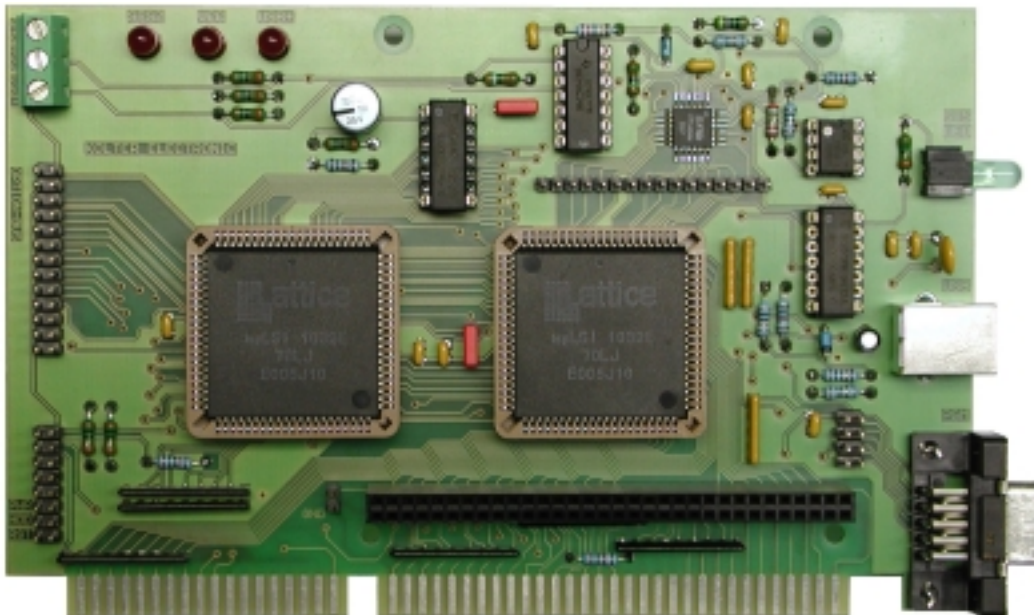


Aktiver Buskonverter USB → ISA

USB nach 8/16-bit-ISA und/oder PC104

Buskonverter-Karte um I/O-Zugriffe über die USB-Schnittstelle auf ein passives ISA-Backplane und/oder ein PC104-Bussystem zu ermöglichen



Industrie-Datenerfassung mit dem PC

KOLTER ELECTRONIC

Tel.: 02235-76707

Fax.: 02235-72048

e-mail: service@pci-card.com

Internet: www.pci-card.com



Inhalt

Sicherheits- und Gefahrenhinweise	3
Anwendung der Buskonverter-Karte	5
Funktion	6
Blockschaltbild	7
Implementierung der Konverterkarte	8
Die DLL-Aufrufe	9
Belegung des ISA-Busses	12
Anschriften und Rufnummernverzeichnis	13



Sehr geehrter Kunde,

wir bedanken uns für den Kauf der USB nach ISA Buskonverter-Karte. Mit dieser Karte haben Sie ein Produkt erworben, welches nach dem heutigen Stand der Technik gebaut wurde.

Dieses Produkt erfüllt die Anforderungen der geltenden europäischen und nationalen Richtlinien. Die Konformität wurde nachgewiesen, die entsprechenden Erklärungen und Unterlagen sind beim Hersteller hinterlegt.

Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen müssen Sie als Anwender diese Bedienungsanleitung beachten!

Bei Fragen wenden Sie sich an unsere Technische Beratung. Rufnummern und Adressen dazu finden Sie unten auf dem Titelblatt oder hinten im Anhang.

Diese Bedienungsanleitung gehört zu diesem Produkt. Sie enthält wichtige Hinweise zur Inbetriebnahme und Handhabung. Achten Sie hierauf, auch wenn Sie dieses Produkt an Dritte weitergeben.

Das Gerät hat den Hersteller in sicherheitstechnisch einwandfreiem Zustand verlassen. Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen, muß der Anwender die Sicherheitshinweise und Warnvermerke beachten, die in dieser Gebrauchsanweisung enthalten sind.

Eine andere Verwendung als die beschriebene führt zur Beschädigung dieses Produktes, darüber hinaus ist dies mit Gefahren, wie z. B. Kurzschluß, Brand, elektrischer Schlag etc. verbunden. Das gesamte Produkt darf nicht geändert bzw. umgebaut und die Gehäuse nicht geöffnet werden!

Besuchen Sie uns unter <http://www.pci-card.com> im Internet

Sicherheits- und Gefahrenhinweise

Allgemein

Achtung! Bei Schäden, die durch Nichtbeachten dieser Bedienungsanleitung verursacht werden, erlischt der Garantieanspruch! Für Folgeschäden übernehmen wir keine Haftung!

Bei Sach- oder Personenschäden, die durch unsachgemäße Handhabung oder Nichtbeachten der Sicherheitshinweise verursacht werden, übernehmen wir keine Haftung! In solchen Fällen erlischt jeder Garantieanspruch.

