätzlich muss eine Trigger-Quelle als Master-Trigger-Quelle definiert werden, die den Zeitpunkt der Datenübertragung bestimmt. Für alle Achsen, die ebenfalls mit der Master-Trigger-Quelle 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 Trigger-Ereignis auftrat. Andernfalls wird der Positionswert als ungültig markiert.

Nicht in allen Betriebsmodi werden alle Möglichkeiten des Trigger-Interfaces unterstützt.

**Weitere Informationen:** "Betriebsmodi", Seite 34

### 2.3.1 Trigger-Eingänge und -Ausgänge

Es werden vier Trigger-Eingänge und -Ausgänge unterstützt. Technische Daten zum Trigger-Eingang können der "Betriebsanleitung" entnommen werden.

#### Trigger-Eingänge

Trigger-Eingänge dienen zur Synchronisation der Positionsabfragen auf externe Ereignisse. Für die Begrenzung der Trigger-Rate siehe "Maximale Trigger-Rate", Seite 29.

Der Abschlusswiderstand 120  $\Omega$  kann per Konfiguration zu- oder abgeschaltet werden.



**Trigger-Ausgänge**

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

Ein Durchschleifen der Trigger-Signale vom Eingang zum Ausgang kann für jeden Kanal separat erfolgen. Der Eingang 1 kann mit Ausgang 1 verbunden werden usw. In der folgenden Abbildung ist ein Trigger-Eingang und der entsprechende Ausgang dargestellt.

**i** Um die Polarität des Signals am Trigger-Ausgang 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 "Betriebsanleitung").

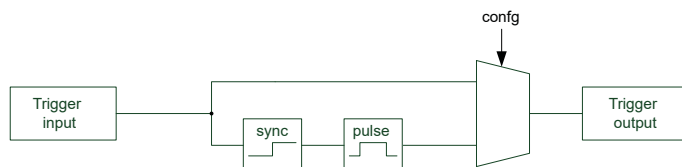


Abbildung 11: Blockschaltbild Weiterleitung von Trigger-Ereignissen

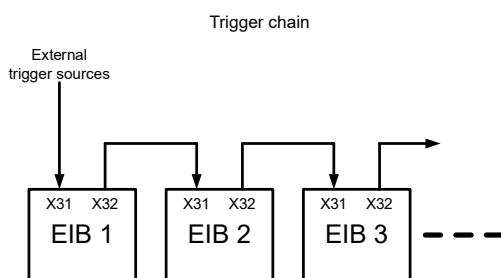


Abbildung 12: Trigger-Kette mit mehreren EIB 74x

**Konfiguration der Trigger-Eingänge und -Ausgänge als logische Ein- und Ausgänge**

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

## 2.3.2 Logische Eingänge und Ausgänge

### Logische Eingänge

Jeder Trigger-Eingang kann individuell auf einen logischen Eingang umgestellt werden. Der Pegel des entsprechenden Eingangs kann per Software-Kommando ausgelesen werden. Der Abschlusswiderstand  $120 \Omega$  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-Eingänge und -Ausgänge im Überblick. Es ist nur ein Kanal dargestellt.

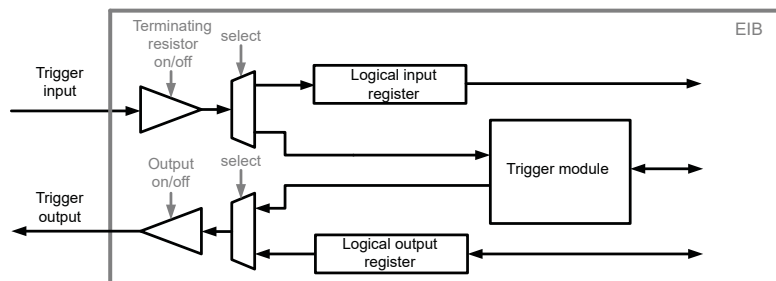


Abbildung 13: Blockschaltbild Trigger-Eingänge und -Ausgänge

### 2.3.3 Trigger-Modul

Das Trigger-Modul ermöglicht die Auswahl und Kontrolle der Trigger-Quellen. Darüber hinaus generiert es interne Trigger-Signale. Externe Trigger-Signale können verzögert werden, wobei die Verzögerungszeit für jeden Eingang separat einstellbar ist.

Der Referenzimpuls eines Messgeräts mit 1  $V_{SS}$ -Schnittstelle kann jeweils für die zugehörige Achse als Trigger-Quelle verwendet werden. Der Referenzimpuls der Achse 1 wird zusätzlich mit den Signalen A und B der Achse 1 logisch UND verknüpft und kann als Trigger-Signal für beliebige Achsen dienen. Die aktive Flanke des Referenzsignals ist dabei jeweils einstellbar. Wird für mehrere Achsen als Trigger-Quelle 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 Trigger-Signale 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 Trigger-Quelle, sondern erlaubt es, die Anzahl der Trigger-Impulse anderer Quellen zu begrenzen. Eine auswählbare Trigger-Quelle kann Impulse liefern, die so lange gesperrt werden, bis mit dem Startsignal das Tor geöffnet wird. Fortan werden alle Trigger-Impulse 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 Trigger-Impulse kann auf diese Weise erhöht werden.

#### Switch-Matrix

Trigger-Quelle	Trigger-Ausgang	Achse	Hilfs-achse	Pulszähler Trigger	Pulszähler Start
Trigger-Eingang	x	x	x	x	x
Referenzimpuls	-	x	x	x	x
Referenzimpuls	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	-

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

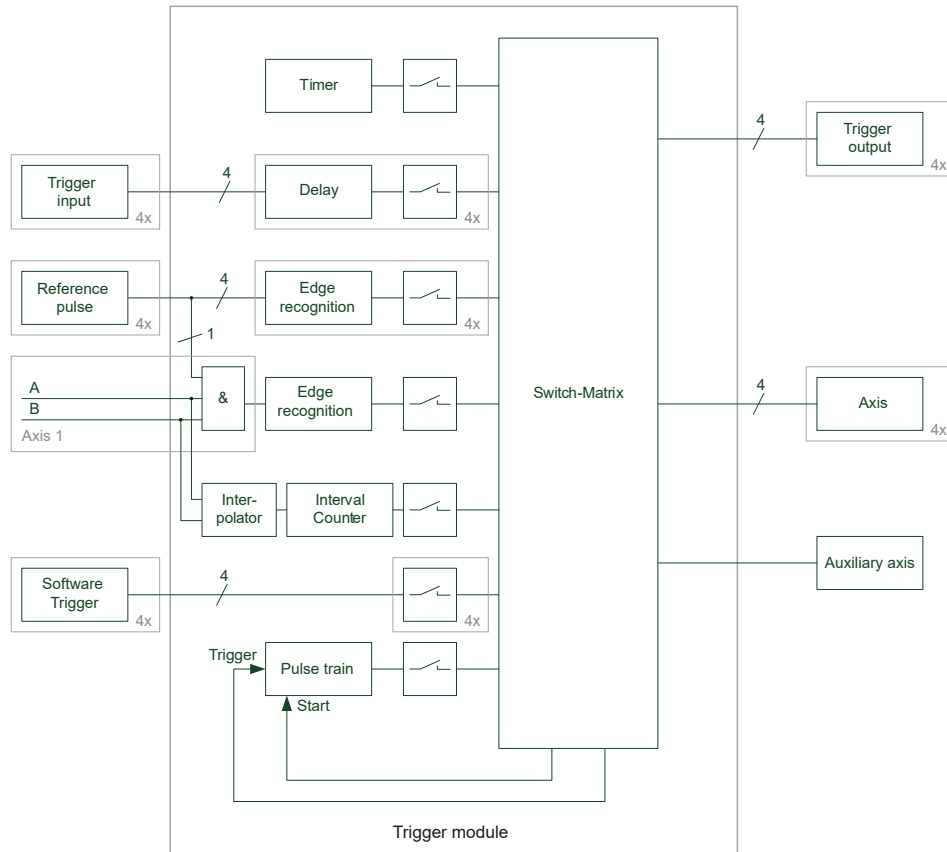


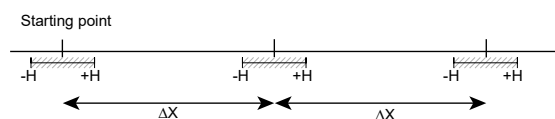
Abbildung 14: Blockschaltbild Trigger-Modul

### 2.3.4 Interval Counter

Der Interval Counter ermöglicht eine positionsabhängige Triggerung in Verbindung mit einem inkrementalen Messgerät an Achse 1. Das Messgerätesignal kann dabei interpoliert werden (siehe "Hilfsachse", Seite 23).

Die Triggerung erfolgt an einer bestimmten Position oder es werden äquidistante Trigger-Impulse mit einem einstellbaren Positionsabstand erzeugt. Die Ausgabe der Trigger-Impulse erfolgt ab dem Überfahren einer einstellbaren Startposition und dann fortlaufend mit dem Positionsabstand in beide Zählrichtungen. Der Positionsabstand  $\Delta X$  muss in Zählschritten angegeben werden (zur Berechnung der Schrittweite siehe "Hilfsachse", Seite 23).

Eine Hysterese verhindert ein mehrfaches Triggern vor allem bei hoher Interpolation des Messgerätesignals. Nachdem ein Trigger-Impuls an einer Position ausgegeben wurde, muss sich der Wert des Positionszählers um  $+H$  oder  $-H$  ändern, bevor an der gleichen Position erneut ein Trigger-Impuls erzeugt wird.



$H = 5$

Abbildung 15: Hysterese des Interval Counters

### 2.3.5 Maximale Trigger-Rate

Die maximale Trigger-Rate der EIB 74x ist abhängig vom eingestellten Betriebsmodus (mit Ausnahme des Betriebsmodus "Polling"):

- Betriebsmodus "Soft Realtime" max. 10 kHz
- Betriebsmodus "Recording" max. 50 kHz
- Betriebsmodus "Streaming" max. 50 kHz

Im Betriebsmodus "Streaming" ist zusätzlich die Datenrate begrenzt auf 1.200.000 Byte/s. Die Datenrate ergibt sich aus der Größe eines Datenpakets und der Trigger-Rate.

$$\frac{\text{Datenpaketgröße}}{\text{Byte}} \cdot \frac{\text{Trigger - Rate}}{\text{Hz}} \leq 1.200.000$$

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. Wenn dieser Zeitabstand nicht eingehalten wird, die Trigger-Rate also zu hoch ist, dann können Trigger-Ereignisse von der EIB 74x nicht akzeptiert werden und gehen verloren (Lost Trigger). Dies wird von der EIB 74x detektiert und im Statuswort des Positionsdatenpakets mit dem Bit "Lost Trigger" angezeigt. Dieses Bit ist solange auf "1" gesetzt, bis es aktiv von der Kunden-Softwareapplikation mit einem Clear-Kommando zurückgesetzt wird.

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



Wenn die maximale Trigger-Rate massiv überschritten wird (z. B. durch Fehlparametrierung oder zu viele Ereignisse am externen Trigger-Eingang), dann kann dies dazu führen, dass die EIB 74x nicht mehr auf externe Kommandos reagiert und nur nach einem Hard-Reset wieder ansprechbar ist.

### 2.3.6 Zähler für akzeptierte Trigger-Ereignisse

Neben der Überwachung auf Lost Trigger verfügt die EIB 74x zur weiteren Fehleraufdeckung über einen Zähler, der mit jedem eintreffenden und akzeptierten Trigger-Ereignis der Master-Trigger-Quelle inkrementiert wird. Ein Trigger-Ereignis wird dann akzeptiert, wenn oben erwähnter Zeitabstand eingehalten wird. Trigger-Ereignisse, die zu Lost Trigger führen, werden nicht gezählt.

