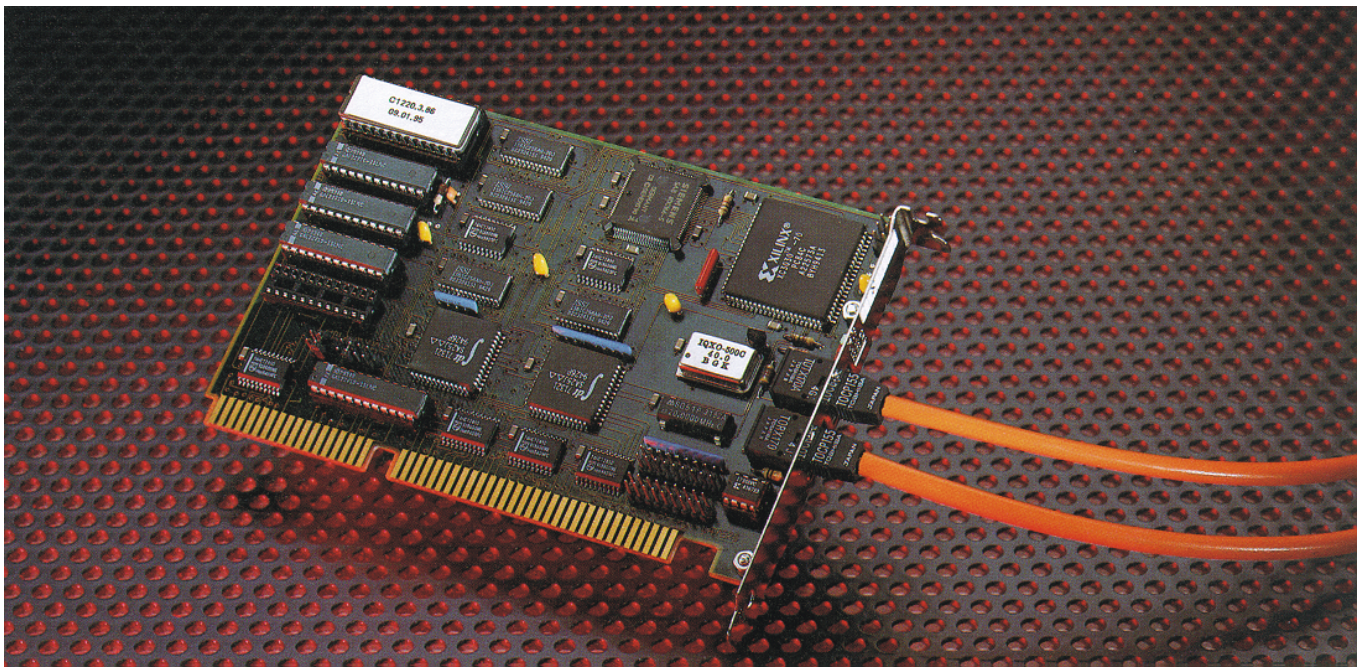


Beckhoff Lightbus - PC Interfacekarte C1220

Technische Hardware Dokumentation

Version 4.00

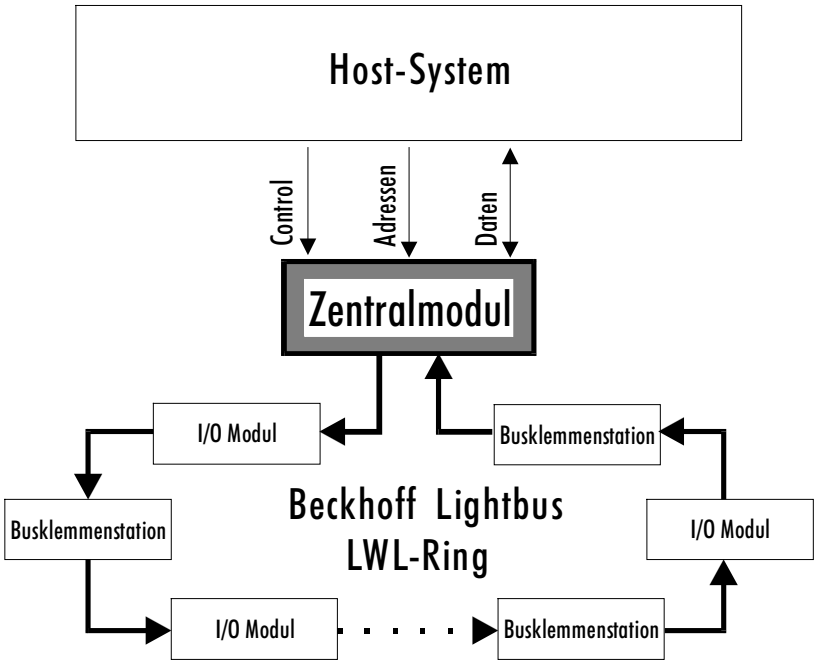
BECKHOFF
INDUSTRIE ELEKTRONIK

Inhaltsverzeichnis

Beckhoff Lightbus Systembeschreibung

Der Beckhoff Lightbus, besteht aus einem intelligenten Zentralmodul und einem Feldbus auf Lichtwellenleiterbasis. h

Beckhoff Lightbus



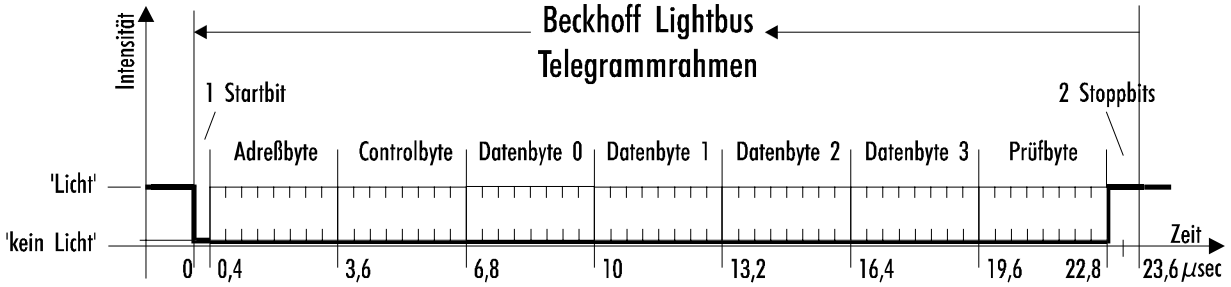
Die Kopplung des Beckhoff Lightbus mit dem Host-System ist über ein DPRAM realisiert. Hierdurch ist eine schnelle und komfortable Kommunikation gewährleistet.

Zur Verarbeitung des Prozessabbildes stehen Buskoppler für Beckhoff Busklemmen und diverse I/O-Module zur Verfügung. Module und Buskoppler sind in einer Ringstruktur miteinander verbunden. Durch den Einsatz des Lichtwellenleiters (LWL) ergibt sich eine geringe Störempfindlichkeit und eine hohe Übertragungsrate von 2,5 MBaud. Im LWL-Ring auftretende Fehler werden vom Zentralmodul erkannt und dem Host-System gemeldet. Implementierte Funktionen zur Ringdiagnose ermöglichen eine schnelle Fehlererkennung und Behebung.

Für die Datenübertragung zwischen Zentralmodul und I/O-Modulen ist ein auf Geschwindigkeit und Einfachheit optimiertes Kommunikationsprotokoll festgelegt. Dieses Kommunikationsprotokoll wird im folgenden Verlauf auch Telegramm genannt.

Die Kommunikation auf dem LWL-Ring wird durch das Zentralmodul gesteuert. Es sendet Telegramme, die die einzelnen Module und Busklemmenstationen im LWL-Ring durchlaufen, und letztlich wieder empfangen und geprüft werden.

Ein Telegramm besteht aus Telegrammrahmen und Telegramminhalt.



Der Telegrammrahmen ist für eine serielle, asynchrone Datenkommunikation erforderlich und besteht aus 1 Startbit, 6 CRC-Prüfbits und 2 Stoppbits. Der Telegrammrahmen wird von der Hardware erzeugt und überprüft. Eine Softwareunterstützung ist nicht notwendig.

