PCI - DAC 416

4 Kanal D/A-Converter 16 bit

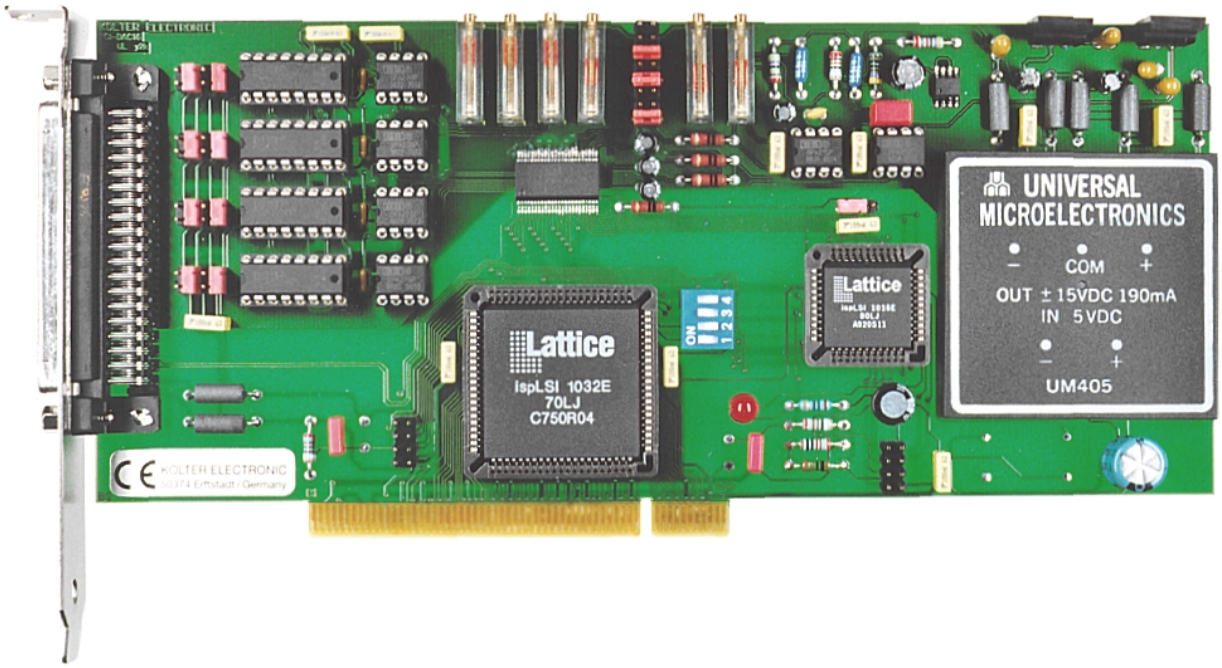
4-Kanal D/A-Karte mit Referenz-Eigenschaften

PCI - DAC 416 AP/E (Standard Version, ≤ 25 µs)

PCI - DAC 416 BP/E (erhöhte Genauigkeit, ≤ 25 µs)

PCI - DAC 416 PA/E (schnelle Version, 2µs)

PCI - DAC 416 P/E (schnelle Version mit erhöhter Genauigkeit)



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

Willkommen

Sehr geehrter Kunde,

wir bedanken uns für den Kauf oder das Interesse an unserer PCI-DAC416 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 EMV-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 Betriebsanleitung sowie weitere Sicherheitsdokumente s.u. beachten.

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

Diese Bedienungsanleitung gehört zu diesem Produkt. Sie enthält wichtige Hinweise zur Inbetriebnahme und Handhabung bei der Installation. Achten Sie hierauf, auch wenn Sie dieses Produkt an Dritte weitergeben. Das Produkt hat den Hersteller in sicherheitstechnisch einwandfreiem Zustand verlassen. Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen, muß der Anwender alle Sicherheitshinweise und Warnvermerke beachten, die in dieser Gebrauchsanweisung enthalten sind. Ggf. müssen weitere Hinweise beachtet werden, die Sie jedoch nur online von unserer Webseite herunterladen können. Beipielsweise haben wir eine FAQ-Seite eingerichtet, um wiederkehrende Fragen ausführlich zu beantworten, die diese Betriebsanleitung vom Umfang her sicher sprengen würde.

Achtung:
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. Die nachfolgenden Sicherheits- und Gefahrenhinweise ergeben sich zu diesem Produkt in der Form, dass der Einbau in/an einem Industrie-PC in industrieller Umgebung als Anlage erfolgt. Somit sind möglicherweise auch übergeordnete Sicherheits- und Gefahrenhinweise relevant, die unser Produkt zwar nicht unmittelbar betreffen, jedoch in ihrer Gesamtheit als industrielle Anlage beachtet werden müssen. Der Einbau, sowie die Inbetriebnahme darf daher nur durch geschultes Fachpersonal, oder durch einen ausgebildeten Techniker erfolgen. Aus Gründen der ständigen Gesetzesänderungen und EU-Richtliniennovellen haben wir uns entschlossen, diese Hinweise als Zusammenfassung in einem separaten Dokument halbjährlich zu aktualisieren und online zu stellen.

Die aktuellen Sicherheits- und Gefahrenhinweise finden Sie auf unserer Webseite unter:

http://www.pci-card.com/SiGef-Hinweise.PDF

Vielen Dank.

Der Einbau in den PC

ACHTUNG:
Einbau und Inbetriebnahme dürfen nur von technisch geschultem Personal erfolgen.

1. Schalten Sie den Rechner und alle daran angeschlossenen Geräte und Anlagen aus.

Bitte beachten Sie:
Potentialunterschiede und statische Aufladung (ESD) kann Ihren Computer und dieses Produkt zerstören!

Entladen Sie sich daher vor dem Weiterarbeiten, indem Sie eine Wasserleitung, ein Heizungsrohr oder ein anderes Metallteil mit Erdverbindung berühren. Die Potentialneutralität ist die Vorraussetzung für jeden Um- und Einbau, sowie die Verbindung mit anderen Anlagen, Komponenten oder Teilen.

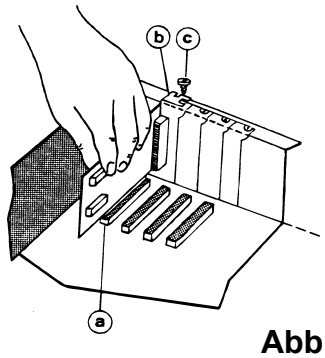


Abb. 1

2. Öffnen Sie den PC. Im allgemeinen müssen dazu auf der Rückseite des Gerätes vier Sicherungsschrauben mit einem Kreuzschlitzschraubendreher gelöst werden. An- schließend können Sie das Gehäuse nach vorne hin wegziehen. Eventuell müssen Sie einige behindernde Kabel entfernen, merken Sie sich jedoch unbedingt die zugehörigen Buchsen bzw. die Steckanordnung (ev. aufschreiben).

3. Die Einsteckplätze befinden sich am hinteren Ende Ihres Rechners. Die Rückwand nicht benutzter Plätze wird von einem Schutzblech verdeckt. Suchen Sie einen freien Einsteckplatz und entfernen Sie das dazugehörige Schutzblech, indem Sie seine Halterungsschraube lösen.

