



**DIN EN ISO 9001:2000
zertifiziert**



**ADDI-DATA GmbH
Dieselstraße 3
D-77833 OTTERSWEIER
+49 (0)7223 / 9493 – 0**

Software-Beschreibung

ADDICOUNT APCI-/CPCI-1710

Digitale Ein- und Ausgänge

5. Ausgabe 12/2004

Produktinformation

Dieses Handbuch enthält die technischen Anlagen, wichtige Anleitungen zur korrekten Inbetriebnahme und Nutzung sowie Produktinformation entsprechend dem aktuellen Stand vor der Drucklegung.

Der Inhalt dieses Handbuchs und die technischen Daten des Produkts können ohne vorherige Ankündigung geändert werden. Die ADDI-DATA GmbH behält sich das Recht vor, Änderungen bzgl. der technischen Daten und der hierin enthaltenen Materialien vorzunehmen.

Gewährleistung und Haftung

Der Nutzer ist nicht berechtigt, über die vorgesehene Nutzung der Karte hinaus Änderungen des Werks vorzunehmen sowie in sonstiger Form in das Werk einzugreifen.

ADDI-DATA übernimmt keine Haftung bei offensichtlichen Druck- und Satzfehlern. Darüber hinaus übernimmt ADDI-DATA, soweit gesetzlich zulässig, weiterhin keine Haftung für Personen- und Sachschäden, die darauf zurückzuführen sind, dass der Nutzer die Karte unsachgemäß installiert und/oder in Betrieb genommen oder bestimmungswidrig verwendet hat, etwa indem die Karte trotz nicht funktionsfähiger Sicherheits- und Schutzvorrichtungen betrieben wird oder Hinweise in der Betriebsanleitung bzgl. Transport, Lagerung, Einbau, Inbetriebnahme, Betrieb, Grenzwerte usw. nicht beachtet werden. Die Haftung ist ferner ausgeschlossen, wenn der Betreiber die Karte oder die Quellcode-Dateien unbefugt verändert und/oder die ständige Funktionsbereitschaft von Verschleißteilen vorwerfbar nicht überwacht wurde und dies zu einem Schaden geführt hat.

Urheberrecht

Dieses Handbuch, das nur für den Betreiber und dessen Personal bestimmt ist, ist urheberrechtlich geschützt. Die in der Betriebsanleitung und der sonstigen Produktinformation enthaltenen Hinweise dürfen vom Nutzer des Handbuchs weder vervielfältigt noch verbreitet und/oder Dritten zur Nutzung überlassen werden, soweit nicht die Rechstübertragung im Rahmen der eingeräumten Produktlizenz gestattet ist. Zuwiderhandlungen können zivil- und strafrechtliche Folgen nach sich ziehen.

ADDI-DATA-Software Produktlizenz

Bitte lesen Sie diese Lizenz sorgfältig durch, bevor Sie die Standardsoftware verwenden. Das Recht zur Benutzung dieser Software wird dem Kunden nur dann gewährt, wenn er den Bedingungen dieser Lizenz zustimmt.

Die Software darf nur zur Einstellung der ADDI-DATA Karten verwendet werden.

Das Kopieren der Software ist verboten (außer zur Archivierung/Datensicherung und zum Austausch defekter Datenträger). Deassemblierung, Dekompilierung, Entschlüsselung und Reverse Engineering der Software ist verboten. Diese Lizenz und die Software können an eine dritte Partei übertragen werden, sofern diese Partei eine Karte käuflich erworben hat, sich mit allen Bestimmungen in diesem Lizenzvertrag einverstanden erklärt und der ursprüngliche Besitzer keine Kopien der Software zurückhält.

Warenzeichen

- ADDI-DATA ist ein eingetragenes Warenzeichen der ADDI-DATA GmbH.
- Turbo Pascal, Delphi, Borland C, Borland C++ sind eingetragene Warenzeichen von Borland Insight Company.
- Microsoft C, Visual C++, Windows XP, 98, Windows 2000, Windows 95, Windows NT, EmbeddedNT und MS DOS sind eingetragene Warenzeichen von Microsoft Corporation.
- LabVIEW, LabWindows/CVI, DasyLab, Diadem sind eingetragene Warenzeichen von National Instruments Corp.
- CompactPCI ist ein eingetragenes Warenzeichen der PCI Industrial Computer Manufacturers Group.
- VxWorks ist ein eingetragenes Warenzeichen von Windriver.

WARNUNG

Bei unsachgemäßem Einsatz und bestimmungswidrigem Gebrauch der Karte können:



◆ **Personen verletzt werden,**



◆ **Baugruppe, PC und Peripherie beschädigt werden,**



◆ **Umwelt verunreinigt werden.**

◆ **Schützen Sie sich, andere und die Umwelt!**

◆ **Sicherheitshinweise unbedingt lesen.**

Liegen Ihnen keine Sicherheitshinweise vor, so fordern Sie diese bitte an.

◆ **Anweisungen des Handbuches beachten.**

Vergewissern Sie sich, dass Sie keinen Schritt vergessen haben. Wir übernehmen keine Verantwortung für Schäden, die aus dem falschen Einsatz der Karte hervorgehen könnten.

◆ **Folgende Symbole beachten:**



WICHTIG!

kennzeichnet Anwendungstipps und andere nützliche Informationen.



WARNUNG!

bezeichnet eine möglicherweise gefährliche Situation.

Bei Nichtbeachten des Hinweises können Karte, PC und/oder Peripherie zerstört werden.