Der Telegramminhalt ist im wesentlichen byteweise organisiert.

AD0 - AD7 bilden das sogenannte Adreßfeld. Über dieses Adreßfeld können bis zu 254 Module und Busklemmenstationen angesprochen werden (die Adressen 0x00 und 0x0ff sind reserviert).

CR0 - CR3 legt den Telegrammtyp fest. Folgende Funktionen können im Telegramm festgelegt werden :

CR3	CR2	CR1	CR0	Funktion	Beschreibung
0	0	0	0	READ	Das adressierte Modul blendet die Eingangsinformation in die Datenfelder D0 - D3 ein.
0	0	0	1	READ/WRITE	Das adressierte Modul blendet die Eingangsinformation in die Datenfelder D0 - D3 ein und übernimmt die Ausgangsinformation.
0	0	1	0	ADREßINITIALI-SIERUNG	Das adressierte Modul übernimmt den Inhalt von D0 als Moduladresse und setzt D0 = 0.
0	0	1	1	RAM	Ein spezieller Telegrammtyp für Buskoppler BK2000
0	1	0	0	ADREßCHECK-UND COUNTBEFEHL	Jedes durchlaufene Modul erhöht den Inhalt von D0 um 1. Das adressierte Modul übernimmt den Inhalt von D0 nach D3
1	0	0	1	LOW INTENSITY BEFEHL	Das adressierte Modul reduziert die Sendeintensität um 20%.
1	0	1	1	BROADCAST	Ein spezieller Telegrammtyp für Buskoppler BK2000

Die Bytes D0 - D3 enthalten die eigentliche Nutzinformation. Die Verarbeitung dieser Nutzdaten ist durch das Controlfeld festgelegt.

Das letzte Byte im Telegramm enthält 2 Reservebits sowie 6 Bits zur Bildung einer CRC Prüfsumme. Bei einer Länge des Inhalts von 50 Bits wird eine Hamming Distanz von d=3 erreicht.

Der Beckhoff Lightbus besteht aus einem physikalischen Ring, der zur Verarbeitung des Prozessabbilds in bis zu 8 logische Ringe aufgeteilt werden kann. Ein logischer Ring arbeitet nur auf ausgewählte Module und Busklemmenstationen, die durch sogenannte Communication Description Lists (CDLs) festgelegt werden. Auf die Übergabe der CDLs vom Host-System an das Zentralmodul wird im späteren Verlauf noch eingegangen.

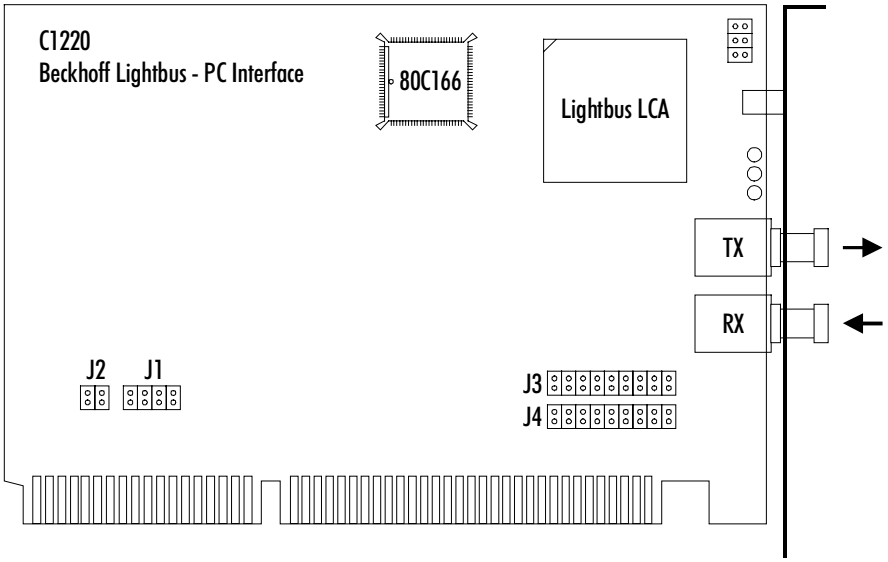
Über das DPRAM wird dem Host-System das Prozeßabbild zur Verfügung gestellt. Das DPRAM ist in drei Bereiche eingeteilt :

- Daten: Input, Output und Merker
- Kommunikation: Initialisierung, Test, Analyse und Konfiguration des Beckhoff Lightbus
- Prozeßkontrolle: Aktualisierung von Prozeßabbildern

Das Zentralmodul benötigt dafür einen Bereich von 4 kByte im Adreßraum des Host-Systems.

Funktionsbeschreibung Hardware

Lightbus - PC Interfacekarte C1220



Das Lightbus - PC Interface C1220 ist ein intelligentes Lightbus Zentralmodul.

PC-Control

Als ISA-Bus PC-Steckkarte realisiert die C1220 die Anbindung des Beckhoff Lightbus an den PC als Host-System und ist damit eine wichtige Komponente des PC-Control Konzepts.

Mit Hilfe der C1220 wird die schnelle Verarbeitung eines von den Sensoren/Aktoren des Beckhoff Lightbus bestimmten Prozeßabbildes ermöglicht.

Funktionsbeschreibung Software

Allgemein

Speicheraufteilung der Schnittstelle

Adreßbereich	Funktion
0x0000 - 0x0BFF	Datenbereich (Eingänge, Ausgänge, Merker) 3 kByte
0x0C00 - 0x0CFF	Handshake-Kanal 0: PC -> C1220 (Konfiguration, Test, Analyse)
0x0D00 - 0x0DFF	Handshake-Kanal 1: C1220 -> PC (Konfiguration, Test, Analyse)
0x0E00 - 0x0FEF	reserviert
0x0FF0 - 0x0FFF	GCB (General Control Block)

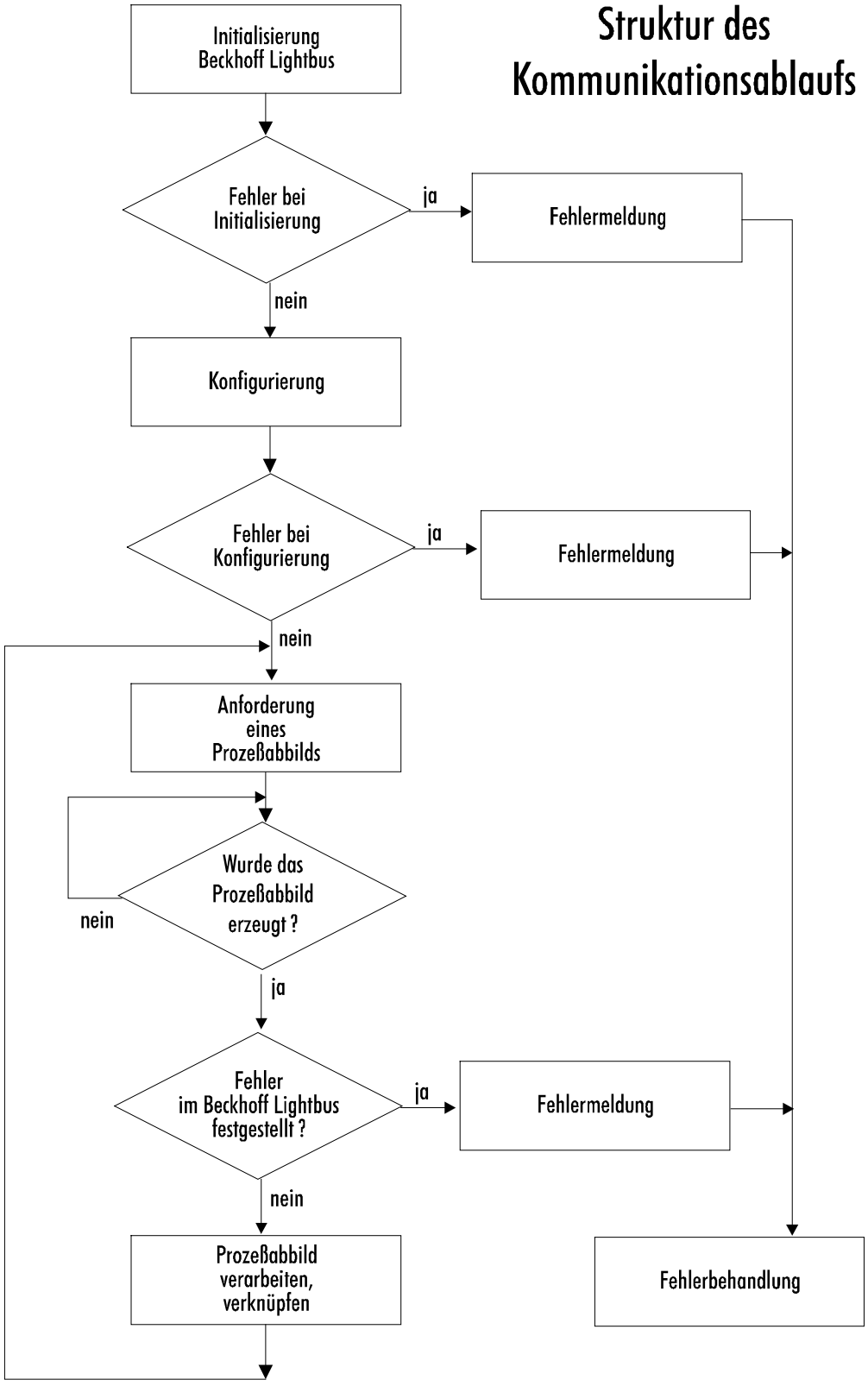
Die Schnittstelle zwischen PC-Bus und C1220 Modul ermöglicht folgende Funktionen :

