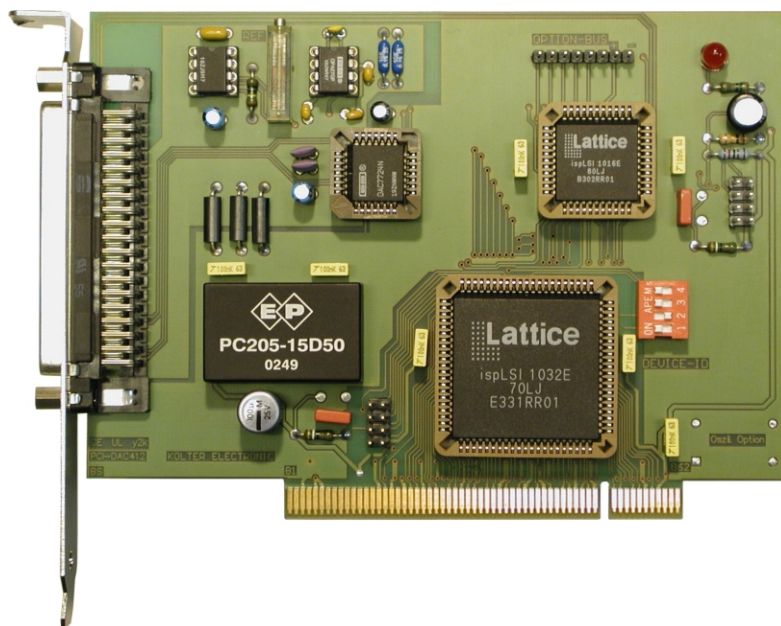


PCI - DAC 412

4 Kanal D/A-Converter 12 bit

Universelle DAC-Karte
mit Spannungsausgang

- | | |
|--------------------------|--|
| PCI - DAC 412-N | (Nr. 990 360 Standard Version, ± 10 Volt) |
| PCI - DAC 412-NB | (Nr. 990 360-B Erhöhte Genauigkeit, ± 10 Volt) |
| PCI - DAC 412-N5 | (Nr. 990 370 Spezial Version, ± 5 Volt) |
| PCI - DAC 412-NB5 | (Nr. 990 370-B Erhöhte Genauigkeit, ± 5 Volt) |



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
Einbau in den PC	5
Allgemeines zu I/O-Karten	6
Funktionsweise der Karte	7
Programmierung / Register	8
Kartenansicht und Bauteile	9
Technische Daten	10
Testprogramm in MS_VC	11
Einbindung in eigene Programme	13
Allgemeine Adressierung unter PCI	15
Vendor und Produkt ID-Informationen	16
Andere Vendor-IDs	17
Steckerbelegungen	18
Anschrift und Rufnummernverzeichnis	19

Sehr geehrter Kunde,

wir bedanken uns für den Kauf der PCI-DAC-Karte. Mit diesem 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.

Der Einbau in den PC

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

Bitte beachten Sie:

Statische Aufladung kann Ihren Computer und die Karte zerstören!

Entladen Sie sich daher vor dem Weiterarbeiten, indem Sie eine Wasserleitung, ein Heizungsrohr oder ein anderes Metallteil mit Erdverbindung berühren.