1	BESTIMMUNGSGEMÄSSE VERWENDUNG	6
1.1	Bestimmungsgemäßer Zweck	6
1.2	Bestimmungswidriger Zweck.....	6
1.3	Technische Dokumentation.....	6
1.4	Funktionsbeschreibung	7
2	DIGITALE EIN- UND AUSGÄNGE	8
2.1.1	Blockdiagramm.....	8
2.1.2	Typische Anwendungen	9
2.2	Benutzte Signale.....	9
2.3	Pinbelegung des Frontsteckers.....	10
2.4	Anschlussbeispiel.....	11
2.5	E/A-Adressbelegung	12
2.6	Beschreibung der E/A-Funktionen	13
2.6.1	Select-Register.....	13
2.6.2	Output-Register	13
2.6.3	Input-Register.....	14
2.6.4	Erkennungs-REGISTER (Base +60).....	14
2.7	Arbeiten mit der "Digitale E/A"-Funktion	14
3	STANDARDSOFTWARE	15
3.1	Einleitung.....	15
3.2	Softwarefunktionen	16
3.2.1	Initialisierung	16
	1) i_APCI1710_InitDigitalIO (...).	16
3.2.2	Digitale E/A lesen	17
	1) i_APCI1710_ReadDigitalIOChlValue (...).	17
	2) i_APCI1710_ReadDigitalIOPortValue (...).	18
3.2.3	Digitale Ausgänge schreiben	19
	1) i_APCI1710_SetDigitalIOMemoryOn (...).	19
	2) i_APCI1710_SetDigitalIOMemoryOff (...).	20
	3) i_APCI1710_SetDigitalIOChlOn (...).	21
	4) i_APCI1710_SetDigitalIOChlOff (...).	22
	5) i_APCI1710_SetDigitalIOPortOn (...).	23
	6) i_APCI1710_SetDigitalIOPortOff (...).	24
3.2.4	Funktionen im Kernel-Mode benutzen.....	25
	Eingänge lesen	25
	1) i_APCI1710_KRNL_ReadDigitalIOChlValue (...).	25
	2) i_APCI1710_KRNL_ReadDigitalIOPortValue (...).	26
	Ausgänge schreiben	27
	3) i_APCI1710_KRNL_SetDigitalIOChlOn (...).	27
	4) i_APCI1710_KRNL_SetDigitalIOPortOn (...).	28

Abbildungen

Abb. 2-1: Blockdiagramm der Funktion "Digital E/A"	8
Abb. 2-2: Pinbelegung des 50-pol. SUB-D Steckers.....	10
Abb. 2-3: Anschlussbeispiel	11

Tabellen

Tabelle 1-1: Mitgelieferte Funktionshandbücher.....	7
Tabelle 2-1: Benutzte Signale.....	9
Tabelle 2-2: E/A-Belegung der "Digitale E/A" Funktion	12
Tabelle 3-1: Define-Wert	15

1 BESTIMMUNGSGEMÄSSE VERWENDUNG

1.1 Bestimmungsgemäßer Zweck

Die Karte **APCI-1710** eignet sich für den Einbau in einen PC mit PCI 5V/32 Bit Steckplätzen, der für elektrische Mess-, Steuer-, Regel- und Labortechnik im Sinne der EN 61010-1 (IEC 61010-1), eingesetzt wird.

Die Karte **CPCI-1710** eignet sich für den Einbau in einen CompactPCI-System mit PCI 5V/32 Bit Steckplätzen, der für elektrische Mess-, Steuer-, Regel- und Labortechnik im Sinne der EN 61010-1 (IEC 61010-1), eingesetzt wird.

1.2 Bestimmungswidriger Zweck

Die Karte **APCI-/CPCI-1710** darf nicht als sicherheitsgerichtetes Betriebsmittel (safety related part, SRP) eingesetzt werden.

Die Karte **APCI-/CPCI-1710** darf nicht in explosionsgefährdeten Atmosphären eingesetzt werden.

1.3 Technische Dokumentation

Dieses Referenzhandbuch bezieht sich sowohl auf die Karte **APCI-1710** als auch auf die Karte **CPCI-1710/1711**. Bitte vergewissern Sie sich, dass Sie außerdem folgendes bekommen haben:

- Die CD1 "Standard Software Drivers" mit dem ADDISET Parametrierprogramm und den benötigten Softwaretreibern.
- Die CD2 "Technical Manuals". Die CD enthält
 - das Handbuch **ADDICOUNT APCI-/CPCI-1710: Funktionsprogrammierbare Zählerkarte für den PCI-Bus**, das allgemeine Informationen für den Betrieb der Karte enthält,
 - ein Referenzhandbuch für jede Funktion, die Sie auf die APCI-/CPCI-1710 programmieren wollen,
- das gelbe Blatt mit den Sicherheitshinweisen.

Je nach verwendeter Funktion finden Sie die notwendigen Belegungs- und Programmierinformationen in den einzelnen Handbüchern.

