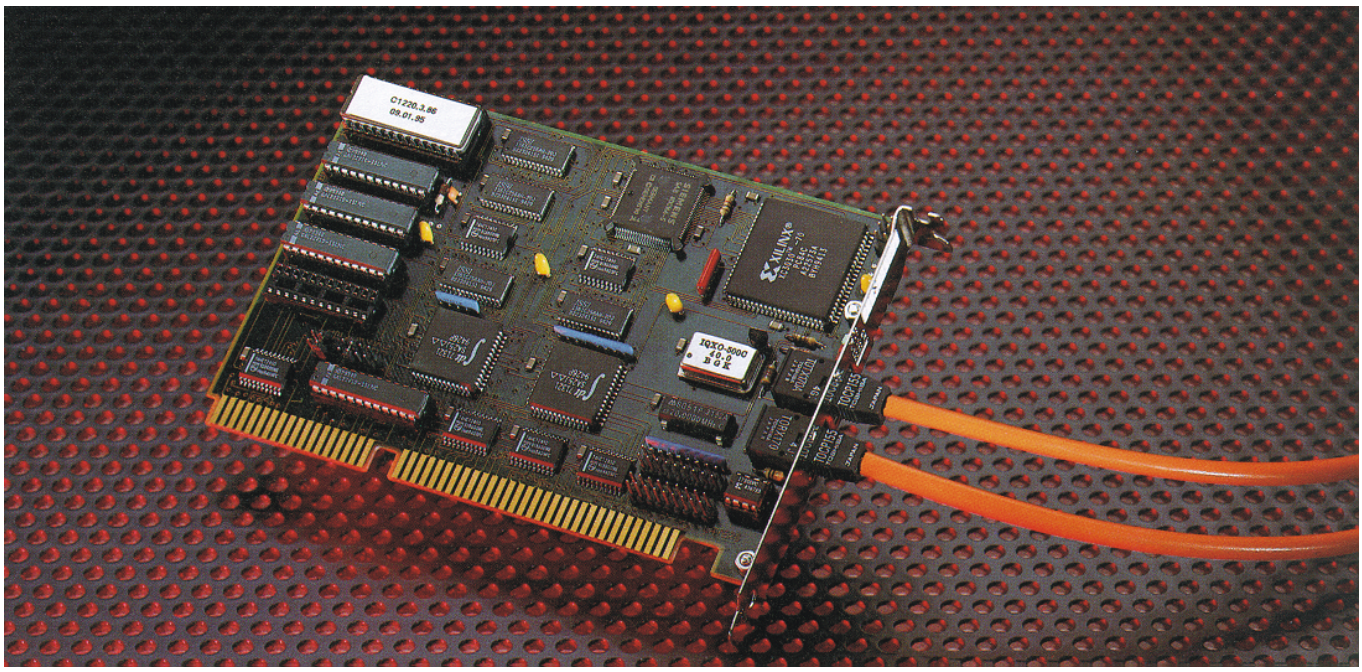


Beckhoff Lightbus - PC interface card C1220

Technical hardware description

Version 4.0

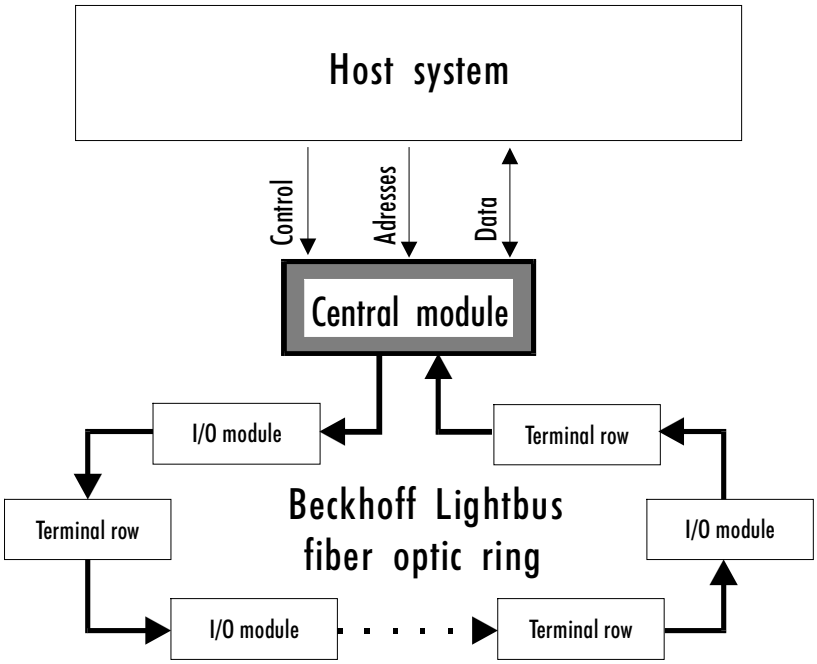
BECKHOFF
INDUSTRIE ELEKTRONIK

Contents

Beckhoff Lightbus System Description

The Beckhoff Lightbus consists of an intelligent central module and a field bus based on fibre-optic conductor.

Beckhoff Lightbus



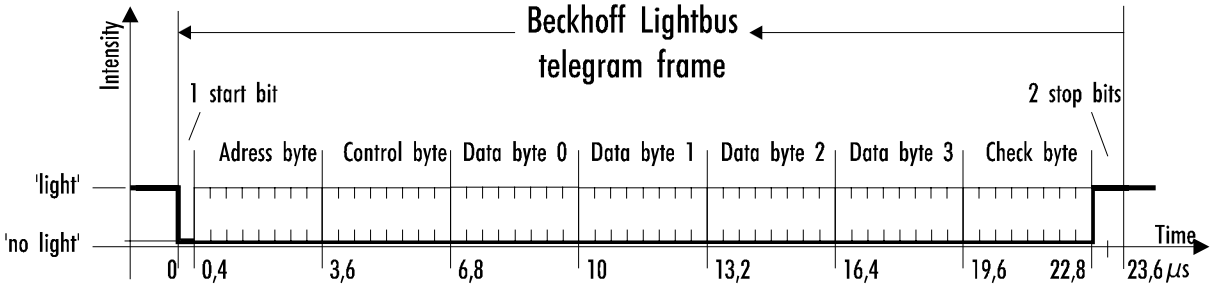
The Beckhoff Lightbus is coupled to the host System via a DPRAM, thus guaranteeing fast and convenient communication.

