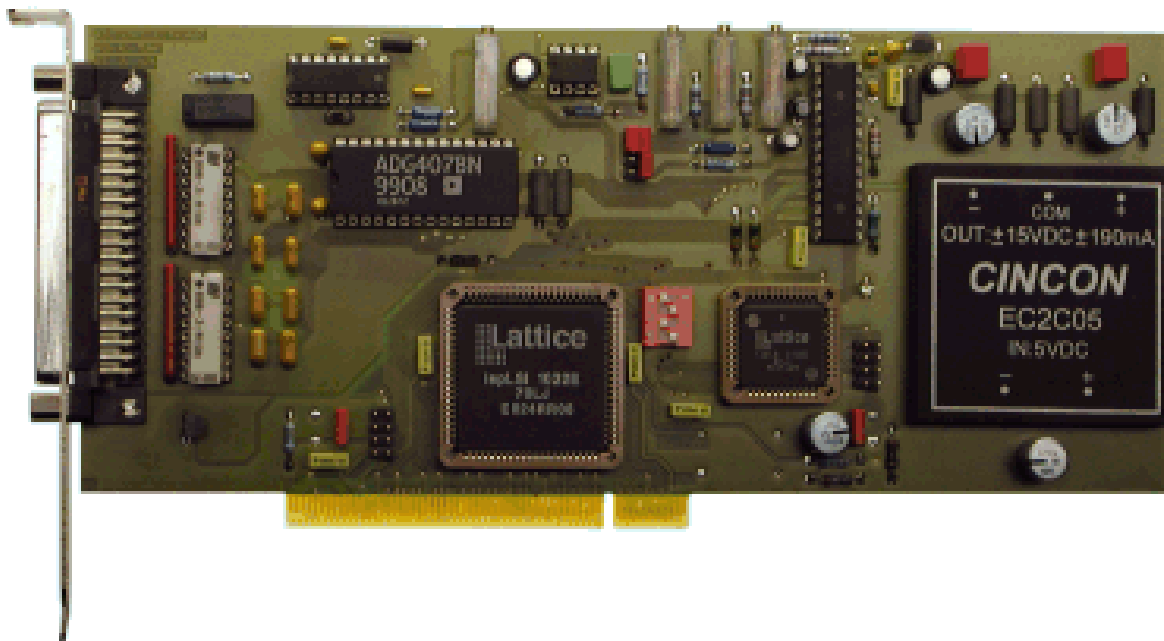


PCI-AD-TEMP

A/D-Messkarte für besondere Aufgaben
zur Kleinsignal-Spannungsmessung



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	4
Allgemeines zu I/O-Karten	5
Funktionsweise der Karte	6
Registerbelegung	7
Blockschaltbild	8
Kartenansicht und Bauteile	9
Jumpereinstellungen	10
Technische Daten	11
Testprogramm in GW-BASIC	12
Einbindung in eigene Programme	14
Allgemeine Adressierung unter PCI	16
Vendor und Produkt ID-Informationen	17
Andere Vendor-IDs	18
Steckerbelegungen	19
Anschriften und Rufnummernverzeichnis	20



Willkommen

Sehr geehrter Kunde,

wir bedanken uns für den Kauf oder das Interesse an unserer PCI-AD-TEMP 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-Richtlinien-novellen 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 Voraussetzung für jeden Um- und Einbau, sowie die Verbindung mit anderen Anlagen, Komponenten oder Teilen.

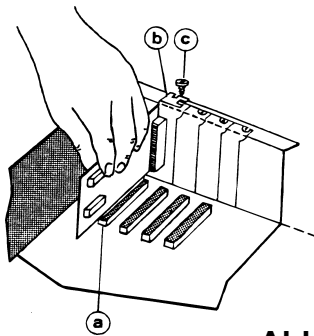


Abb. 1

2. Öffnen Sie den PC. Im allgemeinen müssen dazu auf der Rückseite des Gerätes vier Sicherungsschrauben mit einem Kreuzschlitzschraubendreher gelöst werden. Anschließend 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.

