



**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

Edge Time measurement (ETM)

Ausgabe: 02.01-08/2005

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 unsachgemäßen 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
1.5	Schriftvereinbarung.....	7
2	ETM.....	8
2.1	Allgemeine Beschreibung	8
2.1.1	Blockdiagramm der ETM Funktion	8
2.1.2	Typische Anwendungen.....	8
2.2	Benutzte Signale.....	9
2.3	Pinbelegung des Frontsteckers.....	9
2.4	Anschlussbeispiel.....	10
2.5	E/A-Adressbelegung	11
2.6	Beschreibung der E/A-Funktionen	12
2.6.1	Teilerfaktor Register (Base+0)	12
2.6.2	Modul Register (Kommando/Status; Base+4)	12
2.6.3	Kommando Register ETM Zähler 0 (Base+8).....	12
2.6.4	Status Register ETM Zähler 0 (Base +20)	13
2.6.5	Kommando Register ETM Zähler 0 (Base+24).....	13
2.6.6	Version Register (Base +60)	13
2.7	Arbeiten mit der ETM Funktion	13
3	STANDARDSOFTWARE	14
3.1	Einleitung.....	14
3.2	Interruptmaske	15
3.3	ETM-Initialisierung	17
	1) i_APCI1710_InitETM (...)	17
	2) i_APCI1710_EnableETM (...).....	19
	3) i_APCI1710_DisableETM (...).....	21
	4) i_APCI1710_GetETMInitialisation (...).....	22
3.4	ETM lesen.....	24
	1) i_APCI1710_GetETMProgressStatus (...)	24
	2) i_APCI1710_ReadETMValue (...)	25
	3) i_APCI1710_ReadETMTotalTime	27
	4) i_APCI1710_ConvertETMValue (...)	29
3.5	Funktionen in Kernel-Mode	31
	1) i_APCI1710_KRNL_ReadETMValue (...)	31

Abbildungen

Abb. 2-1: ETM Blockschaltbild	8
Abb. 2-2: Pinbelegung des 50-pol. SUB-D Steckers	9
Abb. 2-3: Anschlussbeispiel.....	10

Tabellen

Tabelle 1-1: Mitgelieferte Funktionshandbücher	7
Tabelle 2-1: Benutzte Signale	9
Tabelle 2-2: E/A-Belegung der ETM Funktion.....	11
Tabelle 3-1: Define-Wert.....	14
Tabelle 3-2: Interruptmaske der Funktion "ETM".....	15
Tabelle 3-3: Rückgabetable für den Zählerwert	16
Tabelle 3-4: Wert der Zeitbasis.....	18

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_IO_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.dpf	Edge time measurement	etm.cfg

Bitte beachten:

Die Karte **CPCI-1710/1711** ist mit der Karte **APCI-1710** kompatibel, was die Softwareinstallation anbelangt. 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.

1.5 Schriftvereinbarung

Die Signale auf dem 50poligen SUB-D Stecker sind alle auf ein Funktionsmodul bezogen. Bitte beachten Sie die folgenden Schriftvereinbarungen:

- UAS: Störungssignal
- CLK: Takt
- REF: Referenzpunkt-Logik
- ENA: Enable

C1+ ist ein Signal für das **Funktionsmodul 1**.

2 ETM

2.1 Allgemeine Beschreibung

Die Funktion "ETM" ist eine Timer-Schnittstelle, die es erlaubt, die Zeit einer Periode und gleichzeitig die High oder Low-Pegelzeit dieser Periode zu messen.

2 Funktionen sind implementiert:

- 1 x 32-Bit Timer, um eine Referenzzeit zu bilden
- 2 x 32-Bit Messtimer, die die Periodenzeit und die Zeit der High oder Low-Pegel messen.