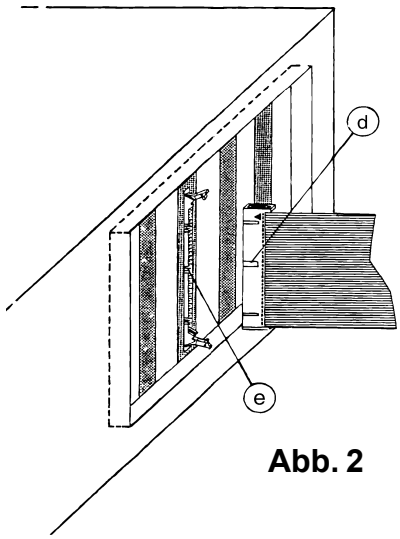


Abb. 2

4. Stecken Sie die Erweiterungskarte in den freien Steckplatz Abb. 1 (a). Achten Sie auf festen Sitz und darauf, daß Sie die Karte beim Einstecken senkrecht halten.

5. Positionieren Sie die Karte mittig über das Befestigungsloch (Gewinde). Befestigen Sie anschließend das Halterungsblech der Karte Abb. 1 (b) mit der Schraube (c) des Schutzbleches.

6. Schließen Sie das Gehäuse Ihres Rechners und befestigen Sie es mit den Sicherungsschrauben. Kabel, die Sie während des Einbaus gelöst haben, sollten Sie nun wieder einstecken. Stecken Sie die/das Anschlußkabel Abb. 2 (d) der Karte in die vorgesehenen Buchse/n (e) und beachten Sie die VDE-Handhabungsvorschriften. Schalten Sie immer zuerst den Rechner ein, um anschließend, beispielsweise eine Spannung zu messen.
Nie umgekehrt !!!

Weitere Informationen finden Sie unter: http://www.pci-card.com/faq.html

Allgemeines zu I/O-Karten

Wenn ein PC zeitlich festgelegte Abläufe innerhalb einer Produktion steuern oder komplexe Prozesse regeln soll, muß man ihn zuerst in die Lage versetzen, die nötigen analogen oder digitalen Meßsignale aufnehmen und ausgeben zu können. Dazu verwendet man am besten eine möglichst exakt auf die jeweilige Aufgabenstellung zugeschnittene Peripherikarte, auf der alle nötigen Ein- und Ausgänge vorhanden sind und mit der auch noch gleich die Pegel anpaßt werden.

Da man, angesichts der Menge der zu automatisierenden Abläufe, diese Karte in der Praxis kaum finden wird, bietet sich als zweitbeste Lösung die Verwendung mehrerer Karten an, die jeweils einen Teilbereich der Aufgabenstellung abdecken.

Häufig werden beispielsweise TTL-I/O-Karten genutzt, die oft viele Signale ein- und ausgeben können, aber nur solche, die im TTL-Pegelbereich von 0...5 V angesiedelt sind. Oder es kommen Timer-Karten zum Einsatz, wenn Taktzeiten leicht zu verändern, aber präzise einstellbar sein müssen.

Optokoppler- und Relais-Karten dienen zur Potentialtrennung zwischen dem PC und der Anlagenseite und können sowohl TTL als auch andere Spannungswerte verarbeiten. Um auch größere Ströme bis zu einigen Ampére schalten zu können, setzt man Karten mit elektro-mechanisch arbeitenden Relais oder sogenannte Halbleiter-Relais ein.

Zur Erfassung physikalischer Größen braucht man analog-/digital-Wandlerkarten, die mit Auflösungen zwischen 8 Bit und 24 Bit und Wandlungsraten von einigen kHz bis zu mehreren MHz verfügbar sind. Mit den in gleicher Variationsbreite lieferbaren digital-/analog-Umsetzern kann man die Steuerspannungen erzeugen, mit denen beispielsweise Sollwertvorgaben an analogen Reglern verändert werden können.

Zur Nutzung einer beliebigen I/O-Karte braucht man immer ein speziell auf die jeweilige Karte zugeschnittenes Steuerprogramm, welches für die Einbindung der Karte in das Betriebssystem des Computers sorgt. Im einfachsten Fall ist das ein mehr oder weniger kleines Treiberprogramm, das beim Booten des Rechners geladen und gestartet wird, während des Betriebs aber nicht mehr weiter in Erscheinung tritt.

Aufwendigere Lösungen beinhalten einen oder mehrere Treiber und ein Anwendungsprogramm, das auf eine spezielle Aufgabenstellung zugeschnitten ist. Der Rechner wird dann üblicherweise auch nur für diese eine Anwendung genutzt.

Die neuen PCI-Karten erlauben eine neue und wesentlich komfortablere Art der I/O-Kartensteuerung, die außerdem noch deutlich flexibler ist, als die herkömmliche Methode. Die Karte enthält quasi ihr eigenes Betriebsprogramm und wird über das Rechner-BIOS initialisiert. Einmal vom BIOS erfasst, wird die Karte unter einer eigenen Zugriffadresse im BIOS geführt und kann mit verschiedenen PCI-Routinen leicht gelesen beziehungsweise verändert werden.

Wenn man Treiberprogramme verwendet, ist man normalerweise an deren Funktionalität gebunden und kann diese nicht weiter verändern. Ein separates Betriebsprogramm für eine oder mehrere Karten lastet den Rechner stark aus und schränkt die Einsatzmöglichkeiten ein.

Die neue Industrie-PCI-Line von Kolter Electronic ermöglicht einen vielfältigen Einsatz in der Automatisierungstechnik. Mit Hilfe der neuen MS_VC++ Programmierbeispielen unter DOS, Windows, und Linux ist der Anwender direkt in der Lage, seine Aufgaben schnell und professionell zu lösen.

Funktionsweise der Karte

Die neue Serie „PCI-DAC 416 mit Sense-Eingängen“ ist für vier verschiedene Grundbestückungsarten bei nur einem Kartendesign ausgelegt. Daraus ergibt sich eine bestmögliche Anpassung an die unterschiedlichsten Aufgaben – sei es zur Erzeugung einer analogen Spannung für Präzisionsanwendungen oder für einen zeitkritischen Einsatz. Dabei kann nochmals, durch die Verwendung von präziseren Ausgangsverstärkern (genauere Offset Spannungen) , zwischen normaler und erhöhter Genauigkeit gewählt werden. Zudem besteht auf Kundenwunsch die Möglichkeit die Präzision der PCI-Karte durch eine Sonderbestückung mit einem ausgesuchten DA-Converterbaustein zu verdoppeln (der lineare Fehler wird von ±3 LSB auf ±2 LSB verringert, der lineare Gleichlauf von ±4 LSB auf ±2 LSB und der lineare Differenzfehler von ±2 auf ±1 LSB.

Mit der PCI-DAC 416-Karte lassen sich softwaregesteuert vier voneinander unabhängige analoge Ausgangsspannungen erzeugen, die, je nach Jumperstellung von JP8 und JP9, bipolar(±) oder unipolar (Bezugspunkt 0 V) sind. Mit dem 16-bit Datenwort ergibt sich so im 10 V-Bereich rechnerisch eine Auflösung von 150 µV. Die Ausgangsspannungsbereiche (2,5 V, 5 V und 10 V) werden für jeden Kanal separat über JP3 ... JP6 festgelegt (siehe Seite 9).