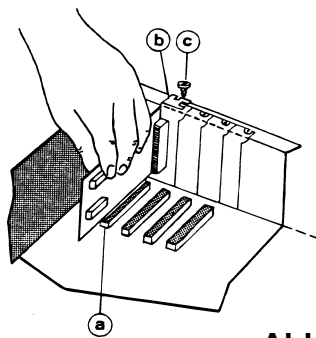


Abb. 1

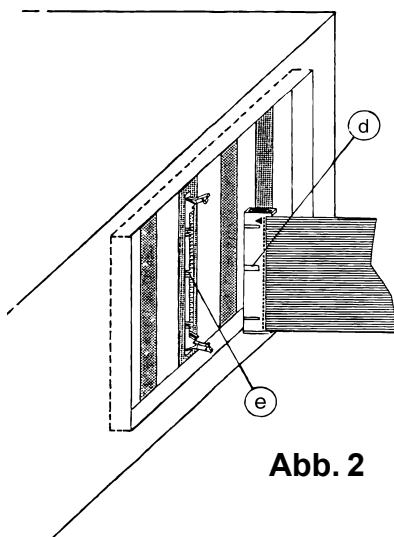


Abb. 2

2. Öffnen Sie den PC. Im allgemeinen müssen dazu auf der Rückseite des Gerätes vier Sicherungsschrauben mit einem Kreuzschlitzschraubendreher gelöst werden. Anschließend können Sie das Gehäuse nach vorne hin wegziehen. Eventuell müssen Sie einige hindernde 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.

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 !!!

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 Ampère 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 9x, Linux und NT ist der Anwender direkt in der Lage, seine Aufgaben schnell und professionell zu lösen.

Funktionsweise der Karte

Die PCI-DAC412 Karte ist für zwei verschiedene Grundbestückungsarten bei nur einem Kartendesign ausgelegt. Dabei kann durch Verwendung einer präziseren Referenzspannungsquelle (REF102BM) sowie einer höheren Garde beim D/A-Wandler, zwischen normaler und erhöhter Genauigkeit gewählt werden. Es besteht (gegen Aufpreis) auf Kundenwunsch die Möglichkeit, die Präzision der PCI-Karte durch eine Sonderbestückung mit einem ausgesuchten D/A-Konverterbaustein zu verdoppeln. Dabei wird der lineare Fehler von ± 2 LSB auf ± 1 LSB verringert und der lineare Gleichlauf von ± 2 LSB auf ± 1 LSB.

Mit einer 5 Volt Spannung wird die DAC-Karte aus dem Netzteil des PCs versorgt. Da eine direkte Verwendung der von dort gelieferten Spannungen für Meßwandler nicht möglich ist (zu hohe Rauschspannung, Ripple 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-Spannungswandler PC205, der daraus eine symmetrische Spannung von ± 15 V generiert. Die Ausgangsspannung des DC/DC-Wandlers wird über Filter-Bauelemente geführt und versorgt zum einen die Operations- und Ausgangs-Verstärker mit ± 15 V und zum anderen die Referenzspannung.

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).

Programmierung

Die PCI-DAC412 ist eine 4-Kanal 12-bit D/A-Wandlerkarte, die über 16-bit-Werte programmiert wird. Somit ist sie softwarekompatibel zur PCI-DAC416 (im ± 10 Volt Bereich). Eine hochwertige Referenzspannungsquelle von $\pm 10,000$ Volt ist die Basis für präzise Ausgangsspannungen. Der Burr-Brown® D/A-Wandler ist ein paralleler 12-bit digital-analog Wandler. Jeder Kanal wird mit nur einem 16-bit Output-Befehl (word mit outw) programmiert. Für die Programmierung stehen insgesamt vier 12-bit Übergaberegister, ein Latch-DAC Register und ein Reset-Register (clear) zur Verfügung. Die letzten bzw. ungenutzten 4-bit des 16-bit Datums werden ignoriert. Damit ist die Karte u.a sofort für ProfiLab-Expert 3.0 verwendbar. Im Einschaltmoment (Power-On), oder während der PC bootet, verhindert eine eigene Auto-Reset-Schaltung ungewollte Spannungszustände an den D/A-Ausgängen.

Anschlüsse

Die vier Spannungsausgänge stehen ungebuffert am Steckverbinder mit jeweils bis zu ± 5 mA Laststrom zur Verfügung. Auf eine aufwendige Feedback-Schaltung (sense) und weitere OP-AMPs bzw. PGAs wurde aus Kostengründen verzichtet. Neben den vier Spannungsausgängen verfügt die Karte noch über zwei programmierbare TTL-Ausgänge. Die Standardversion verfügt über ± 10 Volt Ausgänge, eine Spezialversion (Artikelnummer 990 370) ist mit ± 5 Volt lieferbar. Alle Anschlüsse sind über eine 37-polige Sub-D-Buchse am PC-Slotblech zugänglich. Weiterhin sind zur Versorgung externer, digitaler Schaltungen die GND und +5 Volt-Leitung vom PC-Netzteil an die Buchse geführt.