Bus couplers for Beckhoff Bus terminals and diverse I/O modules are available for processing the process image. Modules and bus couplers are linked to one another in a ring structure. Thanks to the use of fibre-optic conductor, interference sensitivity is low and the data transfer rate of 2.5 Mbaud is high. Errors occurring in the fibre-optic ring are detected by the central module and are reported to the host system. Implemented ring diagnostics functions enable swift error detection and remedying.

A communication protocol optimized for speed and simplicity has been defined for data transfer between the central module and I/O modules. Below, this communication protocol is also referred to as a telegram.

Communication on the fibre-optic ring is controlled by the central module. It sends telegrams which pass through the individual modules and terminal rows in the fibre-optic ring, and which are ultimately received again and checked.

A telegram consists of the telegram frame and contents.



The telegram frame is required for serial, asynchronous data communication and consists of 1 start bit, 6 CRC check bits and 2 stop bits. The telegram frame is generated and checked by the hardware. Software support is not necessary.

The telegram contents are essentially based on a byte organization.

AD0 - AD7 constitute the so called address field. Up to 254 modules and terminal rows can be addressed via this address field (the addresses 0x00 and 0x0ff are reserved).

CR0 - CR3 defines the telegram type. The following functions can be defined in the telegram:

CR3	CR2	CR1	CR0	Function	Description
0	0	0	0	READ	The addressed module inserts the input information in the data fields D0 - D3.
0	0	0	1	READ/WRITE	The addressed module inserts the input information in the data fields D0 - D3 and accepts the output information.
0	0	1	0	ADDRESS INITIALIZATION	The addressed module accepts the contents of D0 as the module address and sets D0 = 0.
0	0	1	1	RAM	A special type of telegram for bus coupler BK2000
0	1	0	0	ADDRESS CHECK AND COUNT COMMAND	Every module that is passed through increments the contents of D0 by 1. The addressed module transfers the contents from D0 to D3.
1	0	0	1	LOW INTENSITY COMMAND	The addressed module reduces the send intensity by 20%.
1	0	1	1	BROADCAST	A special type of telegram for bus coupler BK2000

The bytes D0 - D3 contain the actual user information. Processing of this user information data is defined by the control field.

The last byte in the telegram contains 2 reserve bits and 6 bits for generation of a CRC checksum. A Hamming distance of d=3 is achieved with a length of the contents amounting to 50 bits.

The Beckhoff Lightbus consists of a physical ring which can be split into 8 logical rings for processing the process image. A logical ring only operates on selected modules and terminal rows that are defined by means of so called Communication Description Lists (CDLs). Transfer of the CDLs from the host system to the central module will be discussed in further detail later.

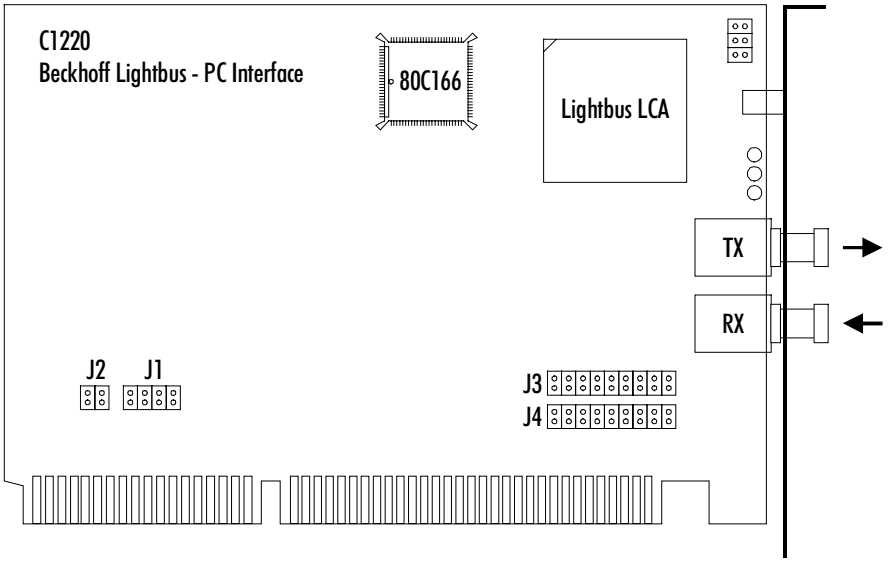
The process image is made available to the host system via the DPRAM. The DPRAM is split into three areas:-

- Data: Input, output and flags
- Communication: Initialization, test, analysis and configuration of the Beckhoff Lightbus
- Process control: Updating of process images

To this end, the central module requires a 4 k byte area in the address space of the host system.

Hardware description of functions

Lightbus PC interface card C1220



The Lightbus - PC interface C1220 is an intelligent Lightbus central module.

PC control

As a plug-in ISA bus PC board, the C1220 links the Beckhoff Lightbus to the PC as the host system and is therefore an important component of the PC control concept.

With the aid of the C1220, fast processing of a process image defined by the sensors / actuators of the Beckhoff Lightbus is enabled.