Der Zählerwert wird im Positionsdatenpaket übertragen und kann auf Stetigkeit überwacht werden. Damit lassen sich verloren gegangene Positionsdatenpakete aufdecken.

## 2.4 Timestamp

Der 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 Positionswernermittlung führt, löst gleichzeitig eine Abspeicherung des aktuellen Timer-Werts in das Timestamp-Register aus. Der Inhalt dieses Registers wird bei aktivierter Timestamp-Funktion mit dem Positionsdatenpaket übertragen. Damit kann die Kunden-Softwareapplikation prüfen, ob der Latch-Zeitpunkt jedes einzelnen Positionswerts dem Erwartungswert entspricht. Bei Applikationen, die nicht über einen periodischen Trigger verfügen, kann mit diesem Register der Zeitpunkt des Trigger-Ereignisses übermittelt werden.

Das Zeitintervall des Timestamp-Zählers ist ein Vielfaches des internen Systemtakts der EIB 74x. Bevor der 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.



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

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

Der Wert für "clock ticks per  $\mu$ s" kann z. B. mit den Funktionen

`EIB7GetTimerTriggerTicks` oder `EIB7GetTimestampTicks` ausgelesen werden.

## 2.5 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 Messgerätekanal separat übermittelt und ist unabhängig vom eingestellten Betriebsmodus.

### Registerinhalt Statuswort

Bit-Nr.	Inkrementelle Position	EnDat-Position	EnDat-Zusatzinformation	Hilfsachse
0	1 = Position gültig	1 = Position 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 Referenzwerts 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

### Bedeutung der Fehlerbits

Name	Bedeutung
Position gültig	1 → kein Fehler aufgetreten Dieses Bit gibt an, ob die übertragene Position gültig ist oder nicht
Zusatzinformation gültig	1 → EnDat-Zusatzinformation wurde empfangen Andernfalls ist keine Zusatzinformation angewählt oder wurde nicht empfangen
Fehler Signalamplitude	1 → Signalamplitude der 1 $V_{SS}$ -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 Messgeräts wurde automatisch abgeschaltet (Überstromabsicherung hat angesprochen)
Fehler Lüfter	1 → Lüfter der EIB 74x arbeitet fehlerhaft
Lost Trigger	Informationen siehe "Maximale Trigger-Rate", Seite 29
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 → Homing-Signal (L1) zum Zeitpunkt der Positionsabfrage aktiv
Limit-Signal	1 → Limit-Signal (L2) 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 Positionsdatenpaket erneut übertragen.

Bei inkrementellen Messgeräten zeigt ein Fehler im Positionsdatenpaket 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 Messgeräts werden auch andere Fehler mit ansprechen. Aus diesem Grund sollte immer zuerst ein eventueller Fehler der Spannungsversorgung zurückgesetzt werden. Nachdem die Spannungsversorgung stabil ist (Wartezeit ca. 1,5 Sekunden), sollten die anderen Fehler rückgesetzt werden.

- **Lost Trigger**  
Dieses Bit zeigt an, dass mindestens ein Trigger-Ereignis aufgrund einer zu kurzen Zeitspanne zwischen zwei Trigger-Ereignissen nicht korrekt verarbeitet wurde. Das Bit kann ebenfalls auftreten, wenn die Trigger-Leitung mit Störungen überlagert ist oder EMV-Einflüsse die Übertragung negativ beeinflussen. Ein Bit `Lost Trigger` bedeutet nicht, dass die Positionswerte falsch sind, es wird lediglich angezeigt, dass Trigger-Ereignisse 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 Positionsdatenpaket 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 Positionsdatenpaket 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.



Weitere Hinweise siehe "Betriebsanleitung". Wird die Lüfter-Überwachung nicht unterstützt, dann ist dieses Bit immer auf "0".

## 2.6 Ethernet-Interface

Das Ethernet-Interface (LAN) wird für die Konfiguration der EIB 74x und für die Übertragung der Positionsdatenpakete genutzt. Die Konfiguration erfolgt über TCP-Kommandos und die Übertragung der Daten im Betriebsmodus "Soft Realtime" 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 Hinweise siehe "Betriebsanleitung" oder siehe "DHCP", Seite 8.

## 2.7 Betriebsmodi

Von der EIB 74x werden folgende Betriebsmodi unterstützt:

- "Polling"
- "Soft Realtime"
- "Streaming"
- "Recording"

### 2.7.1 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 Trigger-Ereignis 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 Betriebsmodus "Recording".

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, wenn 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 Datenpakets. Die Länge eines Datenpakets muss allerdings immer ein Vielfaches von 4 Bytes betragen. Wenn dies bei einer bestimmten Konfiguration nicht erfüllt ist, werden automatisch am Ende entsprechend viele Füllbytes angehängt.

#### Paketkonfiguration

Region	Datenelement	Beschreibung	Länge in Bytes
Global	TriggerCounter	Zähler für Trigger-Ereignisse	2
Achse	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	Statuswort	Status und Fehlermeldungen	2
	Positionswert	Aktueller Positionswert des Messgeräts	4
	Timestamp	Zeitstempel für Positionswert	4
	Referenzposition	Positionswert bei Referenzmarken	4

**Beispiel**

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

- Globale Information: Trigger Counter
- Achse 1: Inkrementale Schnittstelle (1  $V_{SS}$ ), eine Referenzmarke
- Achse 2: Inkrementale Schnittstelle (1  $V_{SS}$ ), eine Referenzmarke

**Paketkonfiguration**

Region	Datenelement	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 Datenpakets von 52 Bytes.

**Default-Konfiguration**

Nach dem Einschalten des Geräts 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 Datenpakets wieder.

**Paketkonfiguration**

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

## 2.7.2 Betriebsmodus "Polling"

Dieser Betriebsmodus 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

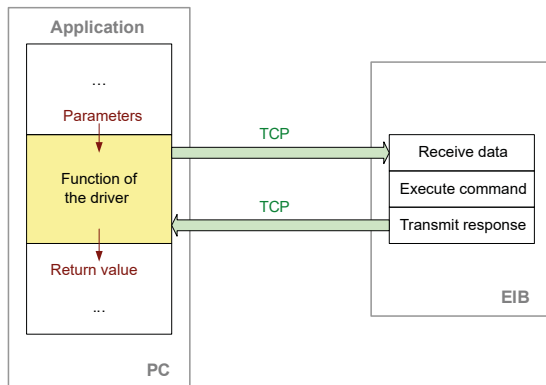


Abbildung 16: Ablauf Positionsabfrage im Betriebsmodus "Polling"

Datenpakete im Betriebsmodus "Polling" sind abhängig von der gewählten Funktion, siehe "Treibersoftware", Seite 45.

### 2.7.3 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 Trigger-Signal erhält. Mit jedem Trigger-Ereignis wird ein Datenpaket automatisch an den PC gesendet. Dort können die Pakete aus einem FIFO gelesen werden.

Für den Betriebsmodus "Soft Realtime" muss die EIB 74x mit den nachfolgend aufgeführten Schritten konfiguriert werden:

- Initialisierung der EIB 74x
- Initialisierung und Konfiguration der Achsen
- Konfiguration des Datenpakets
- Konfiguration der Trigger-Logik
- Auswahl des Betriebsmodus ("Soft Realtime")
- Aktivierung der Trigger-Quelle

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.

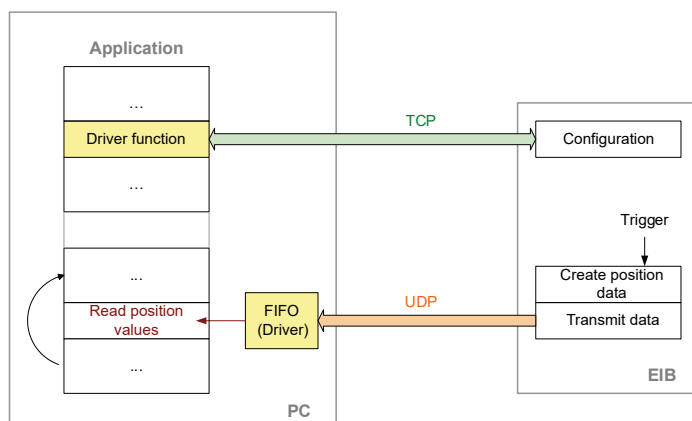


Abbildung 17: Ablauf Positionsanfrage im Betriebsmodus "Soft Realtime"

Sobald die Positionswerte aus dem FIFO am PC gelesen wurden, wird dies an die EIB bestätigt. Wenn 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 Trigger-Quelle zu deaktivieren. Anschließend kann der Betriebsmodus geändert oder die Verbindung zur EIB 74x geschlossen werden.

Verarbeitung von Trigger-Ereignissen:

- Externe Trigger-Eingänge werden unterstützt
- Interne Trigger-Quellen werden unterstützt
- Software-Trigger wird unterstützt

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

Bei den Fehlern "Frequenzüberschreitung" und "Lost Trigger" wird eine Wiederholung der Messung empfohlen. Zum Löschen der Fehler sollte vorübergehend der Betriebsmodus "Polling" gewählt werden.

## 2.7.4 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 Trigger-Signal erhält. Mit jedem Trigger-Ereignis wird ein Datenpaket erzeugt. Je nach Trigger-Rate und Datenvolumen werden mehrere Datenpakete zusammengefasst und zum PC gesendet. Dort können die Pakete aus einem FIFO gelesen werden. Für den Betriebsmodus "Streaming" muss die EIB 74x mit den nachfolgend aufgeführten Schritten konfiguriert werden:

- Initialisierung der EIB 74x
- Initialisierung und Konfiguration der Achsen
- Konfiguration des Datenpakets
- Konfiguration der Trigger-Logik
- Auswahl des Betriebsmodus ("Streaming")
- Aktivierung der Trigger-Quelle

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.

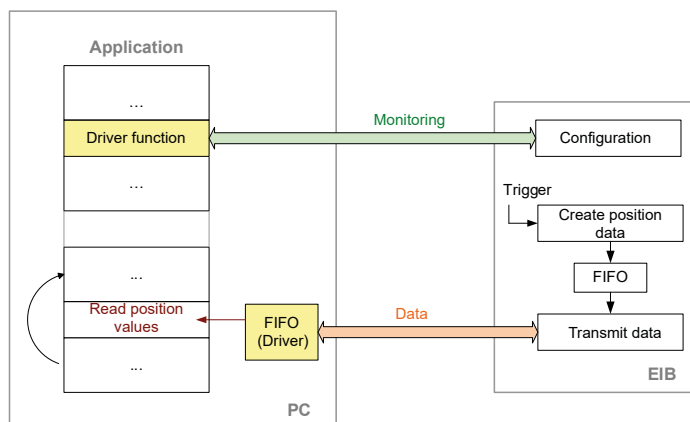


Abbildung 18: Ablauf Positionsabfrage im Betriebsmodus "Streaming"

Sobald die Positionswerte aus dem FIFO am PC gelesen wurden, wird dies an die EIB bestätigt. Wenn 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 Trigger-Quelle zu deaktivieren. Anschließend kann der Betriebsmodus geändert oder die Verbindung zur EIB 74x geschlossen werden.

Verarbeitung von Trigger-Ereignissen:

- Externe Trigger-Eingänge werden unterstützt
- Interne Trigger-Quellen werden unterstützt
- Software-Trigger wird unterstützt

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

Bei den Fehlern "Frequenzüberschreitung" und "Lost Trigger" wird eine Wiederholung der Messung empfohlen. Zum Löschen der Fehler sollte vorübergehend der Betriebsmodus "Polling" gewählt werden.