Tabelle 1-1: Mitgelieferte Funktionshandbücher

Funktion	PDF Datei (CD2 technical manuals)		Funktionsbezeichnung in SET1710	CFG Datei
	deutsch	englisch		
Inkrementalzähler	Inkr_zähler_d.pdf	incr_counter_e.pdf	Incremental counter	inc_cpt.cfg
SSI	SSI_d.pdf	SSI_e.pdf	SSI	ssi.cfg
Chronos	chronos_d.pdf	chronos_e.pdf	Chronos	chronos.cfg
Zähler/Timer	Zähler_timer_d.pdf	counter_timer_e.pdf	counter/timer	82x54.cfg
TOR	TOR_d.pdf	TOR_e.pdf	TOR	tor.cfg
PWM	PWM_d.pdf	PWM_e.pdf	Pulse width modulation	PWM.cfg
TTL	TTL_EA_d.pdf	TTL_IO_e.pdf	TTL I/O	Ttl_io.cfg
Digitale E/A	dig_EA_d.pdf	dig_IO_e.pdf	Digital I/O	dig_IO.cfg
Impulszähler	Impulszähler_d.pdf	pulse_counter_e.pdf	Pulse counter	imp_cpt.cfg
ETM	ETM_d.pdf	ETM_e.pdf	Edge time measurement	etm.cfg

Bitte beachten:

Die Karte **CPCI-1710/1711** ist mit der Karte **APCI-1710** kompatibel, was die Softwareinstallation angeht. Die Programme ADDIREG und SET1710 machen keinen Unterschied zwischen PCI-Karten und CompactPCI-Karten.

Die API-Funktionen der Standardsoftware sind ebenfalls identisch.

1.4 Funktionsbeschreibung

Dieses Handbuch enthält neben einer globalen Beschreibung der Funktionen

- die Pinbelegung des Frontsteckers,
- eine Liste der benutzten Signale,
- den E/A-Bereich,
- ein Kapitel über die mitgelieferten API-Funktionen der Standardsoftware.

2 DIGITALE EIN- UND AUSGÄNGE

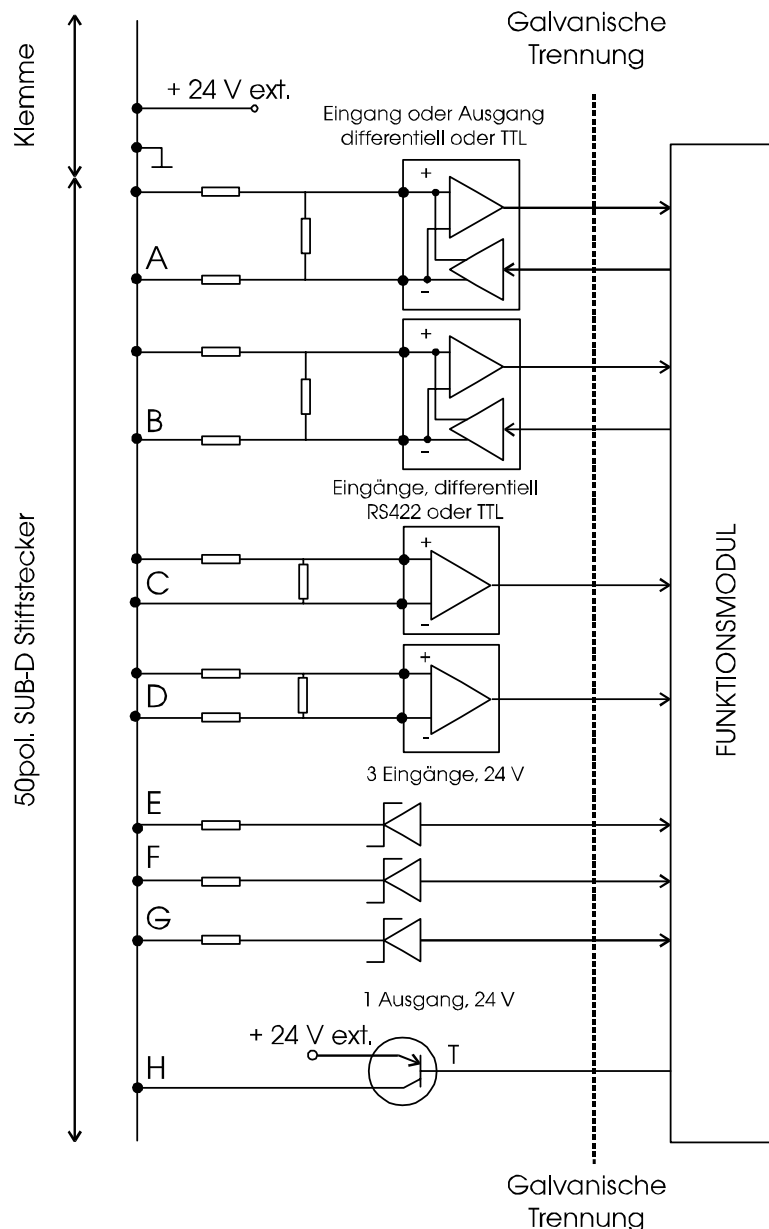
Die Funktion "Digitale Ein-/Ausgänge ermöglicht eine einfache Steuerung bzw. Überwachung einzelner Differenz-, TTL bzw. 24V-Signale.

Eigenschaften:

- 2 x differenzielle RS422/RS485 Eingänge (24 V bei der APCI-1710-24, oder optional für die CPCI-1710 bzw. CPCI-1711), Kanäle C und D
- 2 x differenzielle RS422/RS485 Ein-/Ausgänge (nur für die APCI-1710 oder CPCI-1710); Als Eingang oder Ausgang durch Software einstellbar, Kanäle A und B
- 3 x 24 V Eingänge, Kanäle E, F, G (optional 5 V)
- 1 x 24 V Ausgang (H, optional 5 V)

2.1.1 Blockdiagramm

Abb. 2-1: Blockdiagramm der Funktion "Digital E/A"



2.1.2 Typische Anwendungen

Die Funktion "Digitale Ein-/Ausgänge" ergänzt eine komplexe Zählapplikation mit zusätzlichen 24 V, TTL oder differentiellen RS422 Ein- und Ausgangssignalen.