Die vier Ausgangssignale sowie die zugehörigen Sense-Eingänge sind über eine 37-polige Sub-D-Buchse von außen am PC-Slotblech zugängig. Außerdem sind zur Versorgung externer Schaltungen die GND und +5 Volt-Leitung auf der Sub-D Buchse geführt. Zusätzlich stehen an dieser Buchse zwei Digitalkanäle zur Verfügung, die per Software Schaltfunktionen auszulösen können. Einzelheiten zur Programmierung dieser Digitalausgänge finden Sie in den Erläuterung zur Adressierung (Seite 8) und im Testprogramm (Seite 12).

Herzstück der PCI-DAC-Schaltung ist das PCI-Interface ispLSI1032, das die Daten für die PCI-Schnittstelle des PCs aufbereitet und der PCI-Adreß-Decoder ispLSI1016, der das Management der Ausgaben übernimmt. Vom ispLSI1032 gelangen die Daten an die Digitaleingänge des schnellen präzisions digital zu analog-Konverters DAC7644. Dieser mit vier Analogausgängen ausgestattete DAC zeichnet sich neben besonderes niedrigem Rauschverhalten noch durch die Möglichkeit aus, einen programmierten Reset, entweder auf einen mittigen Eingangsspannungsbereich oder auf den Nullpunkt auszuführen. Im Schaltungsentwurf ist dem Rechnung getragen, indem über die JumperJP8 und JP9 je nach Anwendungsfall eine unipolare- oder bipolare Ausgabe (in den Grenzen der Vorgaben 2,5 V, 5 V und 10 V) initiiert werden kann. Die Steuerung des Konverterbausteins erfolgt mittels der Steuerleitungen RST SEL (rücksetzen Auswahl unipolar/bipolar), RST DAC (Rücksetzen DAC), LD DAC (Laden DAC), R/W DAC (Lesen/Schreiben DAC) und CS DAC (Chip Select DAC). Die vier internen Eingangsregister werden durch ein entsprechendes Bitmuster an den Pins A1 und A2 geladen und danach über die auch für jeden Kanal einmal vorhandenen DAC-Register und die vier Wandler-Einheiten geführt. Von dort wird das analoge DAC-Ausgangssignal gepuffert (mit einem Op-Amp) an den Ausgangspin des ICs geleitet.

Die für den Wandlerbaustein benötigten Referenzspannungen werden über die Low-Power Spannungsreferenz REF 1004 erzeugt. Sie wird zusätzlich über zwei präzisions-Op-Amps gepuffert und über den Jumper JP9 (Auswahl unipolar/bipolar) dem DAC zugeführt. Für eine später eventuell nötig werdende Neujustierung kann man einen Referenzspannungsabgleich mit den Mehrgangtrimmern POT1 auf –2,5 V (1MΩ) und POT6 auf +2,5 V (100 kΩ) vornehmen. **Achtung!** Die DAC-Karte wird mit kalibrierten Referenzspannungen, die eine Toleranz von 10 nV aufweisen, ausgeliefert und braucht deshalb nicht eingestellt zu werden.

Vom DAC-Baustein werden die analogen Spannungen über vier Op-Amps U3...U6 (OPA 27), die erstens zur Pufferung des Signals und zweitens dem Schutz des DAC-Bausteins dienen, zu den Ausgangsverstärker-Bausteinen PGA 205 (High Precision) bzw. PGA 206 (High Speed), die über

eine digital programmierbare Verstärkung von 1-2-4-8-fach verfügen, weitergeleitet. Ein Schutz des DACs ist deshalb sinnvoll, da bei einem Kurzschluß der Ausgänge an den Anschlußbuchsen, was eine Zerstörung der Ausgangsverstärker zur Folge hat, sonst auch der DAC Schaden nehmen würde. Die Verstärkungseinstellung wird mittels der Jumper JP3...JP6 für jeden Kanal getrennt vorgenommen (siehe Jumperdiagramm Seite 9). Eine Offseteinstellung dieser Verstärker ist durch POT2...POT5 (1MΩ) möglich. Beide Karten-Varianten werden allerdings ab Werk abgeglichen ausgeliefert. Bei der Kartenausführung mit dem PGA 205 kann ein Abgleich des Offsets ohnehin unterbleiben, da dieser Baustein von Hause aus extrem genau arbeitet. Die Eingänge FB (Feedback) des Verstärker ICs sind die Sense-Eingänge, die ebenfalls auf die 37-polige Buchsenleiste geführt sind.

Achtung!
Sollten die Sense-Eingänge nicht benötigt werden, so sind sie an der Buchsenleiste unbedingt mit dem Ausgangssignal zu verbinden, da sonst ein einwandfreier Betrieb nicht möglich ist (open loop).

Mit Spannung wird die DAC-Karte aus dem Netzteil des PCs versorgt. Da eine direkte Verwendung der von dort gelieferten Spannung für Meßwandler nicht möglich ist (zu hohe Rauschspannung usw.) müssen relativ aufwendige schaltungstechnische Maßnahmen ergriffen werden, um die Speisespannung zu „säubern“. Über die entsprechenden Stifte des PCI-Slots gelangen Vcc und GND an den DC/DC-Spannungsconverter UM 405, der daraus eine symmetrische Spannung von ±15 V generiert. Die Ausgangsspannung des Converters wird über ein Filter geführt und versorgt zum einen die Operations- und Ausgangs-Verstärker mit ±15 V und zum anderen werden über die nachgeschaltete Spannungsregler ICs (L 7805 und L 7905) die übrigen ICs mit ±5 V versorgt und die Referenzspannung über REF1004 von ±2,5 V gebildet.

Die Einstellung der Kartenadressierung erfolgt automatisch über Plug and Play (PNP). Eine weitergehende Bauteilinitialisierung ist nicht erforderlich. Der DAC-Baustein wird über die entsprechenden I/O-Port-Register direkt programmiert (siehe dazu die Programmbeispiele).

Zur Programmierung/Adressierung der Karte sind grundsätzlich folgende Bauteile-Offset-Adressen zu beachten:

I/O-Port-Adresse:

adr = 6500 Hex, bzw. 25856 Dez
(nur als Beispiel, da PCI die Basis-I/O-Adresse dynamisch vergibt)

Adreß-Offset (fett):

a.) Digitalausgänge Dig 1, Dig 2 (Hex Offset zur Basisadresse):

adr+0x**3C**,1	Funktion: D0 auf „1“ setzen
adr+0x**3C**,2	Funktion: D1 auf „1“ setzen
adr+0x**3C**,0	Funktion: D0 und D1 auf „0“ setzen

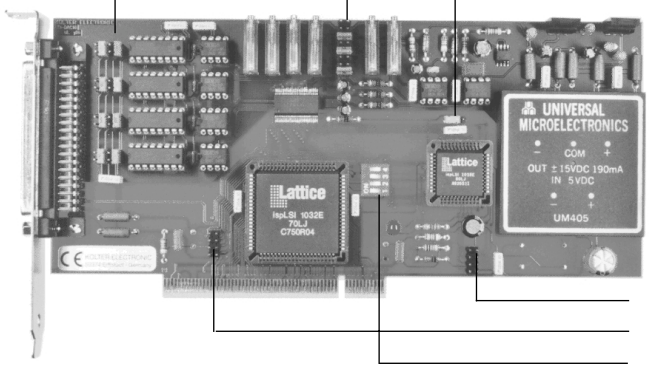
b.) DAC-Steuerung (Hex Offset zur Basisadresse):