- Datenaustausch des Prozeßabbildes
- Test- und Analyse-Funktionen für II/O-System
- Konfiguration
- Steuerung der Prozeßabbilder

Über die Kommunikationskanäle läßt sich der Beckhoff Lightbus durch vier Funktionen konfigurieren. Hierbei werden die Ein-/Ausgänge der dezentralen I/O-Module den Adressen im DPRAM zugeordnet. Ebenfalls über die Kommunikationskanäle können insgesamt neun weitere Funktionen zu Test und Analysefunktionen angefordert werden.

Im unteren 3 kByte Bereich, den das C1220 Modul im Adreßraum belegt, befinden sich die Datenbereiche für die CDLs. Die Aufforderung zur Aktualisierung des Prozeßabbildes geschieht durch Setzen eines Bits in der Anforderungsmaske des GCB (General Control Block). Die Fertigmeldung zu dieser Anforderung erhält man aus dem entsprechenden Bit in der Fertigmaske des GCB.

Struktur des Kommunikationsablaufs



Beschreibung der Kommunikationskanäle

Für die Kommunikation zwischen PC-Bus und C1220 sind zwei Kanäle eingerichtet. Jeder Kanal umfaßt 255 Byte. Der PC schreibt die Daten, die zur Anforderung der gewünschten Funktion erforderlich sind in den Kanal 0 und gibt anschließend ein DV (Data Valid) aus. Nach Übernahme der Daten gibt das Modul C1220 das Signal 'Quit' aus. Der PC nimmt das 'DV' zurück und sobald das Signal 'Quit' auf Null ist, kann eine neue Kommunikation begonnen werden.

Der Kanal 0 vom PC-Bus zur C1220 hat für die Daten den Adreßbereich von 0xC01 bis 0xCFF zur Verfügung. DV ist das MSB von Adresse 0xC00. 'Quit' ist das zweithöchste Bit von Adresse 0xD00.

Kommunikationskanal 0:

Byte 0 0xC00	Byte 1	.....	Byte 254	Byte 255 0xCFF

Der Kanal 1 vom Modul C1220 zum PC-Bus hat für die Daten den Adreßbereich von 0xD01 bis 0xDFF zur Verfügung. DV ist das MSB von Adresse 0xD00. 'Quit' ist das zweithöchste Bit von Adresse 0xC00.

Kommunikationskanal 1:

Byte 0 0xD00	Byte 1	.....	Byte 254	Byte 255 0xDFF

Adressen der Komunikationskanäle

Adresse	Adreßbits								Inhalt
	7	6	5	4	3	2	1	0	
0xC00	1	0	0	0	0	0	0	0	'Data Valid' für Kanal 0 (bei Datentransfer PC -> C1220)
0xC00	0	1	0	0	0	0	0	0	'Quit' für Kanal 1 (bei Datentransfer C1220 -> PC)
0xC01									Länge (von 2 bis 0xFE)
0xC02									Funktionsnummer (1 bis 0xFE)
0xC03									Argument 0
..									..
0xCnn									Argument n
..									..
0xCFF									..
0xD00	1	x	0	0	0	0	0	0	'Data Valid' für Kanal 1 (bei Datentransfer C1220 -> PC)
0xD00	x	1	0	0	0	0	0	0	'Quit' für Kanal 0 (bei Datentransfer PC -> C1220)
0xD01									Länge (von 2 bis 0xFE)

Adresse	Adreßbits	Inhalt
0xD02		Funktionsnummer (0x1 bis 0xFF)
0xD03		Argument 0
..		..
0xDnn		Argument n
..		..
0xDFF		..

Ablauf eines Handshakes

:0C00	0x80	Data Valid Host = 1
:0D00	0x40	DataQuit C1220 = 1
:0C00	0x00	Data Valid Host = 0
:0D00	0x00	Data Quit C1220 = 0
. . .		Funktionsausührung
:0D00	0x80	Data Valid C1220 = 1
:0C00	0x40	Data Quit Host = 1
:0D00	0x00	Data Valid C1220 = 0
:0C00	0x00	Data Quit Host = 0

vorhandene Funktionen

Nr.	Funktion
0x01	LWL-RESET
0x02	Codewort abfragen
0x03	Softwareversion abfragen
0x04	Parity Fehler abfragen
0x05	Dämpfungstest
0x06	Module zählen
0x07	Adreßtest
0x08	Dauersenden
0x09	Software-RESET
0x0a	Bruchstellentest
0x0b	freiprogrammierbare Kommunikation übertragen
0x0c	CDL-Verwaltung reinitialisieren
0x0d	reserviert
0x0e	reserviert
0x0f	Interruptmaske übergeben
0x10	CDL-Konfiguration übertragen
0x11	reserviert
0x12	zyklische Kommunikation
0x13	reserviert
0x14	Stringkommunikation initialisieren
0x15	Stringknoten anmelden
0xff	falsche Funktionsanforderung

Eine Funktionsanforderung setzt sich aus einer Längenangabe, einer Funktionsnummer und den Funktionsargumenten zusammen. Die Längenangabe bezieht sich auf die Anzahl folgender Bytes :

Byte 'Länge' + Byte 'Funktionsnummer' + Anzahl Bytes 'Argument 0' bis 'Argument n'

Test- und Analyse-Funktionen

LWL-Reset

Durch diese Funktion läßt sich der LWL-Ring neu initialisieren. Im Rahmen der Initialisierung wird die Anzahl der Module im Ring bestimmt, die Modul-adressen werden verteilt und getestet, und der Ring wird auf seine Dämpfungsreserve überprüft. Eine eventuell vorhandene Bruchstelle wird ebenfalls erkannt und lokalisiert.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x01				
Antwort	05	0x01	00	00	nn	Funktion korrekt ausgeführt (nn Module im LWL-Ring)
	05	0x01	01	01	00	Maximale Anzahl Sende-wiederholungen überschritten
	05	0x01	01	02	00	Kein Adreß-Setzen möglich
	05	0x01	0a	01	nn	Bruchstelle vor nn-ten Modul vor dem Empfängereingang der C1220
	05	0x01	0a	01	ff	Bruchstelle nicht lokalisierbar (Bruchstelle vor Empfängereingang)
	05	0x01	07	01	nn	Adressen testen :Adreßfehler (Modul nn)
	05	0x01	05	02	00	Dämpfungstest : Fehler bei High-Intensity
	05	0x01	05	03	nn	Dämpfungstest : Fehler bei Low-Intensity schalten (Modul nn)
	05	0x01	05	04	nn	Dämpfungstest : Fehler bei Datenmuster 1 (Muster 00) (Modul nn)
	05	0x01	05	05	nn	Dämpfungstest : Fehler bei Datenmuster 2 (Muster FF) (Modul nn)
	05	0x01	05	06	nn	Dämpfungstest : Fehler bei Datenmuster 3 (Muster AA) (Modul nn)
	05	0x01	05	07	nn	Dämpfungstest : Fehler bei High-Intensity schalten (Modul nn)

Ist der Ring fehlerfrei initialisiert, wird die Anzahl der im Ring vorhandenen Module übergeben. Sollte ein Fehler aufgetreten sein, wird die Fehlerart (siehe Tabelle) sowie die Moduladresse, bei welcher der Fehler aufgetreten ist, zurückgegeben.