### 2.7.5 Betriebsmodus "Recording"

Die Positionsdaten werden im Speicher der EIB 74x abgelegt. Mit jedem Trigger-Ereignis 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 Betriebsmodus "Recording" können die letzten n Samples aus dem Speicher gelesen werden.

Die Aufzeichnungstiefe ist abhängig von der Größe eines Datenpakets und kann ausgelesen werden (siehe "Recording - Speichergröße lesen", Seite 82).

Für den Betriebsmodus "Recording" muss die EIB 74x mit den nachfolgend aufgeführten Schritten konfiguriert werden:

- Initialisierung der EIB 74x
- Initialisierung und Konfiguration der Achsen
- Konfiguration des Datenpakets
- Konfiguration der Trigger-Logik
- Auswahl des Betriebsmodus ("Recording")
- Aktivierung der Trigger-Quelle

Nach dem Abschluss der Aufzeichnungsphase müssen folgende Schritte ausgeführt werden:

- Deaktivieren der Trigger-Quelle
- 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.

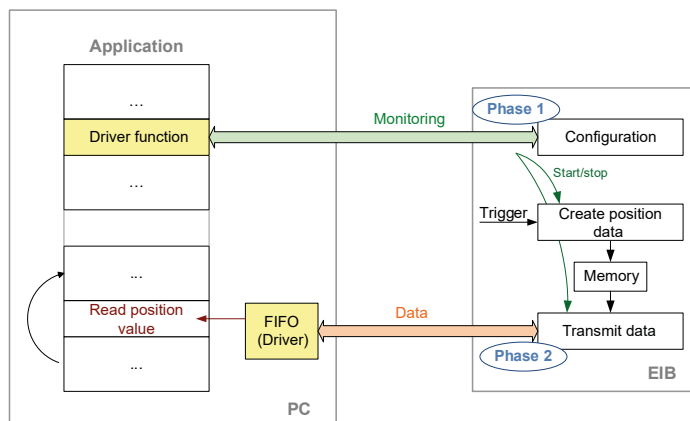


Abbildung 19: Ablauf Positionsabfrage im Betriebsmodus "Recording"

Verarbeitung von Trigger-Ereignissen:

- Externe Trigger-Eingänge werden unterstützt
- Interne Trigger-Quellen werden unterstützt
- Software-Trigger wird unterstützt

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

## 2.8 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.

### 2.8.1 Firmware-Update durchführen

Das folgende Beispiel geht für das Firmware-Update von einem Computer mit dem Betriebssystem Microsoft Windows aus. Die EIB 74x muss über Ethernet mit dem Computer verbunden sein. Der Dateiname für das Update ist in diesem Beispiel **update\_633281-14.flash**. Diese Datei ist gespeichert unter **C:\temp\EIB**.

- ▶ Ausführen der Windows Kommandozeile
- ▶ Speichern des Update-Files unter **C:\temp\EIB\update\_633281-14.flash**
- ▶ Start des TFTP File Transfers > **tftp -i 192.168.1.2**  
**put C:\temp\EIB\update\_633281-14.flash tmp\update.flash**
  - Option **-i** aktiviert den "binary file transfer"
  - IP-Adresse **192.168.1.2** oder kundenspezifische Einstellung
  - Kommando **put** für Transfer vom Host zur EIB 74x
  - Quelldatei in diesem Beispiel **C:\temp\EIB\update\_633281-14.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 den Angaben im Kapitel "Geräte-Resets" der Betriebsanleitung gebootet.



Wenn das Update auf einem Linux-System durchgeführt wird, muss die unterschiedliche Verwendung von "/" (slash) und "\" (backslash) beachtet werden. Die EIB 74x benötigt den backslash, wie oben dargestellt. Andernfalls wird zwar das Update-File zur EIB 74x übertragen, aber von der EIB 74x nicht installiert.

Beispiel für ein Linux-System:

```
user@pc> tftp -v 192.168.1.2 -m binary -c  
put update_633281-14.flash tmp\\update.flash
```

## 2.8.2 TFTP-Client aktivieren

Die nachfolgende Anleitung beschreibt, wie Sie TFTP-Client unter Microsoft Windows 10 aktivieren. Gehen Sie gleichermaßen vor, um TFTP-Client unter Microsoft Windows 7 zu aktivieren.

- ▶ In Microsoft Windows nacheinander öffnen:
  - **Start**
  - **Systemsteuerung**
  - **Programme**
  - **Programme und Features**
- > Der Dialog **Programme und Features** öffnet sich
- ▶ Den Dialog **Windows-Features aktivieren oder deaktivieren** auswählen
- > Der Dialog **Windows-Features aktivieren oder deaktivieren** öffnet sich  
oder
- ▶ Windows-Taste + "R" drücken
- > Der Dialog **Ausführen** öffnet sich
- ▶ Über die Tastatur "appwiz.cpl" eingeben
- ▶ Eingabe mit **OK** bestätigen
- > Der Dialog **Programme und Features** öffnet sich
- ▶ Den Dialog **Windows-Features aktivieren oder deaktivieren** auswählen
- > Der Dialog **Windows-Features aktivieren oder deaktivieren** öffnet sich

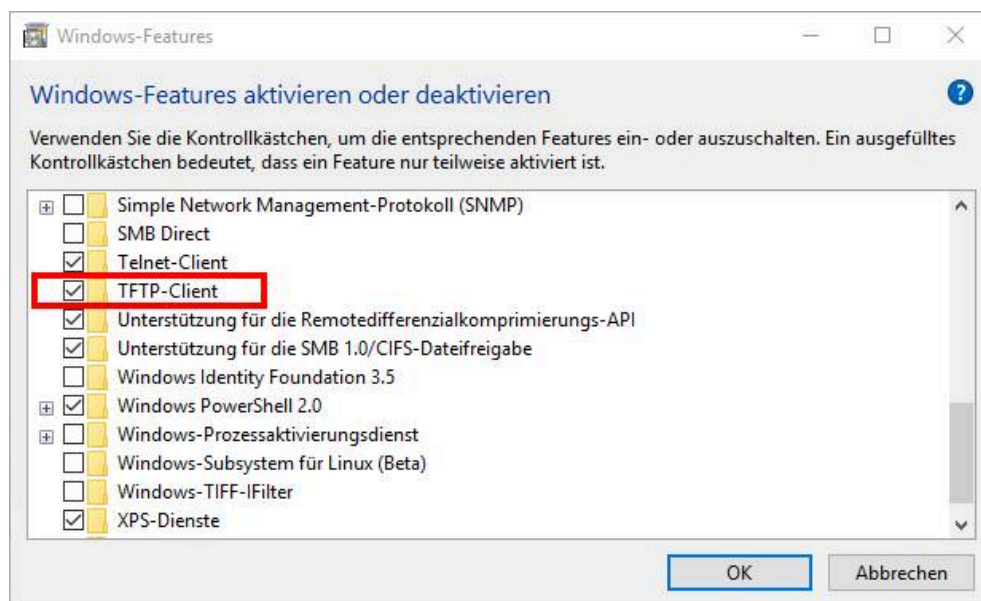




Abbildung 20: Dialog **Windows-Features aktivieren oder deaktivieren**

- ▶ **TFTP-Client** mit Häkchen aktivieren
- ▶ Mit **OK** bestätigen
- > Der **TFTP-Client** steht ohne Neustart des Systems als Kommandozeilenbefehl zur Verfügung

 Gegebenenfalls können zur optimalen Nutzung des TFTP-Clients Anpassungen der Firewall notwendig sein.

 Weitere Informationen erhalten Sie vom Microsoft-Support.

## 2.9 **Reset**

Siehe "Betriebsanleitung"

# 3

**Treibersoftware**

## 3.1 Allgemeine Information

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
- Linux/Unix



Wenn nicht anders aufgeführt, werden 32- und 64-Bit-Versionen unterstützt, wenn diese für die jeweilige Windows-Version verfügbar sind.



Beachten Sie die Systemanforderungen in der beiliegenden ReadMe-Datei. Installieren Sie Funktionen-Pakete, die von Dritten benötigt werden, ggf. nach.

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.



Unterstützung für Visual Basic und C# auf Anfrage.

## 3.2 Installationsanleitung

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

### 3.2.1 Installation unter Microsoft 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 Dateien 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.

### 3.2.2 Installation unter 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 Datei 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.

## 3.3 Überblick

### 3.3.1 Kommunikationsaufbau

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 "Datenpaket erstellen", Seite 56).

Anschließend kann diese Konfiguration über die Funktion `EIB7ConfigDataPacket()` in die EIB 74x geladen werden (siehe "Datenpaket konfigurieren", Seite 58)

### 3.3.2 Betriebsmodus "Polling"

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 Trigger-Quelle auszuwählen. Die Triggerung erfolgt implizit mit dem Aufruf der Funktion `EIB7GetPosition()`.

### 3.3.3 Betriebsmodus "Soft Realtime"

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

Nachdem in den Betriebsmodus "Soft Realtime" gewechselt wurde, kann die Trigger-Quelle 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()` (siehe "Daten aus FIFO lesen", Seite 83) oder `EIB7ReadFIFODataRaw()` (siehe "Daten aus FIFO lesen und konvertieren", Seite 87) erfolgen. 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()` (siehe "Größe eines FIFO-Elements lesen", Seite 84) und `EIB7SizeOfFIFOEntryRaw()` (siehe "Größe eines FIFO-Elements nach Konvertierung lesen", Seite 88) bestimmt werden.

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 "Callback-Mechanismus aktivieren", Seite 91).

### 3.3.4 Betriebsmodus "Streaming"

Zuerst müssen die Achsen mit `EIB7InitAxis()` initialisiert und das Datenpaket sowie die Trigger-Logik konfiguriert werden. Anschließend kann der Betriebsmodus "Streaming" aktiviert werden. Im Betriebsmodus "Streaming" können lediglich die Fehlermeldungen aus dem Statuswort für die Positionswerte zurückgesetzt und der Status des Puffers ausgelesen werden (siehe "Streaming - Status prüfen", Seite 83).

Nachdem in den Betriebsmodus "Streaming" gewechselt wurde, kann die Trigger-Quelle 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()` (siehe "Daten aus FIFO lesen und konvertieren", Seite 87) oder `EIB7ReadFIFODataRaw()` (siehe "Daten aus FIFO lesen", Seite 83) erfolgen. 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()` (siehe "Größe eines FIFO-Elements nach Konvertierung lesen", Seite 88) und `EIB7SizeOfFIFOEntryRaw()` (siehe "Größe eines FIFO-Elements lesen", Seite 84) bestimmt werden.

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 "Callback-Mechanismus aktivieren", Seite 91).



### 3.3.5 Betriebsmodus "Recording"

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

Nach dem Ende der Aufzeichnung können die Daten zum Host übertragen werden. Dies erfolgt im Betriebsmodus "Polling". Mit der Funktion `EIB7TransferRecordingData()` wird die Übertragung gestartet.

Die Kunden-Softwareapplikation am Host kann die Positionsdaten aus dem Empfangspuffer über die Funktionen `EIB7ReadFIFOData()` (siehe "Daten aus FIFO lesen und konvertieren", Seite 87) oder `EIB7ReadFIFODataRaw()` (siehe "Daten aus FIFO lesen", Seite 83) auslesen. 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()` (siehe "Größe eines FIFO-Elements nach Konvertierung lesen", Seite 88) und `EIB7SizeOfFIFOEntryRaw()` (siehe "Größe eines FIFO-Elements lesen", Seite 84) bestimmt werden.

Auf die einzelnen Komponenten eines FIFO-Eintrags kann mit Hilfe der Funktion `EIB7GetDataFieldPtr()` oder `EIB7GetDataFieldPtrRaw()` zugegriffen werden.