Eigenschaften:

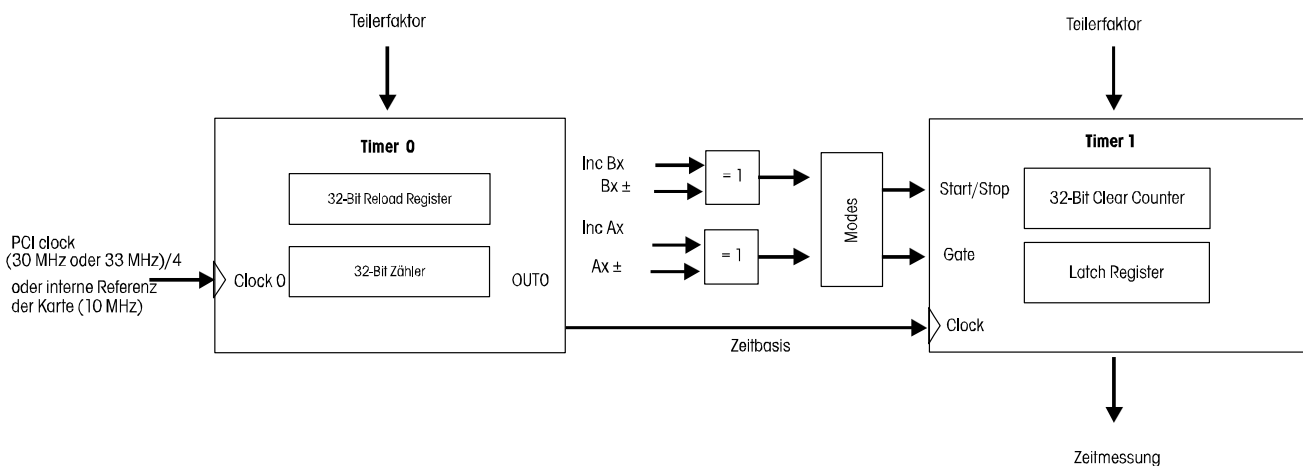
- Zur Vermeidung von Erdschleifen wird eine komplette galvanische Trennung durch Optokoppler für die Ein-/Ausgänge herangezogen.
- Interruptmöglichkeit beim Messende
- Signale bis zu 5 MHz können verarbeitet werden.
- Timer rücklesbar
- Eingänge und Ausgänge können per Software invertiert werden.
- Software GATE möglich

2.1.1 Blockdiagramm der ETM Funktion

Die Schnittstelle enthält:

- 1 Gate-Eingang
- 2 voneinander unabhängige 32-Bit Timer, die über den Datenbus ausgelesen bzw. beschrieben werden können.

Abb. 2-1: ETM Blockschaltbild



2.1.2 Typische Anwendungen

- Periodendauermessung
- Pegeldauermessung

2.2 Benutzte Signale

Die Funktion ETM belegt **4 Eingänge** (A bis D) von dem entsprechenden Funktionsmodul der **APCI-/CPCI-1710**.

Auf einer Karte können Sie maximal 8 ETM (2 pro Modul) nutzen.

Tabelle 2-1: Benutzte Signale

AM STECKER	POLARITÄT	FUNKTION
Ax +/-	Diff./TTL/optional 24 V	Gate-Eingang des ETM Zähler 0
Bx +/-	Diff./TTL/optional 24 V	Eingang des ETM Zähler 0
Cx +/-	Diff./TTL/optional 24 V	Gate des ETM Zähler 1
Dx +/-	Diff./TTL/optional 24 V	Eingang des ETM Zähler 1