Programmierung / Register

Die PCI-DAC412 ist zur PCI-DAC416 registerkompatibel. Daher sind keine weiteren Programmänderungen nötig. Die fehlenden, niederwertigen 4-bit aus dem 16-bit Datum sind nach MSB geschiftet und ergeben damit den gleichen Spannungswert an den D/A-Ausgängen. Zur Programmierung und Adressierung der Karte sind grundsätzlich folgende Offset-Adressen des DAC-Wandlers und weitere I/O-Ports 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):

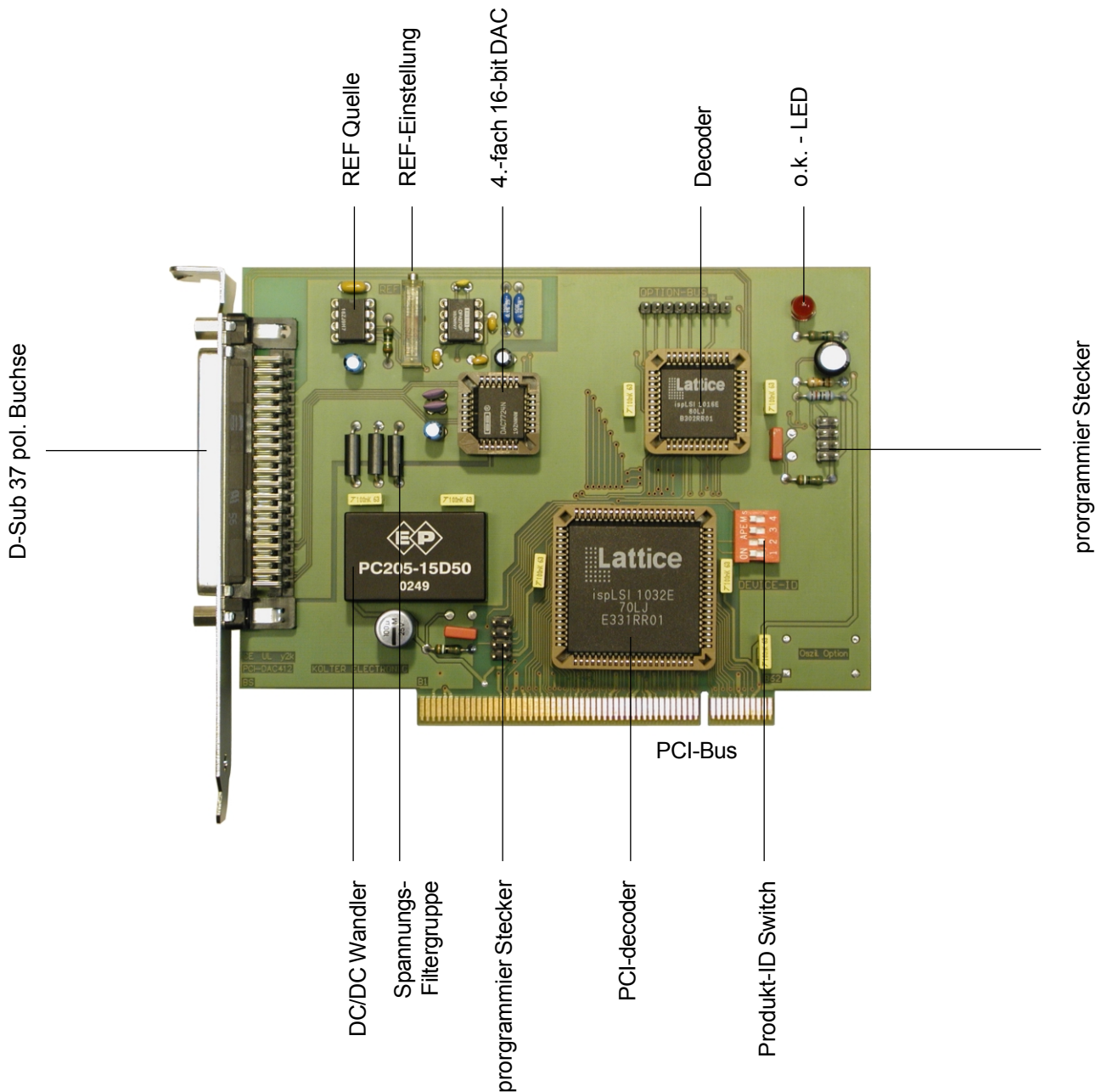
adr+0x 00 ,A	Funktion: schreibe Wert „A“ (Word-Wert, 16 bit) ins DAC-Register 1 (Kanal 1).
adr+0x 04 ,A	Funktion: schreibe Wert „A“ (Word-Wert, 16 bit) ins DAC-Register 2 (Kanal 2).
adr+0x 08 ,A	Funktion: schreibe Wert „A“ (Word-Wert, 16 bit) ins DAC-Register 3 (Kanal 3).
adr+0x 0C ,A	Funktion: schreibe 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.