### 3.3.6 Datentypen und Datenpaket

#### Einfache Datentypen

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



Für die EIB 74x werden für Integer folgende Datenformate in C/C++ angenommen:

- `short`: 16 bit Integer
- `long`: 32 Bit Integer
- `long long`: 64 Bit Integer

Diese Datenformate müssen vor allem bei 64-Bit-Betriebssystemen beachtet werden.

#### EnDat-Zusatzinformation

```
struct ENDAT_ADDINFO
```

Komponente	Beschreibung
<code>status</code>	Statuswort für Zusatzinformation
<code>info</code>	Daten der Zusatzinformation

**Information für TCP-Verbindung**

```
struct EIB7_CONN_INFO
```

Komponente	Beschreibung
<code>id</code>	Identifikationsnummer für diese Verbindung
<code>local ip</code>	Lokale IP-Adresse für diese Verbindung
<code>local port</code>	Lokale Port-Nummer für diese Verbindung
<code>remote ip</code>	IP-Adresse der EIB 74x für diese Verbindung
<code>remote port</code>	Port-Nummer der EIB 74x für diese Verbindung

**Konfiguration für Datenpaket**

```
struct EIB7_DataPacketSection
```

Komponente	Beschreibung
<code>region</code>	Globale Information oder Achse der EIB 74x
<code>items</code>	Datenelemente innerhalb der Region

**3.3.7 Parameter und Rückgabewerte**

Alle Funktionen liefern einen Rückgabewert vom Datentyp `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).

Die Geräte-, Achs- und IO-Funktionen können als Standard-Rückgabewert die nachfolgend aufgeführten Fehlermeldungen liefern:

**Standard-Rückgabewerte**

<code>EIB7_NoError</code>	Funktionsaufruf erfolgreich
<code>EIB7_CMDErr</code>	Funktionsaufruf mit Fehler abgebrochen
<code>EIB7_InvalidHandle</code>	Handle auf EIB 74x bzw. auf Achse der EIB 74x ungültig
<code>EIB7_FuncNotSupp</code>	Funktion von EIB 74x nicht unterstützt
<code>EIB7_InvalidResponse</code>	Fehler bei Datenübertragung
<code>EIB7_AccNotAllowed</code>	Funktion kann nicht ausgeführt werden, da EIB 74x Zugriff nicht erlaubt
<code>EIB7_ConnReset</code>	Verbindung von EIB 74x beendet
<code>EIB7_ConnTimeout</code>	Timeout bei Datenübertragung zur EIB 74x
<code>EIB7_ReceiveError</code>	Fehler beim Empfangen der Daten
<code>EIB7_SendError</code>	Fehler beim Senden der Daten
<code>EIB7_OutOfMemory</code>	Vom System kann nicht genügend Speicher allokiert werden
<code>EIB7_PortDirInv</code>	Signalrichtung des Ports falsch (nur IO-Funktionen)

Zusätzlich können weitere Rückgabewerte geliefert werden, die für jede Funktion separat aufgeführt werden.

## 3.4 Hilfsfunktionen

### 3.4.1 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 z. B. "192.168.1.2" oder "EIB74x-SN1234567" lauten.

#### Funktion

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

#### Parameter

hostname	Pointer auf C-String, der die IP-Adresse oder den Hostnamen der EIB 74x enthält
ip	<i>[Rückgabewert]</i> Pointer auf 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

### 3.4.2 Positionsdatenformat ändern

Das Datenformat eines Positionswerts 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
dest	<i>[Rückgabewert]</i> Pointer auf 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 ungültig

## 3.5 Gerätefunktionen

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

Alle Gerätefunktionen können neben den Standard-Rückgabewerten ("Parameter und Rückgabewerte") zusätzlich weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

### 3.5.1 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

<code>ip</code>	IP-Adresse in "Host Byte Order"
<code>eib</code>	[Rückgabewert] Handle für EIB 74x falls Funktion erfolgreich beendet wurde
<code>timeout</code>	Timeout für folgende Kommandos in Millisekunden (nicht gültig für <code>EIB7Open()</code> )
<code>ident</code>	[Rückgabewert] Pointer auf 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 Firmware-Version der EIB 74x nicht ausgelesen
<code>len</code>	Größe des Zielspeichers in Bytes (0 wenn <code>ident = NULL</code> )

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

### 3.5.2 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 Betriebsmodus "Polling" 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.3 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 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 Datenübertragung aufgetreten
EIB7_CS_ConnectionReset	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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.4 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_IllegalParameter` Timeout kann nicht eingestellt werden

### 3.5.5 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.6 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

<i>eib</i>	EIB-Handle
<i>set</i>	<i>[Rückgabewert]</i> Pointer auf erstes Element des Array
<i>size</i>	Maximale Anzahl der Einträge im Array
<i>len</i>	<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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.7 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 im Parameter `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` und `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

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

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.8 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 ( EIB7_DataPacketSection* packet,
                                     unsigned long       index,
                                     EIB7_DataRegion     region,
                                     unsigned long       items
                                     )
```



**Parameter**

packet            Pointer auf 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 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid            Parameter ungültig

### 3.5.9 Datenpaket konfigurieren

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

#### Funktion

```
EIB7_ERR EIB7ConfigDataPacket      ( EIB7_HANDLE          eib,
                                   EIB7_DataPacketSection*    packet,
                                   unsigned long                size
                                   )
```

#### Parameter

<code>eib</code>	EIB-Handle
<code>packet</code>	Pointer auf Array mit den Konfigurationsdaten für das Datenpaket
<code>size</code>	Anzahl der Einträge im Array <code>packet</code>

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_ParamInvalid</code>	Parameter ungültig
<code>EIB7_PacketTooLong</code>	Konfigurationsdaten beschreiben ein zu langes Datenpaket
<code>EIB7_InvalidPacket</code>	Konfigurationsdaten beschreiben ein ungültiges Datenpaket

### 3.5.10 Betriebsmodus wählen

Der Betriebsmodus der EIB 74x kann eingestellt werden. Es werden die Betriebsmodi "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	Beschreibung
EIB7_OM_Polling	Betriebsmodus "Polling"
EIB7_OM_SoftRealtime	Betriebsmodus "Soft Realtime"
EIB7_OM_Streaming	Betriebsmodus "Streaming"
EIB7_OM_RecordingSingle	Betriebsmodus "Recording" - "Single Shot"
EIB7_OM_RecordingRoll	Betriebsmodus "Recording" - "Rolling"

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

- EIB7\_CantOpenSocket    Interner Fehler (Socket-Fehler)
- EIB7\_CantStartThread    Interner Fehler (Thread-Fehler)
- EIB7\_InvalidOpMode    Gewählter Betriebsmodus nicht unterstützt
- EIB7\_OpModeActive    Gewählter Betriebsmodus bereits aktiv
- EIB7\_OpModeBlocked    Gewählter Betriebsmodus kann nicht aktiviert werden
- EIB7\_InvalidIPAddr    Interner Fehler (IP-Adressfehler)

### 3.5.11 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 Netzwerk in "Host Byte Order"  
gateway              IP-Adresse des Standard Gateway in "Host Byte Order"  
dhcp                  Flag für DHCP-Client in der EIB 74x

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

timeout              Timeout für DHCP-Client in Sekunden

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) 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

### 3.5.12 Netzwerkparameter auslesen

Die Parameter für die Ethernet-Schnittstelle können ausgelesen werden. Es werden immer die benutzerdefinierten Einstellungen ausgegeben, auch wenn mit den Standardeinstellungen 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	<i>[Rückgabewert]</i> Pointer auf Variable für IP-Adresse in "Host Byte Order"
netmask	<i>[Rückgabewert]</i> Pointer auf Variable für Netzwerkmaske in "Host Byte Order"
gateway	<i>[Rückgabewert]</i> Pointer auf Variable für IP-Adresse des Standard Gateway in "Host Byte Order"
dhcp	<i>[Rückgabewert]</i> Pointer auf Variable für Flag des DHCP-Clients

dhcp	Beschreibung
EIB7_MD_Disable	DHCP-Client deaktiviert
EIB7_MD_Enable	DHCP-Client aktiviert

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_NoCustNetwork      Keine kundenspezifischen Einstellungen vorhanden

### 3.5.13 Hostname 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 Hostname auf den Standardwert bei der Auslieferung.

#### Funktion

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

#### Parameter

eib	EIB-Handle
hostname	Pointer auf neuen Hostname

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

**3.5.14 Hostname auslesen**

Der Hostname der EIB 74x wird ausgelesen und als C-String im Zielspeicher abgelegt. Der String ist maximal 32 Zeichen lang (inkl. 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	[Rückgabewert] Pointer auf Zielspeicher für Hostname
len	Größe des Zielspeichers in Bytes

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7_CantRdHostname	Hostname kann nicht gelesen werden
---------------------	------------------------------------

**3.5.15 Seriennummer auslesen**

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

**Funktion**

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

**Parameter**

eib	EIB-Handle
serial	[Rückgabewert] Pointer auf Zielspeicher für Seriennummer
len	Größe des Zielspeichers in Bytes

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

- EIB7\_CantRdHostname      Seriennummer kann nicht gelesen werden
- EIB7\_BufferTooSmall      Zielspeicher zu klein

**3.5.16 Geräte-Identnummer auslesen**

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

**Funktion**

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

**Parameter**

- eib                              EIB-Handle
- ident                            *[Rückgabewert]* Pointer auf Zielspeicher für Geräte-Identnummer
- len                                Größe des Zielspeichers in Bytes

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

- EIB7\_CantRdIdent              Geräte-Identnummer kann nicht gelesen werden
- EIB7\_BufferTooSmall      Zielspeicher zu klein

**3.5.17 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 6 Bytes verwendet. Das niederwertigste Byte der MAC-Adresse wird in das erste Byte des Zielspeichers kopiert.

Beispiel für 00:A0:CD:85:00:01:

Offset	Speicherinhalt
0	0x01
1	0x00
1	0x85
1	0xCD
1	0xA0
1	0x00

**Funktion**

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

**Parameter**

`eib`                            EIB-Handle  
`mac`                            *[Rückgabewert]* Pointer auf Zielspeicher für MAC-Adresse

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.18 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 Zielspeicher für Versionsnummer der Firmware  
`len`                            Größe des Zielspeichers in Bytes  
`select`                        Auswahl der Firmware, deren Versionsnummer ausgelesen wird

<code>select</code>	Beschreibung
<code>EIB7_FW_CurrentlyBooted</code>	Aktuell geladene Firmware
<code>EIB7_FW_Factory</code>	Firmware des Auslieferungszustands
<code>EIB7_FW_User</code>	Firmware des letzten Updates

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.19 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 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.20 Update-Status 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 Variable für Update-Status

select	Beschreibung
EIB7_US_NoUpdate	Kein Update eingespielt
EIB7_US_UpdateFailed	Update konnte nicht durchgeführt werden
EIB7_US_UpdateSuccessful	Update erfolgreich durchgeführt
EIB7_US_VersionIncompatible	Firmware nicht kompatibel mit Hardware der EIB 74x

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.21 Anzahl der offenen Verbindungen auslesen**

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 Variable für Anzahl der offenen Verbindungen

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.22 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. Weitere Informationen zum Inhalt der Verbindungsdaten siehe "Datentypen und Datenpaket", Seite 49.

**Funktion**

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

**Parameter**

`eib`            EIB-Handle  
`info`          *[Rückgabewert]* Pointer auf erstes Element im Array für die Verbindungsdaten  
`size`          Größe des Array `info`  
`cnt`          *[Rückgabewert]* Pointer auf Variable für Anzahl der gültigen Elemente im Array

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.23 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 z. B. aufgrund eines Fehlers am Host nicht ordnungsgemäß beendet wurden. Die ID kann aus den Verbindungsdaten `EIB7_CONN_INFO` entnommen werden (siehe "Verbindungsdaten auslesen", Seite 66).

#### Funktion

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

#### Parameter

<code>eib</code>	EIB-Handle
<code>id</code>	ID der Verbindung, die abgebrochen wird

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_CantTermConn</code>	Verbindung kann nicht beendet werden
<code>EIB7_CantTermSelf</code>	Verbindung kann sich nicht selbst beenden
<code>EIB7_ParamInvalid</code>	Parameter ist kein gültiger Index für eine Verbindung

### 3.5.24 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

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

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.25 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.



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

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

Der Wert für "clock ticks per  $\mu\text{s}$ " kann z. B. mit den Funktionen

`EIB7GetTimerTriggerTicks` oder `EIB7GetTimestampTicks` ausgelesen werden.

#### Funktion

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

#### Parameter

<code>eib</code>	EIB-Handle
<code>period</code>	Ticks pro Timestamp-Periode (> 0), weitere Informationen siehe "Timestamp", Seite 30

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_ParamInvalid</code>	Timestamp-Periode ungültig
--------------------------------	----------------------------

### 3.5.26 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.27 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 Variable für die Takte pro Mikrosekunde

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.28 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 Trigger-Ereignis aus.

**Weitere Informationen:** "Maximale Trigger-Rate", Seite 29



Um den Wert für eine Zeitdauer (z. B. für den Parameter `period` des Funktionsaufrufs `EIB7SetTimestampPeriod`) richtig zu berechnen, muss an die Funktion übergeben werden:  
 $period = \text{Zeitintervall in } \mu s * \text{clock ticks per } \mu s$   
 Der Wert für "clock ticks per  $\mu s$ " kann z. B. mit den Funktionen `EIB7GetTimerTriggerTicks` oder `EIB7GetTimestampTicks` ausgelesen werden.

**Funktion**

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

**Parameter**

eib	EIB-Handle
period	Ticks pro Timer-Trigger-Periode (> 0), weitere Informationen siehe "Timestamp", Seite 30

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

**3.5.29 Zeiteinheit für Verzögerung an Trigger-Eingängen lesen**

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

**Funktion**

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

**Parameter**

eib	EIB-Handle
ticks	<i>[Rückgabewert]</i> 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.30 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.31 Software Trigger

Der Software Trigger erzeugt ein Trigger-Ereignis 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 Betriebsmodus "Polling" ausgeführt werden.

**Funktion**

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

**Parameter**

`eib`                    EIB-Handle  
`source`                 Software-Trigger-Kanal

<code>source</code>	Beschreibung
<code>EIB7_ST_SWtrigger1</code>	Software-Trigger Kanal 1
<code>EIB7_ST_SWtrigger2</code>	Software-Trigger Kanal 2
<code>EIB7_ST_SWtrigger3</code>	Software-Trigger Kanal 3
<code>EIB7_ST_SWtrigger4</code>	Software-Trigger Kanal 4

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid`        Parameter ungültig

### 3.5.32 Master-Trigger-Quelle wählen

Das Master-Trigger-Signal kann aus verschiedenen Quellen gewählt werden. Diese Funktion muss nach der Konfiguration der Trigger-Matrix 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` Trigger-Quelle

<code>src</code>	Beschreibung
<code>EIB7_AT_TrInput1</code>	Trigger-Eingang Kanal 1
<code>EIB7_AT_TrInput2</code>	Trigger-Eingang Kanal 2
<code>EIB7_AT_TrInput3</code>	Trigger-Eingang Kanal 3
<code>EIB7_AT_TrInput4</code>	Trigger-Eingang Kanal 4
<code>EIB7_AT_TrSW1</code>	Software-Trigger Kanal 1
<code>EIB7_AT_TrSW2</code>	Software-Trigger Kanal 2
<code>EIB7_AT_TrSW3</code>	Software-Trigger Kanal 3
<code>EIB7_AT_TrSW4</code>	Software-Trigger Kanal 4
<code>EIB7_AT_TrRI</code>	Referenzimpuls der entsprechenden Achse
<code>EIB7_AT_TrRImaskedCH1</code>	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
<code>EIB7_AT_TrIC</code>	Interval Counter
<code>EIB7_AT_TrPuls</code>	Trigger-Pulszähler
<code>EIB7_AT_TrTimer</code>	Timer Trigger

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid` Parameter ungültig

**3.5.33 Trigger-Quellen aktivieren**

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

**Funktion**