Codewort

Das Codewort wird vom der C1220 jeweils nach erfolgtem Reset auf den Kommunikationskanal 1 ausgegeben. Dies erfolgt hier ohne das Setzen des Data Valid Bits. Durch das Codewort soll dem PC mitgeteilt werden, daß die C1220 Interfacekarte initialisiert und betriebsbereit ist. Das Codewort läßt sich auch jederzeit über die Funktion 0x02 abfragen.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x02				
Antwort	04	0x02	fe	af		richtiges Codewort

Softwareversion

Über Funktion 0x03 läßt sich die Version der EPROM Firmware abfragen.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x03				
Antwort	04	0x03	xx	xx		Version xxxx

Parityfehlerauswertung

Sind die Peripheriemodule mit SPROMs vom Typ 132 bzw. BX415 (BK2000) bestückt ist es möglich Parityfehlerquellen zu lokalisieren. Die Masterkarte erzeugt pro vorhandenem Modul einen (8 Bit breiten) „Parityfehler-Zähler“. Dieser Zähler arbeitet ohne Überlauf. Mittels der Funktion 04 können diese Zähler ausgewertet werden.

Kanal	Länge	Funktion	Argument			Kommentar
			0	.. .	128	
Anforderung	03	0x04	00			Zähler der Module 0 - 127 hochladen
Antwort	130	0x04	n	.. .	y	Zähler der Module 0 – 127 (0 = nicht lokalisierbarer Parityfehler)
Anforderung	03	0x04	01			Zähler der Module 128 - 255 hochladen
Antwort	130	0x04	n	.. .	y	Zähler der Module 128 – 255
Anforderung	03	0x04	02			Zähler zurücksetzen
Antwort	02	0x04				Zähler zurückgesetzt

LWL-Dämpfungstest

Mit dieser Funktion läßt sich die Dämpfungsreserve des LWL-Rings testen. Bei dem Test werden alle Verbindungsstrecken des LWL-Rings partiell mit etwa 80% der normalen Sendeintensität und extremen Testtelegrammen betrieben. Dieser Test läßt sich für alle Module oder nur für ein ausgewähltes Modul durchführen (siehe Tabelle). Die C1220 läßt sich separat über die Moduladresse 0 testen.

Die Tabelle zeigt die Funktionsanforderungen sowie die möglichen Rückmeldungen.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	04	0x05	00	00		alle Module testen
	04	0x05	01	nn		Modul nn testen
Antwort	04	0x05	00	00		Ring hat ausreichende Dämpfungsreserve
	04	0x05	02	00		Fehler bei High-Intensity
	04	0x05	03	nn		Fehler bei Low-Intensity schalten (Modul nn)
	04	0x05	04	nn		Fehler bei Datenmuster 1 (Muster 00)(Modul nn)
	04	0x05	05	nn		Fehler bei Datenmuster 2 (Muster FF)(Modul nn)
	04	0x05	06	nn		Fehler bei Datenmuster 3 (Muster AA)(Modul nn)
	04	0x05	07	nn		Fehler bei High-Intensity schalten (Modul nn)
	04	0x05	09	00		Funktion Dauersenden aktiv

"Fehler bei High_Intensity" bedeutet, daß der Ring bereits im Normalbetrieb eine zu hohe Dämpfung besitzt oder auch eine Bruchstelle vorhanden sein kann.

"Fehler bei Low-Intensity schalten" bedeutet, daß sich die Sendeintensität des betreffenden Moduls nicht reduzieren läßt.

"Fehler bei Datenmuster xx" zeigt an, daß der LWL-Ring hinter dem angegebenen Modul eine zu hohe Dämpfung aufweist. Das Betreiben des Systems ist jedoch noch möglich, so daß die Behebung dieser Störung zu einem geeigneten Zeitpunkt durchgeführt werden kann.

"Fehler bei High-Intensity schalten" bedeutet, daß sich das angegebene Modul nicht mehr auf volle Sendeleistung zurückschalten läßt.

Peripherie-Module zählen

Mit dieser Funktion läßt sich die Anzahl der Module im Ring bestimmen.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x06				
Antwort	04	0x06	00	nn		Module zählen : nn Module im Ring
	04	0x06	01	00		Module zählen : Ring unterbrochen

Peripherie-Modul Adressen testen

Über diese Funktion wird geprüft, ob die Module ihre bei der Initialisierung erhaltenen Adressen noch halten.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x07				
Antwort	04	0x07	00	00		Adressen korrekt
	04	0x07	01	nn		Fehler bei Adresse nn

Um eine maximale Betriebssicherheit zu gewährleisten, kann diese Funktion im Normalbetrieb auch zyklisch im Hintergrund durchgeführt werden. Die Funktion wird dabei durch Setzen eines Bits im GCB aktiviert. Im Fehlerfall wird eine Meldung über den GCB an den PC abgesetzt.

Dauersenden

Die Funktion Dauersenden steuert nur die 'Cycle'-LED auf den Modulen an. Hiermit läßt sich feststellen, wieviele Module noch mit dem Sendeausgang der C1220 verbunden sind. Diese Funktion sollte nur aktiviert werden, wenn die Funktion 0x0a (Bruchstellentest) kein zufriedenstellendes Ergebnis liefert. Das Dauersenden kann softwareseitig nur durch RESET gestoppt werden.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x08				
Antwort	03	0x08	01			Dauersenden durch RESET abschaltbar

Software-RESET

Durch diese Funktion läßt sich die C1220 zurücksetzen. Neben der Neuinitialisierung des LWL-Ringes wird auch der Controller und das Dual Ported RAM neu initialisiert. Der erfolgte RESET wird durch das Codewort (ohne Data Valid) quittiert.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x09				
Antwort	04	0x02	fe	af		

LWL-Bruchstellentest

Eine Bruchstelle im LWL-Ring kann durch diese Funktion lokalisiert werden. Der Test gibt abhängig vom Ergebnis die Anzahl der Boxen im Ring bzw. den Ort der Bruchstelle an.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x0a				
Antwort	04	0x0a	00	nn		keine Bruchstelle, nn Module im Ring
	04	0x0a	01	nn		Bruchstelle vor nn-ten Modul vor dem Empfängereingang der C1220
	04	0x0a	01	ff		Bruchstelle nicht lokalisierbar (Bruchstelle vor Empfängereingang)

Sollte die Bruchstelle als nicht lokalisierbar angegeben sein, so liegt sie vermutlich zwischen dem letzten Modul und dem Empfangseingang der C1220.

Fehlerhafte Funktionswahl

Wird über Handshake-Kanal 0 eine Funktion angefordert, die reserviert bzw. nicht vorhanden ist, wird sie mit der Funktion 0x0ff quittiert, die als Argument 0 die fehlerhafte Funktionsnummer enthält.