Software description of functions

General

Memory breakdown of the interface

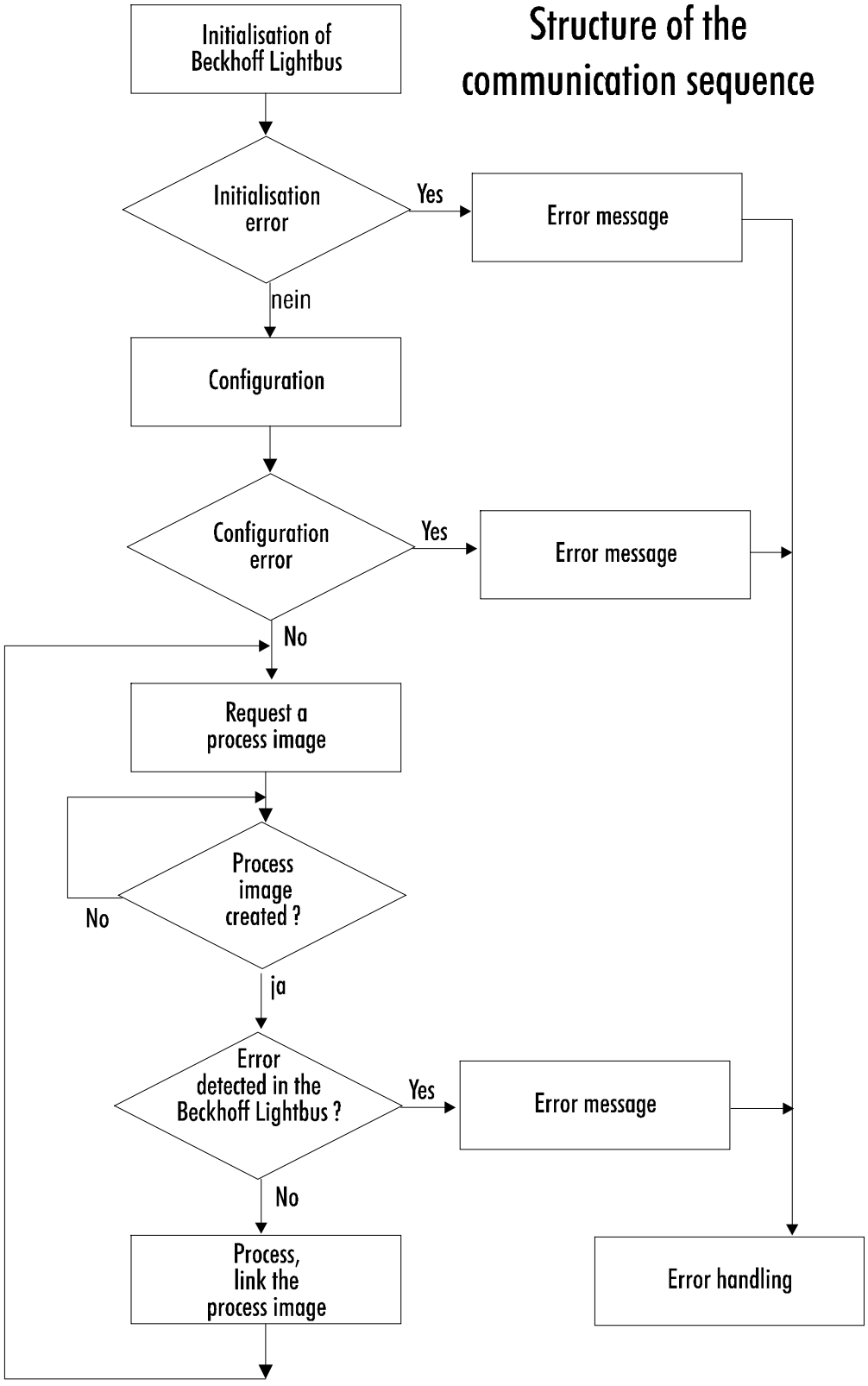
Address area	Function
0x0000 - 0x0BFF	Data area (Inputs, outputs, flags) 3 kbytes
0x0C00 - 0x0CFF	Handshake channel 0: PC -> C1220 (configuration, test, analyse)
0x0D00 - 0x0DFF	Handshake channel 1: C1220 -> PC (configuration, test, analyse)
0x0E00 - 0x0FEF	reserved
0x0FF0 - 0x0FFF	GCB (General control block)

The interface between the PC bus and the C1220 module enables the following functions:

- Data transfer of the process image
- Test and analysis functions for the II/O system
- Configuration
- Control of process images

By way of the communication channels, the Beckhoff Lightbus can be configured by means of four functions. In doing so, the inputs / outputs of the decentralized I/O modules are assigned to the addresses in the DPRAM. A total of nine further functions can also be requested via the communication channels for test and analysis functions.

The data areas for the CDLs are located in the bottom 3 kbyte area that the C1220 module occupies in the address space. The request to update the process image is realized by setting a bit in the request mask of the GCB (General Control Block). The ready message for this request is obtained from the corresponding bit in the ready mask of the GCB.


Structure of the communication sequence
Initialisation of Beckhoff Lightbus
Initialisation error
Yes
Error message
nein
Configuration
Configuration error
Yes
Error message
No
Request a process image
Process image created ?
No
ja
Error detected in the Beckhoff Lightbus ?
Yes
Error message
No
Process, link the process image
Error handling

Description of the communication channels

Two channels are configured for communication between the PC bus and C1220. Each channel embraces 255 bytes. The PC writes the data required for requesting the required function into the channel 0 and then outputs a DV (Data Valid). After acceptance of the data, the C1220 module outputs the ‘Quit’ signal. The PC withdraws the ‘DV’ and a new communication can be commenced as soon as the ‘Quit' signal is 0.

Channel 0 from the PC bus to the C1220 provides the address area from 0xC01 to 0xCFF for the data. DV is the MSB of the address 0xC00. 'Quit' is the second highest bit of the address 0xD00.

Communication channel 0:

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

Channel 1 from the C1220 module to the PC bus provides the address area from 0xD01 to 0xDFF for the data. DV is the MSB of the address 0xD00. 'Quit' is the second highest bit of the address 0xC00.

Communication channel 1:

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

Addresses of the Komunikationskanäle

Address	Address bits								Contents
	7	6	5	4	3	2	1	0	
0xC00	1	0	0	0	0	0	0	0	’Data Valid’ for Channel 0 (in the case of PC -> C1220 data transfer)
0xC00	0	1	0	0	0	0	0	0	’Quit’ for Channel 1 (in the case of C1220 -> PC data transfer)
0xC01									Length (von 2 bis 0xFE)
0xC02									Functionsnummer (1 bis 0xFE)
0xC03									Argument 0
..									..
0xCnn									Argument n
..									..
0xCFF									..
0xD00	1	x	0	0	0	0	0	0	’Data Valid’ for Channel 1 (in the case of C1220 -> PC data transfer)
0xD00	x	1	0	0	0	0	0	0	’Quit’ for Channel 0 (in the case of PC -> C1220 data transfer)
0xD01									Length (from 2 to 0xFE)

Address	Address bits	Contents
0xD02		Function number (1 to 0xFF)
0xD03		Argument 0
..		..
0xDnn		Argument n
..		..
0xDFF		..