```
EIB7_ERR EIB7GlobalTrigger-Enable ( EIB7_HANDLE eib,
                                   EIB7_MODE mode,
                                   unsigned long src
                                   )
```



**Parameter**

`eib` EIB-Handle  
`mode` Trigger-Quellen aktivieren oder deaktivieren

<code>mode</code>	Beschreibung
<code>EIB7_MD_Enable</code>	Trigger-Quelle aktivieren
<code>EIB7_MD_Disable</code>	Trigger-Quelle deaktivieren

`src` Trigger-Quelle

<code>src</code>	Beschreibung
<code>EIB7_TS_TrInput1</code>	Trigger-Eingang Kanal 1
<code>EIB7_TS_TrInput2</code>	Trigger-Eingang Kanal 2
<code>EIB7_TS_TrInput3</code>	Trigger-Eingang Kanal 3
<code>EIB7_TS_TrInput4</code>	Trigger-Eingang Kanal 4
<code>EIB7_TS_TrRI1</code>	Referenzimpuls Achse 1
<code>EIB7_TS_TrRI2</code>	Referenzimpuls Achse 2
<code>EIB7_TS_TrRI3</code>	Referenzimpuls Achse 3
<code>EIB7_TS_TrRI4</code>	Referenzimpuls Achse 4
<code>EIB7_TS_TrRImaskedCH1</code>	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
<code>EIB7_TS_TrIC</code>	Interval Counter
<code>EIB7_TS_TrPuls</code>	Trigger-Pulszähler
<code>EIB7_TS_TrTimer</code>	Timer Trigger
<code>EIB7_TS_All</code>	Alle Trigger-Quellen

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid` Parameter ungültig

### 3.5.34 Pulszähler konfigurieren

Für den Pulszähler können ein Trigger-Signal und ein Startsignal gewählt werden. Das Startsignal gibt den Pulszähler frei. Dieser wird fortan mit jedem Impuls am Trigger-Signal dekrementiert, bis der Wert Null erreicht ist. Anschließend werden alle weiteren Trigger-Impulse gesperrt. Wenn die Funktion erneut ausgeführt wird, 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 Pulszähler

start	Beschreibung
EIB7_PS_Tr Input1	Trigger-Eingang Kanal 1
EIB7_PS_Tr Input2	Trigger-Eingang Kanal 2
EIB7_PS_Tr Input3	Trigger-Eingang Kanal 3
EIB7_PS_Tr Input4	Trigger-Eingang Kanal 4
EIB7_PS_Tr RI1	Referenzimpuls Achse 1
EIB7_PS_Tr RI2	Referenzimpuls Achse 2
EIB7_PS_Tr RI3	Referenzimpuls Achse 3
EIB7_PS_Tr RI4	Referenzimpuls Achse 4
EIB7_PS_Tr SW1	Software-Trigger Kanal 1
EIB7_PS_Tr SW2	Software-Trigger Kanal 2
EIB7_PS_Tr SW3	Software-Trigger Kanal 3
EIB7_PS_Tr SW4	Software-Trigger Kanal 4
EIB7_PS_Tr RI maskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_PS_Tr IC	Interval Counter

trigger

Trigger-Signal für den Pulszähler

trigger	Beschreibung
EIB7_PT_TrInput1	Trigger-Eingang Kanal 1
EIB7_PT_TrInput2	Trigger-Eingang Kanal 2
EIB7_PT_TrInput3	Trigger-Eingang Kanal 3
EIB7_PT_TrInput4	Trigger-Eingang Kanal 4
EIB7_PT_TrRI1	Referenzimpuls Achse 1
EIB7_PT_TrRI2	Referenzimpuls Achse 2
EIB7_PT_TrRI3	Referenzimpuls Achse 3
EIB7_PT_TrRI4	Referenzimpuls Achse 4
EIB7_PT_TrRImaskedCH1	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
EIB7_PT_TrIC	Interval Counter
EIB7_PT_TrTimer	Timer Trigger

count

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

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid      Parameter ungültig

### 3.5.35 Interpolationsfaktor für 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 Flankenbewertung.

#### 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                    Flankenbewertung

edge	Beschreibung
EIB7_ICE_1x	Flankenbewertung 1x
EIB7_ICE_2x	Flankenbewertung 2x
EIB7_ICE_4x	Flankenbewertung 4x

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid      Parameter ungültig

### 3.5.36 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 Trigger-Impuls ausgegeben. Diese Position ist einstellbar. Der zweite Modus ermöglicht eine Triggerung in festen Abständen. Der erste Trigger-Impuls wird an der Startposition ausgegeben. Anschließend wird in festen, über den Parameter `interval` einstellbaren Abständen ein Trigger-Impuls erzeugt. Alternativ lässt sich auch die aktuelle Position als Startposition verwenden. Bevor der Interval Counter Trigger neu konfiguriert werden kann, muss die Trigger-Funktion mit dem Modus `EIB7_ICM_Disable` deaktiviert werden.

#### Funktion

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

#### Parameter

`eib`                    EIB-Handle  
`mode`                    Trigger-Modus

mode	Beschreibung
<code>EIB7_ICM_Disable</code>	Keine Triggerung
<code>EIB7_ICM_Single</code>	Triggerung nur an einer festen Position
<code>EIB7_ICM_Periodic</code>	Periodische Triggerung in festen Abständen

`start`                    Start der Triggerung

start	Beschreibung
<code>EIB7_ICS_Current</code>	Triggerung startet bei aktueller Position
<code>EIB7_ICS_StartPos</code>	Triggerung startet bei Startposition

`startpos`                Positionswert für ersten Trigger-Impuls

`interval`                Abstand zwischen zwei Trigger-Impulsen in Zählschritten (6)

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid`      Parameter ungültig

### 3.5.37 Abschlusswiderstände einstellen

Die Abschlusswiderstände für die Inkrementalsignale der Messgeräteeingänge können deaktiviert werden. Diese Einstellung gilt immer für alle  $1V_{SS}$ -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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_CantChIncInpTrm    Modus kann nicht verändert werden

### 3.5.38 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.39 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

### 3.5.40 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, sollten 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`                   Abschlusswiderstände aktivieren oder deaktivieren

mode	Beschreibung
EIB7_MD_Disable	Datenübertragung stoppen
EIB7_MD_Enable	Datenübertragung starten

`offset`                Offset für erstes 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_CantOpenSocket`    Interner Fehler (Socket-Fehler)  
`EIB7_CantStartThread`    Interner Fehler (Thread-Fehler)  
`EIB7_OpModeBlocked`    EIB 74x befindet sich nicht im Betriebsmodus "Polling"  
`EIB7_RecDataReadErr`    Daten können nicht aus der EIB 74x gelesen werden  
`EIB7_ParamInvalid`      Parameter ungültig



### 3.5.41 Recording - Status prüfen

Der Status im Betriebsmodus "Recording" kann ausgelesen werden. Ferner kann der aktuelle Inhalt im Pufferspeicher bestimmt werden. Dies ist auch während der Aufzeichnung im Betriebsmodus "Recording" 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 Zielvariable für Anzahl der Datenpakete im Puffer  
**status**                            *[Rückgabewert]* Pointer auf Zielvariable für Status

status	Beschreibung
0	Betriebsmodus "Recording" deaktiviert
1	Betriebsmodus "Recording" aktiviert
2	Daten werden übertragen
3	Warten auf Verbindungsaufbau für Datenübertragung

**progress**                            *[Rückgabewert]* Pointer auf Zielvariable für 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.42 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 Datenpakets. Aus diesem Grund muss zuerst das Datenpaket konfiguriert werden.

#### Funktion

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

#### Parameter

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

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7_InvalidPacket	Aktuelle Konfiguration für Datenpaket ungültig
--------------------	--

### 3.5.43 Streaming - Status prüfen

Während der Betriebsmodus "Streaming" 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 Betriebsmodus "Streaming" in Bytes angegeben.

#### Funktion

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

#### Parameter

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

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.44 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. Wenn 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-Einträgen aufnehmen kann. Der Inhalt eines Eintrags im FIFO entspricht dem aktuell konfigurierten Datenpaket ohne angehängte 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**

<code>eib</code>	EIB-Handle
<code>data</code>	<i>[Rückgabewert]</i> Pointer auf Zielspeicher
<code>cnt</code>	Anzahl der zu lesenden Einträge ( $\geq 0$ )
<code>entries</code>	<i>[Rückgabewert]</i> Anzahl der kopierten Einträge
<code>timeout</code>	Timeout in Millisekunden

<code>timeout</code>	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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_FIFOEmpty</code>	Keine Daten im FIFO
<code>EIB7_ElementTypeInv</code>	Interner Fehler
<code>EIB7_FIFOoverflow</code>	FIFO-Überlauf seit dem letzten Aufruf der Funktion (Daten gingen verloren)

**3.5.45 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**

<code>eib</code>	EIB-Handle
<code>size</code>	<i>[Rückgabewert]</i> Pointer auf Variable für Größe eines FIFO-Elements in Bytes

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.46 Zugriff auf 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 Datenpakets 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 Feinauswahl 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 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 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_FieldNotAvail Angegebenes Feld kann nicht gefunden werden

### 3.5.47 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-Einträgen aufnehmen kann. Der Inhalt eines Eintrags entspricht dem aktuell konfigurierten Datenpaket ohne angehängte 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) 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)

### 3.5.48 Größe eines FIFO-Elements nach 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 Größe eines FIFO-Elements in Bytes

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.49 Zugriff auf 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 Datenpakets 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 Feinauswahl 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 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 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_FieldNotAvail      Angegebenes Feld kann nicht gefunden werden

### 3.5.50 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 Anzahl der FIFO-Elemente

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.51 FIFO löschen

Der Inhalt des FIFOs wird gelöscht. Dieses Kommando hat keine Auswirkung, wenn der Betriebsmodus "Polling" 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.52 FIFO-Größe einstellen

Die Größe des FIFOs wird neu festgelegt. Alle Daten im FIFO werden gelöscht. Die Größe kann nur im Betriebsmodus "Polling" 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_SoftRTen` Betriebsmodus "Soft Realtime" ist aktiviert

**3.5.53 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 Variable für FIFO-Größe in Bytes

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.54 Callback-Mechanismus aktivieren**

Der Callback-Mechanismus wird aktiviert bzw. deaktiviert und gegebenenfalls der Funktions-Pointer 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
<code>EIB7_MD_Disable</code>	Callback-Mechanismus deaktivieren
<code>EIB7_MD_Enable</code>	Callback-Mechanismus aktivieren

`threshold` Anzahl der Elemente im FIFO, ab dem der Callback-Mechanismus auslöst (> 0)  
`handler` Pointer auf 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**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**

<code>eib</code>	EIB-Handle
<code>cnt</code>	Anzahl der Elemente im FIFO
<code>data</code>	Pointer auf Benutzerdaten

**3.5.55 Trigger-Quelle für Hilfsachse wählen**

Das Trigger-Signal 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` Trigger-Quelle

<code>src</code>	Beschreibung
<code>EIB7_AT_TrInput1</code>	Trigger-Eingang Kanal 1
<code>EIB7_AT_TrInput2</code>	Trigger-Eingang Kanal 2
<code>EIB7_AT_TrInput3</code>	Trigger-Eingang Kanal 3
<code>EIB7_AT_TrInput4</code>	Trigger-Eingang Kanal 4
<code>EIB7_AT_TrSW1</code>	Software-Trigger Kanal 1
<code>EIB7_AT_TrSW2</code>	Software-Trigger Kanal 2
<code>EIB7_AT_TrSW3</code>	Software-Trigger Kanal 3
<code>EIB7_AT_TrSW4</code>	Software-Trigger Kanal 4
<code>EIB7_AT_TrRI</code>	Referenzimpuls der entsprechenden Achse
<code>EIB7_AT_TrRImaskedCH1</code>	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
<code>EIB7_AT_TrIC</code>	Interval Counter
<code>EIB7_AT_TrPuls</code>	Trigger-Pulszähler
<code>EIB7_AT_TrTimer</code>	Timer Trigger

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid` Parameter ungültig

**3.5.56 Position der Hilfsachse abfragen**

Der aktuelle Positionswert wird ausgelesen. Zusätzlich wird ein Statuswort übertragen, aus dem mögliche Positionsfehler hervorgehen. Die Positionsabfrage kann nur im Betriebsmodus "Polling" 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 Statuswort
pos	[Rückgabewert] Pointer auf Zielvariable für Position