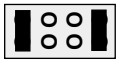
adr+0x**00**,A	Funktion: schreibe Wert „A“ (Word-Wert, 16 bit) ins DAC-Register 1 (Kanal 1).
adr+0x**04**,A	Funktion: schreibe Wert „A“ (Word-Wert, 16 bit) ins DAC-Register 2 (Kanal 2).
adr+0x**08**,A	Funktion: schreibe Wert „A“ (Word-Wert, 16 bit) ins DAC-Register 3 (Kanal 3).
adr+0x**0C**,A	Funktion: schreibe Wert „A“ (Word-Wert, 16 bit) ins DAC-Register 4 (Kanal 4).
adr+0x**38**,0	Übergabebefehl; die DAC-Latch-Register werden geladen. Erst jetzt beginnt die Wandlung.
adr+0x**10**,0	Resetbefehl; alle DAC-Register werden gelöscht.
	Der Maximalwert von A beträgt $FFFF_{Hex}$, der Minimalwert ist $0000_{Hex.}$

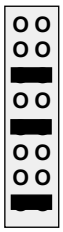
Andere Beispiele zur Kartenprogrammierung entnehmen Sie bitte dem Testprogramm oder den Texten auf der Diskette bzw. CD.

Jumpereinstellungen

Um die einzelnen Betriebs-Modi (Ausgangsspannungsbereich, unipolar/bipolar) auszuwählen werden Steckbrücken bzw. Jumper gesteckt.



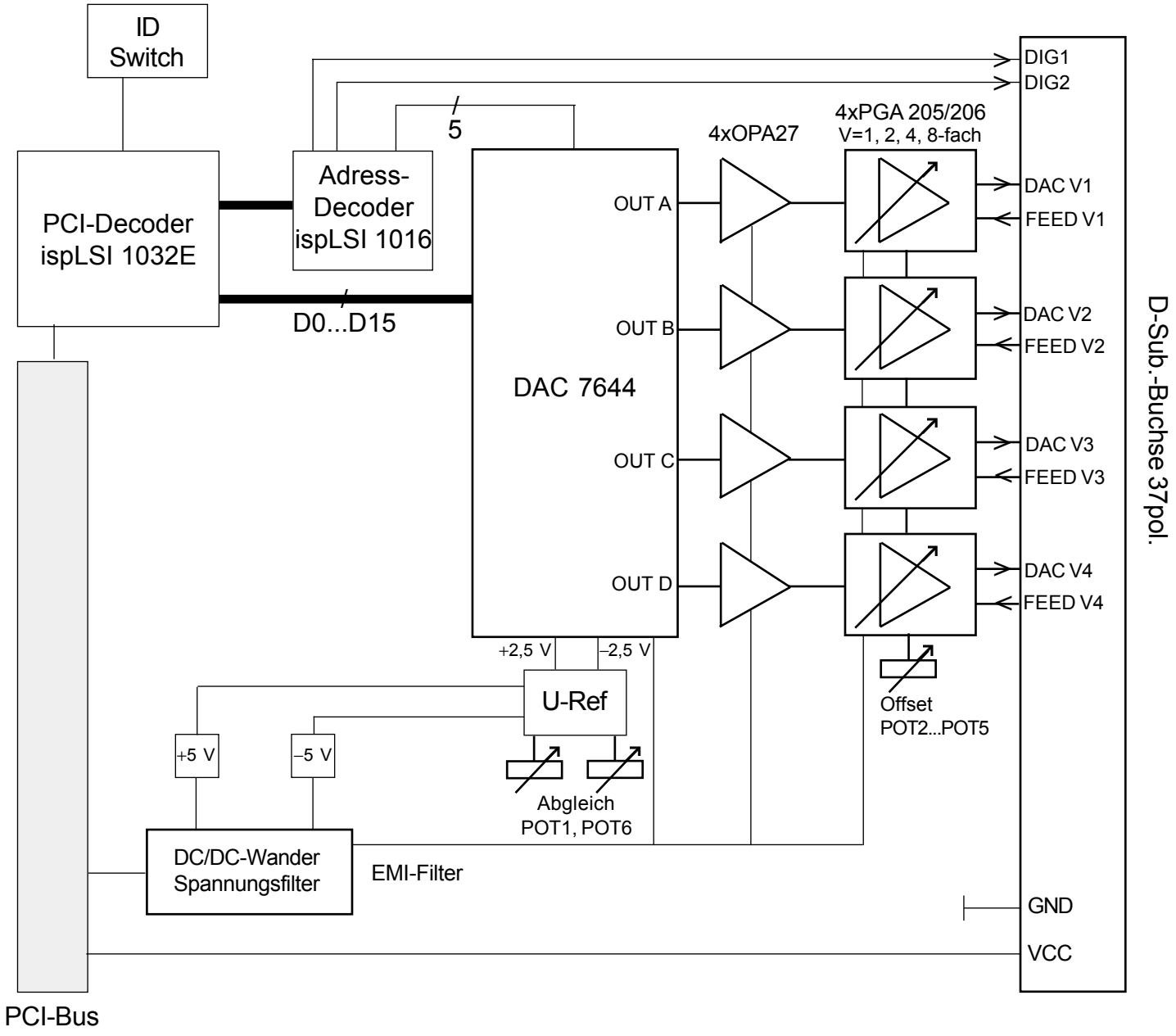
Ausgangsspannung	2,5 V	5 V	10 V	Bemerkung
				