Sequence of a handshake

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

Existing functions

No.	Function
0x01	FIBRE-OPTIC RESET
0x02	Query code word
0x03	Query software version
0x04	Query parity error
0x05	Attenuation test
0x06	Count modules
0x07	Address test
0x08	Continuous sending
0x09	Software RESET
0x0a	Fracture point test
0x0b	Transfer freely programmable communication
0x0c	Reinitialize CDL management
0x0d	reserved
0x0e	reserved
0x0f	Interrupt mask
0x10	Transfer CDL configuration
0x11	reserved
0x12	Cyclic communication
0x13	reserved
0x14	Initialize string communication
0x15	Log in string node **???**
0xff	Invalid function request

A function request is composed of a length entry, a function number and the function arguments. The length entry refers to the number of following bytes:

Byte 'Length' + Byte 'Function number' + Number of Bytes 'Argument 0' to 'Argument n'

Test and analysis functions

Fibre-optic reset

The fibre-optic ring can be reinitialized by means of this function. Within the scope of initialization, the number of modules in the ring is defined, the module addresses are distributed and tested and the ring is checked with regard to its attenuation reserve. Any existing fracture point is also detected and located.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x01				
Reply	05	0x01	00	00	nn	Function correctly executed (nn modules in the fibre-optic ring)
	05	0x01	01	01	00	Maximum number of send repetitions exceeded
	05	0x01	01	02	00	No address setting possible
	05	0x01	0a	01	nn	Fracture point before nn-th module before the receiver input of the C1220
	05	0x01	0a	01	ff	Fracture point cannot be located (Fracture point before receiver input)
	05	0x01	07	01	nn	Test addresses: Address error (module nn)
	05	0x01	05	02	00	Attenuation test: Error with high intensity
	05	0x01	05	03	nn	Attenuation test: Switch error with low intensity (module nn)
	05	0x01	05	04	nn	Attenuation test: Error with data pattern 1 (pattern 00)(module nn)
	05	0x01	05	05	nn	Attenuation test Error with data pattern 2 (pattern FF) (module nn)
	05	0x01	05	06	nn	Attenuation test : Error with data pattern 3 (pattern AA) (module nn)
	05	0x01	05	07	nn	Attenuation test : Switch error with high intensity (module nn)

The number of modules in the ring is communicated if the ring is initialized without errors. If an error should have occurred, the error type (see table) and the module address where the error occurred are returned.

Code word

The C1220 outputs the code word after every reset of communication channel 1. Here, this takes place without setting the Data Valid bit. The purpose of the code word is to inform the PC that the C1220 interface card is initialized and ready. The code word can also be queried at any time by way of the 0x02 function.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x02				
Reply	04	0x02	fe	af		Correct code word

Software version

The version of the EPROM firmware can be queried by way of the 0x03 function.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x03				
Reply	04	0x03	xx	xx		Version xxxx

Evaluation of Parity Errors

If the peripheral modules are fitted with type 132 or BX415 (BK2000) SPROMs, it is possible to localise the sources of parity errors. The master card produces a “parity error counter” (8 bits wide) for every module present. This counter works without overflow. The counter can be read by means of function 04.

	Length	Function	Argument			Comment
			0	.. .	128	
Request	03	0x04	00			Transmit counter for modules 0 - 127
Reply	130	0x04	n	.. .	y	Counter for modules 0 – 127 (0 = non-localisable parity error)
Request	03	0x04	01			Transmit counter for modules 128 -255
Reply	130	0x04	n	.. .	y	Counter for modules 128 -255
Request	03	0x04	02			Reset counter
Reply	02	0x04				Counter reset

Fibre-optic attenuation test

The attenuation reserve of the fibre-optic ring can be tested with this function. In this test, all leaks of the fibre-optic ring are partially operated with approximately 80% of the normal transmission intensity and extreme test telegrams. This test can be run for all modules or for only one selected module (see table). The C1220 can be tested separately via the module address 0.

The table shows the function requests and the possible acknowledgements.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	04	0x05	00	00		Test all modules
	04	0x05	01	nn		Test module nn
Reply	04	0x05	00	00		Ring has adequate attenuation reserve
	04	0x05	02	00		Error with high intensity
	04	0x05	03	nn		Switch error with low intensity (module nn)
	04	0x05	04	nn		Error with data pattern 1 (pattern 00)(module nn)
	04	0x05	05	nn		Error with data pattern 2 (pattern FF)(module nn)
	04	0x05	06	nn		Error with data pattern 3 (Pattern AA)(module nn)
	04	0x05	07	nn		Switch error with high intensity (module nn)
	04	0x05	09	00		Continuous sending function active

"Error with high intensity" means that the ring already has an excessive attenuation during normal operation or that there may be a fracture point.

"Switch error with low intensity" means that the transwhere intensity of the module concerned cannot be reduced.

"Error with data pattern xx" indicates that the fibre-optic ring after the specified module has an excessive attenuation. It is nevertheless possible to operate the system, with the result that this malfunction can be remedied at a suitable point in time.

"Switch error with high intensity" means that the specified module can no longer be switched back to the full transwhere power.

Count peripheral modules

The number of modules in the ring can be defined with this function.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x06				
Reply	04	0x06	00	nn		Count modules: nn modules in the ring
	04	0x06	01	00		Count modules: Ring interrupted

Test peripheral module addresses

By means of this function, a check is made as to whether the modules are still keeping to the addresses they received on initialization.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x07				
Reply	04	0x07	00	00		Addresses correct
	04	0x07	01	nn		Error at address nn

To guarantee maximum operating reliability, during normal operation this function can also be run cyclically in the background. In doing so, the function is activated by setting a bit in the GCB. In the event of an error, a message is sent to the PC via the GCB.

Continuous sending

The continuous sending function only controls the 'Cycle' LEDs on the modules to determine how many modules are still connected to the transwhere output of the C1220. This function should only be activated if the 0x0a (Fracture point test) does not return a satisfactory result. On the software end, continuous sending can only be stopped by a RESET.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x08				
Reply	03	0x08	01			Continuous sending can be stopped by RESET

Software-RESET

The C1220 can be reset by means of this function. Besides reinitialization of the fibre-optic ring, the controller and the dual ported RAM are also reinitialized. Completion of RESET is acknowledged by the code word (without Data Valid).