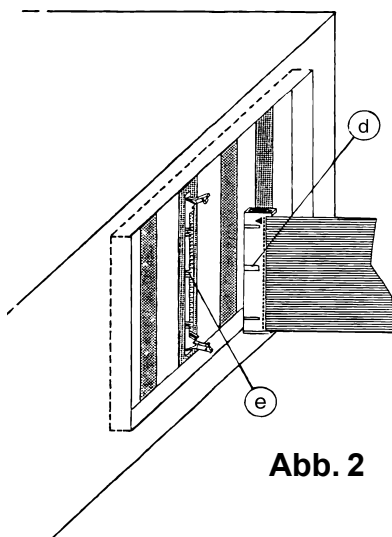


Abb. 2

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 Anlagen-seite 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 Steuerungsspannungen 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 Messkarte „PCI-AD-TEMP“ ist mit einem 16-bit A/D-Wandler (ADS 7807) und einem programmierbaren Vorverstärker (PGA 204) ausgestattet. Der Messbereich des Wandlers ist auf 5 Volt festgelegt, die Vorverstärkung kann je nach Bitmuster A0/A1 auf 1,10,100 oder 1000-fach eingestellt werden. Somit lassen sich auch noch kleinste Spannungen von NiCr-Ni Thermoelementen u.a. Sensoren mit +/- 5mV (full-range) präzise messen.

Damit Störsignale, Eingangsruschen und 50 Hz Brummeinstreuungen am Messeingang das Nutzsinal nicht unnötig belastet, wurde ein 20ms R-C-Tiefpassfilter vor jeden analogen Messkanal geschaltet, bevor das Signal dem Multiplexer zugeführt wird. Da die gesockelten Reihenwiderstände als Array ausgeführt sind, kann die Integrationszeit der R-C-Kombination nachträglich verändert werden, wodurch sich die Messgenauigkeit erheblich verbessern lässt. Alle Eingänge sind in Differentialtechnik (d.e.) ausgeführt und mit 1 MOhm - SIL (ebenfalls gesockelt) nach analog-GND vorterminiert.

Eine Besonderheit ist die programmierbare Nullpunktkorrektur, die durch Umschalten eines speziellen Relais auf Kanal 8, den Messkanal auf einen virtuellen Nullpunkt kurzschließt. Um die Abweichung zum tatsächlichen Nullpunkt zu kompensieren, wird der ermittelte Messwert der quasi „neuen“ Nullpunktspannung bei Folgemessungen dann lediglich als negativen Offset mit einbezogen. Den Messvorgang der Nullpunkterfassung kann der Anwender jederzeit selbst bestimmen.

Die Karte verfügt zusätzlich über einen digitalen Ein- und Ausgang (TTL). Bis auf die Ansteuerung der PGA-Leitungen A0/A1 und der Programmierung des Relais, sind alle I/O-Register zur PCI-AD16LC kompatibel.

PCI-AD-TEMP Merkmale:

- Hochpräzise Kleinsignal-Messkarte
- z.B. für NiCr-Ni Temperaturfühler
- oder für Messungen an Richtkopplerdioden...

- 8-Kanal-Differential-Eingang (symmetrisch)
- für Thermoelemente oder andere Kleinsignale
- +/- 5mV, 50 mV, 500 mV, 5 Volt bzw. 0..2,5 mV, 25 mV, 250 mV, 2,5 Volt
- 16-bit ADC Burr Brown Wandler, 25 µs
- 20 ms low-pass R-C Eingangfilter
- Nullpunktgleich per Relais (an Kanal 8)
- kleinste Abweichungen von nur 1..2 µV

Alle I/O Signale sind über eine 37-polige Sub-D-Buchse von außen am PC-Blech zugänglich. Zur Versorgung externer Schaltungen sind zusätzlich, neben den beiden Digitalkanälen, aus dem PC die GND und +5 Volt-Leitung auf der Sub-D Buchse geführt.

Bei der Verwendung der externen BNC-Box, die speziell für die PCI-AD-Karten entwickelt wurde und ebenfalls über KOLTER ELECTRONIC bezogen werden kann, ist der Transientenschutz (TS) auf der Karte überflüssig, da sich die entsprechenden Schutzbauteile bereits in der BNC-Box befinden.

Die Einstellung der Kartenadressierung erfolgt automatisch über Plug&Play (PnP). Eine weitergehende Bauteilinitialisierung ist nicht erforderlich, A/D- und D/A-Wandler werden über die entsprechenden I/O-Port-Register direkt programmiert (siehe dazu die Programmbeispiele). Andere Beispiele zur Kartenprogrammierung entnehmen Sie bitte den Quelltexten auf der CD.