Spannung	bipolar	unipolar	Bemerkung
			

			DAC-RESET Null/Mid-Range

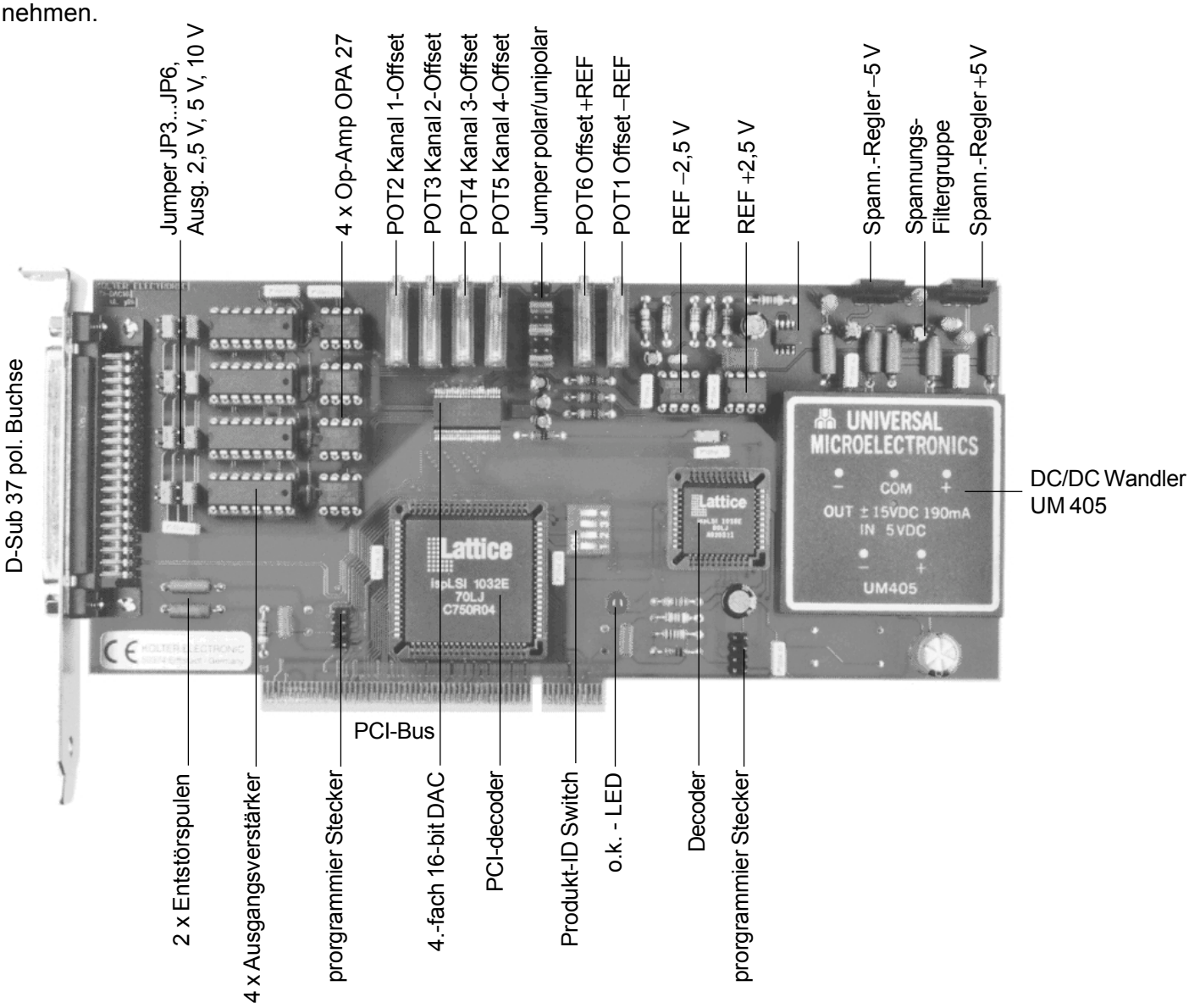
Blockschaltbild

Hier eine einfache Übersicht über die einzelnen Funktionsblöcke der PCI-DAC 416-Karte.



Kartenansicht und Bauteile

Die übliche Bestückung der Karte sowie dieBauteilepositionen können Sie dem folgenden Bild entnehmen.



Allgemeines

Die Einstellung des DIP-Schalters für die Produkt-ID sollte nie verändert werden. Der DIP-Schalter legt die Produkt-ID von 0x0015 Hex fest, damit das Rechner-BIOS eine entsprechende I/O-Adresse zuweisen kann und die Karte per Software identifiziert wird. Die LED zeigt an, ob die Karte richtig funktioniert bzw. das PLSI richtig geladen wurde – also ein quasi Selbsttest.

Technische Daten

D/A-Kanäle	4 Kanäle	
Polarität	unipolar / bipolar (Jumper)	
DAC-Auflösung	16 bit	
linearer Fehler	typ. ±3, max. ±4 beim DAC 7644E	
	typ. ±2, max. ±3 beim DAC 7644EB	
linearer Gleichlauf	typ. ±4 beim DAC 7644E	
	typ. ±2 beim DAC 7644EB	
linearer Differenzfehler	typ. ±2, max. ±3 beim DAC 7644E	
	typ. ±1, max. ±2 beim DAC 7644EB	
Ausgangsspannungen	2,5 V, 5 V, 10 V, unipolar / bipolar	
Ausgangsstrom	max. +23 mA, –17 mA	
D/A Wandelgeschwindigkeit	typ. 8 µs, max. 25 µs	
Verzögerungszeit	PGA 205, max. 25 µs	
	PGA 206, max. 2 µs	
Slew Rate	PGA 205, max. 0,7 V/µs	
	PGA 206, max. 25 V/µs	
Last Ausgangskapazität	max. 1 nF	
Digital-Output	2 Kanäle, TTL (0/5 Volt), Pegel siehe Datenblatt zum ispLSI1016E	
Betriebsarten	TTL-OUT, D/A	port-polling, programmgesteuert, byte-weise
PCI-Decoder	1	ispLSI 1032E, Lattice IC
Vendor-ID	KOLTER	0x1001
Product-ID	KOLTER	0x0015
Adressierung	var.	PCI PNP
Bus	32/64 bit	32bit PCI-Bus, gem Spec. 2.1 - 2.3
Anschlüsse	37 pol.	Sub-D Buchse
	2 x 8 pol. IDC	Programmier-Stecker (intern)
Abmessungen	100 x 216 mm	Kartenmaße (ohne Halteblech)
Temperaturbereich	0...70 °C	typ. Betriebszustand / Dauerbetrieb
Lagertemperatur	–20...+85 °C	

Zulassungen und Eigenschaften

- EMV (CE) konform
- UL Platine, yellow-card
- Year 2000 compliance
- Schwingprüfung, gerüttelt nach DIN 61010
- Einzeltest, 100 % geprüft
- RoHS-konform auf Anfrage

Testprogramm in MS_VC

```
// ======================================================================
// DAC416.CPP
// Programm für direkte I/O-Zugriffe auf Hardware unter Windows 95/98
// Geschrieben für PCI-DAC 16 Messkarte
// Copyright by KOLTER ELECTRONIC 1999
// geprüft mit C-Compiler MSVC++6.0 am 30.08.1999 / ko.
// ======================================================================

#include "windows.h"    // required for all Windows applications
#include <time.h>
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

unsigned short A;
unsigned short adr;

main()
{
// I/O-Port-Adresse vergeben. Hier 6500 HEX
adr = 25856;

printf ("Teste TTL Ausgang... Taste fuer weiter \n");
do {
// TTL-OUT Hardware_Port Ausgabe schreiben: D0 und D1
// _outp(adr+0,A); _outp... syntax  ist hier zwingend bei C++ 6.0
     _outp (adr+60,1);
Sleep(1000);
     _outp (adr+60,2);
Sleep(1000);
     _outp (adr+60,0);
Sleep(1000);
} while (!_kbhit());    // wiederholen, bis Taste gedrückt

// ..........................................................

printf ("DAC TEST >> Taste druecken \n");
getch();
// DAC Test, nur als 16bit-Zugriff möglich !!!
printf ("Ausgang 1 = 0xFFFF \n");
A=0xFFFF;
_outpw(adr+0x00,A);     // schreibe word nach Kanal 1 (DAC-Register)
_outpw(adr+0x04,A);     // schreibe word nach Kanal 2 (DAC-Register)
_outpw(adr+0x08,A);     // schreibe word nach Kanal 3 (DAC-Register)
_outpw(adr+0x0C,A);     // schreibe word nach Kanal 4 (DAC-Register)
_outp (adr+56,0);       // Uebergabe an DAC-Output, latch all register

getch();
printf ("Ausgang 1 = 0xC000 \n");
A=0xC000;
_outpw(adr+0x00,A);     // schreibe word nach Kanal 1 (DAC-Register)
_outpw(adr+0x04,A);     // schreibe word nach Kanal 2 (DAC-Register)
```

```
_outpw(adr+0x08,A);     // schreibe word nach Kanal 3 (DAC-Register)
_outpw(adr+0x0C,A);     // schreibe word nach Kanal 4 (DAC-Register)
_outp (adr+56,0);       // Uebergabe an DAC-Output, latch all register

getch();
printf ("Ausgang 1 = 0x8000 \n");
A=0x8000;
_outpw(adr+0x00,A);     // schreibe word nach Kanal 1 (DAC-Register)
_outpw(adr+0x04,A);     // schreibe word nach Kanal 2 (DAC-Register)
_outpw(adr+0x08,A);     // schreibe word nach Kanal 3 (DAC-Register)
_outpw(adr+0x0C,A);     // schreibe word nach Kanal 4 (DAC-Register)
_outp (adr+56,0);       // Uebergabe an DAC-Output, latch all register

getch();
printf ("Ausgang 1 = 0x4000 \n");
A=0x4000;
_outpw(adr+0x00,A);     // schreibe word nach Kanal 1 (DAC-Register)
_outpw(adr+0x04,A);     // schreibe word nach Kanal 2 (DAC-Register)
_outpw(adr+0x08,A);     // schreibe word nach Kanal 3 (DAC-Register)
_outpw(adr+0x0C,A);     // schreibe word nach Kanal 4 (DAC-Register)
_outp (adr+56,0);       // Uebergabe an DAC-Output, latch all register

getch();
printf ("Ausgang 1 = 0x0000 \n");
A=0x0000;
_outpw(adr+0x00,A);     // schreibe word nach Kanal 1 (DAC-Register)
_outpw(adr+0x04,A);     // schreibe word nach Kanal 2 (DAC-Register)
_outpw(adr+0x08,A);     // schreibe word nach Kanal 3 (DAC-Register)
_outpw(adr+0x0C,A);     // schreibe word nach Kanal 4 (DAC-Register)
_outp (adr+56,0);       // Uebergabe an DAC-Output, latch all register

getch();
printf ("Clear and Reset all DAC register \n");
_outp (adr+16,0);       // RESET all DAC register auf Adresse 0x10

 return 0;
}
```