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x09				
Reply	04	0x02	fe	af		

Fibre-optic fracture point test

A fracture point in the fibre-optic ring can be localized by this function. Depending on the result, the test specifies the number of boxes in the ring or the location of the fracture point.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x0a				
Reply	04	0x0a	00	nn		No fracture point, nn modules in the ring
	04	0x0a	01	nn		Fracture point before nn th-module before the receiver input of the C1220
	04	0x0a	01	ff		Fracture point cannot be located (fracture point before receiver input)

If the fracture point should be specified as not being capable of location, it is probably located between the last module and the receive input of the C1220.

Invalid function selection

If a function is requested via handshake channel 0 that is reserved or is not available, it is acknowledged with the function 0x0ff, which contains the invalid function number as Argument 0.

Example:

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	03	0x04	01			Request function 4 (reserved)
Reply	03	0x0ff	04			

Configuration

A total of four functions is available for description of the configuration, of affiliation of the inputs or outputs in the Beckhoff Lightbus to the addresses in the DPRAM, and the affiliations of the modules to the processor groups. The configuration is also transwhereted via the handshake channels.

The management part of the communication functions must be reinitialized at the start of a new configuration.

Each of the maximum number of 8 communication functions can be optionally configured as CDL communication or as freely programmable communication.

A further function configures the interrupt channels for the address-independent interrupts.

Reinitializing communication management

Both the CDLs and also the freely programmable communication functions consist of two parts, a data part and a management part. The management parts must be reset before new configurations are communicated. The management parts of all 8 communication functions are reset by activating the function 0ch.

Channel	Length	Function	Argument			Comment
			0	1	2	
Request	02	0x0c				
Reply	03	0x0c	00			

CDL communication

A CDL is generated for each group of modules whose process image is to updated jointly. This CDL is composed of so-called descriptors. A descriptor describes a telegram for a module and is structured as follows:

Bytes	Contents
0,1	II/O module address (1 - FE)
2,3	Control Word : 0x0000: READ 0x0010: READ/WRITE 0x0030: RAM 0x00B0: BROADCAST
4,5	Pointer to byte for output in D0 of a message
6,7	Pointer to byte for output in D1 of a message
8,9	Pointer to byte for output in D2 of a message
10,11	Pointer to byte for output in D3 of a message
12,13	Pointer to byte for input in D0 of a message
14,15	Pointer to byte for input in D1 of a message
16,17	Pointer to byte for input in D2 of a message
18,19	Pointer to byte for input in D3 of a message

Example of a descriptor:

Telegram to I/O module 1 : D0 - D2 Outputs
D3 Input

The data for the output in D0 - D2 is fetched from the addresses 0x400, 0x302 and 0x210 in the DPRAM.

The data item for the input in D3 is stored at the address 0x30 in the DPRAM.

Bytes	Contents
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

Constants

At the DPRAM adress offset 0xEF0 - 0xFEF the constants 0x00 - 0xFF can be found. To insert constants into the data bytes of the Lightbus telegrams, you just have to set the descriptor to the corresponding offset.

The above-mentioned CDLs are split into parts so they can be transferred via the handshake channel 0. In doing so, the information for a message must not be split. Transfer can be activated with the function 0x10.

Channel	Length	Function	blank	Argument				
				0	1	2	...	n
Request	nn	0x10	00	aa	bb	db1,0		dbn,19

Channel	Length	Function	Argument			Comment
			0	1	2	
Reply	04	0x10	aa	00		o.k.
	04	0x10	aa	01		Error in CDL data (e.g.: Pointer not in the DPRAM's data area)
	04	0x10	aa	02		CDL overflow
	04	0x10	aa	03		Invalid descriptor length

where:

aa	00 = start of a CDL transfer 01 = further descriptors of the same CDL 02 = last transfer of the same CDL
bb	Process image No bb (1 ... 8)
db1,0	Descriptor 1, byte 0 of a CDL
...	...
dbn,19	Descriptor n, byte 19 of a CDL (n = 2 ... 13)

The module address, the control byte and the pointers to the data bytes of a message are transferred in Intel notation (least significant byte at least significant address).If a pointer to a data byte in a message is not needed, a dummy pointer 0x0ffff must be entered here.

The arguments 2 - n can be dropped when CDL transfer is concluded (Argument aa = 02).

Freely programmable communication

With this mode of communication, telegrams are stored as from a previously defined address in the DPRAM and are combined in a process image. The input data is transferred to the PC system as from an address that is also defined beforehand.

With this function, the parameters necessary for initialization are transferred to the C1220.

Channel	Length	Function	blank	Argument			
				0	1	2	3
Request	09	0x0b	00	pan	at	oa 0,1	ia 0,1

Channel	Length	Function	Argument 0	Comment
Reply	03	0x0b	00	ok
	03	0x0b	01	Error

where:

pan	Process image number
at	Number of telegrams
oa 0,1	Base address of output area
ia 0,1	Base address of input area

The base address of the output area defines the memory area in the DPRAM as from which the user-defined telegrams are stored. In doing so, only the address byte, the control byte and four data bytes are entered. The check byte is not entered. This entry is made internally by the controller.

As from the base address for the input area, the C1220 enters 0x00 for the address and control byte and the input data is stored.

Example:

Initialization of communication 3 as free communication with 2 telegrams. The base address for the output area is 0x400, and the base address for the input area is 0x210.

Channel	Length	Function	blank	Argument			
				0	1	2	3
Request	09	0x0b	00	03	02	00,04	10,02

Channel	Length	Function	Argument 0	Comment
Reply	03	0x0b	00	ok

By means of this structure it is also possible to modify the module address and the control byte during the run time. One restriction here, however, is that this must not take place while communication is active.

Cyclic communication

With the function 0x12, it is possible to trigger communication cyclically by the central module. In doing so, the otherwise necessary Handshake by way of the GCB is dropped.

Channel	Length	Function	Argument 1	Argument2
Request	04	0x12	k	pan

Channel	Length	Function	Argument 0	Comment
Reply	03	0x12	00	ok
	03	0x12	01	Error

where:

pan	Process image number
k	Status 0 = Communication passive 1 = Communication active

In this mode of communication, however, only byte-oriented I/O functions should be executed because the timing behavior is no longer deterministic.

Transmitting the interrupt mask

The module C1220 features 4 interrupt channels through which the address-independent interrupts are transferred to the PC. Transfer to the PC is realized via the GCB.