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

**3.5.57 Daten der Hilfsachse auslesen**

Die aktuelle Position und einige zusätzliche Parameter werden bestimmt und ausgelesen. Die Funktion darf nur im Betriebsmodus "Polling" 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 Trigger-Quelle 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	[Rückgabewert] Pointer auf Zielvariable für Statuswort
pos	[Rückgabewert] Pointer auf Zielvariable für Position
ref	[Rückgabewert] Pointer auf Zielvariable für Referenzposition
timestamp	[Rückgabewert] Pointer auf Zielvariable für Timestamp-Wert
counter	[Rückgabewert] Pointer auf Zielvariable für Trigger Counter

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

### 3.5.58 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.59 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.60 Trigger-Fehler der Hilfsachse quittieren

Die Fehlermeldungen für nicht erkannte Trigger-Ereignisse 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.61 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.62 Status der Referenzfahrt für Hilfsachse prüfen

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

active	Beschreibung
EIB7_MD_Disable	Referenzfahrt inaktiv
EIB7_MD_Enable	Referenzfahrt aktiv

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).



### 3.5.63 Referenzfahrt für Hilfsachse starten

Nach Aufruf dieses Kommandos wird die Referenzposition beim Überfahren der nächsten Referenzmarke 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.64 Referenzfahrt für 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.5.65 Timestamp für 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_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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.5.66 Trigger-Flanke für Referenzimpuls der Hilfsachse einstellen**

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

**Funktion**

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

**Parameter**

eib EIB-Handle  
 edge Aktive Trigger-Flanke für 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid Parameter ungültig

## 3.6 Achsfunktionen

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

Alle Achsfunktionen können neben den Standard-Rückgabewerten ("Parameter und Rückgabewerte") zusätzlich weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

### 3.6.1 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.

Die Parameter `bandwidth` und `comp` wirken 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 Referenzwerts 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 Inkrementalsignalen (1 V <sub>SS</sub> )
EIB7_IT_Incremental_11u	Messgerät mit Inkrementalsignalen (11 µA)
EIB7_IT_EnDat21	Messgerät mit EnDat 2.1-Interface
EIB7_IT_EnDat01	Messgerät mit EnDat 2.1-Interface und Inkrementalsignalen (1 V <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						
	<table border="1"> <thead> <tr> <th>homing</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>EIB7_HS_None</td> <td>Homing-Signal nicht ausgewertet</td> </tr> <tr> <td>EIB7_HS_Available</td> <td>Homing-Signal ausgewertet</td> </tr> </tbody> </table>	homing	Beschreibung	EIB7_HS_None	Homing-Signal nicht ausgewertet	EIB7_HS_Available	Homing-Signal ausgewertet
homing	Beschreibung						
EIB7_HS_None	Homing-Signal nicht ausgewertet						
EIB7_HS_Available	Homing-Signal ausgewertet						
limit	Limitsignalauswertung aktivieren oder deaktivieren						
	<table border="1"> <thead> <tr> <th>limit</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>EIB7_LS_None</td> <td>Limitsignal nicht ausgewertet</td> </tr> <tr> <td>EIB7_LS_Available</td> <td>Limitsignal ausgewertet</td> </tr> </tbody> </table>	limit	Beschreibung	EIB7_LS_None	Limitsignal nicht ausgewertet	EIB7_LS_Available	Limitsignal ausgewertet
limit	Beschreibung						
EIB7_LS_None	Limitsignal nicht ausgewertet						
EIB7_LS_Available	Limitsignal ausgewertet						
compensation	Online-Kompensation aktivieren oder deaktivieren						
	<table border="1"> <thead> <tr> <th>compensation</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>EIB7_CS_None</td> <td>Signalkompensation deaktiviert</td> </tr> <tr> <td>EIB7_CS_CompActive</td> <td>Signalkompensation aktiviert</td> </tr> </tbody> </table>	compensation	Beschreibung	EIB7_CS_None	Signalkompensation deaktiviert	EIB7_CS_CompActive	Signalkompensation aktiviert
compensation	Beschreibung						
EIB7_CS_None	Signalkompensation deaktiviert						
EIB7_CS_CompActive	Signalkompensation aktiviert						
bandwidth	Eingangsbandbreite für Inkrementalsignale (high/low)						
	<table border="1"> <thead> <tr> <th>bandwidth</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>EIB7_BW_High</td> <td>Hohe Eingangsbandbreite für 1 V<sub>SS</sub>-Signale</td> </tr> <tr> <td>EIB7_BW_Low</td> <td>Niedrige Eingangsbandbreite für 1 V<sub>SS</sub>-Signale</td> </tr> </tbody> </table>	bandwidth	Beschreibung	EIB7_BW_High	Hohe Eingangsbandbreite für 1 V <sub>SS</sub> -Signale	EIB7_BW_Low	Niedrige Eingangsbandbreite für 1 V <sub>SS</sub> -Signale
bandwidth	Beschreibung						
EIB7_BW_High	Hohe Eingangsbandbreite für 1 V <sub>SS</sub> -Signale						
EIB7_BW_Low	Niedrige Eingangsbandbreite für 1 V <sub>SS</sub> -Signale						
EnDatclock	EnDat-Taktrate						
recovery	"Recovery Time I" für EnDat 2.2						
	<table border="1"> <thead> <tr> <th>recovery</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>EIB7_RT_Long</td> <td>Lange "Recovery Time I" nach EnDat-Spezifikation</td> </tr> <tr> <td>EIB7_RT_Short</td> <td>Kurze "Recovery Time I" nach EnDat-Spezifikation</td> </tr> </tbody> </table>	recovery	Beschreibung	EIB7_RT_Long	Lange "Recovery Time I" nach EnDat-Spezifikation	EIB7_RT_Short	Kurze "Recovery Time I" nach EnDat-Spezifikation
recovery	Beschreibung						
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						
	<table border="1"> <thead> <tr> <th>calculation</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>EIB7_CT_Long</td> <td>Lange "Calculation Time"</td> </tr> <tr> <td>EIB7_CT_Short</td> <td>Kurze "Calculation Time"</td> </tr> </tbody> </table>	calculation	Beschreibung	EIB7_CT_Long	Lange "Calculation Time"	EIB7_CT_Short	Kurze "Calculation Time"
calculation	Beschreibung						
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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_ParamInvalid</code>	Parameter ungültig
<code>EIB7_InvInterface</code>	Schnittstellentyp ungültig
<code>EIB7_InvRefMarkOpt</code>	Referenzmarke ungültig
<code>EIB7_InvDistCodeRef</code>	Parameter für abstandscodierte Referenzmarken ungültig ( <code>linecount</code> , <code>increment</code> )
<code>EIB7_ConfOptIncons</code>	Parameter in dieser Form nicht kombinierbar
<code>EIB7_AccNotAllowed</code>	Zugriff verweigert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)



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

### 3.6.2 Trigger-Quelle für Achse wählen

Das Trigger-Signal 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`             Trigger-Quelle

<code>src</code>	Beschreibung
<code>EIB7_AT_TrInput1</code>	Trigger-Eingang Kanal 1
<code>EIB7_AT_TrInput2</code>	Trigger-Eingang Kanal 2
<code>EIB7_AT_TrInput3</code>	Trigger-Eingang Kanal 3
<code>EIB7_AT_TrInput4</code>	Trigger-Eingang Kanal 4
<code>EIB7_AT_TrSW1</code>	Software-Trigger Kanal 1
<code>EIB7_AT_TrSW2</code>	Software-Trigger Kanal 2
<code>EIB7_AT_TrSW3</code>	Software-Trigger Kanal 3
<code>EIB7_AT_TrSW4</code>	Software-Trigger Kanal 4
<code>EIB7_AT_TrRI</code>	Referenzimpuls der entsprechenden Achse
<code>EIB7_AT_TrRImaskedCH1</code>	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
<code>EIB7_AT_TrIC</code>	Interval Counter
<code>EIB7_AT_TrPuls</code>	Trigger-Pulszähler
<code>EIB7_AT_TrTimer</code>	Timer Trigger

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid`     Parameter ungültig

### 3.6.3 Trigger-Flanke für Referenzimpuls einstellen

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

**Funktion**

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

**Parameter**

axis                    AXIS-Handle  
 edge                    Aktive Trigger-Flanke für 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid        Parameter ungültig

**3.6.4 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 zurückgesetzt. 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_InvInterface        Schnittstellentyp ungültig

**3.6.5 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 Betriebsmodus "Polling" 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**

<code>axis</code>	AXIS-Handle
<code>status</code>	<i>[Rückgabewert]</i> Pointer auf Zielvariable für Statuswort
<code>pos</code>	<i>[Rückgabewert]</i> Pointer auf Zielvariable für Position

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_CantLatchPos</code>	Position kann nicht bestimmt werden
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (nur EnDat-Messgeräte)
<code>EIB7_NotInitialized</code>	Achse nicht konfiguriert

**3.6.6 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 Betriebsmodus "Polling" 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 Trigger-Quelle 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**

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

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_ParamInvalid</code>	Parameter ungültig (evtl. Achse nicht für inkrementale Messgeräte konfiguriert)
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (nur für EnDat-Messgeräte)
<code>EIB7_NotInitialized</code>	Achse nicht konfiguriert

**3.6.7 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**

<code>axis</code>	AXIS-Handle
-------------------	-------------

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_CantClearEnc</code>	Fehler kann nicht gelöscht werden
--------------------------------	-----------------------------------

### 3.6.8 Trigger-Fehler quittieren

Die Fehlermeldung für das Trigger-Interface wird quittiert. Der Trigger-Fehler 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.6.9 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.6.10 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 Betriebsmodus "Polling" 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat konfiguriert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts
<code>EIB7_NotInitialized</code>	Achse nicht konfiguriert

### 3.6.11 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für inkrementale Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht konfiguriert

### 3.6.12 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 Referenzwerts"

Dieses Kommando ist nur für Achsen zulässig, die für Messgeräte mit abstandscodierten 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für inkrementale Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht konfiguriert

### 3.6.13 Referenzfahrt starten

Nach Aufruf dieses Kommandos wird die Referenzposition beim Überfahren der nächsten Referenzmarke 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

<code>ref</code>	Beschreibung
<code>EIB7_RP_RefPos1</code>	Eine Referenzposition speichern
<code>EIB7_RP_RefPos2</code>	Zwei Referenzpositionen speichern

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_InvInterface`      Achse nicht für inkrementale Messgeräte konfiguriert  
`EIB7_NotInitialized`    Achse nicht konfiguriert  
`EIB7_ParamInvalid`      Parameter ist keine gültige Option für Referenzmarken

### 3.6.14 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

- EIB7\_InvInterface        Achse nicht für inkrementale Messgeräte konfiguriert
- EIB7\_NotInitialized    Achse nicht konfiguriert
- EIB7\_ParamInvalid      Ungültiger Parameter für Achse

### 3.6.15 Referenzfahrt Status prüfen

Der Status der Referenzfahrt wird ausgegeben. Hiermit lässt sich prüfen, ob die Referenzfahrt noch aktiv ist, oder ob 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

active	Beschreibung
EIB7_MD_Enable	Referenzfahrt aktiv
EIB7_MD_Disable	Referenzfahrt abgeschlossen

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.6.16 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_InvInterface        Achse nicht für inkrementale Messgeräte konfiguriert  
EIB7\_ParamInvalid        Parameter ungültig

### 3.6.17 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 Betriebsmodus "Polling" 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 EIB7EnDat21GetPosition ( EIB7_AXIS      axis,
                                   unsigned short*   status,
                                   ENCODER_POSITION* pos
                                   )
```