- einfache Überwachung von 24 V Signalen, RS422, TTL Signalen,
- einfache Steuerung von 24 V, TTL, RS422, differentiellen Signalen.

2.2 Benutzte Signale

Die Funktion "Digitale Ein- und Ausgänge" belegt **5 bis 7 Eingänge (A bis G) und 1 bis 3 Ausgänge (A, B und H)** von dem entsprechenden Funktionsmodul der APCI-/CPCI-1710.

Tabelle 2-1: Benutzte Signale

AM STECKER	POLARITÄT	FUNKTION
A x +/-	Diff./TTL/Opt. 24V*	Digitaler Ein-/Ausgang, per Software umschaltbar
B x +/-	Diff./TTL/ Opt. 24V*	Digitaler Ein-/Ausgang, per Software umschaltbar
C x +/-	Diff./TTL/Opt. 24V	Digitaler Eingang
D x +/-	Diff./TTL/Opt. 24V	Digitaler Eingang
E x	24V/Opt. 5V	Digitaler Eingang
F x	24V/Opt. 5V	Digitaler Eingang
G x	24V/Opt. 5V	Digitaler Eingang
H x	24 V/Opt. TTL	Digitaler Ausgang

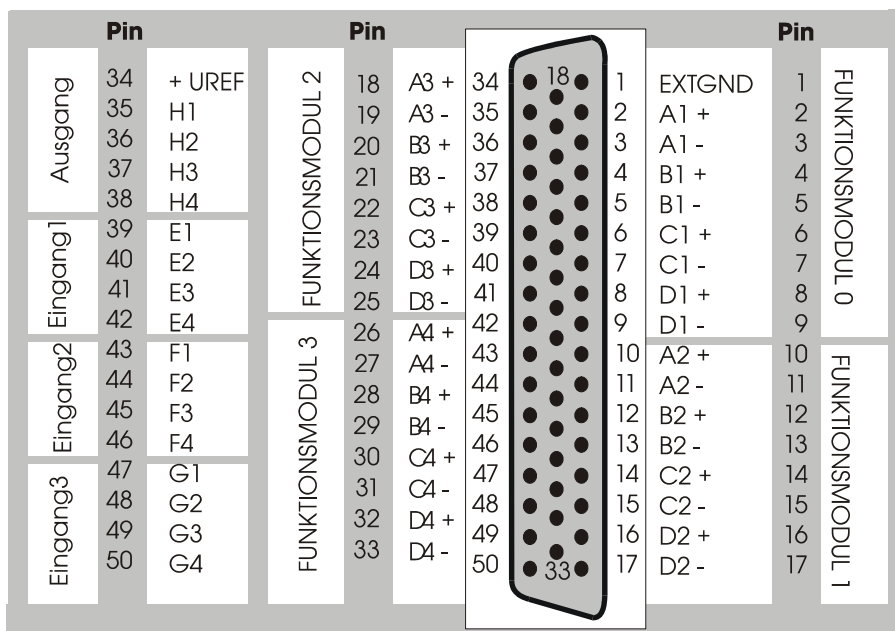
*: nur für die APCI-1710 oder CPCI-1710/1711

x: Nummer des Funktionsmoduls.

2.3 Pinbelegung des Frontsteckers

Die untere Abbildung ist ein Anschlussbeispiel: Die Funktion "Digitale E/A" ist auf allen Funktionsmodulen implementiert.

Abb. 2-2: Pinbelegung des 50-pol. SUB-D Steckers



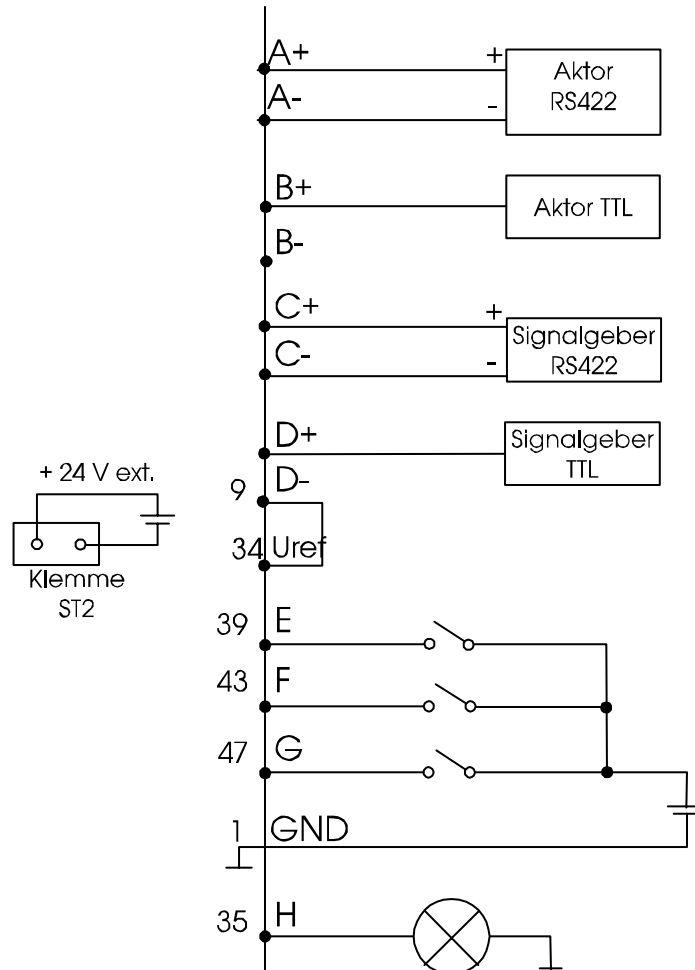
2.4 Anschlussbeispiel

Abb. 2-3: Anschlussbeispiel

Programmierung (Modul 1)