- Sollten Sie sich über den korrekten Anschluß nicht im klaren sein oder sollten sich Fragen ergeben, die nicht im Laufe der Bedienungsanleitung abgeklärt werden, so setzen Sie sich bitte mit unserer technischen Support oder einem anderen Fachmann in Verbindung.
- Es ist vor der Inbetriebnahme eines Gerätes generell zu prüfen, ob dieses Gerät oder Modul grundsätzlich für den Anwendungsfall, für den es vorgesehen werden soll, geeignet ist.
- Aus Sicherheits- und Zulassungsgründen (CE) ist das eigenmächtige Umbauen und/oder Verändern des Gerätes nicht gestattet.
- Beim Öffnen von Abdeckungen oder Entfernen von Teilen, außer wenn dies von Hand möglich ist, können spannungsführende Teile freigelegt werden. Auch können Anschlußstellen spannungsführend sein. Vor einem Abgleich, einer Wartung, einer Instandsetzung oder einem Austausch von Teilen muß das Gerät von allen Spannungsquellen getrennt sein, wenn ein Öffnen des Gerätes erforderlich ist. Wenn danach ein Abgleich, eine Wartung oder eine Reparatur am geöffneten Gerät unter Spannung unvermeidlich ist, darf das nur durch eine Fachkraft geschehen, die mit den damit verbundenen Gefahren bzw. den einschlägigen Vorschriften dafür vertraut ist.
- Kondensatoren im Gerät können noch geladen sein, selbst wenn das Gerät von allen Spannungsquellen getrennt wurde.
- Elektrische Geräte gehören nicht in Kinderhände. Lassen Sie in Anwesenheit von Kindern besondere Vorsicht walten.



- Lassen Sie das Verpackungsmaterial nicht achtlos liegen, Kunststoffolien bzw. -tüten, Styroporsteile, etc. könnten für Kinder zu einem gefährlichen Spielzeug werden.
- Das Gerät ist nicht für die Anwendung an Menschen oder Tieren zugelassen.
- Gießen Sie nie Flüssigkeiten über den Geräten aus. Es besteht höchste Gefahr eines Brandes oder lebensgefährlichen elektrischen Schlags. Sollte dennoch Flüssigkeit ins Geräteinnere gelangt sein, ziehen Sie sofort das Steckernetzteil aus der Netzsteckdose, bzw. entfernen Sie die Batterien und wenden Sie sich an eine Fachkraft.
- Vermeiden Sie eine starke mechanische Beanspruchung der Geräte.
- Setzen Sie die Geräte keinen extremen Temperaturen, starken Vibrationen oder hoher Feuchtigkeit aus.
- Schalten Sie die Geräte niemals gleich dann ein, wenn sie von einem kalten Raum in einen warmen Raum gebracht wurden. Das dabei entstehende Kondenswasser kann unter Umständen die Geräte zerstören. Lassen Sie die Geräte ausgeschaltet auf Zimmertemperatur kommen. Warten Sie bis das Kondenswasser verdunstet ist.
- Im Fehlerfall können Netzgeräte Spannungen über 50 V Gleichspannung abgeben, von der Gefahren ausgehen, auch dann, wenn die angegebenen Ausgangsspannungen der Geräte niedriger liegen.
- In gewerblichen Einrichtungen sind die Unfallverhütungsvorschriften des Verbandes der gewerblichen Berufsgenossenschaften für elektrische Anlagen und Betriebsmittel zu beachten.
- In Ausbildungseinrichtungen (Schulen) sowie Hobby- und Selbsthilfwerkstätten ist der Umgang mit elektrischen Geräten und deren Zubehör durch geschultes Personal verantwortlich zu überwachen.
- Betreiben Sie das Gerät (oder die Baugruppe) nicht in Räumen oder bei widrigen Umgebungsbedingungen, in/ bei welchen brennbare Gase Dämpfe oder Stäube vorhanden sind oder vorhanden sein können. Vermeiden Sie den Betrieb in unmittelbarer Nähe von elektrostatischen Feldern (Auf-/Entladungen) und Sendeantennen, da es dadurch zu fehlerhaften Anwendungen kommen kann.
- Bei einer mutwilligen mechanischen Beeinträchtigung oder elektrischen Änderung (Umbau) des Meßgerätes erlischt der Garantieanspruch.
- Wenn ein gefahrloser Betrieb nicht mehr möglich ist, so ist das Gerät außer Betrieb zu setzen und gegen unbeabsichtigten Betrieb zu sichern. Es ist anzunehmen, daß ein gefahrloser Betrieb nicht mehr möglich ist, wenn a) das Gerät sichtbare Beschädigungen aufweist, b) das Gerät nicht mehr arbeitet c) nach längerer Lagerung unter ungünstigen Verhältnissen d) nach schweren Transportbeanspruchungen.
- Beachten Sie beim Betrieb des Geätes oder der Baugruppe unbedingt die Umgebungsbedingungen (Arbeits-temperaturbereich, Luftfeuchtigkeit).
- Vermeiden Sie den Betrieb in stark feuchter und nasser Umgebung.