2.3 Pinbelegung des Frontsteckers

i

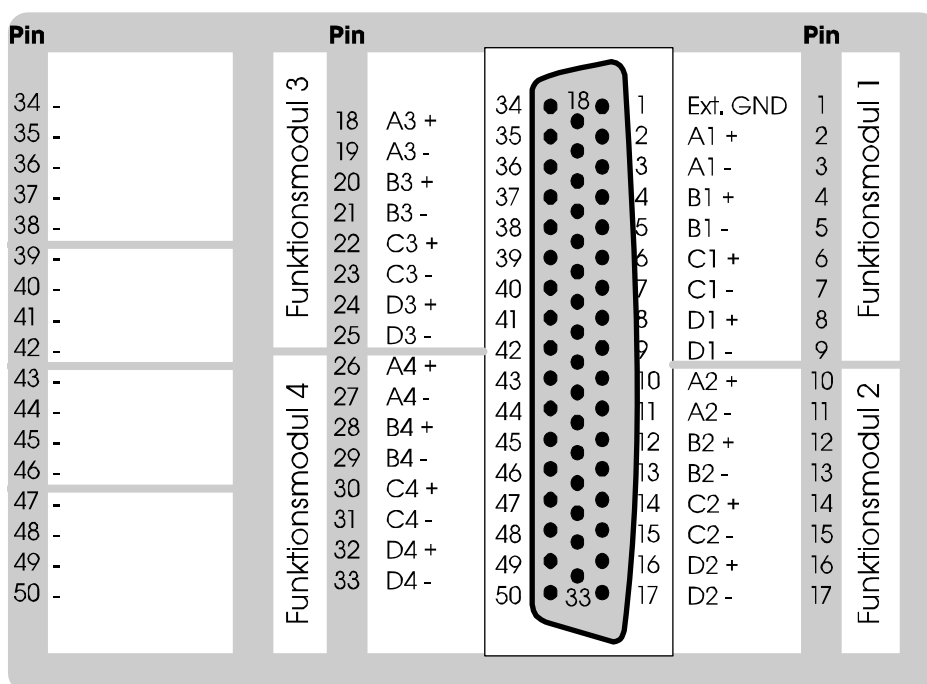
WICHTIG!

Die Funktionsmodule weisen unterschiedliche Bezeichnungen in der Hardware- bzw. Software-Beschreibungen auf.

Für die Steckerbelegung (Hardware) werden die Module von 1 bis 4 nummeriert. Für das SET1710 Programm oder die Softwarefunktionen (Software) **BEGINNT** die Modulnummerierung mit 0.

Die untere Abbildung ist ein Anschlussbeispiel: Die Funktion "ETM" ist auf allen Funktionsmodulen implementiert.

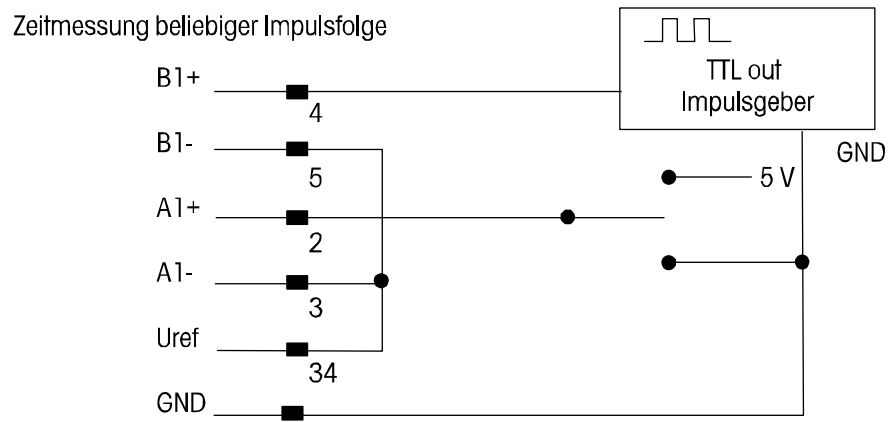
Abb. 2-2: Pinbelegung des 50-pol. SUB-D Steckers



-: nicht belegt

2.4 Anschlussbeispiel

Abb. 2-3: Anschlussbeispiel



2.5 E/A-Adressbelegung

Tabelle 2-2: E/A-Belegung der ETM Funktion

			D31...D24	D23...D16	D15.....D8	D7.....D0
BYTES	Rd	Wr				
BASE _x + 0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Teilerfaktor		
BASE _x + 4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				Kommando/Status Modulx
BASE _x + 8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				Kommando ETM Zähler 0
BASE _x + 12	<input checked="" type="checkbox"/>			Flankenzeitauswertung ETM Zähler 0		
BASE _x + 16	<input checked="" type="checkbox"/>			Gesamtzeit Zähler 0		
BASE _x + 20	<input checked="" type="checkbox"/>					Status ETM Zähler 0
BASE _x + 24	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				Kommando ETM Zähler 1
BASE _x + 28	<input checked="" type="checkbox"/>			Flankenzeitauswertung ETM Zähler 1		
BASE _x + 32	<input checked="" type="checkbox"/>			Gesamtzeit Zähler 1		
...			-	-	-	-
BASE _x + 60	<input checked="" type="checkbox"/>		FUNKNBR2	FUNKNBR1	REVBYTE2	REVBYTE1

-: keine Funktion; **x**: Nummer des Funktionsmoduls.