A: Ausgang

B: Eingang



2.5 E/A-Adressbelegung

Tabelle 2-2: E/A-Belegung der "Digitale E/A" Funktion

IORD				
	D31...D24	D23...D16	D15.....D8	D7.....D0
BYTES				
BASE _x + 0	-	-	-	Input-Register
BASE _x + 4	-	-	-	-
.....	-	-	-	-
BASE _x + 60	FUNKNBR2	FUNKNBR1	REVBYTE2	REVBYTE1

IOWR				
	D31...D24	D23...D16	D15.....D8	D7.....D0
BYTES				
BASE _x + 0	-	-	-	Output-Register
BASE _x + 4	-	-	-	Select-Register
.....	-	-	-	-
BASE _x + 60	-	-	-	-

-: keine Funktion ; y: keine relevanten Daten , x: Nummer des Funktionsmoduls.

Die Zugriffe werden immer in 32-Bit breite gelesen oder geschrieben.

2.6 Beschreibung der E/A-Funktionen

2.6.1 Select-Register

Basisadresse + 4:

8-Bit Register; setzt die Polarität der Ein-/Ausgangskanäle A und B.

Dieses Register kann nur geschrieben werden. Nach einem Reset wird der Wert auf "0" gesetzt, d.h. die Leitung sind als Eingangskanal gesetzt

Bit	Logischer Wert	Bedeutung
BIT D0	0	Die I/O Leitungen A sind als Eingangskanal gesetzt
	1	Die I/O Leitungen A sind als Ausgangskanal gesetzt
BIT D1	0	Die I/O Leitungen B sind als Eingangskanal gesetzt
	1	Die I/O Leitungen B sind als Ausgangskanal gesetzt

2.6.2 Output-Register

Basisadresse + 0:

8-Bit Register zur Steuerung der digitalen Ausgänge.

Dieses Register kann nur geschrieben werden. Nach einem Reset wird der Wert auf "0" gesetzt, d.h. die Ausgänge sind auf Low gesetzt.

Bit	Logischer Wert	Bedeutung
BIT D0	0	Ausgang H ist auf "Low" gesetzt
	1	Ausgang H ist auf "High" gesetzt
BIT D1	0	Ausgang A ist auf "Low" gesetzt (differentiell/digital)
	1	Ausgang A ist auf "High" gesetzt (differentiell/digital)
BIT D2	0	Ausgang B ist auf "Low" gesetzt (differentiell/digital)
	1	Ausgang B ist auf "High" gesetzt (differentiell/digital)

2.6.3 Input-Register

Basisadresse + 0:

8-Bit Register zum Lesen der digitalen Eingänge. Dieses Register kann nur gelesen werden und gibt Information über den Status der digitalen Eingänge.

Bit	Logischer Wert	Bedeutung
BIT D0	0	Eingang C ist auf "Low" gesetzt (diff.)
	1	Eingang C ist auf "High" gesetzt (diff.)
BIT D1	0	Eingang D ist auf "Low" gesetzt (diff.)
	1	Eingang D ist auf "High" gesetzt (diff.)
BIT D2	0	Eingang E ist auf "High" gesetzt (>17 V)
	1	Eingang E ist auf "Low" gesetzt oder Eingangsspannung ist <14 V
BIT D3	0	Eingang F ist auf "High" gesetzt (>17 V)
	1	Eingang F ist auf "Low" gesetzt oder Eingangsspannung ist <14 V
BIT D4	0	Eingang G ist auf "High" gesetzt (>17 V)
	1	Eingang G ist auf "Low" gesetzt oder Eingangsspannung ist <14 V
BIT D5	0	Eingang A ist auf "Low" gesetzt (diff.)
	1	Eingang A ist auf "High" gesetzt (diff.)
BIT D6	0	Eingang A ist auf "Low" gesetzt (diff.)
	1	Eingang A ist auf "High" gesetzt (diff.)

2.6.4 Erkennungs-REGISTER (Base + 60)

Funktionsbezeichnung und Revision. (Lesebefehl, ASCII Format)

BASE + 60 "D" "I" "1" "1"

Bedeutet: Digitale E/A Revision 1.1

2.7 Arbeiten mit der "Digitale E/A"-Funktion

1. Signale anschließen.
2. A und B Kanäle als Eingang oder Ausgang initialisieren.
3. Ausgänge steuern.
4. Eingänge lesen.

3 STANDARDSOFTWARE

3.1 Einleitung



WICHTIG!

Merken Sie sich die folgenden Schriftweisen im Text:

Funktion: "i_APCI1710_SetBoardInformation"

Variable *ui_Address*

Tabelle 3-1: Define-Wert

Define name	Decimal value	Hexadecimal value
DLL_COMPILER_C	1	1
DLL_COMPILER_VB	2	2
DLL_COMPILER_PASCAL	3	3
DLL_LABVIEW	4	4

3.2 Softwarefunktionen

3.2.1 Initialisierung

1) i_APCI1710_InitDigitalIO (...)