Bei Anschluß an Netzspannung

- Die Geräte sind in Schutzklasse I aufgebaut. Sie sind mit einer VDE-geprüften Netzleitung mit Schutzleiter ausgestattet und dürfen daher nur an 230-V-Wechselspannungsnetzen mit Schutzerdung betrieben bzw. angeschlossen werden.
- Es ist darauf zu achten, daß der Schutzleiter (gelb/grün) weder in der Netzleitung noch im Gerät bzw. im Netz unterbrochen wird, da bei unterbrochenem Schutzleiter Lebensgefahr besteht.
- Bei Arbeiten an Geräten oder Baugruppen, die mit der Netzspannung verbunden sind, ist das Tragen von metallischem oder leitfähigem Schmuck wie Ketten, Armbändern, Ringen o.ä. verboten.
- Bei Arbeiten unter Spannung darf nur dafür ausdrücklich zugelassenes Werkzeug verwendet werden.
- Reparatur- und Wartungsarbeiten an Geräten, die in irgendeiner Form mit der Netzspannung verbunden sind dürfen nur vom Hersteller selbst oder einem Fachmann, der mit den verbundenen Gefahren und den einschlägigen Vorschriften dafür vertraut ist, durchgeführt werden.

Anwendung der Buskonverter-Karte

Allgemeines

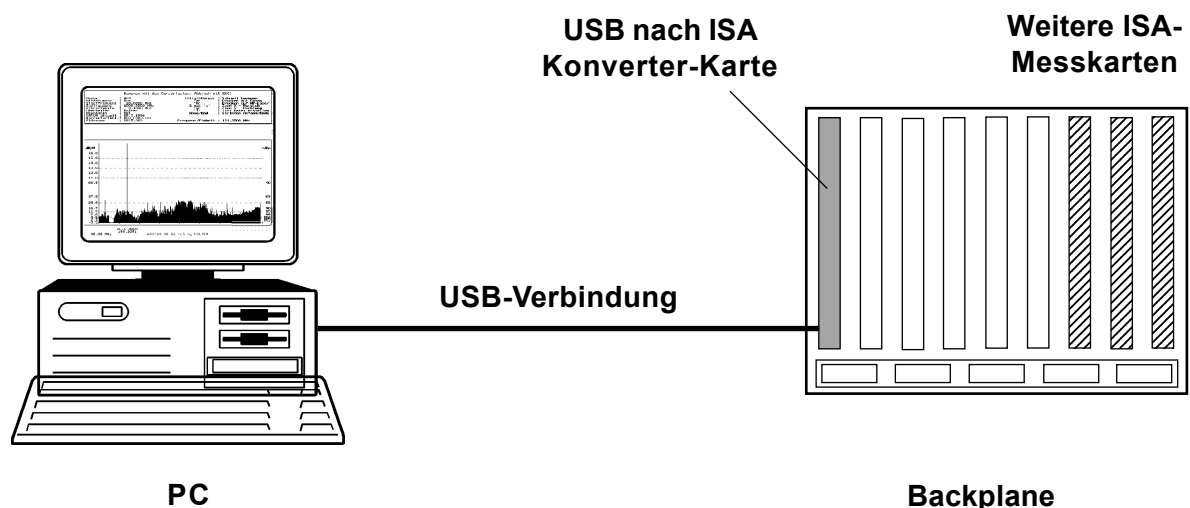
Mainboards der neueren Generation stellen ausnahmslos nur noch PCI-Steckplätze zur Verfügung. Dies ist deshalb so, weil PCs nicht in erster Linie für Messanwendungen sondern für den Office- und Homebereich produziert werden, bei denen Aspekte wie die Verfügbarkeit und Langlebigkeit (zum Beispiel von Schnittstellendefinitionen) nur eine untergeordnete Rolle spielen, da die Computerhardware meist komplett ausgetauscht wird. Überlegungen wie ein bestehendes Umfeld integriert werden kann müssen hier nicht angestellt werden. Mit dem Wegfall des ISA-Busses sehen sich viele professionelle Anwender mit dem Problem konfrontiert, dass nach einem Austausch eines Computersystems die vorhandene Messumgebung mit hohem Kostenaufwand an den PCI-Bus angepasst werden muss. Dabei spielen die relativ hohen Hardwarekosten (PCI-Messkarten) noch nicht einmal die entscheidende Rolle, sondern der Zeitaufwand für die Implementierung der neuen Software (Anpassung selbstentwickelter Software, nötige Testreihen usw.). Zudem möchte man die Erfahrungen nutzen, die aus dem langjährigen Einsatz der ISA-Karten gewachsen sind, anstatt sich auf ein System einzulassen mit dem man hardwaremäßig und programmiertechnisch Neuland betritt. Zudem kann es vorkommen, dass es eine geeignete PCI-Messkarte für die angestrebte Anwendung (noch) gar nicht gibt.

Systembeschreibung

Die Buskonverter-Karte bietet die Möglichkeit, die oben genannten Probleme zu umgehen, indem die „alten“ ISA-Bus-Karten auch an den neuen Rechnern, die nur einen PCI-Bus verfügen, weiterhin genutzt werden können. Voraussetzung ist lediglich dass diese PCs über eine USB-Schnittstelle verfügen, da die Anbindung über diese Schnittstelle realisiert wurde.

Die Buskonverter-Karte wird dabei, wie die anzusteuern ISA-Messkarten, in ein passives ISA-Backplane (es gibt sie mit bis zu 20 ISA-Steckplätzen) eingesteckt und über das USB-Kabel mit dem Computer verbunden.

Als Besonderheit muss noch herausgestellt werden, dass auf der Konverter-Karte die ISA-Bussignale auch für die PC/104-Spezifikation zur Verfügung stehen, das heißt PC/104-Erweiterungskarten können direkt auf die Konverterplatine gesteckt werden.