Registerbeschreibung

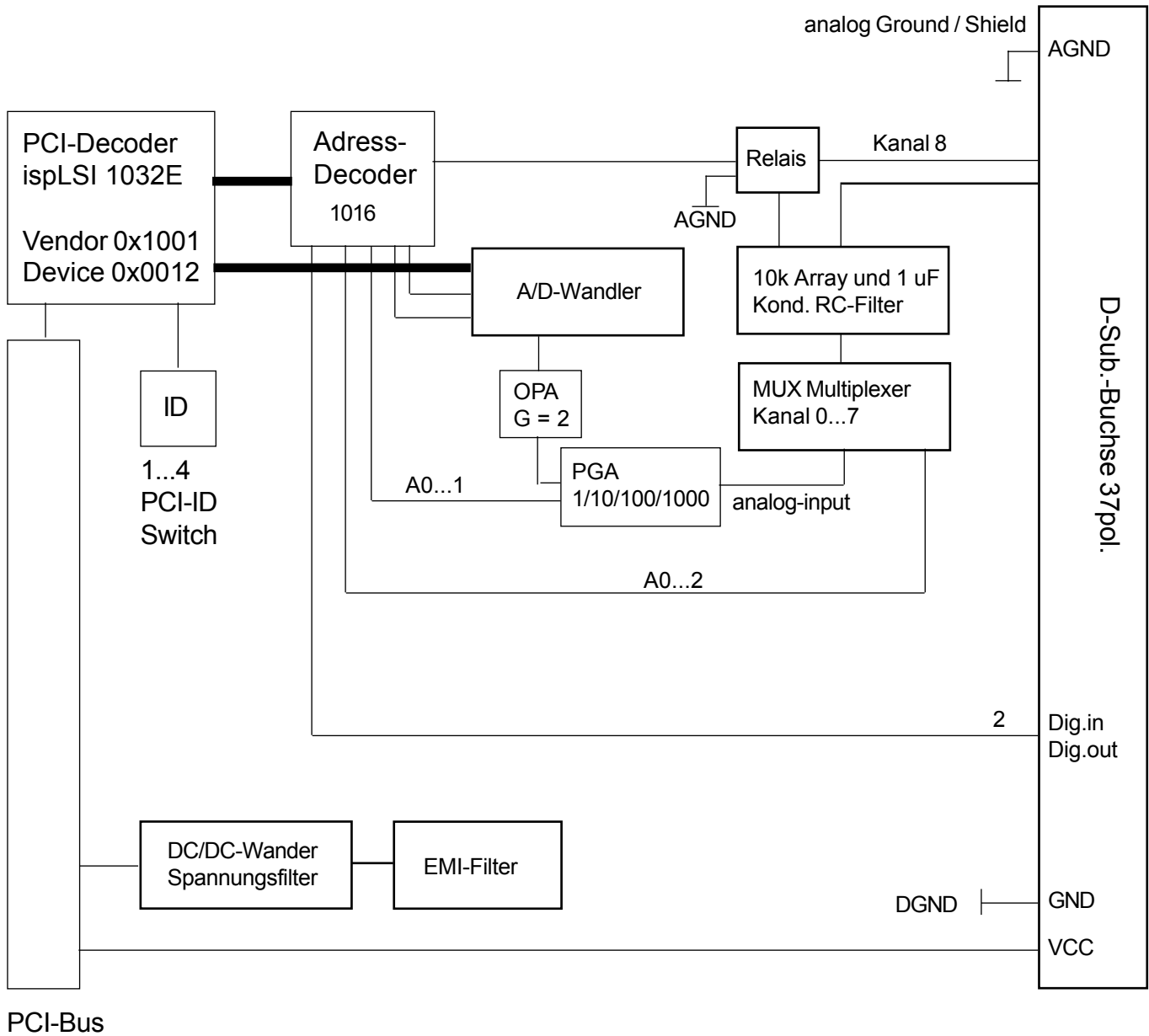
Zur Programmierung bzw. Adressierung der I/O-Port-Register sind grundsätzlich folgende Offset-Adressen zu beachten:

in/out	Adresse	Data	Funktion	sonstiges
Digital I/O				
in	adr + 17	0/2	digital Input auf bit 1	restl. bits ohne Verwendung
out	adr + 16	0/1	digital Output auf bit 0	restl. bits ohne Verwendung
Multiplexer				
out	adr + 4	0..7	Multiplexer-Kanal setzen (ADC)	warten 0,25 bis 10 us
A/D-Wandler				
in	adr + 1	0/1	EOC-bit end-of-conversion	A/D-Wandlungsende abfragen
out	adr + 0	1	setze ADC auf lesen	R/C bit
out	adr + 0	0	convert ADC	A/D-Wandlung
out	adr + 0	1	high-byte von ADC aktiv	HB setzen
out	adr + 0	2	low-byte von ADC aktiv	LB setzen
in	adr + 0	0...255	ADC byte Register (Data-Port)	HB/LB lesen
PGA-mode				
out	adr + 28	0	mode ± 5 Volt	PGA: A0=0 A1=0
out	adr + 28	1	mode ± 500 mV	PGA: A0=1 A1=0
out	adr + 28	2	mode ± 50 mV	PGA: A0=0 A1=1
out	adr + 28	3	mode ± 5 mV	PGA: A0=1 A1=1
Nullpunkt-Relais				
out	adr + 4	7	Mux-Kanal für Relais einstellen	
out	adr + 28	mode + 4	Relais auf virtuellen Nullpunkt schalten	

Alle Angaben in dezimal.