Einbindung in eigene Programme

Programmierbeispiel zur Karteneinbindung

Um die entsprechende Vergabe der I/O-Adresse braucht sich der Anwender also nicht zu kümmern. Wohl aber um die Einbindung der Adresse in sein Programm. Damit das eingene Programm weiß, wo sich die neue Karte im Adressraum befindet, müssen Parameter übergeben werden. Das Programm PCIGETIO.C (im Ordner UTILS auf der Diskette) ist in der Lage, dem nachzukommen. Bei Aufruf liefert es einfach die I/O-Adresse zurück.

Das Programm PCIGETIO.C

```
/*   Beispiele:
Die folgende Funktion zeigt eine Möglichkeit auf, wie die I/O-Adresse einer
KOLTER-Karte ermittelt werden kann. Als übergabewert der Funktion ist die Device
ID der entsprechenden Karte anzugeben (Beispiel: PCI16/16 = 0x0010). Der Rückgabewert
der Funktion beinhaltet die I/O-Adresse der Karte oder 0, wenn keine Karte
gefunden wurde bzw. keine Plug'n Play-Adresse vergeben wurde.
*/


#define VENDOR_KOLTER 0x1001

extern unsigned long far pascal INT_1A( unsigned long reax,
                        unsigned long rebx,
                        unsigned long recx,
                        unsigned long redi);

unsigned long int PciGetIO(Device) {

unsigned long reax;
unsigned long rebx;
unsigned long recx;
unsigned long redi;
unsigned long ret_ecx;

unsigned int slot_no;
unsigned long io_adr;
unsigned long ven_dev;
unsigned long K_ven_dev;

io_adr = 0;

/* Device und VendorID werden verknüpft */
K_ven_dev = ((long) Device << 16) | VENDOR_KOLTER;

for (slot_no = 0;(slot_no <= 0x00F8) && (io_adr == 0);slot_no += 8) {
     ven_dev=INT_1A(0xb10aL,(long) slot_no,0L,0L);

         if (K_ven_dev == ven_dev)               /* Karte von KOLTER wurde gefunden*/
            io_adr=INT_1A(0xb10aL,(long) slot_no,0L,0x0015L);
}

io_adr &= 0xFFFFFFF0;                                  /* I/O-Adresse wird maskiert */
return(io_adr);                                        /* Rückgabe der I/O-Adresse  */

}
```

In eigener Software kann diese Routine wie im folgenden Beispiel eingesetzt werden:

Um das Programm PCIGETIO einzubinden ist es allerdings nötig, vorher ein Make-File zu bilden. MS VC++ erledigt das für Sie.

Das Quick-C Programm:

```
#include <stdio.h>

void main()
{

unsigned long ret;    // Variable definieren

ret=PciGetIO(0x15);   // in der long-Variable "ret" steht die I/O-Basis-Adresse

printf("%lx\n",ret);  // Adresse nur auf Monitor anzeigen

}
```

Das Programm PCIUTILS liefert über die Produkt ID und Vendor ID die Adresse zurück, die die Karte belegt. Das Programm befindet sich ebenfalls auf der beigelgten Diskette im Ordner UTILS.

PCIUTIL.C

```
extern unsigned long far pascal INT_1A( unsigned long reax,
                                        unsigned long rebx,
                                        unsigned long recx,
                                        unsigned long redi);


unsigned long pci_rd_cfg (unsigned int einheit, unsigned int adresse)
{
unsigned long inhalt;

  inhalt=INT_1A(0xb10aL,(long) einheit,0L , (long) adresse);
  return (inhalt);
}

void pci_wr_cfg(unsigned int einheit, unsigned int adresse, unsigned long
wert)
{
  INT_1A(0xb10dL, (long) einheit, (long) wert, (long) adresse);
}
```

Der PCI-Bus

Seit vor Jahren die ersten „Peripheral Component Interconnect“- Spezifikationen (PCI) herauskamen, haben sich viele Computer- und vor allem Meßtechnik-Zusatzkarten-Hersteller diesem schnellen 32-Bit-Standard zugewandt. Sein Vorteil: Im Vergleich mit dem „alten“ ISA-Bus lassen sich deutlich höhere Bus-Transferraten erzielen, was in der modernen Multimediatechnik natürlich einen entscheidenden Pluspunkt bedeutet. Und obwohl es u.a. schon viele interessante Meßtechnik-Boards für PCI gibt, so steht der richtige Boom diesem Marktsegment erst noch bevor. Mittlerweile ist der PCI-Bus in jedem neuen PC als Haupt-Bus-Architektur zu finden. Der wesentliche Vorteil des PCI-Bus ist die um den Faktor 20 bis 100 schnellere Datenübertragung im Vergleich mit dem ISA-Bus. Hinzu kommt die 32-Bit-Architektur, was der heutigen Windows-Betriebssystem- und -Meßtechnik-Software entgegenkommt und überdies eine Leistungssteigerung darstellt, die jedoch für den größten Anteil an Messapplikationen kaum benötigt wird.