Syntax:

```
<Return value> = i_APCI1710_InitDigitalIO
                    (BYTE b_BoardHandle,
                    BYTE b_ModulNbr,
                    BYTE b_ChannelAMode,
                    BYTE b_ChannelBMode)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_ChannelAMode	Modus des Kanals A 0: Kanal wird als digitaler Eingang verwendet 1: Kanal wird als digitaler Ausgang verwendet
BYTE	b_ChannelBMode	Modus des Kanals B 0: Kanal wird als digitaler Eingang verwendet 1: Kanal wird als digitaler Ausgang verwendet

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Konfiguriert den Betriebsmode der digitale Ein-/Ausgangskanäle vom ausgewählten Modul (*b_ModulNbr*).

Die Funktion ist als erste aufzurufen bevor eine andere Funktionen aufgerufen wird, die auf die digitalen E/A Kanäle zugreift.

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
```

```
i_ReturnValue = i_APCI1710_InitDigitalIO
                (b_BoardHandle,
                0,
                0,
                1);
```

Return Wert:

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Modulauswahl ist falsch

-3: Das ausgewählte Modul ist kein "Digital I/O" Modul.

-4: Konfiguration des A Kanals ist falsch

-5: Konfiguration des B Kanals ist falsch

3.2.2 Digitale E/A lesen

1) i_APCI1710_ReadDigitalIOChIValue (...)

Syntax:

```
<Return value> = i_APCI1710_ReadDigitalIOChIValue
                    (BYTE b_BoardHandle,
                     BYTE b_ModulNbr,
                     BYTE b_InputChannel,
                     PBYTE pb_ChannelStatus)
```

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_InputChannel	Auswahl des digitalen Eingangs (0 bis 6)
		0: Kanal C
		1: Kanal D
		2: Kanal E
		3: Kanal F
		4: Kanal G
		5: Kanal A
		6: Kanal B

- Ausgabe:

PBYTE	pb_ChannelStatus	Status des digitalen Eingangs.
		0: Kanal ist nicht aktiv
		1: Kanal ist aktiv

Aufgabe:

Gibt den Status des ausgewählten digitalen Eingangs (*b_InputChannel*) vom ausgewählten Module (*b_ModulNbr*).

Funktionsaufruf:

```
ANSI C:
int          i_ReturnValue;
unsigned char b_BoardHandle;
unsigned char b_ChannelStatus;
i_ReturnValue = i_APCI1710_ReadDigitalIOChIValue
                (b_BoardHandle,
                 0,
                 0,
                 &b_ChannelStatus);
```

Return Wert:

0: Kein Fehler
-1: Der Handle-Parameter der Karte ist falsch
-2: Modulauswahl ist falsch
-3: Das ausgewählte Modul ist kein "Digital I/O" Modul.
-4: Der ausgewählte digitale Eingang ist falsch
-5: Digital I/O Funktion nicht initialisiert. Siehe "i_APCI1710_InitDigitalIO".
-6: Kanal A ist als digitaler Ausgang geschaltet
-7: Kanal B ist als digitaler Ausgang geschaltet.

2) i_APCI1710_ReadDigitalIOPortValue (...)

Syntax:

```
<Return value> = i_APCI1710_ReadDigitalIOPortValue  
                (BYTE b_BoardHandle,  
                BYTE b_ModulNbr,  
                PBYTE pb_PortValue)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)

- Ausgabe:

PBYTE	pb_PortValue	Status des digitalen Eingangsports
-------	--------------	------------------------------------

Aufgabe:

Gibt den Status des digitalen Eingangsports vom ausgewählten Modul (*b_ModulNbr*).

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;  
unsigned char b_BoardHandle;  
unsigned char b_PortValue;
```

```
i_ReturnValue = i_APCI1710_ReadDigitalIOPortValue  
                (b_BoardHandle,  
                0,  
                &b_PortValue);
```

Return Wert:

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Modulauswahl ist falsch

-3: Das ausgewählte Modul ist kein "Digital I/O" Modul.

-4: Digital I/O Funktion nicht initialisiert. Siehe Funktion "i_APCI1710_InitDigitalIO".

3.2.3 Digitale Ausgänge schreiben

1) i_APCI1710_SetDigitalIOMemoryOn (...)

Syntax:

```
<Return value> = i_APCI1710_SetDigitalIOMemoryOn  
                (BYTE    b_BoardHandle  
                BYTE    b_ModulNbr)
```

Parameter**- Eingabe:**

BYTE	b_BoardHandle	Handle-Parameter der APCI1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Aktiviert den digitalen Ausgangsspeicher. Nach dem Aufruf der Funktion, werden die Ausgänge, die mit der Funktion "i_APCI1710_SetDigitalIOOutputXOn" gesetzt wurden, nicht zurückgesetzt. Sie können diese mit der Funktion "_APCI1710_SetDigitalIOOutputXOff" zurücksetzen.

Funktionsaufruf:ANSI C:

```
int          i_ReturnValue;  
unsigned char b_BoardHandle;
```

```
i_ReturnValue = i_APCI1710_SetDigitalIOMemoryOn (b_BoardHandle, 0);
```

Return Wert:

- 0: Kein Fehler.
- 1: Der Handle-Parameter der Karte ist falsch.
- 2: Modulauswahl ist falsch
- 3: Das ausgewählte Modul ist kein "Digital I/O" Modul.
- 4: Digital I/O Funktion nicht initialisiert. Siehe Funktion "i_APCI1710_InitDigitalIO".

2) i_APCI1710_SetDigitalIOMemoryOff (...)