#### Parameter

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



**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)

**3.6.18 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 Betriebsmodus "Polling" 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 EIB7EnDat21SelectMemRange ( EIB7_AXIS      axis,
                                     unsigned char  mrs
                                     )
```

**Parameter**

<code>axis</code>	AXIS-Handle
<code>mrs</code>	MRS-Code für Speicherbereich

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)

### 3.6.19 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 Betriebsmodus "Polling" 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

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

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)

### 3.6.20 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 Betriebsmodus "Polling" 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**

<code>axis</code>	AXIS-Handle
<code>addr</code>	Speicheradresse innerhalb des aktiven Speicherblocks
<code>data</code>	<i>[Rückgabewert]</i> Pointer auf Variable für empfangenes Datenwort

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)

**3.6.21 EnDat 2.1 - Messgerät Reset**

Das EnDat-Reset-Kommando wird zum Messgerät gesendet. Das Messgerät führt einen Reset durch und ist für eine bestimmte Zeit nicht ansprechbar. Weitere Angaben dazu können dem Datenblatt des Messgeräts entnommen werden.

Diese Funktion darf nur im Betriebsmodus "Polling" 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 EIB7EnDat21ResetEncoder ( EIB7_AXIS axis )
```

**Parameter**

<code>axis</code>	AXIS-Handle
-------------------	-------------

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)

### 3.6.22 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.

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

Diese Funktion darf nur im Betriebsmodus "Polling" 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 EIB7EnDat21ReadTestValue ( EIB7_AXIS          axis,
                                     unsigned long*      high,
                                     unsigned long*      low
                                     )
```

#### Parameter

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

#### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

```
EIB7_InvInterface        Achse nicht für EnDat-Messgeräte konfiguriert
EIB7_NotInitialized     Achse nicht initialisiert
EIB7_EncPwrSuppErr      Fehler in Spannungsversorgung des Messgeräts
                        (Messgerät nicht betriebsbereit)
EIB7_EnDatErrII         EnDat-Fehler Typ II aufgetreten
EIB7_EnDatIfBusy        EnDat-Master nicht betriebsbereit
EIB7_EnDatXmitErr       Fehler bei Datenübertragung (Messgerät evtl. nicht
                        angeschlossen)
```

### 3.6.23 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 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 EIB7EnDat21WriteTestCommand ( EIB7_AXIS          axis,
                                         unsigned char      port
                                         )
```

**Parameter**

`axis`                    AXIS-Handle  
`port`                     Port-Adresse für Testkommando

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_InvInterface`        Achse nicht für EnDat-Messgeräte konfiguriert  
`EIB7_NotInitialized`     Achse nicht initialisiert  
`EIB7_EncPwrSuppErr`      Fehler in Spannungsversorgung des Messgeräts  
                               (Messgerät nicht betriebsbereit)  
`EIB7_EnDatErrII`         EnDat-Fehler Typ II aufgetreten  
`EIB7_EnDatIfBusy`        EnDat-Master nicht betriebsbereit  
`EIB7_EnDatXmitErr`       Fehler bei Datenübertragung (Messgerät evtl. nicht  
                               angeschlossen)

**3.6.24 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 Betriebsmodus "Polling" 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 Statuswort  
`pos`                     [Rückgabewert] Pointer auf Variable für Positionswert  
`ai1`                    [Rückgabewert] Pointer auf Struktur für EnDat-Zusatzinformation 1  
`ai2`                    [Rückgabewert] Pointer auf Struktur für EnDat-Zusatzinformation 2

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7_InvInterface	Achse nicht für EnDat-Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
EIB7_EnDatErrII	EnDat-Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat-Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)
EIB7_EnDat22NotSupp	Messgerät unterstützt keine EnDat 2.2-Befehle oder Achse nicht für EnDat 2.2-Betrieb konfiguriert