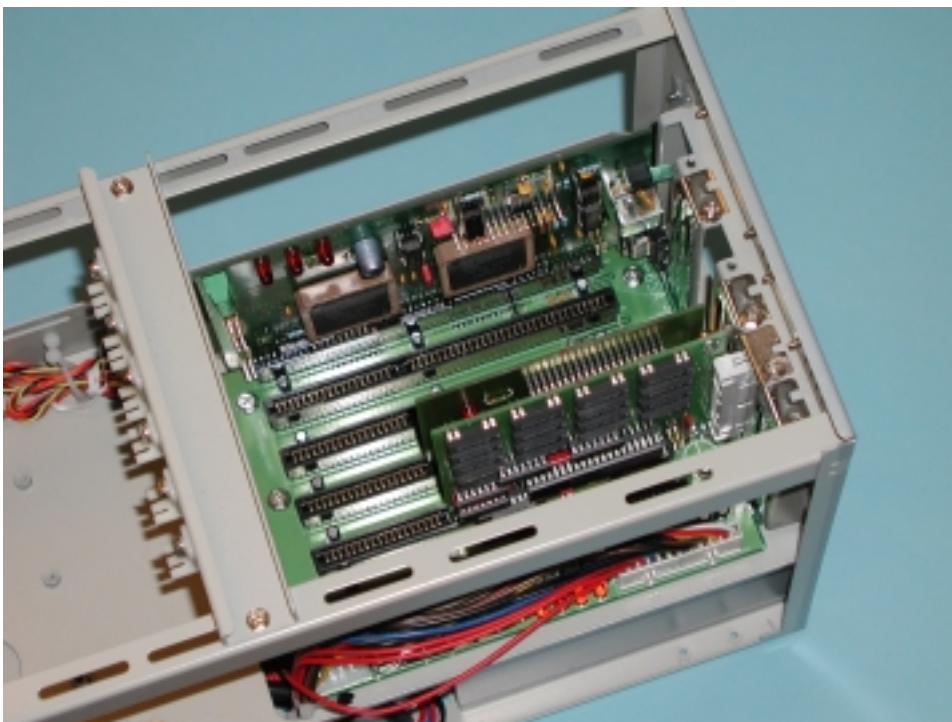


Funktion

Die Buskonverter-Karte USB nach ISA stellt dem Anwender reine Port- und Memory-Zugriffe mit IRQ zur Verfügung, damit er seine ISA-Karten am USB-Port betreiben kann.

Die neue USB-zu-ISA-Bus Interface-Karte koppelt dazu einen passiven ISA-Bus (ISA-Backplane) an eine USB-Schnittstelle, also unter Ansteuerung von USB-Signalen, in beide Richtungen. Dabei entfällt die typische Verwendung einer all-in-one CPU-Karte völlig, da die I/O-Befehle von einem Windows-Hostrechner übernommen werden und sie durch die USB-ISA Brücken (2x ispLSI1032E) auf dem Bus in Echtzeit emuliert wird. Das ISA-Bus-Timing gilt als absolut unkritisch und entspricht in etwa einem 6 MHz AT286. Mit diesem langsamen Timing werden somit auch alle älteren ISA-Karten der ersten Generation noch voll unterstützt. Obgleich die USB1.1-Verbindung mit einer Datenrate von etwa 1 MB/s arbeitet (12Mbit/s lt. USB) liegt die tatsächliche Worst-Case Ausführungszeit (von der Anwendung bis zur ISA-Karte) von 8- oder 16-bit Befehlen bei nur 1000 Anweisungen pro Sekunde.

Die Verbindung zwischen Anwendung und ISA-Karte wird durch eine Dynamic Link Library (DLL) erreicht, die zum eigenen Programmsource über CALL-Funktionen, angebunden beziehungsweise angesteuert wird. Über einen SYS-Treiber wird der Kontakt vom Windows-Betriebssystem zum USB-Interface aufgenommen. Anschliessend durchlaufen die Daten eine ISA-Bus-Emulation bestehend aus zwei ispLSI1032E cPLD-ICs. Sie übernehmen die aktive Koordinierung und stellen die Verbindung zwischen USB-Interface und dem eigentlichen ISA-Bus her. Verschiedene State-Machines, Zähler und Vergleicher verarbeiten diverse Frames, in denen I/O- beziehungsweise MEM-Adressen, Daten sowie die Steuersignale und Interruptsignale, nach einem vorgegebenen Ablaufschema gesammelt und erzeugt werden. Damit wird der ISA-Bus quasi transparent über die DLL abgebildet, so dass nur minimaler Entwicklungsaufwand nötig ist, um eine bestehende Anwendung an dieses System anzupassen. Eine typische Anwendung ist beispielsweise die Ankopplung an ein passives ISA-Bus-System, wie sie im Bereich von Industrie-Rechnern sogenannten IPCs her bekannt sind. Hierzu bieten wir entsprechende Gehäuse mit Netzteil separat an, die bereits über einen passiven ISA-Bus mit drei, vier, sechs oder mehr 16-bit ISA-Steckplätze verfügen.