Der ISA-Bus hat hervorragende Dienste als Universal-PC-Bus geleistet, aber für die modernen CPUs, die schnellen Netzwerke, Grafikkarten, SCSI-Platten und Speichermodule sind 2,5 MByte pro Sekunde einfach zu langsam. PCI verspricht deutlich über 100 MByte/s, was natürlich auch für meßtechnische Anwendungen manchmal von entscheidender Bedeutung sein kann. „Plug and play“ vereinfacht die Installation und Inbetriebnahme. Der meist auf den PCI-tauglichen Meßdatenerfassungs- oder Multifunktions-Boards vorhandene digitale Signalprozessor schafft weitere Leistungsreserven und entlastet den Host-PC. Hinzu kommt, daß schnelle A/D-Wandler nun auch Abtastraten erlauben, die um den Faktor 3...10 höher liegen als noch vor einigen Jahren und durchaus schon bei 100 MS/s (Megasamples/Sekunde), was einer Abtastfrequenz von100 MHz entspricht, angelangt sind.

Grundsätzlich erlaubt der Hardware- und Protokoll-Aufbau des PCI-Bus die Übertragung von 32-Bit-Datenwörtern mit einer Taktfrequenz von 33/66 MHz. Dies führt zu einem optimalen Datendurchsatz nicht nur für meßtechnische, sondern beispielsweise auch für Video- oder High-speed SCSI-Anwendungen. Ein weiterer Vorteil ist das „Plug and play“-Konzept, bei dem man die installierte PCI-Zusatzkarte nicht mehr umständlich mit DIP-Schaltern bezüglich Adreßraum und Interrupts konfigurieren muß. Vielmehr wird beim Booten des Rechners das PCI-BIOS aktiviert, das alle im Rechner installierten Karten abfragt und ihre Adreßbelegungen sowie Speicherplatz-Bedürfnisse festlegt. Ein eigener PCI-Chipsatz auf dem Motherboard erledigt diese Aufgaben. Auch das Bus-Mastering-Konzept erweist sich bei der schnellen Datenübertragung als nützlich: Eine Karte kann – entweder dauernd oder nur für einen bestimmten Zeitraum – die Kontrolle über den Datentransfer übernehmen und zielgerichtet die Weiterleitung der Daten steuern. Dies ist besonders für meßtechnische Aufgaben sinnvoll, da dadurch die CPU (die ihrerseits natürlich auch als Bus-Master fungieren kann) entlastet wird.

Da es sich bei den Datentransfers sowohl im meßtechnischen Bereich wie auch bei anderen Applikationen hauptsächlich um Übertragungen in bestimmte Speicherbereiche handelt, sind zwei Betriebsarten von besonderer Bedeutung: der „Memory Mode“ und der „Real Time Mode“ (hier nur für meßtechnische Applikationen betrachtet). Im Memory-Modus digitalisieren die A/D-Wandler der Meßdatenerfassungskarte die Analogwerte und schreiben die Digitalworte in ein eigenes RAM, das sich auch auf dieser Karte befindet. Dieser Speicher kann nun von der CPU (oder einer anderen PCI-Karte) abgefragt werden – das Ganze allerdings nur nach Abschluß der A/D-Umsetzung. Mit diesem Verfahren kann man die höchsten Speicher-Transferraten erreichen, die durchaus bei über 500 MBit/s liegen können. Selbstverständlich ist die Länge der aufzuzeichnenden Datensätze abhängig von dem zur Verfügung stehenden On-Board-Speicher auf der PCI-Karte. Für viele Applikationen ist diese Art der Datenerfassung weitaus übertrieben. Die Regel in der Meßtechnik zeigt jedoch, daß Datenraten von unter 1 MHz völlig ausreichend sind.

Im „Real Time“-Modus schreiben die A/D-Wandler ihre digitalisierten Meßdaten nicht auf ein On-Board-Memory, sondern über den PCI-Bus entweder in den Motherboard-Speicher oder in das Memory einer anderen PCI-Karte im Rechner. Da hier die Übertragung auf dem PCI-Bus korreliert werden muß, lassen sich keine so hohen Transferraten wie im Memory-Modus erreichen, dafür nutzt man aber die Flexibilität der beliebigen Übertragung in alle dem Rechner zur Verfügung stehenden Speicherbereiche – auch Festplatten. Immerhin erreicht man bei einer 8-Bit-A/D-Wandlung in diesem Modus noch eine effektive PCI-Bus-Übertragungsrate von 100 MS/s. Übrigens: Den eben beschriebenen Real-Time-Modus sollte man nur verwenden, wenn die Abtastrate kleiner ist als die maximale PCI-Bus-Übertragungskapazität von 100 MS/s für 8-Bit-Abtast-Auflösung. Wenn sehr lange Datenströme aufzuzeichnen sind, ist der Real-Time-Modus im Zusammenspiel mit einer schnellen SCSI-Festplatte die einzige Alternative. Will man mehrere PCI-Datenerfassungskarten in einem Rechner betreiben, ist zu berücksichtigen, daß sich die maximale Bus-Bandbreite dann auf die einzelnen Karten aufteilt, was bedeutet, daß beispielsweise beim Einsatz von zwei PCI-Erfassungskarten in einem Rechner für jedes der beiden Boards nur die Hälfte der Übertragungskapazität zur Verfügung steht.

Bustakt und Datenbreite

Gegenüber den herkömmlichen Bussen weist der PCI-Bus eine ganze Reihe von Vorteilen auf. Wesentlich ist, daß er mit Busfrequenzen von 33, 66 und > 66 MHz sowie einer Datenbreite von 32 Bit arbeitet und somit schneller Daten zu den I/O-Karten übertragen kann. Besonders vorteilhaft ist dies z. B. bei den neuen A/D und D/A- Karten, obwohl diese oft nur im kHz-Bereich arbeiten.

Automatische Adress-Vergabe

Alle PCI-Karten sind mit einem „PLUG and PLAY" Mechanismus versehen, was eine Einstellung der I/O-Adresse per Hand (Jumper) erübrigt und mögliche Adreßkonflikte vermeidet. Beim Hochfahren des Rechners vergibt das PCI-BIOS allen eingesteckten Karten automatisch eine freie I/O-Adresse. Durch eine im Lieferumfang enthaltene Software (die mit jeder PCI-Karte ausgeliefert wird) kann die automatisch vergebene I/O-Adresse erfragt werden bzw. verändert werden. Hierzu führen wir einige Tools, die Sie auf den Beispieldisketten finden werden.

Mitgelieferte Treiber/Utilities

- Abfrage aller eingesteckten KOLTER-PCI-Karten
- Abfrage der I/O-Adressen von KOLTER-PCI-Karten
- Windows NT Treiber
- Windows Treiber
- Utils unter DOS
- Linux-Beispiel-source

Software-Identifikation

Alle Karten können softwaremäßig identifiziert werden, da diese neben einer fest eingestellten Hersteller-ID (für KOLTER = 0x1001) auch über eine Produkt-ID verfügen.

Funktionsweise der Adressvergabe

Alle PCI-Karten verfügen über einen 64 Byte großen Konfigurationsheader, in dem eine Hersteller-ID, Produkt-ID, sowie Informationen enthalten sind, ob die PCI-Karte Memory und/oder I/O-Bereiche benötigt- und wie groß dieser Bereich sein sollte. Beim Hochfahren (booten) des Rechners, checkt das

BIOS des Motherboards alle PCI-Karten durch und verteilt systematisch freie I/O- bzw. Memory-Adressen auf den Karten. Die Adresslage jeder im PC befindlichen Karte wird also vom Motherboard bzw. BIOS vorbestimmt und nicht, wie sonst üblich, per Jumper auf der Steckkarte zwangsadressiert. Die zugeteilten Adressen werden in den dafür vorgesehenen Registern auf den PCI-Karten selber redundant zwischengespeichert. Anschließend werden die PCI-Karten entsprechend aktiviert und dekodieren sich auf den eingestellten bzw. zugewiesenen Adressraum.