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

2.6 Beschreibung der E/A-Funktionen

Die ETM Funktion enthält 2 Zeitmesser. Jedem Zeitmesser stehen zwei Signale zur Verfügung:

- das Gate-Signal (A oder C Kanal)
- das zu messende Signal (B oder D Kanal)

Beim Trigger-Signal wird der Zeitmesser gelatcht, auf 0 zurückgesetzt und eine neue Dauermessung wird gestartet.

Die Zeitbasis des Zeitmessers kann per Software definiert werden. Sie ist aber gleich für beide Zähler. Diese Zeit liegt zwischen 25 ns und 0.4s. Als Zeitbasis wird der 40 MHz Quarz der Karte benutzt.

2.6.1 Teilerfaktor Register (Base+0)

Teilerfaktor für die Zeitbasis

2.6.2 Modul Register (Kommando/Status; Base+4)

D0:	0	PCI-Bus Clock wird als Zeitbasis benutzt.
	1	Interner Quarz wird als Zeitbasis benutzt (40MHz).
D1:	0	Teilungsfaktor für die Zeitbasis noch nicht initialisiert.
	1	Teilungsfaktor für die Zeitbasis initialisiert.
D2:	0	Kein Interrupt vorhanden auf dem ersten ETM Zähler
	1	Interrupt vorhanden auf dem ersten ETM Zähler
D3:	0	Kein Interrupt vorhanden auf dem zweiten ETM Zähler
	1	Interrupt vorhanden auf dem zweiten ETM Zähler
D4:	0	Kein Softwarereset erzeugen
	1	Softwarereset auf alle 3 ETM Zähler erzeugen

2.6.3 Kommando Register ETM Zähler 0 (Base+8)

D0:	0	Misst die Low-Zeit
	1	Misst die High-Zeit
D1:	0	Der Low-Pegel startet/beendet die Messung
	1	Der High-Pegel startet/beendet die Messung
D2:	0	Nur eine Messung wird gestartet
	1	Zyklische Messung
D3:	0	Die Messung wird gleich nach der Startfunktion gestartet
	1	Die Messung wird nach dem nächsten Triggersignal gestartet
D4:	0	Es erfolgt kein Interrupt nach der Messung
	1	Ein Interrupt wird nach jeder Messung erzeugt
D5:	0	Messung nicht starten
	1	Messung starten

2.6.4 Status Register ETM Zähler 0 (Base + 20)

D0:	0 = Messung noch nicht gestartet
	1 = Messung gestartet (Triggersignal vorhanden)
D1:	0 = Messung noch nicht beendet
	1 = Messung beendet (Triggersignal vorhanden)
D2:	0 = Kein Übergang
	1 = Übergang

2.6.5 Kommando Register ETM Zähler 0 (Base+ 24)

D0:	0	Misst die Low-Zeit
	1	Misst die High-Zeit
D1:	0	Der Low-Pegel startet/beendet die Messung
	1	Der High-Pegel startet/beendet die Messung
D2:	0	Nur eine Messung wird gestartet
	1	Zyklische Messung
D3:	0	Die Messung wird gleich nach der Startfunktion gestartet
	1	Die Messung wird nach dem nächsten Triggersignal gestartet
D4:	0	Es erfolgt kein Interrupt nach der Messung
	1	Ein Interrupt wird nach jeder Messung erzeugt
D5:	0	Messung nicht starten
	1	Messung starten

2.6.6 Version Register (Base + 60)

Enthält die Funktionsbezeichnung und die Revision. (Lesebefehl, ASCII Format)