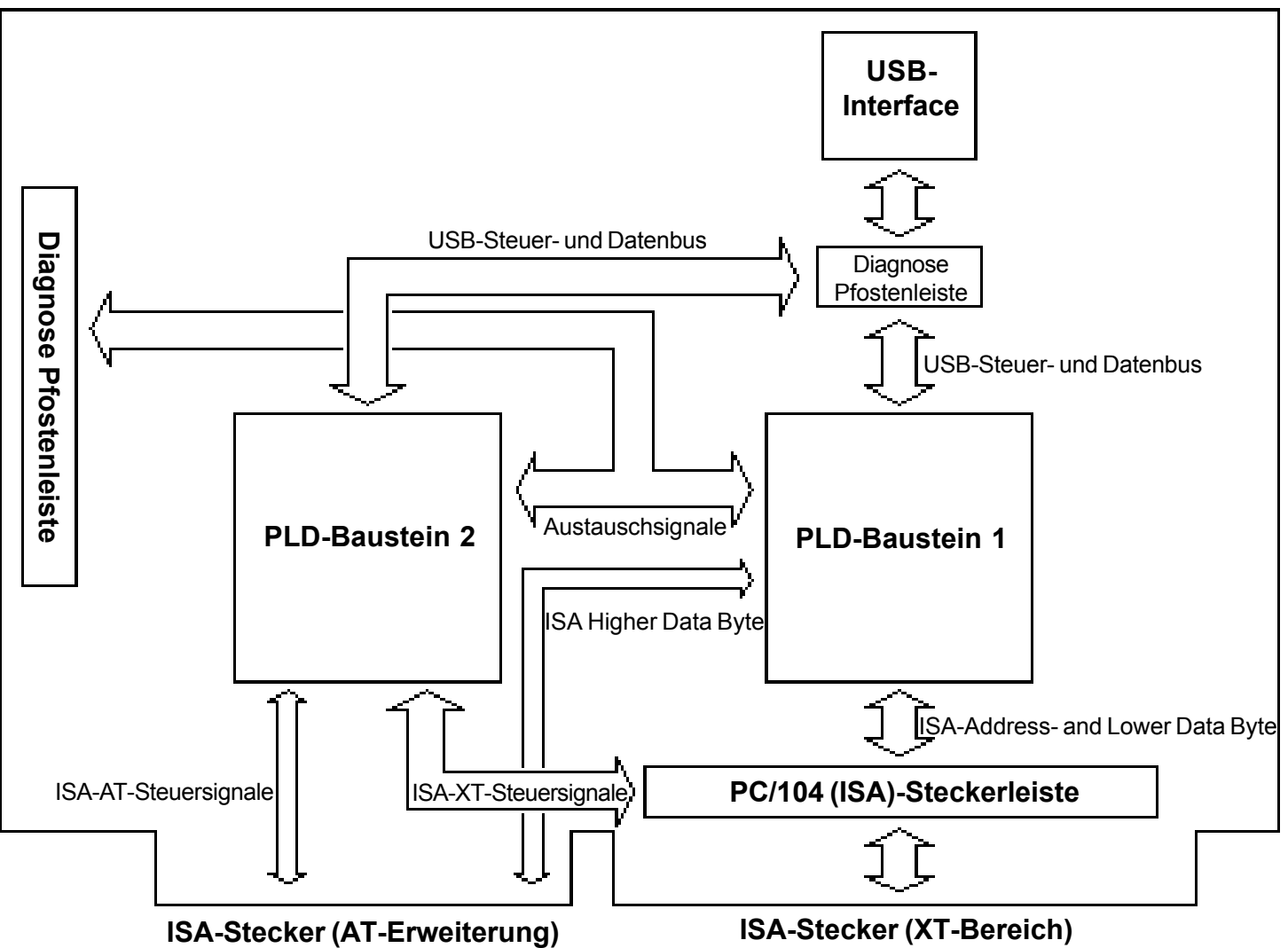


Ansicht der USB nach ISA Konverter-Karte auf einer ISA-Backplane.

Über den USB-Bus und die Konverterkarte werden dann die ISA-Karten angesteuert.

Für diese Kombination sind spezielle Gehäuse erhältlich, die bereits mit einem Netzteil ausgerüstet sind.