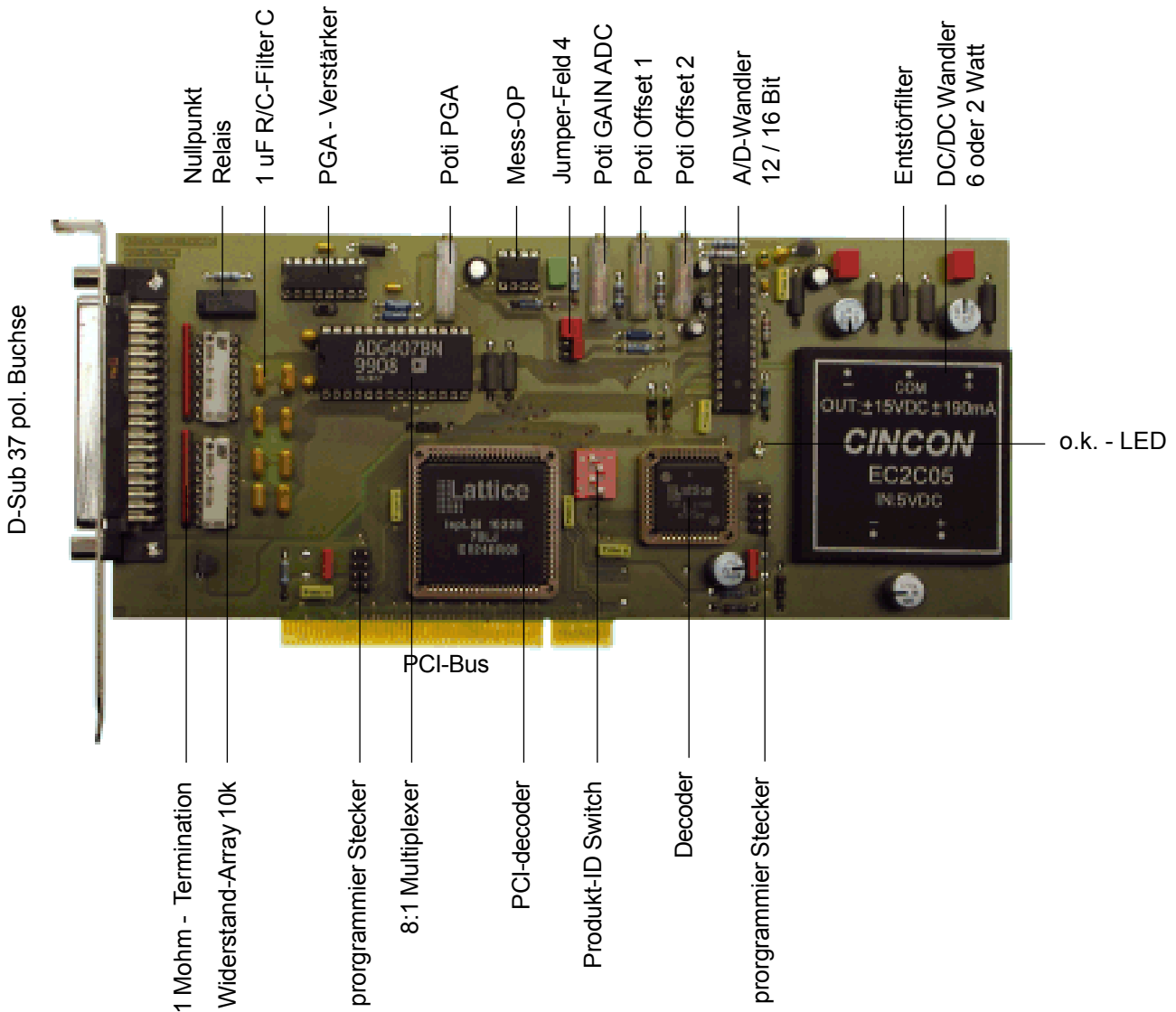
Blockschaltbild

Hier eine vereinfachte Übersicht über die einzelnen Funktionsblöcke der A/D-Karte:



Kartenansicht und Bauteile

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



Allgemein

Die Einstellung des DIP-Schalters darf nicht verändert werden. Der DIP-Schalter legt die Produkt-ID von 0x0012 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 beziehungsweise das PLSI richtig geladen wurde (ein Selbsttest).

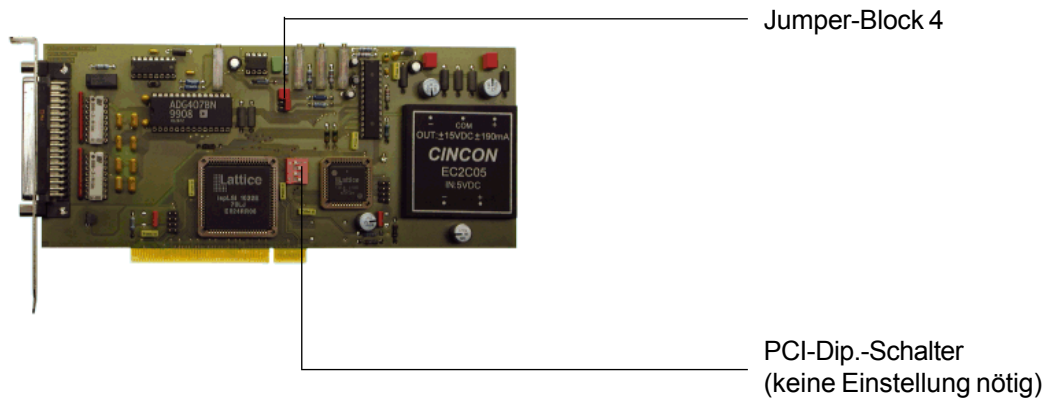
Filter - Eingangsanpassung

Die PCI-AD-TEMP Karte kann mit verschiedenen Widerstandarrays ausgerüstet werden, um so die Filtereigenschaften der Eingangsspannung optimal an den A/D-Wandler und Multiplexer anzupassen. Werkseitig werden für den RC-Filter 10 k und 1 uF vorbestückt. Die daraus berechnete Zeitkonstante lässt sich wie folgt berechnen:

$$t (63\%) = C * R \quad \text{oder} \quad 20 \text{ ms} = 1 \text{ uF} * 20 \text{ kohm}$$

Jumpereinstellungen

Um die einzelnen Betriebs-Modi (OP-Verstärkung, ADC-Einstellung, slew-rate-Filter...) auszuwählen werden Steckbrücken bzw. Jumper gesteckt.



Alle hier aufgeführten Messspannungsbereiche beziehen sich auf den A/D-Wandler-Teil inkl. der Vorverstärkerschaltung des OPA 227 (G=2). Folgende A/D-Einstellungen (gezeigt ist die Standard-Version) können durch umstecken des Jumperblocks 4 erreicht werden:

Eingang / Volt	Block 4	Bemerkung
± 5		bipolar - mode
0...2,5		unipolar - mode

Je nach Vorverstärkereinstellung des PGA (über A0 / A1) wird der jeweilige Spannungsmessbereich mit der Verstärkung der PGA multipliziert.
PGA = programmable gain amp

ACHTUNG ! Neu-Abgleich bei Bereichsänderung

Durch Einstellen eines neuen Meßspannungsbereichs (*default* = ± 5 Volt) kann ein Neuabgleich der Einstellungen „Gain und Offset“ erforderlich werden.