Kartenansicht und Bauteile

Die übliche Bestückung der PCI-DAC412 Karte sowie die Bauteilepositionen 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

Ausgangsspannungen	± 10 Volt unipolar, Sonderbestückung ± 5 Volt
D/A-Kanäle	vier Kanäle, 12 bit unipolar
Crosstalk, Kanal-zu-Kanal	0,25 LSB (full scale step)
Gesamtabweichung	typ. 5 mV
Ausgangsstrom	max. ± 5 mA, siehe Datenblatt
Kurzschlußstrom	20 mA, Dauer unbekannt
D/A Wandelgeschwindigkeit	typ. 8 ms, max. 10 μ s ($\pm 0,012\%$ 20V)
Last-Kapazität	max. 500 pF
Ausgangsrauschen	typ. ± 5 mV bei < 20 MHz
linearer Fehler	max. ± 2 LSB mit DAC7724N max. ± 1 LSB mit DAC7724NB
linearer Gleichlauf	max. ± 2 LSB mit DAC7724N max. ± 1 LSB mit DAC7724NB
linearer Differenzfehler	typ. ± 1 LSB bei beiden Typen
Digital-Output Betriebsarten	2-fach, TTL-Pegel (0/5 Volt) I/O Port-R/W, programmgesteuert TTL-OUT, D/A port-polling, byte-weise
PCI-Decoder	1x ispLSI 1032E, Lattice IC
Vendor-ID	KOLTER 0x1001
Product-ID	KOLTER 0x0015
Adressierung	var. PNP
PCI-Bus	32 bit, PCI 2.1 (2.2) compliance
User-Bus	16 bit, quasi x80 Standard
Anschlüsse	37 pol. Sub-D Buchse 2 x 8 pol. IDC Programmier-Stecker (intern)
Abmessungen	141 x 103 mm, Kartenmaße o. PC-Blech
Temperaturbereich	0...70 °C, typ. Betriebszustand, Dauer
Lagertemperatur max.	-20...+85 °C

Zulassungen und Eigenschaften

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

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 &= 0xFFFFFFFF; /* 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 beigelegten 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);
}

```

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;

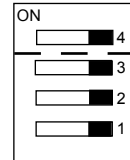
      outp(port,wert&0xff); // auf Port schreiben
      outp(port+4,(wert>>8)&0xff); // dito

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

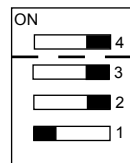
Vendor- und Produkt ID-Informationen

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



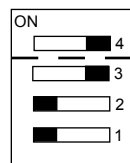
Produkt-ID: 0x0011_{Hex}
 OPTO-PCI /N
 OPTO-PCI /P



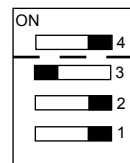
Produkt-ID: 0x0012_{Hex}
 PCI-ADxx
 PCI-ADxx-DAC4
 PCI-DAC4
 PCI-ADTERM



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



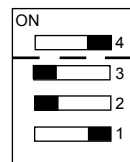
Produkt-ID: 0x0014_{Hex}
 PCI-Counter-1, 3 x 25 bit U/D
 PCI-Timer



Produkt-ID: 0x0015_{Hex}
PCI-DAC 416 und PCI-DAC 412



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 Spezifikation

DIL-Switch 4: OFF = PCI 2.1 Spezifikation

Andere Vendor-IDs