### 3.6.25 EnDat 2.2 - Position und Zusatzinformation lesen und Speicherbereich wählen

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 Betriebsmodus "Polling" 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*     ail,
                                       ENDAT_ADDINFO*     ai2,
                                       unsigned char      mrs,
                                       unsigned char      block
                                       )
```

#### Parameter

axis	AXIS-Handle
status	[Rückgabewert] Pointer auf Variable für Statuswort
pos	[Rückgabewert] Pointer auf Variable für Positionswert
ail	[Rückgabewert] Pointer auf Struktur für EnDat-Zusatzinformation 1
ai2	[Rückgabewert] Pointer auf Struktur für EnDat-Zusatzinformation 2
mrs	MRS-Code für Speicherbereich
block	Block-Adresse für Speicherbereiche der "Sektion 2"

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)
<code>EIB7_EnDat22NotSupp</code>	Messgerät unterstützt keine EnDat 2.2-Befehle oder Achse nicht für EnDat 2.2-Betrieb konfiguriert

**3.6.26 EnDat 2.2 - Position und Zusatzinformation lesen und Daten senden**

Die Position und Zusatzinformation eines EnDat22-Messgeräts wird übertragen (siehe "EnDat 2.2 - Position und Zusatzinformation lesen", Seite 117). 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 Betriebsmodus "Polling" 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**

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

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7_InvInterface	Achse nicht für EnDat-Messgeräte konfiguriert
EIB7_NotInitialized	Achse nicht initialisiert
EIB7_EncPwrSuppErr	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
EIB7_EnDatErrII	EnDat-Fehler Typ II aufgetreten
EIB7_EnDatIfBusy	EnDat-Master nicht betriebsbereit
EIB7_EnDatXmitErr	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)
EIB7_EnDat22NotSupp	Messgerät unterstützt keine EnDat 2.2-Befehle oder Achse nicht für EnDat 2.2-Betrieb konfiguriert

### 3.6.27 EnDat 2.2 - Position und Zusatzinformation lesen und Daten senden

Die Position und Zusatzinformation eines EnDat22-Messgeräts wird übertragen (siehe "EnDat 2.2 - Position und Zusatzinformation lesen", Seite 117). 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 Betriebsmodus "Polling" 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

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



**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)
<code>EIB7_EnDat22NotSupp</code>	Messgerät unterstützt keine EnDat 2.2-Befehle oder Achse nicht für EnDat 2.2-Betrieb konfiguriert

**3.6.28 EnDat 2.2 - Position und Zusatzinformation lesen und Testkommando senden**

Die Position und Zusatzinformation eines EnDat22-Messgeräts wird übertragen (siehe "EnDat 2.2 - Position und Zusatzinformation lesen", Seite 117). Der Parameter `port` beinhaltet die Port-Adresse für das Testkommando.

Diese Funktion darf nur im Betriebsmodus "Polling" 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**

<code>axis</code>	AXIS-Handle
<code>status</code>	<i>[Rückgabewert]</i> Pointer auf Variable für Statuswort
<code>pos</code>	<i>[Rückgabewert]</i> Pointer auf Variable für Positionswert
<code>ai1</code>	<i>[Rückgabewert]</i> Pointer auf Struktur für EnDat-Zusatzinformation 1
<code>ai2</code>	<i>[Rückgabewert]</i> Pointer auf Struktur für EnDat-Zusatzinformation 2
<code>port</code>	Port-Adresse für Testkommando

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)
<code>EIB7_EnDat22NotSupp</code>	Messgerät unterstützt keine EnDat 2.2-Befehle oder Achse nicht für EnDat 2.2-Betrieb konfiguriert

### 3.6.29 EnDat 2.2 - Position und Zusatzinformation lesen und Fehler-Reset senden

Die Position und Zusatzinformation eines EnDat22-Messgeräts wird übertragen (siehe "EnDat 2.2 - Position und Zusatzinformation lesen", Seite 117). Außerdem wird der Fehlerspeicher des EnDat22-Messgeräts gelöscht.

Diese Funktion darf nur im Betriebsmodus "Polling" 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

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

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

<code>EIB7_InvInterface</code>	Achse nicht für EnDat-Messgeräte konfiguriert
<code>EIB7_NotInitialized</code>	Achse nicht initialisiert
<code>EIB7_EncPwrSuppErr</code>	Fehler in Spannungsversorgung des Messgeräts (Messgerät nicht betriebsbereit)
<code>EIB7_EnDatErrII</code>	EnDat-Fehler Typ II aufgetreten
<code>EIB7_EnDatIfBusy</code>	EnDat-Master nicht betriebsbereit
<code>EIB7_EnDatXmitErr</code>	Fehler bei Datenübertragung (Messgerät evtl. nicht angeschlossen)
<code>EIB7_EnDat22NotSupp</code>	Messgerät unterstützt keine EnDat 2.2-Befehle oder Achse nicht für EnDat 2.2-Betrieb konfiguriert

**3.6.30 EnDat 2.2 - Zusatzinformation wählen**

Die Zusatzinformationen für ein EnDat 2.2-Messgerät können konfiguriert werden. Die Konfiguration muss im Betriebsmodus "Polling" erfolgen. Die Zusatzinformationen werden in den Betriebsmodi "Soft Realtime", "Streaming" und "Recording" übertragen.

Die entsprechenden Zusatzinformationen werden im Messgerät ausgewählt, wenn aus dem Betriebsmodus "Polling" in einen anderen Betriebsmodus 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

Zusatzinformation 1 für EnDat 2.2

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

Zusatzinformation 2 für EnDat 2.2

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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

- EIB7\_ParamInvalid      Parameter ungültig
- EIB7\_InvInterface      Achse nicht für EnDat-Messgeräte konfiguriert
- EIB7\_NotInitialized    Achse nicht initialisiert
- EIB7\_EnDat22NotSupp    Messgerät unterstützt keine EnDat 2.2-Befehle oder Achse nicht für EnDat 2.2-Betrieb konfiguriert

### 3.6.31 EnDat 2.2 - Sequenz für Zusatzinformation wählen

Die Zusatzinformationen für ein EnDat 2.2-Messgerät können konfiguriert werden. Die Konfiguration muss im Betriebsmodus "Polling" erfolgen. Die Zusatzinformationen werden in den Betriebsmodi "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 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 ( $\leq 9$ )

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

### 3.6.32 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 Betriebsmodus "Polling" 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 Statuswort der EnDat-Position
posEnDat	[Rückgabewert] Pointer auf Variable für EnDat-Positionswert
statusIncr	[Rückgabewert] Pointer auf Variable für Statuswort der Inkrementalposition
posIncr	[Rückgabewert] Pointer auf Variable für inkrementalen Positionswert

### Rückgabewert

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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



### 3.6.33 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.6.34 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 Status der Spannungsversorgung

<code>power</code>	Beschreibung
<code>EIB7_MD_Disable</code>	Spannungsversorgung ausgeschaltet
<code>EIB7_MD_Enable</code>	Spannungsversorgung eingeschaltet

`err` [Rückgabewert] Pointer auf Variable für Überstromfehler

<code>err</code>	Beschreibung
<code>EIB7_PF_None</code>	Kein Fehler
<code>EIB7_PF_Overcurrent</code>	Spannungsversorgung wegen Überstrom deaktiviert

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

**3.6.35 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

<code>mode</code>	Beschreibung
<code>EIB7_MD_Disable</code>	Timestamp deaktivieren
<code>EIB7_MD_Enable</code>	Timestamp aktivieren

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Alle möglichen Werte sind bei den Standard-Rückgabewerten aufgelistet (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50).

### 3.7 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 neben den Standard-Rückgabewerten ("Parameter und Rückgabewerte") zusätzlich weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

#### 3.7.1 Eingangsport konfigurieren

Mit dieser Funktion kann der Modus eines Eingangsports konfiguriert werden. Der Port kann als Trigger-Eingang 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 Trigger-Eingang oder logischer Eingangsport

mode	Beschreibung
EIB7_IOM_Trigger	Trigger-Eingang
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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid Parameter ungültig

### 3.7.2 Ausgangsport konfigurieren

Mit dieser Funktion kann der Modus eines Ausgangsports konfiguriert werden. Der Port kann als Trigger-Ausgang oder als logischer Ausgang verwendet werden. Außerdem 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` Trigger-Ausgang oder logischer Ausgangsport

mode	Beschreibung
EIB7_IOM_Trigger	Trigger-Ausgang
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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid      Parameter ungültig

### 3.7.3 Trigger-Quelle für Trigger-Ausgang wählen

Das Trigger-Signal für den Trigger-Ausgang kann aus verschiedenen Quellen gewählt werden. Diese Einstellung ist nur im Betriebsmodus "Polling" und für Trigger-Ausgänge möglich.

#### Funktion

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

**Parameter**

`io` IO-Handle  
`src` Trigger-Quelle

<code>src</code>	Beschreibung
<code>EIB7_OT_TrgInSync</code>	Trigger-Eingang synchronisiert
<code>EIB7_OT_TrgInAsync</code>	Trigger-Eingang nicht synchronisiert
<code>EIB7_OT_TrgSW1</code>	Software-Trigger Kanal 1
<code>EIB7_OT_TrgSW2</code>	Software-Trigger Kanal 2
<code>EIB7_OT_TrgSW3</code>	Software-Trigger Kanal 3
<code>EIB7_OT_TrgSW4</code>	Software-Trigger Kanal 4
<code>EIB7_OT_TrgRImaskedCH1</code>	Verknüpfter Referenzimpuls von Achse 1 (A&B&RI)
<code>EIB7_OT_TrgIC</code>	Interval Counter
<code>EIB7_OT_TrgPuls</code>	Trigger-Pulszähler
<code>EIB7_OT_TrgTimer</code>	Timer Trigger

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid` Parameter ungültig

**3.7.4 Verzögerungszeit für Trigger-Eingang einstellen**

Für jeden Trigger-Eingang kann separat eine Zeit eingestellt werden, um die ein Trigger-Impuls 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.

**i** Um den Wert für eine Zeitdauer (z. B. für den Parameter `period` des Funktionsaufrufs `EIB7SetTimestampPeriod`) richtig zu berechnen, muss an die Funktion übergeben werden:  
 $period = \text{Zeitintervall in } \mu\text{s} * \text{clock ticks per } \mu\text{s}$   
 Der Wert für "clock ticks per  $\mu\text{s}$ " kann z. B. mit den Funktionen `EIB7GetTimerTriggerTicks` oder `EIB7GetTimestampTicks` ausgelesen werden.

**Funktion**

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

**Parameter**

`io` IO-Handle (nur für Eingänge), weitere Informationen siehe "Timestamp", Seite 30  
`dly` Verzögerungszeit in Taktzyklen ( $\leq 256$ )

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid` Parameter ungültig

**3.7.5 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 Trigger-Eingang oder Trigger-Ausgang 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 Betriebsmodus

mode	Beschreibung
<code>EIB7_IOM_Trigger</code>	Trigger-Port
<code>EIB7_IOM_Logical</code>	Logik-Port

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

level	Beschreibung
<code>EIB7_MD_Disable</code>	Ausgangstreiber deaktivieren
<code>EIB7_MD_Enable</code>	Ausgangstreiber aktivieren

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

`EIB7_ParamInvalid` Parameter ungültig

**3.7.6 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 Trigger-Ausgang eingesetzt, erzeugt die Funktion eine Fehlermeldung.

**Funktion**

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

**Parameter**

io                    IO-Handle  
 level                Logischer Pegel des Ausgangs

level	Beschreibung
0	Low-Pegel
1	High-Pegel

**Rückgabewert**

Der Rückgabewert liefert einen Status für den Funktionsaufruf. Neben den Standard-Rückgabewerten (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

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

**3.7.7 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 Betriebsmodus

mode	Beschreibung
EIB7_IOM_Trigger	Trigger-Eingang
EIB7_IOM_Logical	Logik-Eingang

termination        *[Rückgabewert]* Pointer auf Variable für 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid      Parameter ungültig

## 3.7.8 Konfigurationsdaten für Ausgang lesen

Die Konfigurationsdaten für einen Ausgangsport werden ausgelesen. 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 Betriebsmodus

mode	Beschreibung
EIB7_IOM_Trigger	Trigger-Ausgang
EIB7_IOM_Logical	Logik-Ausgang

`enable`              *[Rückgabewert]* Pointer auf Variable für 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 (siehe "Tabelle 15: Standard-Rückgabewerte", Seite 50) können die nachfolgend aufgelisteten Fehlermeldungen auftreten:

EIB7\_ParamInvalid      Parameter ungültig



## 3.8 Allgemeine Funktionen

Alle allgemeinen Funktionen können neben den Standard-Rückgabewerten ("Parameter und Rückgabewerte") zusätzlich weitere Werte zurückgeben, die für jede Funktion separat aufgeführt werden.

### 3.8.1 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

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

#### Rückgabewert

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

<code>EIB7_NoError</code>	Funktionsaufruf erfolgreich
<code>EIB7_OutOfMemory</code>	Vom System kann nicht genügend Speicher allokiert werden
<code>EIB7_BufferTooSmall</code>	Zielspeicher zu klein

### 3.8.2 Fehlermeldung in Text umwandeln

Ein Fehlercode wird in eine Textmeldung umgewandelt und als C-String zurückgegeben. Im System sind für alle bekannten Fehlercodes eine Kurzbezeichnung und ein beschreibender Text definiert.

Über den Parameter `mnemonic` wird eine Kurzbeschreibung der Fehlermeldung in Textform zurückgegeben (ca. 30 bis 40 Zeichen). Der Parameter `message` enthält eine ausführlichere Beschreibung (ca. 100 bis 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>code</code>	Fehlercode, der in Text umgewandelt wird
<code>mnemonic</code>	<i>[Rückgabewert]</i> Pointer auf Zielspeicher für Kurzbeschreibung
<code>mnemlen</code>	Größe des Zielspeichers <code>mnemonic</code> in Bytes
<code>message</code>	<i>[Rückgabewert]</i> Pointer auf Zielspeicher für Fehler- text
<code>msglen</code>	Größe des Zielspeichers <code>message</code> in Bytes

#### Rückgabewert

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

<code>EIB7_NoError</code>	Funktionsaufruf erfolgreich
<code>EIB7_OutOfMemory</code>	Vom System kann nicht genügend Speicher allokiert werden
<code>EIB7_IllegalParameter</code>	Ungültiger Fehlercode

# HEIDENHAIN

---

## DR. JOHANNES HEIDENHAIN GmbH

Dr.-Johannes-Heidenhain-Straße 5

**83301 Traunreut, Germany**

☎ +49 8669 31-0

FAX +49 8669 32-5061

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)

**NC 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)

**APP programming** ☎ +49 8669 31-3106

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

---

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