Syntax:

```
<Return value> = i_APCI1710_SetDigitalIOMemoryOff  
                (BYTE b_BoardHandle  
                BYTE b_ModulNbr)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Deaktiviert den digitalen Ausgangsspeicher.

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;  
unsigned char b_BoardHandle;
```

```
i_ReturnValue = i_APCI1710_SetDigitalIOMemoryOff (b_BoardHandle, 0);
```

Return Wert:

- 0: Kein Fehler.
- 1: Der Handle-Parameter der Karte ist falsch.
- 2: Modulauswahl ist falsch
- 3: Das ausgewählte Modul ist kein "Digital I/O" Modul.
- 4: Digital I/O Funktion nicht initialisiert. Siehe Funktion "i_APCI1710_InitDigitalIO".

3) i_APCI1710_SetDigitalIOChlOn (...)

Syntax:

```
<Return value> = i_APCI1710_SetDigitalIOChlOn
                    (BYTE   b_BoardHandle,
                    BYTE   b_ModulNbr,
                    BYTE   b_OutputChannel)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_OutputChannel	Auswahl des digitalen Ausgangs (0 bis 2) 0: Kanal H 1: Kanal A 2: Kanal B

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Setzt den Ausgang, der im Parameter *b_Channel* eingegeben ist.
Einen Ausgang setzen bedeutet den Ausgang auf High setzen

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
```

```
i_ReturnValue = i_APCI1710_SetDigitalIOChlOn
                (b_BoardHandle,
                0,
                0);
```

Return Wert:

0: Kein Fehler
-1: Der Handle-Parameter der Karte ist falsch
-2: Modulauswahl ist falsch
-3: Das ausgewählte Modul ist kein "Digital I/O" Modul.
-4: Der ausgewählte digitale Ausgang ist falsch
-5: Digital I/O Funktion nicht initialisiert. Siehe Funktion "i_APCI1710_InitDigitalIO".
-6: Das digitale Kanal A wird als Eingang verwendet.
-7: Das digitale Kanal B wird als Eingang verwendet.

4) i_APCI1710_SetDigitalIOChlOff (...)

Syntax:

```
<Return value> = i_APCI1710_SetDigitalIOChlOff
                    (BYTE b_BoardHandle,
                     BYTE b_ModulNbr,
                     BYTE          b_OutputChannel)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_OutputChannel	Selection of the digital output channel (0 to 2)
		0: Kanal H
		1: Kanal A
		2: Kanal B

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Setzt den Ausgang zurück, der im Parameter *b_Channel* eingegeben ist. Zurücksetzen bedeutet den Ausgang auf LOW setzen.

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
```

```
i_ReturnValue = i_APCI1710_SetDigitalIOChlOff
                (b_BoardHandle,
                 0,
                 0);
```

Return Wert:

- 0: Kein Fehler
- 1: Der Handle-Parameter der Karte ist falsch
- 2: Modulauswahl ist falsch
- 3: Das ausgewählte Modul ist kein "Digital I/O" Modul.
- 4: Der ausgewählte digitale Ausgang ist falsch
- 5: Digital I/O Funktion nicht initialisiert. Siehe Funktion "i_APCI1710_InitDigitalIO".
- 6: Das digitale Kanal A wird als Eingang verwendet.
- 7: Das digitale Kanal B wird als Eingang verwendet.
- 8: Digitaler Ausgangsspeicher ist ausgeschaltet. Rufen Sie zuerst die Funktion "i_APCI1710_SetDigitalIOMemoryOn" off.

5) i_APCI1710_SetDigitalIOPortOn (...)

Syntax:

```
<Return value> = i_APCI1710_SetDigitalIOPortOn
                    (BYTE  b_BoardHandle,
                     BYTE   b_ModulNbr,
                     BYTE   b_PortValue)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_PortValue	Ausgangswert (0 bis 7)

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Setzt einen oder mehrere Ausgangsports. Einen Ausgang setzen bedeutet einen Ausgang auf High setzen. Wenn Sie den digitalen Ausgangsspeicher ausgeschaltet haben (OFF), werden alle anderen Ausgänge auf "0" gesetzt.

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
```

```
i_ReturnValue = i_APCI1710_SetDigitalIOPortOn
                (b_BoardHandle,
                 0,
                 5);
```

Return Wert:

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Modulauswahl ist falsch

-3: Das ausgewählte Modul ist kein "Digital I/O" Modul.

-4: Ausgangswert falsch

-5: Digital I/O Funktion nicht initialisiert. Siehe Funktion "i_APCI1710_InitDigitalIO".

-6: Das digitale Kanal A wird als Eingang verwendet.

-7: Das digitale Kanal B wird als Eingang verwendet.

6) i_APCI1710_SetDigitalIOPortOff (...)

Syntax:

```
<Return Wert> = i_APCI1710_SetDigitalIOPortOff
                    (BYTE      b_BoardHandle,
                    BYTE      b_ModulNbr,
                    BYTE      b_PortValue)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle-Parameter der APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_PortValue	Ausgangswert (0 to 7)

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Setzt einen oder mehrere Ausgangsports zurück (auf LOW).

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
```

```
i_ReturnValue = i_APCI1710_SetDigitalIOPortOff
                (b_BoardHandle,
                0,
                5);
```

Return Wert:

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Modulauswahl ist falsch

-3: Das ausgewählte Modul ist kein "Digital I/O" Modul.

-4: Ausgangswert falsch

-5: Digital I/O Funktion nicht initialisiert. Siehe Funktion "i_APCI1710_InitDigitalIO".

-6: Das digitale Kanal A wird als Eingang verwendet.

-7: Das digitale Kanal B wird als Eingang verwendet.

-8: Digitaler Ausgangsspeicher ist ausgeschaltet. Rufen Sie zuerst die Funktion "i_APCI1710_SetDigitalIOMemoryOn".

3.2.4 Funktionen im Kernel-Mode benutzen

i

WICHTIG!