Wo ist die Kartenadresse ?

Jede PCI-Karte kann durch die Hersteller-ID und Produkt-ID identifiziert werden. Unseren Karten liegt dazu ein spezielles Programm bei (PCI_INF.EXE), das die Informationen sichtbar macht. Das Programm gibt folgende Informationen auf dem Monitor aus:

Nr.	**Nummer des PCI-Slot (nicht sichtbar)**
Hersteller-ID	**Name des Herstellers**
Produkt-ID	**Name bzw. Nummer der Karte**
I/O-Adresse	**Hardware-Adresse, unter der die Karte angesprochen wird**
MEM-Adresse	**Basisspeicheradresse (nicht sichtbar)**
Produkt	**Name der PCI-Karte**

Allgemeine Adressierung unter PCI

In selbstgeschriebener Software kann beispielsweise auch die folgende Routine eingesetzt werden. Es handelt sich um einen allgemeinen Beispiel-Source zur Ansteuerung von Relais oder Optokopplern oder TTL-Signalen. Der Source-Code ist auf keine besondere Karte zugeschnitten und dient lediglich zur Erläuterung.

Um das Programm einzubinden ist vorher ein Make-File zu bilden. MS VC++ erledigt das für Sie. PCIGETIO.EXE muss im gleichen Verzeichnis wie TEST.EXE sein.

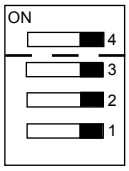
Das C-Programm, z. B. mit MS VC++ unter Windows 95/98:

```
// Test-Programm

#include <stdio.h>

void main()
{
unsigned long ret;
unsigned int port;

int i;
long int j;
unsigned int wert;

port=ret=PciGetIO(0x015);

printf("Die Port-Adresse ist:%lx\n",ret);   // Adresse holen
if(port==0) exit(0);                        // abbruch wenn nicht o.k.

while(!kbhit()) {                           // wiederholen bis Taste
for(i=0;i<16;++i) {
   wert=1<<i;

   outp(port,wert&0xff);                    // auf Port schreiben
   outp(port+4,(wert>>8)&0xff);             // dito

   for(j=0;j!=200000;++j);                  // oder Sleep(500);
   printf("%x %x \n",inp(port),inp(port+4)); // auf Monitor mitschreiben
}
}
}
```

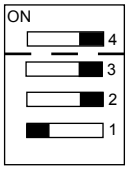
Vendor- und Produkt ID-Informationen

Zur VENDOR-ID 0x1001, die exklusiv der Firma KOLTER ELECTRONIC zugeordnet ist, verwenden wir für unsere PCI-Karten folgende Produkt-IDs:

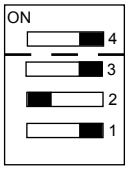
Produkt-ID: $0x0010_{Hex}$
PCI-1616 TTL I/O
und für zukünftige TTL-I/O-Karten



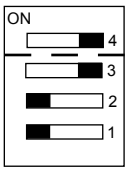
Produkt-ID: $0x0011_{Hex}$
OPTO-PCI /N
OPTO-PCI /P



Produkt-ID: $0x0012_{Hex}$
PCI-ADxx
PCI-ADxx-DAC4
PCI-DAC4
PCI-ADTERM

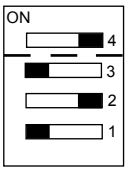


Produkt-ID: $0x0013_{Hex}$
PCI-OPTOREL
PCI-Relais

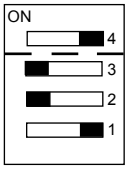


Produkt-ID: $0x0014_{Hex}$
PCI-Counter-1, 3 x 25 bit U/D
PCI-Timer

Produkt-ID: $0x0015_{Hex}$
PCI-DAC 416



Produkt-ID: $0x0016_{Hex}$
PCI-MFB

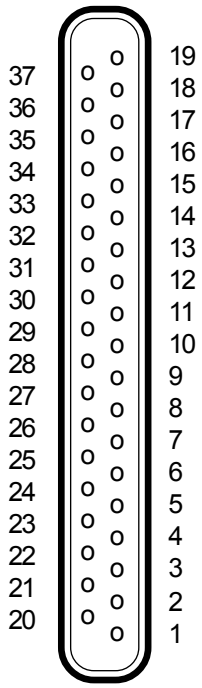


Produkt-ID: $0x0017_{Hex}$
PCI-PROTO3

Der für die Eingabe der Produkt-ID nicht benötigte vierte Schalter dient zur Umschaltung auf die gewünschte PCI-Spezifikation:
DIL-Switch 4: ON = PCI 2.2 / 2.3 Spezifikation
DIL-Switch 4: OFF = PCI 2.1 Spezifikation

Steckerbelegung

Ansicht auf den Stecker (37pol. Sub-D Buchse an der Karte)
Relais-Kontakte



1	GND	20	Vcc
2	DIG1	21	DIG2
3	n.c.	22	n.c.
4	n.c.	23	n.c.
5	n.c.	24	n.c.
6	n.c.	25	n.c.
7	n.c.	26	n.c.
8	n.c.	27	n.c.
9	n.c.	28	n.c.
10	n.c.	29	AGND
11	AGND	30	FEED V4
12	DAC V4	31	AGND
13	AGND	32	FEED V3
14	DAC V3	33	AGND
15	AGND	34	FEED V2
16	DAC V2	35	AGND
17	AGND	36	FEED V1
18	DAC V1	37	AGND
19	AGND		

GND	= digital Masse
AGND	= analog Masse
FEED Vx	= analog Eingang x = 1...4
DAC Vx	= analog Ausgang x = 1...4
DIGx	= digital Ausgang (0/5 V) x = 1...2

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
Geschäftsleitung	95 37 32

Internet

E-Mail - Service	service@pci-card.com
E-Mail - Technik	technik@pci-card.com
Haupt-Domains	http//www.pci-card.com http//www.kolter.de

CE

EMV-Konformität:

Die EMV-Konformität gilt für industielle Einrichtungen bzw. ortsfeste Anlagen.
Der Einsatz im priv. Haushalt ist auf Grund der Prüfungsvorschriften untersagt.

Die elektromagnetische Verträglichkeit wurde nach 2004/108/EG
(vormals 89/336/EWG) nachgewiesen.

Folgende Fachgrundnormen wurden bei der EMV-Prüfung angewandt:

- DIN EN 61326:1997 + A1:1998 + A2:2001 (Test, Measurement, Control and Laboratory Equipment)
- DIN EN 50 081-2 (EMV Störaussendung - Industrie)
- DIN EN 50 082-2 (EMV Störfestigkeit - Industrie)

Die komplette EG-Konformitätserklärung können Sie auch unter folgender
URL als PDF-Dokument herunterladen: http://www.pci-card.com/ce.pdf

Diese Erklärung bescheinigt die Übereinstimmung mit den genannten Richtlinien, ist jedoch keine Zusicherung von Eigenschaften im Sinne des Produkthaftungsgesetzes. Die Sicherheitshinweise auf unserer Webseite, sowie in der mitgelieferten Produktinformation sind zu beachten. Weitere Informationen unter: http://www.pci-card.com/faq015.html