The address-independent interrupts can be generated by the peripheral modules. In doing so, they are inserted in the interrupt field of the control byte.

The function 0x0f is used to communicate to the module C1220 which interrupt channels are to be activated and which interrupt criteria are to lead to interrupt transmission to the PC.

Channel	Length	Function	Argument				
			0	1	2	3	4
Request	07	0x0f	0m	Criterion for interrupt channel 0	Criterion for interrupt channel 1	Criterion for interrupt channel 2	Criterion for interrupt channel 3
Reply	03	0x0f	0m				

The LOW nibble in the argument 0 specifies which of the 4 possible interrupt channels is to be enabled.

Example:
m = 0x00 all interrupt channels disabled (Default)
m = 0x01 Interrupt channel 0 enabled
m = 0x06 Interrupt channels 1 and 2 enabled
m = 0x0f Interrupt channels 0, 1, 2 and 3 enabled

Each interrupt channel can be characterized by way of a criterion.

The following criteria can be selected:

Criterion	Interrupt-Channel$_{(0,1,2,3)}$
no Interrupt	0
Interrupt on positive edge	1
Interrupt on negative edge	2
Interrupt on edge change	3

The respective criteria are assigned to the interrupt channels by way of the arguments 1 to 4.

Example:

Channel	Length	Function	Argument				
			0	1	2	3	4
Request	07	0x0f	0x0c	00	00	01	03
Interrupt enabling for channels 2 and 3							
Interrupt criterion for interrupt channel 0							
no interrupt transfer							
Interrupt criterion for interrupt channel 1							
no interrupt transfer							
Interrupt criterion for interrupt channel 2							
Interrupt transfer with a positive edge							
Interrupt criterion for interrupt channel 3							
Interrupt transfer with an edge change							

Before the interrupts are activated, a read access must take place to the cell IRQ inputs in the General Control Block (see Chapter 3.5).

String Communication

General

String communication is used for packet oriented data exchange with peripheral modules. Usually, parameter data is exchanged with the modules (e.g. parameterisation of a BK2000 by register interface).
This type of communication also permits slave to slave communication, as well as communication between master and slave. The master card then functions simply as a relay station.

The following resources are required to carry out string communication:
– 2 CDLs for sending or receiving the strings.
– 2 buffers in DPRAM for string storage, the buffer size being parameterisable.

String structure

A data string consists of a four-byte string header and a string data area. The header contains the necessary routing information, and the data area contains the user data itself. The entire string can have a maximum length of 255 bytes.

A string has the following structure:

Offset	Description
0x00	Address of the sender (TX)
0x01	Address of the receiver (RX)
0x02	Channel / priority (only relevant to the BK2000)
0x03	String length
0x04	
...	String data
0xFF	

Initialisation of string communication

String communication is initialised via the handshake channel with function 0x14.

	Length	Function	Argument			
			0	1	2	3
Request	0x0A	0x14	Init StringComm. (0x01)	CDL no. string Trns	CDL no. string receive	Max. string length
Request	0x03	0x14	Deinit StringComm. (0x00)	Deactivation of string communication also deactivates all the string slaves.		
Reply	0x03	0x14	0x00	No error.		
			0x01	Wrong CDL number for string transmit CDL.		
			0x02	String transmit CDL already occupied.		
			0x03	Wrong CDL number for string receive CDL.		
			0x04	String receive CDL already occupied.		
			0x05	Wrong transmit string base address.		
			0x06	Wrong receive string base address.		

Continuation of the table

Argument			
4	5	6	7
Offset string transmit buffer		Offset string receive buffer	
Deactivation of string communication also deactivates all the string slaves.			
No error.			
Wrong CDL number for string transmit CDL.			
String transmit CDL already occupied.			
Wrong CDL number for string receive CDL.			
String receive CDL already occupied.			
Wrong transmit string base address.			
Wrong receive string base address.			

Registration of a string slave

Before string communication with a string slave is possible, it must be registered with the master card.
This is done by means of function 0x15.

	Length	Function	Argument			
			0	1	2	
Request	0x0A	0x15	SubFnc	Physical slave address	Logical slave address	
Request	0x03	0x15	01	Mn	xy	Enter string slave without string reset.
			02	Mn	Xy	Enter string slave with string reset.
			03	Mn	Xy	Enter string slave without string reset. Transmission of the string without triggering an interrupt at the slave.
			04	Mn	xy	Enter string slave with string reset. Transmission of the string without triggering an interrupt at the slave.
			00	Mn	xy	Deactivate string slave.
Reply	0x03	0x15	0x00	No error.		
			0x01	Wrong slave address.		
			0x02	Error during string reset at the slave.		
			0x03	Optical fibre error.		

Before communication with a slave is possible, a string reset must be successfully carried out. The string reset at a slave synchronises the handshake bits between master and slave. There are two ways to trigger a string reset:

- The reset is initiated by the master when the slave is registered.
- The reset is initiated at a later time by the slave (see also “Triggering a String Reset by the Slave”).

A string slave is only addressed for string transmission by means of its logical slave address (although the logical address can be the same as the physical address).

Structure of the buffer for string communication

Transmit / receive buffer	Description	
0x00.0	Active flag	
0x00.1 - 0x00.7	Error field	0x00: String transmitted without error. 0x04: Optical fibre error. 0x08: String slave not initialised. 0x10: String slave not yet ready for communication. 0x20: Timeout during string transmission. 0x40: String length error.
0x01	Empty	
0x02	Address of the sender (TX)	
0x03	Address of the receiver (RX)	
0x04	Channel / priority	
0x05	String length	
0x06 - 0xFF	String data	

Sending a string

To send a string to a string slave, the string data (header and user data) are placed in the C1220's transmit buffer. If the active flag is now set, the master card is induced to send the string. Once this has occurred, the master card now resets the active flag. If any error has occurred during the string transmission, this is indicated in the error field.

Receiving a string

If a string is received from a string slave, it is placed in the C1220's receive buffer, and the active flag is set. As long as a string that has been received has not been acknowledged by resetting the active flag, no further string will be fetched from a string slave.

Slave to slave string communication

Slave to slave communication (received string has RX not equal to "0") is processed entirely by the master card.