BASE + 60 "E" "T" "1" "0"

Bedeutet: ETM, Revision 1.0

2.7 Arbeiten mit der ETM Funktion

1. Anschluss des Signalgebers an die Karte
2. Parametrierung der API Funktion (Signalpegel-Auswahl, Zeitreferenz, Single oder Continuous Mode)
3. Status der Messung per Polling oder Interrupt auswerten
4. Zeitmessungs-Timer auslesen
5. Wert aus dem Zeitmessungs-Timer und Zeitreferenz ergibt den Pegel und Periodenzeit

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
APCI1710_30MHZ	30	1E
APCI1710_33MHZ	33	21
APCI1710_40MHZ	40	28

3.2 Interruptmaske

Jeder ETM-Zähler kann einen Interrupt generieren. Um diesen Interrupt zu bekommen, sollen Sie den Interrupt aktivieren und die Interruptroutine mit der Funktion "i_APCI1710_SetBoardIntRoutineX" Funktion.

Tabelle 3-2: Interruptmaske der Funktion "ETM"

b_ModuleMask	ul_InterruptMask	Bedeutung
0000 0001	00010 0000 0000 0000 0000	Interrupt auf ETM Zähler 0, Modul 0
0000 0001	00100 0000 0000 0000 0000	Interrupt auf ETM Zähler 1, Modul 0
0000 0010	00010 0000 0000 0000 0000	Interrupt auf ETM Zähler 0, Modul 1
0000 0010	00100 0000 0000 0000 0000	Interrupt auf ETM Zähler 1, Modul 1
0000 0100	00010 0000 0000 0000 0000	Interrupt auf ETM Zähler 0, Modul 2
0000 0100	00100 0000 0000 0000 0000	Interrupt auf ETM Zähler 1, Modul 2
0000 1000	00010 0000 0000 0000 0000	Interrupt auf ETM Zähler 0, Modul 3
0000 1000	00100 0000 0000 0000 0000	Interrupt auf ETM Zähler 1, Modul 3

Tabelle 3-3: Rückgabefabelle für den Zählerwert

b_ModuleMask	ul_InterruptMask	Quelle	ul_CounterLatchValue
b_ModuleMask = 1	ul_InterruptMask = 20000Hex	Interrupt durch ETM Zähler 0 ausgelöst Modul 0	Gemessene Flankenzeit (24-Bit)
b_ModuleMask = 1	ul_InterruptMask = 40000Hex	Interrupt durch ETM Zähler 1 ausgelöst Modul 0	Gemessene Flankenzeit (24-Bit)
b_ModuleMask = 2	ul_InterruptMask = 20000Hex	Interrupt durch ETM Zähler 0 ausgelöst Modul 1	Gemessene Flankenzeit (24-Bit)
b_ModuleMask = 2	ul_InterruptMask = 40000Hex	Interrupt durch ETM Zähler 1 ausgelöst Modul 1	Gemessene Flankenzeit (24-Bit)
b_ModuleMask = 4	ul_InterruptMask = 20000Hex	Interrupt durch ETM Zähler 0 ausgelöst Modul 2	Gemessene Flankenzeit (24-Bit)
b_ModuleMask = 4	ul_InterruptMask = 40000Hex	Interrupt durch ETM Zähler 1 ausgelöst Modul 2	Gemessene Flankenzeit (24-Bit)
b_ModuleMask = 8	ul_InterruptMask = 20000Hex	Interrupt durch ETM Zähler 0 ausgelöst Modul 3	Gemessene Flankenzeit (24-Bit)
b_ModuleMask = 8	ul_InterruptMask = 40000Hex	Interrupt durch ETM Zähler 1 ausgelöst Modul 3	Gemessene Flankenzeit (24-Bit)

3.3 ETM-Initialisierung

1) i_APCI1710_InitETM (...)