Blockschaltbild



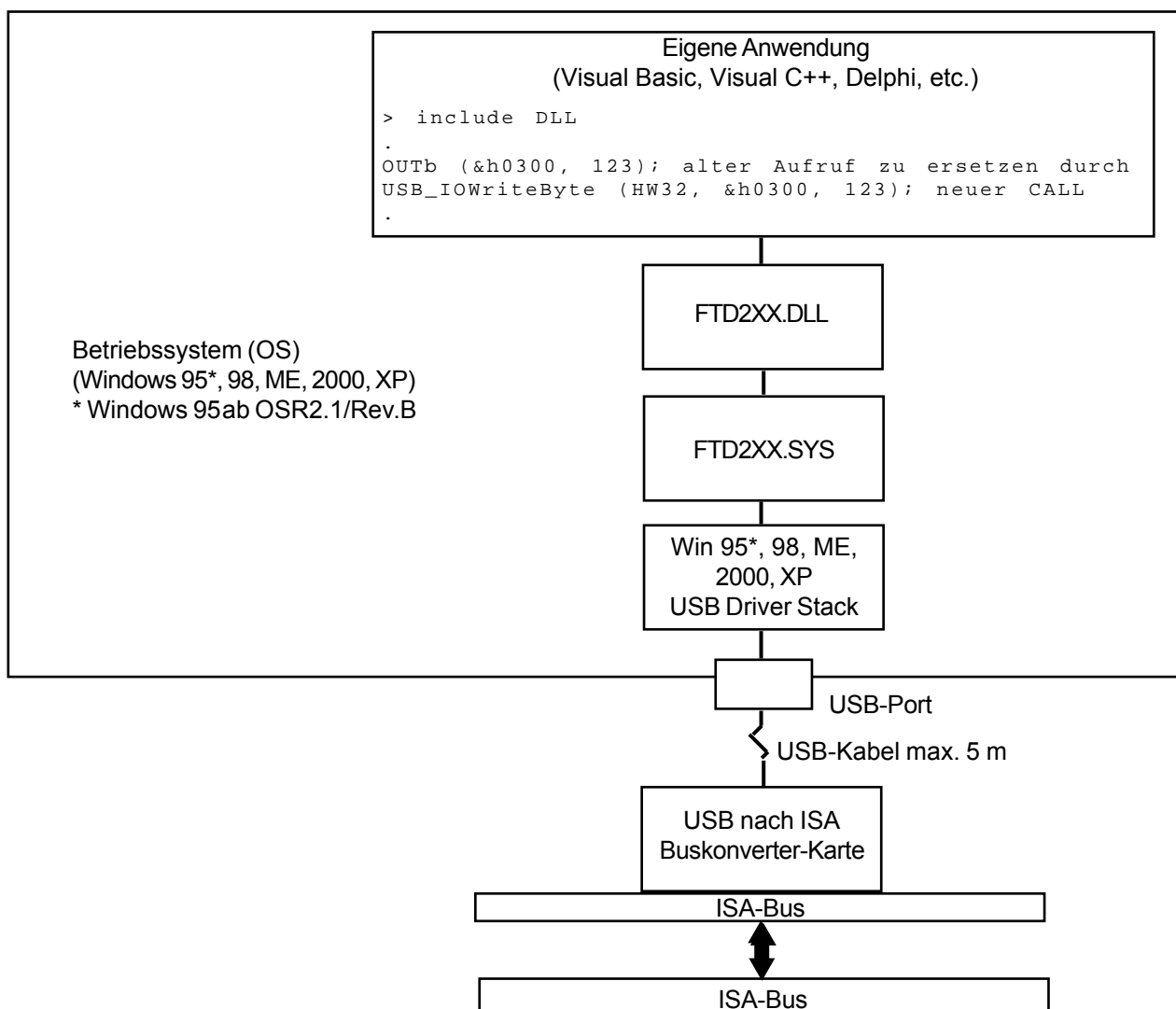


Implementierung der Konverterkarte

Ein weiterer Pluspunkt, zu den Vorteilen der Weiternutzung der „alten“ ISA-Karten in bestehenden Systemumgebungen, ist die einfache softwaremäßige Einbindung in die vorhandene Anwendungsprogramme. Wegen des durchdachten Konzepts sind dazu nur wenige Schritte nötig.

1. In der bereits vorhandenen Anwendungssoftware (Visual C++, Visual Basic, Delphi usw.) werden lediglich die IN/OUT-Befehle durch einen CALL-Aufruf für die entsprechende DLL ersetzt (am einfachsten per „Suchen und Ersetzen“). Für Visual Basic-Anwendungen muss vor der Neucompilierung das Modul „USB_ISA.BAS“ mit eingebunden werden.
2. Kopieren der FTD2XX.DLL in das Verzeichnis, in dem sich die Anwendungssoftware befindet.
3. Einbindung der FTD2XX.SYS. Diese Datei wird folgendermaßen installiert: Zuerst wird das Kontextmenü der Datei FTD2XX.INF auf der CD (Pfad: \Drivers\Usb\Usb-2-ISA) durch einen Rechtsklick auf den Dateinamen geöffnet – dann im Kontextmenü „Installieren“ wählen.
4. Fertig.

Um die Arbeitsweise zu verdeutlichen, hier eine Übersicht zum Ablauf der ISA-Bus-Zugriffe über den USB-Port.