Diese Funktionen stehen nur für die Benutzer Interruptroutine unter Windows NT und Windows 95/98 im synchronen Mode zur Verfügung. Siehe Funktion "i_APCI1710_SetBoardIntRoutineWin32".

Eingänge lesen

1) i_APCI1710_KRNL_ReadDigitalIOChIValue (...)

Syntax:

```
<Return value> = i_APCI1710_KRNL_ReadDigitalIOChIValue
                    (UINT    ui_BaseAddress,
                     BYTE    b_ModulNbr,
                     BYTE    b_InputChannel,
                     PBYTE   pb_ChannelStatus)
```

Parameter

- Eingabe:

UINT	ui_BaseAddress	Basisadresse der Karte
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_InputChannel	Auswahl des digitalen Eingangs (0 bis 6)

- Ausgabe:

PBYTE	pb_ChannelStatus	Status des digitalen Eingangs
		0: Kanal deaktiviert
		1: Kanal aktiviert.

Aufgabe:

Gibt den Status des ausgewählten digitalen Eingangskanal zurück (*b_InputChannel*) für das angegebene Modul (*b_ModulNbr*).

Funktionsaufruf:

ANSI C:

```
int    i_ReturnValue;
unsigned int  ui_BaseAddress;
unsigned char b_ChannelStatus;
```

```
i_ReturnValue = i_APCI1710_KRNL_ReadDigitalIOChIValue
                (ui_BaseAddress,
                 0,
                 0,
                 &b_ChannelStatus);
```

Return Wert:

0: Kein Fehler
-1: Modulauswahl ist falsch
-2: Das ausgewählte Modul ist kein "Digital I/O" Modul.
-3: The selected digital input Kanal is wrong
-4: Kanal A ist als Ausgang geschaltet
-5: Kanal B ist als digitaler Ausgang geschaltet.

2) i_APCI1710_KRNL_ReadDigitalIOPortValue (...)**Syntax:**

```
<Return value> = i_APCI1710_KRNL_ReadDigitalIOPortValue
                    (UINT          ui_BaseAddress,
                     BYTE          b_ModulNbr,
                     PBYTE pb_PortValue)
```

Parameter**- Eingabe:**

UINT	ui_BaseAddress	Basisadresse der Karte
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)

- Ausgabe:

PBYTE	pb_PortValue	Status des digitalen Eingangsports.
-------	--------------	-------------------------------------

Aufgabe:

Gibt den Status des ausgewählten digitalen Eingangsports für das angegebene Modul (*b_ModulNbr*) zurück.

Funktionsaufruf:ANSI C:

```
int          i_ReturnValue;
unsigned int ui_BaseAddress;
unsigned char b_PortValue;
```

```
i_ReturnValue = i_APCI1710_KRNL_ReadDigitalIOPortValue
                (ui_BaseAddress,
                 0,
                 &b_PortValue);
```

Return Wert:

0: Kein Fehler
-1: Modulauswahl ist falsch
-2: Das ausgewählte Modul ist kein "Digital I/O" Modul.

Ausgänge schreiben

3) i_APCI1710_KRNL_SetDigitalIOChlOn (...)

Syntax:

```
<Return value> = i_APCI1710_KRNL_SetDigitalIOChlOn
                                     (UINT      ui_BaseAddress,
                                      BYTE      b_ModulNbr,
                                      BYTE      b_OutputChannel)
```

Parameter

- Eingabe:

UINT	ui_BaseAddress	Basisadresse der Karte
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_OutputChannel	Selection of the digital output Kanal (0 to 2) 0: Kanal H 1: Kanal A 2: Kanal B

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Setzt den Ausgang, der im Parameter *b_Channel* eingegeben ist. Einen Ausgang setzen bedeutet den Ausgang auf High setzen

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned int ui_BaseAddress;
```

```
i_ReturnValue = i_APCI1710_KRNL_SetDigitalIOChlOn
                (ui_BaseAddress,
                 0,
                 0);
```

Return Wert:

0: Kein Fehler
 -1: Modulauswahl ist falsch
 -2: Das ausgewählte Modul ist kein "Digital I/O" Modul.
 -3: Der ausgewählte digitale Ausgang ist falsch
 -4: Das digitale Kanal A wird als Eingang verwendet.
 -5: Das digitale Kanal B wird als Eingang verwendet.

4) i_APCI1710_KRNL_SetDigitalIOPortOn (...)**Syntax:**

```
<Return value> = i_APCI1710_KRNL_SetDigitalIOPortOn
                                (UINT          ui_BaseAddress,
                                BYTE          b_ModulNbr,
                                BYTE          b_PortValue)
```

Parameter**- Eingabe:**

UINT	ui_BaseAddress	Basisadresse der Karte
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_PortValue	Ausgangswert (0 to 7)

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Setzt einen oder mehrere Ausgangsports. Setzen bedeutet den Ausgang auf High setzen
Alle andere Ausgänge werden auf "0" gesetzt.

Funktionsaufruf:ANSI C:

```
int    i_ReturnValue;
unsigned int  ui_BaseAddress;
```

```
i_ReturnValue = i_APCI1710_KRNL_SetDigitalIOPortOn
                (ui_BaseAddress,
                0,
                5);
```

Return Wert:

0: Kein Fehler
-1: Modulauswahl ist falsch
-2: Das ausgewählte Modul ist kein "Digital I/O" Modul.
-3: Ausgangswert falsch
-4: Das digitale Kanal A wird als Eingang verwendet.
-5: Das digitale Kanal B wird als Eingang verwendet.