Mit dem Poti „Offset“ wird zuerst, bei kurzgeschlossenem Eingang, der „Nullpunkt“ justiert. Dann mit dem Poti „Gain“ auf die anliegende Spannung abgeglichen. Diese Spannung sollte im Endbereich des Skalenwertes liegen. Dazu sollte unbedingt eine ausreichend genaue Referenzspannung verwendet werden. Ist diese nicht verfügbar, dann kann auch ein kalibriertes Digitalvoltmeter parallel zu einer Batterie geschaltet werden, um die Spannungswerte während der Justage zu vergleichen.



Technische Daten

A/D-Kanäle	8, double ended, wahlweise unipolar oder bipolar (Jumper)	
A/D-Wandler Auflösung	16 bit (je nach Garde: ADS7807 P oder ADS7807 PB)	
Meßspannungen	0...2,5 bzw. ± 5 Volt (ohne Instrumentenverstärker)	
Instrumentenvorverstärker	über PGA-Verstärker, Gain = 1 : 10 : 100 : 1000 -fach	
PGA Toleranz	siehe Datenblatt zu PGA204AP (BP)	
Überspannungsschutz	< 44 Vss, während Betrieb (weitere Daten siehe ADG 407 BN)	
Eingangswiderstand	> 1 M Ω (ohne Shunt) je A/D-Kanal	
A/D Geschwindigkeit	typ. 25 μ s. + Multiplexerzeit	
MUX Multiplexerzeit	typ. 4..10 μ s. (min. 250 ns)	
A/D Toleranz	lin. 16 bit, typ. ± 2 LSB	
R-C-Filter	je Kanal $2x R_v = 10k + C_x = 1\mu F$	
Grenzfrequenz	$t(63\%) = C * R = 20$ ms	
Digital - Input	1 Kanal, TTL (0/5 Volt), Pegel siehe Datenblatt ispLSI1016	
Digital - Output	1 Kanal, TTL (0/5 Volt), Pegel siehe Datenblatt ispLSI1016	
Betriebsarten	I/O, A/D, PGA	port-polling, programmgesteuert, byte-weise
PCI-Decoder	1	ispLSI 1032E, Lattice IC
Vendor-ID	KOLTER	0x1001
Product-ID	KOLTER	0x0012
Adressierung	var.	PCI PNP
Bus	32 bit	32bit PCI Bus, gem. Spec. 2.1 - 2.3
Anschlüsse	37 pol. 2x 8 pol. IDC	Sub-D Buchse Programmier-Stecker (für interne Zwecke)
Abmessungen	100 x 216 mm	Kartenmaße (ca.-Maße ohne Halteblech)
Temperaturbereich	10...50 °C	typ. Betriebszustand / Dauerbetrieb
Temperaturbereich	0...70 °C	max. 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 GW-BASIC

```

1 ADR = (&H6500): REM hier absolute BASIS-I/O-Adresse
2 MODE = 0      : REM Verstaerkung von PGA einstellen
3 NULLP = 4     : REM Bit fuer Nullpunkt-Abgleich
4 KORR = 0

10 CLS: PRINT „PCI-AD-TEMP GW-BASIC Testprogramm (c) KOLTER ELECTRONIC“
20 PRINT „1...Bereich +/- 5 Volt „
30 PRINT „2...Bereich +/- 0,5 Volt „
40 PRINT „3...Bereich +/- 50 mV „
50 PRINT „4...Bereich +/- 5 mV „
60 A$=INKEY$: IF A$="" THEN GOTO 60

70 IF A$= „1“ THEN MODE = 0 : OUT (ADR + 28),MODE
75 IF A$= „2“ THEN MODE = 1 : OUT (ADR + 28),MODE
80 IF A$= „3“ THEN MODE = 2 : OUT (ADR + 28),MODE
85 IF A$= „4“ THEN MODE = 3 : OUT (ADR + 28),MODE
89 CLS

90 IF A$= „1“ THEN PRINT„Aktueller Messbereich +/- 5 Volt“
91 IF A$= „2“ THEN PRINT„Aktueller Messbereich +/- 0,5 Volt“
92 IF A$= „3“ THEN PRINT„Aktueller Messbereich +/- 50 mV“
93 IF A$= „4“ THEN PRINT„Aktueller Messbereich +/- 5 mV“

100 REM
110 MVD = .0003051804379#
120 MAX = 200
130 Locate 4 : Gosub 1000
133 Print „ „

135 PRINT „          korrigiert   unkorrigiert“
136 PRINT „_____“
140 Locate 8

150 FOR KANAL = 0 TO 7: REM Messkanal 0 bis 7 am MUX einstellen
160 OUT (ADR + 4), KANAL: REM Output auf MUX
170 GOSUB 360: REM warten bis MUX eingestellt ist
172 DIGIT = 0
173 ALLE = 0
175 FOR MWERT = 1 TO MAX
180 GOSUB 500
185 ALLE = ALLE + DIGIT
190 NEXT MWERT
200 DIGIT = (ALLE / MAX)
260 VOLT = ((DIGIT * MVD) - 10!) / 2 : REM Spannung berechnen
270 PRINT USING „DIGIT = ##### „ ;DIGIT,
275 PRINT „ Kanal=“; KANAL, USING „ mV = ###.### „; VOLT-KORR; VOLT
280 NEXT KANAL