Die DLL-Aufrufe

Damit die DLL-Aufrufe nicht ins Leere laufen, ist je nach Programm-Entwicklungsumgebung für Visual Basic, Visual C++, Delphi usw. eine spezielles Modul notwendig, mit der die einzelnen CALLs definiert aufgerufen werden können. Für Visual Basic ist im Folgenden das Listing dieses Moduls explizit angegeben. Kommen andere Programm-Entwicklungsumgebungen zum Einsatz kann dieses Listing als Grundlage zur Umsetzung in die entsprechenden Module (zum Beispiel für C++) genutzt werden.

```
Attribute VB_Name = "Module1"
'Declare Function FT_Open Lib "FTD2XX.DLL" (ByVal intDeviceNumber As Integer,
ByRef lngHandle As Long) As Long
'Declare Function FT_Close Lib "FTD2XX.DLL" (ByVal lngHandle As Long) As Long
'Declare Function FT_Read Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal
lpszBuffer As String, ByVal lngBufferSize As Long, ByRef lngBytesReturned As
Long) As Long
'Declare Function FT_Write Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal
lpszBuffer As String, ByVal lngBufferSize As Long, ByRef lngBytesWritten As
Long) As Long
'Private Declare Function FT_SetBaudRate Lib "FTD2XX.DLL" (ByVal lngHandle As
Long, ByVal lngBaudRate As Long) As Long
'Private Declare Function FT_SetDataCharacteristics Lib "FTD2XX.DLL" (ByVal
lngHandle As Long, ByVal byWordLength As Byte, ByVal byStopBits As Byte, ByVal
byParity As Byte) As Long
'Private Declare Function FT_SetFlowControl Lib "FTD2XX.DLL" (ByVal lngHandle
As Long, ByVal intFlowControl As Integer, ByVal byXonChar As Byte, ByVal byXoffChar
As Byte) As Long
'Private Declare Function FT_ResetDevice Lib "FTD2XX.DLL" (ByVal lngHandle As
Long) As Long
'Private Declare Function FT_SetDtr Lib "FTD2XX.DLL" (ByVal lngHandle As Long)
As Long
'Private Declare Function FT_ClrDtr Lib "FTD2XX.DLL" (ByVal lngHandle As Long)
As Long
'Private Declare Function FT_SetRts Lib "FTD2XX.DLL" (ByVal lngHandle As Long)
As Long
'Private Declare Function FT_ClrRts Lib "FTD2XX.DLL" (ByVal lngHandle As Long)
As Long
'Private Declare Function FT_GetModemStatus Lib "FTD2XX.DLL" (ByVal lngHandle
As Long, ByRef lngModemStatus As Long) As Long
Declare Function FT_Purge Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal
lngMask As Long) As Long
'Private Declare Function FT_GetStatus Lib "FTD2XX.DLL" (ByVal lngHandle As
Long, ByRef lngRxBytes As Long, ByRef lngTxBytes As Long, ByRef lngEventsDWord
As Long) As Long
Declare Function FT_GetQueueStatus Lib "FTD2XX.DLL" (ByVal lngHandle As Long,
ByRef lngRxBytes As Long) As Long
'Private Declare Function FT_GetEventStatus Lib "FTD2XX.DLL" (ByVal lngHandle
As Long, ByRef lngEventsDWord As Long) As Long
'Private Declare Function FT_SetChars Lib "FTD2XX.DLL" (ByVal lngHandle As
Long, ByVal byEventChar As Byte, ByVal byEventCharEnabled As Byte, ByVal byErrorChar
As Byte, ByVal byErrorCharEnabled As Byte) As Long
'Private Declare Function FT_SetTimeouts Lib "FTD2XX.DLL" (ByVal lngHandle As
Long, ByVal lngReadTimeout As Long, ByVal lngWriteTimeout As Long) As Long
'Private Declare Function FT_SetBreakOn Lib "FTD2XX.DLL" (ByVal lngHandle As
Long) As Long
```



```
'Private Declare Function FT_SetBreakOff Lib "FTD2XX.DLL" (ByVal lngHandle As Long) As Long
Declare Function FT_Read Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal lpszBuffer As String, ByVal lngBufferSize As Long, ByRef lngBytesReturned As Long) As Long
Declare Function FT_Write Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal lpszBuffer As String, ByVal lngBufferSize As Long, ByRef lngBytesWritten As Long) As Long
Declare Function USB_Open Lib "FTD2XX.DLL" (ByVal intDeviceNumber As Integer, ByRef lngHandle As Long) As Long
Declare Function USB_Close Lib "FTD2XX.DLL" (ByVal lngHandle As Long) As Long
Declare Function USB_Init Lib "FTD2XX.DLL" (ByVal lngHandle As Long) As Long
Declare Function USB_ReadIRQ Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByRef IRQ As Long) As Long
Declare Function USB_ResetISA Lib "FTD2XX.DLL" (ByVal lngHandle As Long) As Long
Declare Function USB_IOWriteByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByVal Daten As Long) As Long
Declare Function USB_IOWriteHighByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByVal Daten As Long) As Long
Declare Function USB_IOWriteWord Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByVal Daten As Long) As Long
Declare Function USB_IOReadByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByRef Daten As Long) As Long
Declare Function USB_IOReadHighByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByRef Daten As Long) As Long
Declare Function USB_IOReadWord Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByRef Daten As Long) As Long
Declare Function USB_MEMWriteByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByVal Daten As Long) As Long
Declare Function USB_MEMWriteHighByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByVal Daten As Long) As Long
Declare Function USB_MEMWriteWord Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByVal Daten As Long) As Long
Declare Function USB_MEMReadByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByRef Daten As Long) As Long
Declare Function USB_MEMReadHighByte Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByRef Daten As Long) As Long
Declare Function USB_MEMReadWord Lib "FTD2XX.DLL" (ByVal lngHandle As Long, ByVal Addr As Long, ByRef Daten As Long) As Long

' Return codes
Const FT_OK = 0
Const FT_INVALID_HANDLE = 1
Const FT_DEVICE_NOT_FOUND = 2
Const FT_DEVICE_NOT_OPENED = 3
Const FT_IO_ERROR = 4
Const FT_INSUFFICIENT_RESOURCES = 5

' Globale Variablen
Public Handle As Long
Public HW32 As Long
Public ActiveHW As Boolean ' Treiber gestartet
Public AddressRangeLimitLow, AddressRangeLimitHigh, DataRangeLimitLow As Long
Public DataRangeLimitHigh As Single ' !Long und Integer sind vorzeichenbehaftet!
Public adr As Long
Public dummy As Long
Public us As Long
Public value_b As Byte
Public value_w As Long
```



```
Function CloseHandle()  
  
If USB_Close(Handle) <> FT_OK Then  
    StatusFenster.AddItem "Handle konnte nicht geschlossen werden!"  
End If  
'StatusFenster.AddItem "Handle geschlossen!"  
  
End Function  
  
Function OpenHandle()  
If USB_Open(0, Handle) <> FT_OK Then  
    StatusFenster.AddItem "Handle konnte nicht geöffnet werden!"  
End If  
'StatusFenster.AddItem "Handle geschlossen!"  
  
End Function  
Public Function HexToInt(ByVal strMyString As String) As Long  
    Dim lngMyInteger As Long  
    lngMyInteger = 0  
    On Error Resume Next  
    lngMyInteger = "&h" & strMyString  
    HexToInt = lngMyInteger  
End Function  
  
Public Function IntToHex2(ByVal MyVal As Byte) As String  
    Dim s As String  
    s = Hex(MyVal)  
    If Len(s) = 1 Then s = "0" & s  
    IntToHex2 = s  
End Function  
Public Function IntToHex4(ByVal MyVal As Integer) As String  
    Dim s As String  
    s = Hex(MyVal)  
    While Len(s) < 4  
        s = "0" & s  
    Wend  
    IntToHex4 = s  
End Function  
Public Function IntToHex8(ByVal MyVal As Long) As String  
    Dim s As String  
    s = Hex(MyVal)  
    While Len(s) < 8  
        s = "0" & s  
    Wend  
    IntToHex8 = s  
End Function
```