Register communication

String communication can be used to access the register interface of a bus coupler or of a terminal in a simple manner.
To trigger register communication, channel 8 must simply be entered into the string header. In the string data area an additional header, 6 bytes large, is necessary.

High Byte	Low Byte	I/O Address	
Register data		127	
		--	
		5	User data
Number of words	Register (base)	4	
R/W | Table	Terminal number	3	Header for register communication
Message Ident		2	
Size	Priority | 8	1	
RX_address	TX_address	0	Header for string communication

Process image control functions

The General Control Block serves to control and check updating of the individual process images. When a bit is set in the request mask the corresponding process image is updated and is reported as complete via the ready mask. After the complete message, the bit first has to be deleted from the request mask before communication can be restarted. It is possible to interrupt updating of a process image. If the request by a higher-priority update is triggered during an ongoing process updating in the request mask, the current operation is interrupted.

The corresponding bits are set in the error mask if errors in the fibre-optic ring are to be detected during normal operation.

General Control Block

Adress	Contents	Comment
0x0FFF	Request mask	
0x0FFE	IRQ outputs	
0x0FFD	Ready mask	
0x0FFC	IRQ inputs	
0x0FFB		reserved
0x0FFA	Error-Mask	
0x0FF9	Control-Mask	
0x0FF8	-	reserved
0x0FF7	-	reserved
0x0FF6	-	reserved
0x0FF5	Firmware Revision	
0x0FF4	Fimrware Release	
0x0FF3	-	reserved
0x0FF2	-	reserved
0x0FF1	-	reserved
0x0FF0	-	reserved

Request mask:

P7							P0

high <= Priority <= low

Ready mask:

P7							P0

where: P7 => Process 08
...
P0 => Process 01

Error mask:

E7						E1	E0

E0 set = general fibre-optic error
E1 set = Address error in resident address check
E7 set = CPU error in the C1220

IRQ output:

-	-	-	-	IO3	IO2	IO1	IO0

If the PC modifies this mask, it is inserted into the interrupt fields of the next telegrams. The nibble is inserted into the interrupt field until it is withdrawn again by the PC.

IRQ inputs:

-	-	-	-	II3	II2	II1	II0

If an address-independent interrupt is generated by an I/O module, it is transferred to the PC via this mask provided it is enabled by the interrupt mask.

Pending interrupts are buffered by the C1220, i.e. only ever one interrupt is transferred the PC via the GCB. Any other pending interrupt is not transferred until this one has been recognized by the PC.

Control mask:

-	-	-	-	-	-	C1	C0

Bit C0 can be used to allow the PC to switch off the resident address test, or to reactivate it. In order to be able to localise parity errors, the address check must be active.

C0 set : Address test active.
C1 reset : Address test active even with optical fibre errors.

C0 is set by default.

C1220 II/O Error Counter

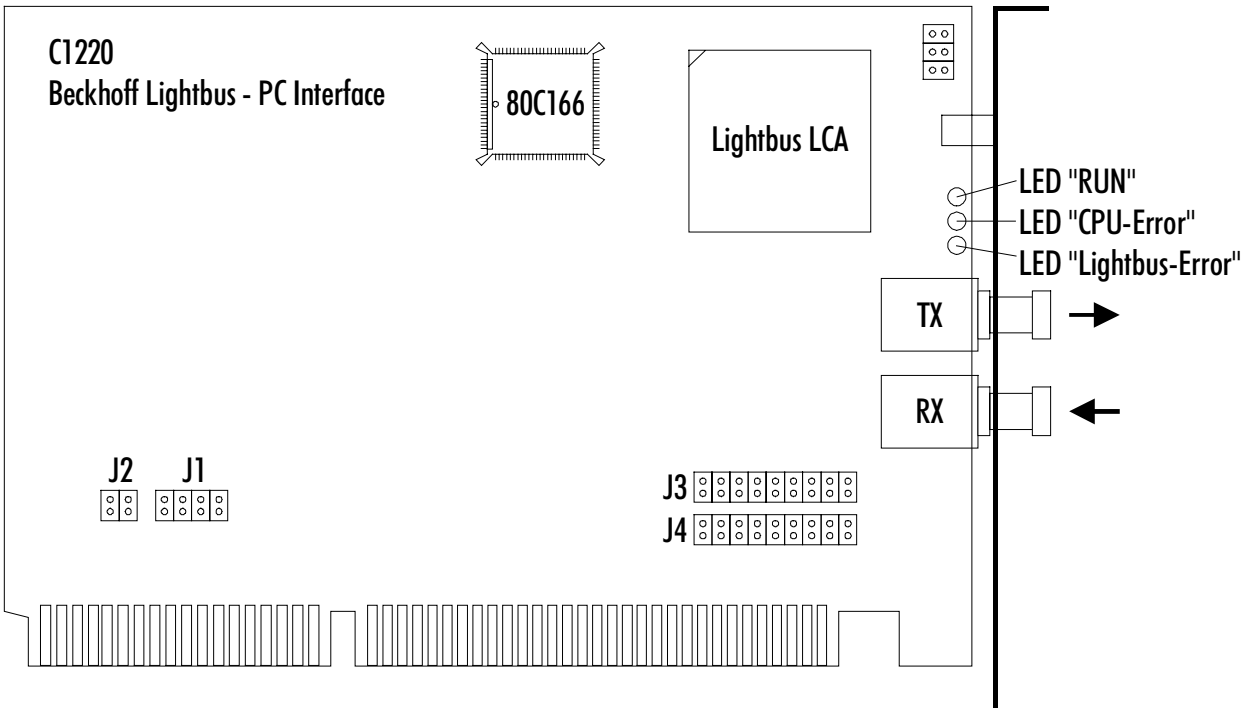
The C1220 has several counters to register II/O problems. The counters are stored in the DPRAM from offset 0xEE0, as 16-bit values. There is no overflow processing, and no erasure of the counters by the C1220.

The counter which registers the errors of the internal address check (0xEEA) is designed as an 8-bit counter, and does not increment the total error.