```



```

290 FOR T = 0 TO 5000!: NEXT T: REM Verzoeigerung nur fuer Konsolenanzeige
300 SOUND 15000, .02: REM Lautsprecher-Click erzeugen
310 DIN = INP(ADR + 17): REM digitaler Eingang lesen
320 PRINT
330 IF (DIN AND 2) = 2 THEN PRINT „Digital-Input = HIGH „: OUT (ADR + 16), 1
340 IF (DIN AND 2) = 0 THEN PRINT „Digital-Input = LOW „: OUT (ADR + 16), 0
350 GOTO 140: REM Schleifenende beginnt von vorne

355 REM _____

360 FOR DELAY = 0 TO 2000: NEXT DELAY: RETURN: REM min. 20ms wegen 50Hz !!!
500 REM ADC wandeln
510 OUT (ADR + 0), 1: REM RC = read ADC, Vorbereitung zur Wandlung
520 OUT (ADR + 0), 0: REM RC = convert ADC, jetzt Wandlung !!!
530 IF (INP(ADR + 1) AND 1) = 0 THEN GOTO 530: REM EOC-Bit ADC abfragen
540 OUT (ADR + 0), 0 + 1: REM Set HSB byte fuer ADC
550 HB = INP(ADR + 0): REM high-byte lesen
560 OUT (ADR + 0), 2 + 1: REM Set LSB byte fuer ADC
570 LB = INP(ADR + 0): REM low-byte lesen
580 DIGIT = (HB * 256) + LB: REM 12/16bit digital-word bilden
590 RETURN

1000 REM Nullpunkt messen
1010 OUT (ADR + 4), 7: REM letzter Messkanal einstellen
1020 OUT (ADR + 28),MODE + 4 : REM Kurzschlussbrücke mit Relais schalten
1030 FOR DELAY = 0 TO 100000: NEXT DELAY : REM ca. 200 ms warten
1040 ALLE = 0: REM ..... AD-Wert holen .....
1050 FOR MWERT = 1 TO 500
1060 GOSUB 500
1070 ALLE = ALLE + DIGIT
1080 NEXT MWERT
1090 DIGIT = (ALLE / 500)
1100 VOLT = ((DIGIT * MVD) - 10!) / 2 : REM Spannung berechnen
1110 PRINT USING „DIGIT = ##### „;DIGIT,
1120 PRINT „ Nullpunkt =“; USING „###.### mV „; VOLT
1130 OUT (ADR + 28),MODE : REM Kurzschlussbrücke abschalten
1140 FOR DELAY = 0 TO 10000: NEXT DELAY : REM ca. 200 ms warten
1150 KORR = VOLT
1160 Return
  
```



Einbindung in eigene Programme

Programmierbeispiel zur Karteneinbindung

Um die entsprechende Vergabe der I/O-Adresse braucht sich als Anwender bei PCI-Karten nicht zu kümmern. Wohl aber um Installation der Device-Treiber und die Einbindung der Adresse in sein Programm.