```
#if 0
PCICODE.H
```

Created automatically from the web using the following URL:
<http://www.halcyon.com/scripts/jboemler/pci/pcicode>
 Software to create and maintain the PCICODE List written by:
 Jim Boemler (jboemler@halcyon.com)
 This is header number 547, generated 08-21-96.

Too many people have contributed to this list to acknowledge them all, but a few have provided the majority of the input and deserve special mention:

Frederic Potter (frederic@cao-vlsi.ibp.fr), who maintains a list for Linux.
 Chris Aston (caston@madge.com) at Madge Networks, who furnished his companys list.
 Thomas Dippon of Hewlett-Packard GmbH, who added a huge list of vendors and devices _after_ it seemed the list was nearly complete.

```
#endif
```

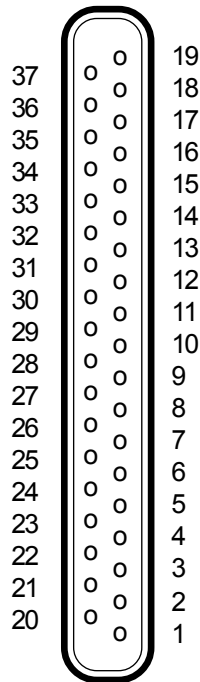
```
typedef struct _PCI_VENTABLE
{
    unsigned short VenId ;
    char * VenShort ;
    char * VenFull ;
} PCI_VENTABLE, *PPCI_VENTABLE ;

PCI_VENTABLE PciVenTable [] =
{
    { 0x003D, „M-M“, „Martin-Marietta Corporation“ },
    { 0x0E11, „Compaq“, „Compaq“ },
    { 0x1000, „SYM“, „Symbios Logic Inc.“ },
    { 0x1001, „KOLTER“, „KOLTER ELEC. Germany“ },
    { 0x1002, „ATI“, „ATI Technologies“ },
    { 0x1003, „ULSI“, „ULSI“ },
    { 0x1004, „VLSI“, „VLSI Technology“ },
    { 0x1005, „Avance“, „Avance Logic Inc.“ },
    { 0x1006, „Reply“, „Reply Group“ },
    { 0x1007, „NetFrame“, „Netframe Systems“ },
    { 0x1008, „Epson“, „Epson“ }, { 0x100A, „“, „“ },
    { 0x100A, „Phoenix“, „Phoenix Technologies Ltd.“ },
    { 0x100A, „“, „“ },
    { 0x100B, „NSC“, „National Semiconductor“ },
    { 0x100C, „Tseng“, „Tseng Labs“ },
    { 0x100C, „“, „tseng labs“ },
    { 0x100D, „AST“, „AST Research“ },
    { 0x100E, „Weitek“, „Weitek“ },
    { 0x1010, „VLogic“, „Video Logic Ltd.“ },
    { 0x1011, „DEC“, „Digital Equipment Corporation“ },
    { 0x1012, „Micronics“, „Micronics Computers Inc.“ },
    { 0x1013, „Cirrus“, „Cirrus Logic“ },
    { 0x1014, „IBM“, „IBM“ },
    { 0x1015, „LSIL“, „LSI Logic Corp of Canada“ },
    { 0x1016, „ICL“, „ICL Personal Systems“ },
    { 0x1017, „Spea“, „Spea Software AG“ },
```

```
USW
```

Steckerbelegung

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



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	n.c.
12	DAC V4	31	AGND
13	AGND	32	n.c.
14	DAC V3	33	AGND
15	AGND	34	n.c.
16	DAC V2	35	AGND
17	AGND	36	n.c.
18	DAC V1	37	AGND
19	AGND		

GND	= digital Masse
AGND	= analog Masse (nur für D/A-Ausgänge)
DAC Vx	= analog Ausgang x = 1...4
DIGx	= digital Ausgang (0/5 V) x = 1...2
Vcc	= 5 Volt Ausgang, ungesichert

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

Internet

E-Mail - Service service@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>