Belegung des ISA-Busses

A		B		C		D	
CHKCH	1	1	GND	SBHE	1	1	M16
SD7	2	2	RES DRV	LA23	2	2	IO16
SD6	3	3	+5V	LA22	3	3	IRQ10
SD5	4	4	IRQ9	LA21	4	4	IRQ11
SD4	5	5	-5V	LA20	5	5	IRQ12
SD3	6	6	DRQ2	LA19	6	6	IRQ15
SD2	7	7	-12V	LA18	7	7	IRQ14
SD1	8	8	NOVS	LA17	8	8	DAK0
SD0	9	9	+12V	MRDC	9	9	DRQ0
CHRDY	10	10	GND	MWTC	10	10	DAK5
AEN	11	11	SMWTC	SD8	11	11	DRQ5
SA19	12	12	SMRDC	SD9	12	12	DAK6
SA18	13	13	IOWC	SD10	13	13	DRQ6
SA17	14	14	IORC	SD11	14	14	DAK7
SA16	15	15	DAK3	SD12	15	15	DRQ7
SA15	16	16	DRQ3	SD13	16	16	+5V
SA14	17	17	DAK1	SD14	17	17	MASTER16
SA13	18	18	DRQ1	SD15	18	18	GND
SA12	19	19	REFRESH				
SA11	20	20	BCLK				
SA10	21	21	IRQ7				
SA9	22	22	IRQ6				
SA8	23	23	IRQ5				
SA7	24	24	IRQ4				
SA6	25	25	IRQ3				
SA5	26	26	DAK2				
SA4	27	27	TC				
SA3	28	28	BALE				
SA2	29	29	+5V				
SA1	30	30	OSC				
SA0	31	31	GND				

ISA-Bus Belegung (AT-Erweiterung)

ISA-Bus Belegung (XT-Bereich)



Anschriften und Rufnummernverzeichnis

Anschriften

Postfach 1127 D-50362 Erftstadt
Steinstraße 22 D-50374 Erftstadt

Rufnummern

Auslandsvorwahl ++49 22 35
Inlandsvorwahl 0 22 35

Vertrieb und Service 7 67 07
Fax 7 20 48

Werkstatt und Prüffeld 69 18 52
BBS Mailbox-Modem 95 37 30
Pressestelle 95 37 31
Geschäftsleitung 95 37 32
ISDN (nur auf Anfrage) 69 18 52
E-Fax 0 40 36 03 - 13 99 39

Fax-Abruf-Service

Hauptkatalog, 32 Seiten 0 22 35 - 68 91 19
aktuelle Preisliste, 8 Seiten 0 22 35 - 68 91 27
OPTO-PCI-Karte, 20 Seiten 0 22 35 - 68 91 28
PCI-1616-Karte, 19 Seiten 0 22 35 - 68 91 29
neue Produkte, Kurzvorstellung 0 22 35 - 68 91 33
PCI-Karten, K98/99 Antenne 0 22 35 - 95 36 69

Internet

E-Mail - Service service@pci-card.com
E-Mail - Technik technik@pci-card.com
E-Mail - Info info@pci-card.com
E-Mail - Webmaster webmaster@pci-card.com
E-Mail - Herr Kolter hkolter@pci-card.com

Haupt-Domains <http://www.pci-card.com>
<http://www.kolter.de>

Redirects <http://www.emv-messtechnik.de>
<http://www.pci-messtechnik.de>
<http://www.messkarten.de>
<http://www.pc-messkarten.de>

Server mit Frames <http://www.pci-card.com/index.htm>
Server ohne Frames <http://www.pci-card.com/home2.htm>