Die Device-Treiber werden, über ein Setup-Programm, von der CD aus in „C:\Programme“ installiert. Die CD liegt jedem KOLTER-Produkt bei kann aber auch gegen eine Schutzgebühr separat bezogen werden.

Zur Einbindung in eigene Projekte beziehungsweise zur Weitergabe einer Anwendungssoftware mit den Device-Treibern, können die entsprechenden Dateien direkt in den Projekt-Ordner kopiert werden (aus dem Verzeichnis \Pci\Utils_PCI\Visual_BASIC\Sourcen). Das Setup befindet sich auf der aktuellen CD im Pfad \Pci\Utils_PCI\Visual_BASIC\Program.

Damit das eingene Programm weiß, wo sich die neue Karte im Adressraum befindet, müssen Parameter übergeben werden. Das Programm PCIGETIO.C (im CD-Ordner:\PCI\Utils_PCI\pci_tools) 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 rcx,
                                       unsigned long redi);

unsigned long int PciGetIO(Device) {
unsigned long reax;
unsigned long rebx;
unsigned long rcx;
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(0x12);  // 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 dem beigelegten Datenträger.

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 sich dazu im gleichen Verzeichnis wie TEST.EXE befinden.

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(0x012);

  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;

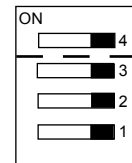
      outp(port,wert&0xff); // auf Port schreiben
      outp(port+4,(wert>>8)&0xff); // dito

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

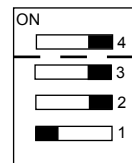

Vendor- und Produkt ID-Informationen

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

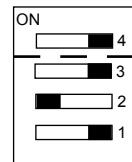
Produkt-ID: 0x0010_{Hex}
 PCI-1616 TTL I/O
 und für zukünftige TTL-I/O-Karten



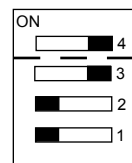
Produkt-ID: 0x0011_{Hex}
 OPTO-PCI /N
 OPTO-PCI /P



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



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



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



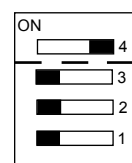
Produkt-ID: 0x0015_{Hex}
 PCI-DAC 416



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



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
```

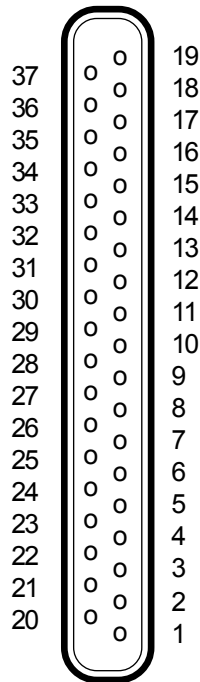
```
typedef struct _PCI_VENTABLE
{
    unsigned short VenId ;
    char * VenShort ;
    char * VenFull ;
} PCI_VENTABLE, *PPCI_VENTABLE ;

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.
```

Steckerbelegungen

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



1	n.c.	20	n.c.
2	n.c.	21	n.c.
3	n.c.	22	n.c.
4	n.c.	23	n.c.
5	n.c.	24	n.c.
6	AGND	25	AGND
7	AD0 +	26	AD1 +
8	AD2 +	27	AD3 +
9	AD4 +	28	AD5 +
10	AD6 +	29	AD7 +
11	AGND	30	AD0 -
12	AD1 -	31	AD2 -
13	AD3 -	32	AD4 -
14	AD5 -	33	AD6 -
15	AD7 -	34	AGND
16	AGND	35	n.c.
17	n.c.	36	DIG out
18	DIG inp	37	+ 5 Volt
19	D.GND		(nicht abgesichert)

D.GND	= digital Masse
AGND	= analog Masse
ADx	= analog Eingang x=0..7
DIG inp	= digital Eingang (0/5 V)
DIG out	= digital Ausgang (0/5 V)
+ 5 Volt	= Spannungsausgang vom PC

Anschriften und Rufnummernverzeichnis



Anschriften

Postfach 1127
Steinstraße 22

D-50362 Erftstadt
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



EMV-Konformität:

Die EMV-Konformität gilt für industrielle 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>