Beispiel:

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	03	0x04	01			Anforderung Funktion 4 (reserviert)
Antwort	03	0x0ff	04			

Konfiguration

Für die Beschreibung der Konfiguration, der Zuordnung der Ein- bzw. Ausgänge im Beckhoff Lightbus zu den Adressen im DPRAM, sowie die Zuordnung der Module zu den Prozeßgruppen stehen insgesamt vier Funktionen zur Verfügung. Die Übertragung der Konfiguration erfolgt ebenfalls über die Handshake-Kanäle.

Zu Beginn einer neuen Konfigurierung ist der Verwaltungsteil der Kommunikationen zu reinitialisieren.

Jede der maximal 8 Kommunikationen kann wahlweise als CDL-Kommunikation oder als freiprogrammierbare Kommunikation konfiguriert werden.
Eine weitere Funktion konfiguriert die Interruptkanäle für die adreßunabhängigen Interrupts.

Kommunikationsverwaltung reinitialisieren

Sowohl die CDLs als auch die freiprogrammierbaren Kommunikationen bestehen aus zwei Teilen, einem Daten- und einem Verwaltungsteil. Bevor neue Konfigurationen übergeben werden, müssen die Verwaltungsteile zurückgesetzt werden. Das Zurücksetzen der Verwaltungesteile aller 8 Kommunikationen erfolgt durch Aktivieren der Funktion 0x0c.

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Anforderung	02	0x0c				
Antwort	03	0x0c	00			

CDL-Kommunikation

Zu jeder Gruppe von Modulen, deren Prozeßabbild gemeinsam aktualisiert werden soll, wird eine CDL erzeugt. Diese CDL setzt sich aus sogenannten Descriptoren zusammen. Ein Descriptor beschreibt ein Telegramm zu einem Modul und ist wie folgt aufgebaut:

Bytes	Inhalt
0,1	Lightbus Moduladresse (1 - FE)
2,3	Control Word : 0x0000: READ 0x0010: READ/WRITE 0x0030: RAM 0x00B0: BROADCAST
4,5	Pointer auf Byte für Output in D0 einer Message
6,7	Pointer auf Byte für Output in D1 einer Message
8,9	Pointer auf Byte für Output in D2 einer Message
10,11	Pointer auf Byte für Output in D3 einer Message
12,13	Pointer auf Byte für Input in D0 einer Message
14,15	Pointer auf Byte für Input in D1 einer Message
16,17	Pointer auf Byte für Input in D2 einer Message
18,19	Pointer auf Byte für Input in D3 einer Message

Beispiel für einen Descriptor:

Telegramm an I/O-Modul 1 : D0 - D2 Ausgänge
D3 Eingang

Die Daten für den Output in D0 - D2 werden von den Adressen 0x400, 0x302 und 0x210 im DPRAM geholt.

Das Datum für den Input in D3 wird auf der Adresse 0x30 im DPRAM abgelegt.

Bytes	Inhalt
0,1	0x01, 0x00
2,3	0x10, 0x00
4,5	0x00, 0x04
6,7	0x02, 0x03
8,9	0x10, 0x02
10,11	0xff, 0xff
12,13	0xff, 0xff
14,15	0xff, 0xff
16,17	0xff, 0xff
18,19	0x30, 0x00

Konstanten

Am DPRAM-Adreßoffset 0xEF0 - 0xFEF befinden sich Konstanten 0x00 - 0xFF.
Um Konstanten in die Datenbytes der Lightbus-Telegramme einzusetzen muß im Descriptor lediglich der entsprechende Offset eingetragen werden.

Die oben genannten CDLs werden so in Teile zerlegt, daß sie über den Handshake-Kanal 0 übertragen werden können. Die Informationen für eine Message dürfen dabei nicht geteilt werden. Mit der Funktion 0x10 kann die Übertragung aktiviert werden.

Kanal	Länge	Funktion	leer	Argument				
				0	1	2	...	n
Anforderung	nn	0x10	00	aa	bb	db1,0		dbn,19

Kanal	Länge	Funktion	Argument			Kommentar
			0	1	2	
Antwort	04	0x10	aa	00		o.k.
	04	0x10	aa	01		Fehler in CDL-Daten (z.B.: Pointer nicht im Datenbereich des DPRAMs)
	04	0x10	aa	02		CDL-Überlauf
	04	0x10	aa	03		Falsche Descriptorlänge

mit:

aa	00 = Beginn eines CDL-Transfers 01 = weitere Descriptoren der gleichen CDL 02 = letzte Übertragung der gleichen CDL
bb	Prozeßabbild-Nr. bb (1 ... 8)
db1,0	Desciptor 1, Byte 0 einer CDL
...	...
dbn,19	Desciptor n, Byte 19 einer CDL (n = 2 ... 13)

Die Übergabe der Moduladresse, des Controlbytes und der Pointer auf die Datenbytes einer Message erfolgen in Intel Notation. (Niederwertiges Byte auf niederwertiger Adresse). Wird ein Pointer auf ein Datenbyte in einer Message nicht benötigt, so ist hier ein Dummypointer 0x0ffff einzutragen.

Wird die CDL-Übertragung abgeschlossen (Argument aa = 02) können die Argumente 2 - n entfallen.

Freiprogrammierbare Kommunikation

Bei dieser Art von Kommunikation werden ab einer vorher festgelegten Adresse im DPRAM Telegramme abgelegt und zu einem Prozeßabbild zusammengefaßt. Die Eingangsdaten werden dem PC System ab einer ebenfalls vorher festgelegten Adresse übergeben.

Mit dieser Funktion werden der C1220 die notwendigen Parameter zur Initialisierung übergeben.

Kanal	Länge	Funktion	leer	Argument			
				0	1	2	3
Anforderung	09	0x0b	00	pan	at	oa 0,1	ia 0,1

Kanal	Länge	Funktion	Argument 0	Kommentar
Antwort	03	0x0b	00	ok
	03	0x0b	01	Fehler

mit:

pan	Prozeßabbildnummer
at	Anzahl Telegramme
oa 0,1	Basisadresse Outputbereich
ia 0,1	Basisadresse Inputbereich

Die Basisadresse Outputbereich legt den Speicherbereich im DPRAM fest, ab der die selbstdefinierten Telegramme abgelegt werden. Es werden hierbei lediglich Adreßbyte, Contolbyte sowie vier Datenbytes eingetragen. Das Prüfbyte wird nicht eingetragen. Dieser Eintrag erfolgt intern durch den Controller.

Ab der Basisadresse Inputbereich werden von der C1220 für Adreß- und Controlbyte 0x00 eingetragen, und die Eingangsdaten abgelegt.

Beispiel:

Initialisierung der Kommunikation 3 als freie Kommunikation mit 2 Telegrammen. Basisadresse für den Outputbereich 0x400, Basisadresse für den Inputbereich 0x210.

Kanal	Länge	Funktion	leer	Argument			
				0	1	2	3
Anforderung	09	0x0b	00	03	02	00,04	10,02

Kanal	Länge	Funktion	Argument 0	Kommentar
Antwort	03	0x0b	00	ok

Durch diese Struktur besteht außerdem die Möglichkeit die Moduladresse und das Controlbyte während der Laufzeit zu verändern. Einschränkend gilt hier allerdings, daß dies nicht bei aktiver Kommunikation geschehen darf.

Zyklische Kommunikation

Mit der Funktion 0x12 besteht die Möglichkeit eine Kommunikation zyklisch vom Zentralmodul antriggern zu lassen. Der sonst notwendige Handshake über den GCB entfällt hierbei.

Kanal	Länge	Funktion	Argument 1	Argument2
Anforderung	04	0x12	k	pan

Kanal	Länge	Funktion	Argument 0	Kommentar
Antwort	03	0x12	00	ok
	03	0x12	01	Fehler

mit:

pan	Prozeßabbildnummer
k	Status 0 = Kommunikation passiv 1 = Kommunikation aktiv

Bei dieser Kommunikationsart sollten allerdings nur byteorientierte E/A-Funktionen ausgeführt werden, da kein deterministisches Zeitverhalten mehr vorliegt.