DPRAM-Offset	Meaning	Function
0xEE0	Total error	Delivers the number of error handing operations as the sum total of the individual error triggers (following)
0xEE2	Error in receiver 1	Address and/or control unequal to transmitted bytes
0xEE4	Error in receiver 2	Address and/or control unequal to transmitted bytes
0xEE6	Timeout error	Timeout in telegram reception
0xEE8	Parity error	Telegram received with CRC error
----------	**----------**	**----------**
0xEEA	Error in internal address check	When this counter is incremented an address check and count telegram with logically false content (AD <> D3) is received.
0xEEC	Module address in case of internal address check error	If bit 1 is set in the C1220 error mask, this cell contains the module address of the box which caused the error.

Technical Data

Interface processor	Siemens SAB 80C166-S
Data connection	Beckhoff Lightbus
Data transfer rate	2,5 MBaud, 32 Bits of user information in 25µsec
Supply voltage	5 V
Current consumption	800 mA
Dimensions	161mm x 107mm



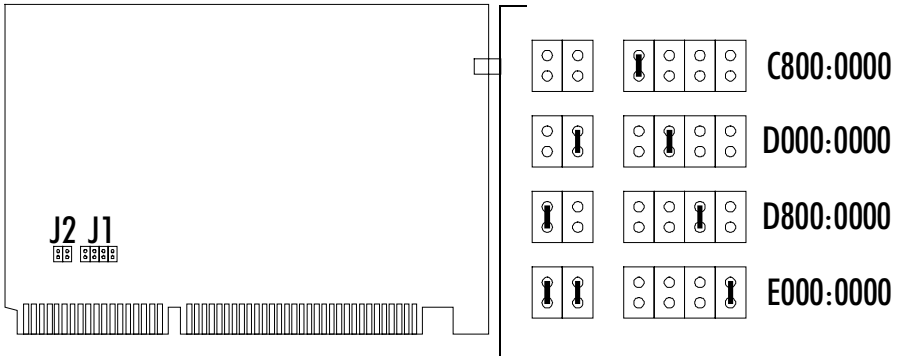
Installation notes

Jumper configuration

The C1220 interface card occupies one ISA bus slot on the PC's bus board. The fibre-optic ring is connected with two fibre optic connecters on the panel.

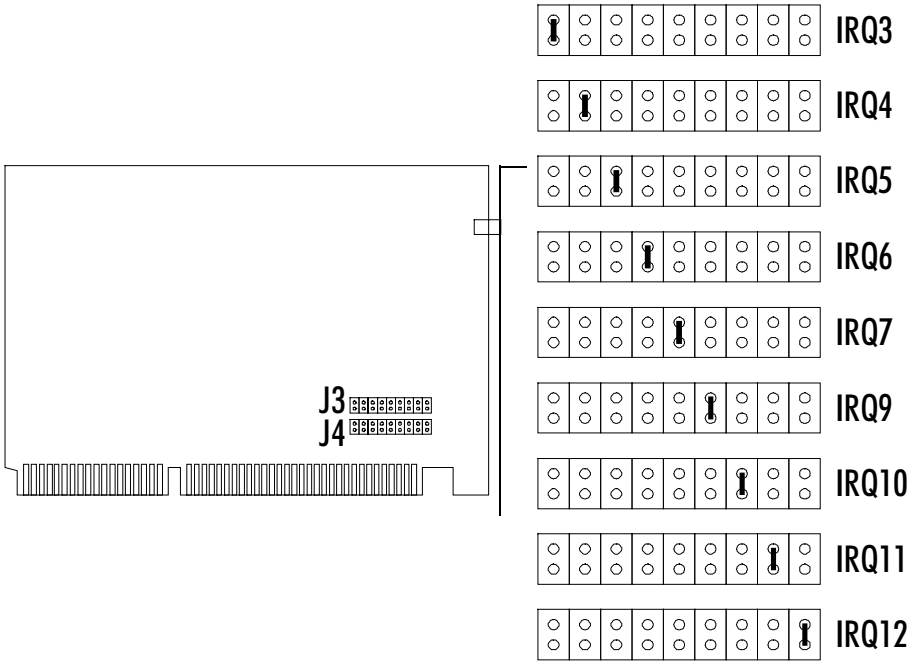
Jumpers J1 and J2

The base address for the required 4 kbyte area of the PC address space is set by means of jumpers J2 and J1:



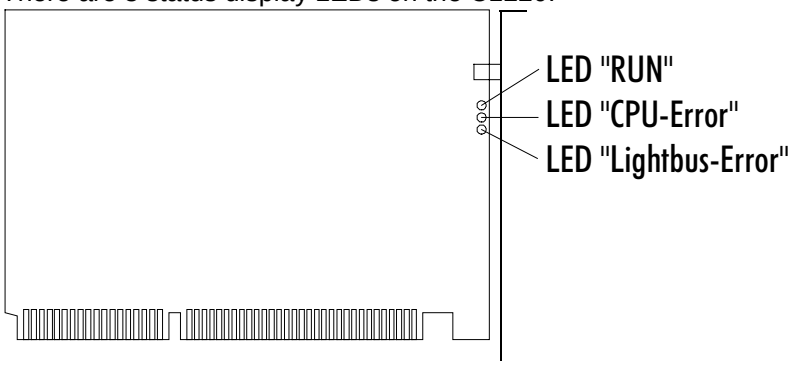
Jumpers J3 and J4

The IRQ number of the Ready interrupt is defined by means of jumper J3. Jumper J4 defines the IRQ number of the fast interrupt inputs.



Status display

There are 3 status display LEDs on the C1220.



"RUN" LED

The 'RUN' LED indicates that the C1220 has initialized without errors and is ready for operation.

"CPU-Error" LED

An irrecoverable hardware fault has occurred if only this LED lights up. If the 'RUN' LED also lights up, a program error has occurred which it might be possible to remedy by means of a hardware reset.

"Lightbus-Error" LED

'LWL-FAIL' LED is activated if a defect occurs in the fibre optic ring during operation. The LED flashes if a general fibre-optic error has occurred. The LED is statically activated if the error has occurred during the resident address check. Updating of the process image is interrupted. The cause of the error can be determined by means of the available diagnostic functions.

Installation in the PC

1. Switch off the PC and any external power supplies.

2. Insert the C1220 interface card in a 16-Bit ISA bus slot on the PC's bus board.

The C1220 does not require an external power supply. The card is powered directly by the PC. Therefore, when the PC is switched on, the C1220 also assumes operation. Before the C1220 can assume operation, however, the fibre-optic connections must be established and the jumpers of the C1220 must be configured correctly.