Interruptmaske übergeben

Das Modul C1220 verfügt über 4 Interruptkanäle über welche die adreßunabhängigen Interrupts dem PC übergeben werden. Die Übergabe an den PC erfolgt über den GCB.

Die adreßunabhängigen Interruptbits können von den Peripherimodulen erzeugt werden. Sie werden dabei in das Interruptfeld des Controlbytes eingeblendet.

Mit der Funktion 0x0f wird dem Modul C1220 mitgeteilt, welche Interruptkanäle aktiviert werden sollen und welche Interruptkriterien zur Interruptübergabe an den PC führen sollen.

Kanal	Länge	Funktion	Argument				
			0	1	2	3	4
Anforderung	07	0x0f	0m	Kriterium Interrupt-kanal 0	Kriterium Interrupt-kanal 1	Kriterium Interrupt-kanal 2	Kriterium Interrupt-kanal 3
Antwort	03	0x0f	0m				

Das LOW-Nibble in Argument 0 gibt an, welche der 4 möglichen Interruptkanäle freigegeben werden sollen.

Beispiel:
- m = 0x00 alle Interruptkanäle gesperrt (Defaultwert)
- m = 0x01 Interruptkanal 0 freigegeben
- m = 0x06 Interruptkanäle 1 und 2 freigegeben
- m = 0x0f Interruptkanäle 0, 1, 2 und 3 freigegeben

Jeder Interruptkanal läßt sich über ein Kriterium charakterisieren.

Folgende Kriterien können ausgewählt werden:

Kriterium	Interrupt-Kanal$_{(0,1,2,3)}$
kein Interrupt	0
Interrupt auf positive Flanke	1
Interrupt auf negative Flanke	2
Interrupt auf Flankenwechsel	3

Über die Argumente 1 bis 4 werden die jeweiligen Kriterien den Interrupt-Kanälen zugeordnet.

Beispiel:

Kanal	Länge	Funktion	Argument				
			0	1	2	3	4
Anforderung	07	0x0f	0x0c	00	00	01	03
Interruptfreigabe für die Kanäle 2 und 3							
Interruptkriterium für Interruptkanal 0 keine Interruptübergabe							
Interruptkriterium für Interruptkanal 1 keine Interruptübergabe							
Interruptkriterium für Interruptkanal 2 Interruptübergabe bei positiver Flanke							
Interruptkriterium für Interruptkanal 3 Interruptübergabe bei Flankenwechsel							

Bevor die Interrupts aktiviert werden, muss ein Lesezurgriff auf die Zelle IRQ-Eingänge im General Control Block (siehe Kapitel 3.5) durchgeführt werden.

Stringkommunikation

Allgemein

Die Stringkommunikation dient zum paketorientierten Datenaustausch mit Peripheriemodulen. Im allgemeinen werden Parameterdaten mit den Modulen ausgetauscht (z.B. Parametrierung eines BK2000 per Registerinterface).
Neben der Kommunikation zwischen Master und Slave ist bei dieser Kommunikationsart auch eine Slave zu Slave Kommunikation möglich. Die Masterkarte dient hierbei lediglich als Relaisstation.

Zur Durchführung der Stringkommunikation sind die folgenden Ressourcen notwendig.
– 2 CDLs zum Senden bzw. Empfangen der Strings.
– 2 Puffer im DPRAM zur Ablage der Strings, wobei die Puffergröße parametrierbar ist.

Struktur des Strings

Ein Datenstring besteht aus einem vier Byte großen String-Header und einem String-Datenbereich. Der Header enthält die notwendigen Routing Informationen, der Datenbereich die eigentlichen Nutzdaten. Der gesamte String kann eine maximale Länge von 255 Byte haben.

Ein String ist wie folgt aufgebaut :

Offset	Beschreibung
0x00	Adresse des Absenders (TX)
0x01	Adresse dem Empfängers (RX)
0x02	Kanal / Priorität (nur relevant für BK2000)
0x03	Stringlänge
0x04	
...	Stringdaten
0xFF	

Initialisierung der Stringkommunikation

Die Stringkommunikation wird über den Handshakekanal mit der Funktion 0x14 initialisiert

Kanal	Länge	Funktion	Argument			
			0	1	2	3
Anforderung	0x0A	0x14	Init StringComm. (0x01)	CDL-Nr String-Trns	CDL-Nr String-Receive	Max. Stringlänge
Anforderung	0x03	0x14	Deinit StringComm. (0x00)	Das Deaktivieren der Stringkommunikation deaktiviert auch alle String-Slaves		
Antwort	0x03	0x14	0x00	Kein Fehler		
			0x01	Falsche CDL Nummer für String-Transmit CDL		
			0x02	String- Transmit CDL bereits belegt		
			0x03	Falsche CDL Nummer für String-Receive CDL		
			0x04	String-Receive CDL bereits belegt		
			0x05	Falsche Basisadresse Transmit-String		
			0x06	Falsche Basisadresse Receive-String		

Fortsetzung der Tabelle

Argument			
4	5	6	7
Offset String-Transmitpuffer		Offset String-Receivepuffer	
Das Deaktivieren der Stringkommunikation deaktiviert auch alle String-Slaves			
Kein Fehler			
Falsche CDL Nummer für String-Transmit CDL			
String- Transmit CDL bereits belegt			
Falsche CDL Nummer für String-Receive CDL			
String-Receive CDL bereits belegt			
Falsche Basisadresse Transmit-String			
Falsche Basisadresse Receive-String			

Bekanntgabe eines String-Slaves

Bevor eine Stringkommunikation zu einem String-Slave möglich ist muss dieser der Masterkarte bekannt gegeben werden. Dies geschieht mittels der Funktion 0x15.

Kanal	Länge	Funktion	Argument			
			0	**1**	**2**	
Anforderung	0x0A	0x15	SubFnc	Physikalische Slave-Adresse	Logische Slave-Adresse	
Anforderung	0x03	0x15	01	Mn	xy	Eintrag String-Slave ohne String Reset
			02	Mn	Xy	Eintrag String-Slave mit String Reset
			03	Mn	Xy	Eintrag String-Slave ohne String Reset Übertragung der Strings ohne Auslösen eines Interrupts auf dem Slave
			04	Mn	xy	Eintrag String-Slave mit String Reset Übertragung der Strings ohne Auslösen eines Interrupts auf dem Slave
			00	Mn	xy	String-Slave Deaktivieren
Antwort	0x03	0x15	0x00	Kein Fehler		
			0x01	Falsche Slaveadresse		
			0x02	Fehler bei String Reset auf Slave		
			0x03	LWL – Fehler		

Bevor eine Kommunikation mit einem Slave möglich ist muß ein String Reset erfolgreich durchgeführt worden sein. Der String Reset auf einem Slave dient zur Synchronisation der Handshakebits zwischen Master und Slave. Es gibt zwei Möglichkeiten einen String Reset auszulösen:

- Der Reset wird bei Bekanntgabe des Slaves vom Master initiiert.
- Der Reset wird zu einem späteren Zeitpunkt vom Slave initiiert (siehe auch „Auslösen eines String-Resets durch den Slave“).

Die Adressierung eines String-Slaves bei der Stringübertragung geschieht nur über seine logische Slaveadresse (wobei die logische Adresse gleich der physikalischen sein kann).

Struktur der Puffer für die Stringkommunikation

Transmit-/ Receivepuffer	Beschreibung	
0x00.0	Aktivflag	
0x00.1 - 0x00.7	Fehlerfeld	0x00: String fehlerfrei übertragen 0x04: LWL Fehler 0x08: String Slave nicht initialisiert 0x10: String Slave noch nicht kommunikationsbereit 0x20: Timeout bei Stringübertragung 0x40: String Längenfehler
0x01	Leer	
0x02	Adresse des Absenders (TX)	
0x03	Adresse dem Empfängers (RX)	
0x04	Kanal / Priorität	
0x05	Stringlänge	
0x06 - 0xFF	Stringdaten	

Senden eines Strings

Um einen String an einen Stringslave zu Senden werden zuerst die Stringdaten (Header und Daten) in den Transmitpuffer der C1220 eingetragen. Wird nun das Aktivflag gesetzt wird die Masterkarte veranlasst den String abzusenden. Ist dies geschehen setzt die Masterkarte ihrerseits das Aktivflag zurück. Ein eventuell aufgetretender Fehler bei der Stringübertragung wird im Fehlerfeld gemeldet

Empfangen eines Strings

Ist ein String von einem String-Slave empfangen worden wird dieser in den Receivepuffer der C1220 abgelegt und das Aktivflag gesetzt. Solange ein empfangener String nicht durch Rücksetzen des Aktivflags quittiert ist wird kein weiterer String von einem Stringslave abgeholt.

Slave zu Slave Stringkommunikation

Die Slave zu Slave Kommunikation (empfangener String mit RX ungleich „0“) wird komplett von der Masterkarte bearbeitet.

Registerkommunikation

Per Stringkommunikation kann auf einfache Weise das Registerinterface eines Buskopplers bzw. einer Klemme angesprochen werden. Um eine Registerkommunikation anzustoßen muß im Stringheader lediglich der Kanal 8 eingetragen werden. Im Stringdatenbereich ist ein weiterer, 6 Byte großer, Header notwendig.

High Byte	Low Byte	I/O Adresse	
Registerdaten		127	Nutzdaten
		--	
		5	
Anzahl Worte	Register(-basis)	4	Header zur Registerkommunikation
R/W Tabelle	Klemmennummer	3	
Message Ident		2	
Size	Priorität 8	1	Header zur Stringkommunikation
RX_Adresse	TX_Adresse	0	

Prozeßabbild-Kontrollfunktionen

General Control Block

Der General Control Block dient zur Steuerung und zur Kontrolle der Aktualisierung der einzelnen Prozeßabbilder. Mit dem Setzen eines Bits in der Anforderungsmaske wird das entsprechende Prozeßabbild aktualisiert und über die Fertigmaske als fertig gemeldet. Nach der Fertigmeldung muß erst das Bit aus der Anforderungsmaske gelöscht werden, bevor die Kommunikation erneut gestartet werden kann. Das Aktualisieren eines Prozeßabbilds ist unterbrechbar. Wird bei laufender Prozeßaktualisierung in der Anforderungsmaske die Anforderung einer höher priorisierten Aktualisierung ausgelöst, so wird die laufende unterbrochen.

Sollten während des Normalbetriebs Fehler auf dem LWL-Ring erkannt werden, so werden die entsprechenden Bits in der Error-Maske gesetzt.

General Control Block

Adresse	Inhalt	Kommentar
0x0FFF	Anforderungsmaske	
0x0FFE	IRQ-Ausgänge	
0x0FFD	Fertigmaske	
0x0FFC	IRQ-Eingänge	
0x0FFB		reserviert
0x0FFA	Error-Maske	
0x0FF9	Control-Maske	
0x0FF8	-	reserviert
0x0FF7	-	reserviert
0x0FF6	-	reserviert
0x0FF5	Firmware Revision	
0x0FF4	Fimrware Release	
0x0FF3	-	reserviert
0x0FF2	-	reserviert
0x0FF1	-	reserviert
0x0FF0	-	reserviert

Anforderungsmaske:

P7							P0

hoch <= Priorität <= niedrig

Fertigmaske:

P7							P0

wobei: P7 => Prozess 08
...
P0 => Prozess 01

Errormaske:

E7						E1	E0

E0 gesetzt = allgemeiner LWL Fehler
E1 gesetzt = Adreßfehler bei residentem Adreßcheck
E7 gesetzt = CPU-Fehler auf der C1220

IRQ-Ausgänge:

-	-	-	-	IO3	IO2	IO1	IO0

Wird vom PC diese Maske modifiziert, so wird sie in die Interruptfelder der nächsten Telegramme eingeblendet. Das Nibble wird solange in das Interruptfeld eingeblendet, bis es vom PC wieder zurückgenommen wird.

IRQ-Eingänge:

-	-	-	-	II3	II2	II1	II0

Wird von einem I/O-Modul ein adreßunabhängiger Interrupt generiert, wird er über diese Maske dem PC übergeben, sofern er durch die Interruptmaske freigegeben ist.

Anstehende Interrupts werden von der C1220 gepuffert, d.h. es wird dem PC jeweils nur ein Interrupt über den GCB übergeben. Erst wenn dieser vom PC erkannt worden ist, wird ein eventuell noch anstehender Interrupt übergeben.

Control-Maske:

-	-	-	-	-	-	C1	C0

Über Bit C0 läßt sich der residente Adreßtest vom PC abschalten bzw. wieder aktivieren. Um Parityfehler lokalisieren zu Können muss der Adresscheck aktiviert sein.

C0 gesetzt : Adreßtest aktiv
C1 gesetzt : Adreßtest auch bei LWL Fehler aktiv

C0 ist als Default gesetzt.

C1220 II/O Fehlerzähler

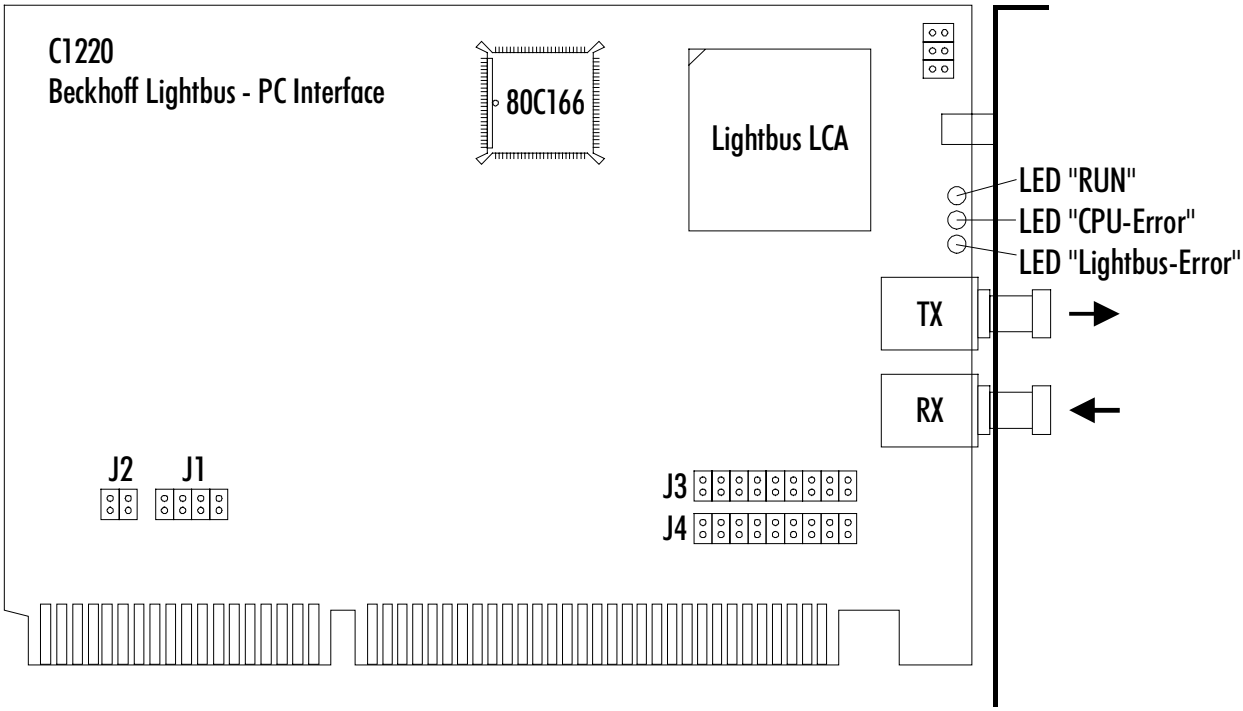
Die C1220 verfügt über mehrere Zähler zur Erfassung von II/O Problemen. Die Zähler sind im DPRAM ab dem Offset 0xEE0 als 16 Bit Werte abgelegt. Es erfolgt keine Überlaufverarbeitung bzw. kein Löschen der Zähler durch die C1220.

Der Zähler, der die Fehler des internen Adreßchecks erfaßt (0xEEA) inkrementiert nicht den Summenfehler.

DPRAM-Offset	Bedeutung	Funktion
0xEE0	Summenfehler	Summe der einzelnen Fehlertrigger (nachfolgend)
0xEE2	Fehler im Receiver 1	Adreß- und / oder Control ungleich der gesendeten Bytes
0xEE4	Fehler im Receiver 2	Adreß- und / oder Control ungleich der gesendeten Bytes
0xEE6	Timeoutfehler	Zeitüberlauf bei Telegrammempfang
0xEE8	Parityfehler	Telegramm mit CRC Fehler empfangen
----------	**----------**	**----------**
0xEEA	Fehler bei internem Adresscheck	Bei Inkrementierung dieses Zählers wird ein Adresscheck und Count Telegramm mit logisch falschem Inhalt (AD <> D3) empfangen.
0xEEC	Moduladresse bei internem Adresscheckfehler	Wird in der C1220 Fehlermaske das Bit 1 gesetzt, enthält diese Zelle die Moduladresse der Box, die den Fehler verursacht hat.

Technische Daten

Schnittstellenprozessor	Siemens SAB 80C166-S
Datenanschluß	Beckhoff Lightbus
Übertragungsrate	2,5 MBaud, 32 Bit Nutzinformation in 25 µsec
Versorgungsspannung	5 V
Stromaufnahme	800 mA
Abmessungen	161mm x 107mm



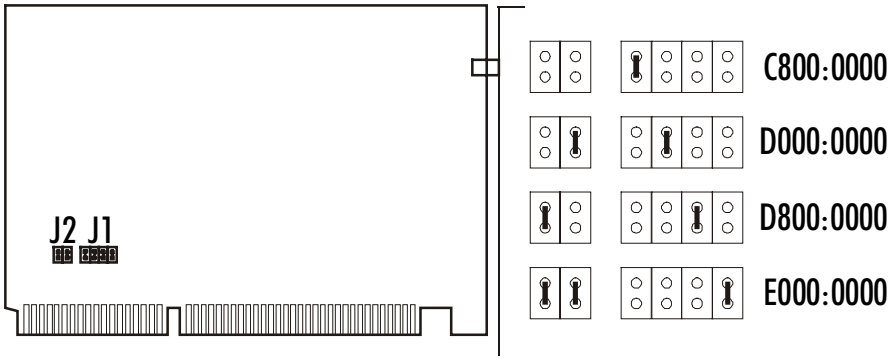
Installationshinweise

Jumperkonfiguration

Die Interfacekarte C1220 belegt einen ISA-Bus Steckplatz auf der PC Busplatine. Der Anschluß des LWL-Rings erfolgt mit zwei LWL-Steckern über die Blende.

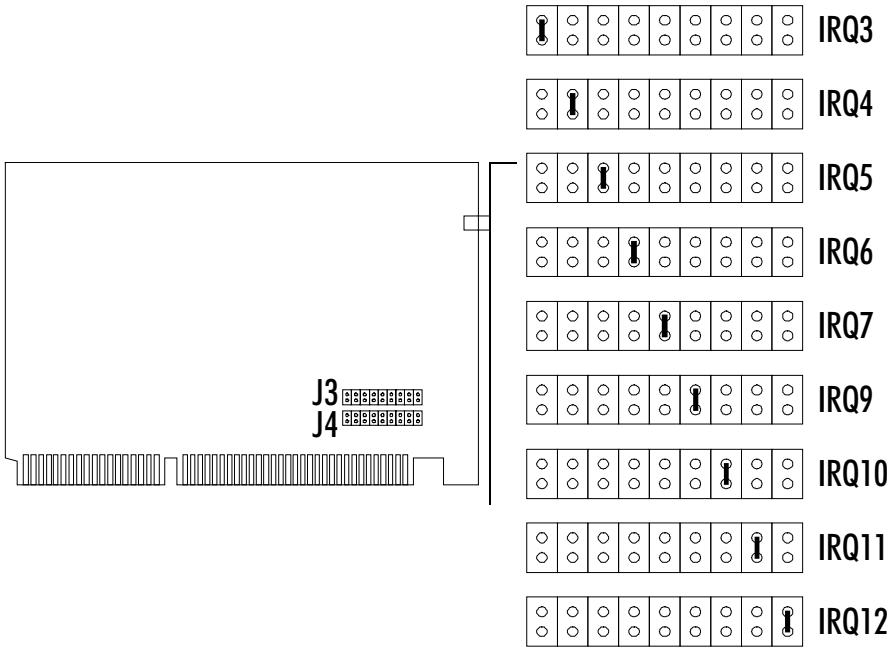
Jumperfeld J1 und J2

Die Einstellung der Basisadresse für den benötigten 4-kByte-Bereich des PC-Adreßraums erfolgt über die Jumperfelder J2 und J1:



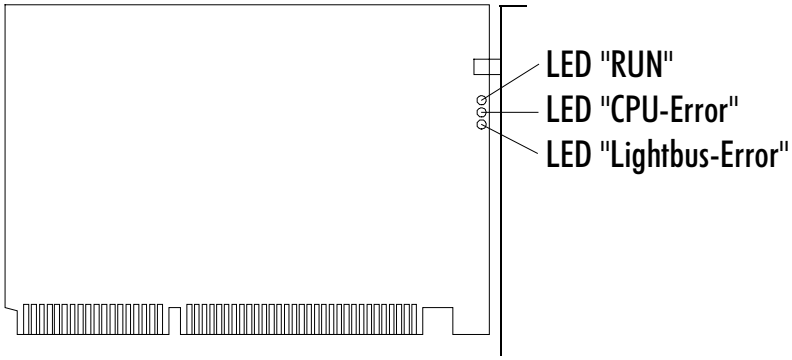
Jumperfeld J3 und J4

Über Jumperfeld J3 wird die IRQ-Nr des Ready-Interrupts festgelegt. Jumperfeld J4 legt die IRQ-Nr der schnellen Interrupteingänge fest.



Statusanzeige

Auf der C1220 befinden sich 3 LEDs zur Statusanzeige.



LED “RUN“

Die LED 'RUN' zeigt an, daß die C1220 fehlerfrei initialisiert und betriebsbereit ist.

LED “CPU-Error“

Leuchtet nur diese LED auf, so liegt ein nicht behebbarer Hardwarefehler vor. Leuchtet ebenfalls die LED ’RUN’ auf, so liegt ein Programmfehler vor, der eventuell durch einen Hardwarereset behoben werden kann.

LED “Lightbus-Error“

Tritt während des Betriebs ein Defekt im LWL Ring auf, so wird die LED 'LWL-FAIL' aktiviert. Liegt ein allgemeiner LWL-Fehler vor, blinkt die LED. Ist der Fehler beim residenten Adreßtest aufgetreten ist die LED statisch eingeschaltet. Die Aktualisierung des Prozessabbilds wird unterbrochen. Durch die vorhandenen Diagnosefunktionen kann die Fehlerursache ermittelt werden.

Montage im PC

1. Schalten Sie den PC und eventuelle externe Spannungsversorgungen ab.

2. Die Interfacekarte C1220 wird in einen 16 Bit ISA-Bus Steckplatz auf der PC Busplatine gesteckt.

Die C1220 benötigt keine externe Spannungsversorgung. Die Karte wird direkt vom PC gespeist. Beim Einschalten des PC geht damit auch die C1220 in Betrieb. Bevor die C1220 jedoch den Betrieb aufnehmen kann, müssen die Lichtleiterverbindungen hergestellt, und die Jumper der C1220 korrekt konfiguriert werden.