Syntax:

<Return Wert> = i_APCI1710_InitETM
 (BYTE b_BoardHandle,
 BYTE b_ModulNbr,
 BYTE b_ClockSelection,
 BYTE b_TimingUnit,
 ULONG ul_Timing,
 PULONG pul_RealTiming)

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle der APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_ClockSelection	Auswahl des PCI Bus-Takts - APCI1710_30MHZ: Die Karte benutzt einen PCI Bus Takt von 30 MHz - APCI1710_33MHZ: Die Karte benutzt einen PCI Bus Takt von 33 MHz - APCI1710_40MHZ: Die Karte benutzt den 40 MHz Quarz Takt.
BYTE	b_TimingUnit	Einheit der Zeitbasis (0 bis 2) 0: ns 1: µs 2: ms
ULONG	ul_Timing	Wert der Zeitbasis. Siehe Tabelle "Wert des Zeitbasis"

- Ausgabe:

PULONG	pul_RealTiming	Richtiger Wert der Zeitbasis. Gibt den Wert zurück, der dem in ul_Timing eingegebenen Wert so genau wie möglich entspricht.
--------	----------------	---

Tabelle 3-4: Wert der Zeitbasis

PCI Bus-Takt	<i>b_TimingUnit</i>	<i>ul_Timing</i> Minimum-Wert	<i>ul_Timing</i> Maximum-Wert
APCI1710_30MHz	ns (0)	33	559240500
	µs (1)	1	559240
	ms (2)	1	559
APCI1710_33MHz	ns (0)	30	508400454
	µs (1)	1	508400
	ms (2)	1	508
APCI1710_40MHz	ns (0)	25	419430375
	µs (1)	1	419430
	ms (2)	1	419

Aufgabe:

Konfiguriert alle ETM Zähler des ausgewählten Moduls (*b_ModulNbr*). Die Parameter *ul_Timing* und *ul_TimingUnit* legen die Zeitbasis für die Messung fest. *pul_RealTiming* gibt den richtigen Zeitwert zurück.

Diese Funktion soll aufgerufen werden, bevor Sie eine andere Funktion aufrufen, die auf den ETM Zähler zugreift.

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
unsigned long ul_RealTiming;
```

```
i_ReturnValue = i_APCI1710_InitETM (b_BoardHandle,
                                     0,
                                     APCI1710_40MHZ,
                                     1,
                                     100,
                                     &ul_RealTiming);
```

Return-Wert:

0: Kein Fehler

-1: Handle Parameter der Karte ist falsch

-2: Die ausgewählte Modulnummer ist falsch.

-3: Das ausgewählte Modul ist kein "ETM"-Modul.

-4: Der ausgewählte Eingangstakt ist falsch.

-5: Die ausgewählte Zeiteinheit ist falsch.

-6: Die ausgewählte Zeitbasis ist falsch.

-7: Der 40MHz Takt kann nicht auf Ihrer Karte konfiguriert werden.

2) i_APCI1710_EnableETM (...)

Syntax:

```
<Return Wert> = i_APCI1710_EnableETM
                    (BYTE      b_BoardHandle,
                    BYTE      b_ModulNbr,
                    BYTE      b_ETM,
                    BYTE      b_EdgeLevel,
                    BYTE      b_TriggerLevel,
                    BYTE      b_CycleMode,
                    BYTE      b_FirstTriggerMode,
                    BYTE      b_InterruptEnable)
```

Parameter:

-Eingabe

BYTE	b_BoardHandle	Handle der APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 to 3)
BYTE	b_ETM	Auswahl des ETM Zählers (0 oder 1)
BYTE	b_EdgeLevel	Flankenauswahl zu Zeitmessung 0: Misst die Low-Pegelzeit 1: Misst die High-Pegelzeit
BYTE	b_TriggerLevel	Auswahl des Trigger-Pegels 0: Trigger beim Low-Pegel 1: Trigger beim High-Pegel
BYTE	b_CycleMode	Modeauswahl 0: Einzel-Modus 1: Continuous Mode. Jeder Trigger stoppt die Messung und startet einen neuen Zyklus.
BYTE	b_FirstTriggerMode	Modus des ersten Triggers 0: Die Messung der ersten Flankenzeit startet nachdem die Funktion "i_APCI1710_EnableETM" aufgerufen ist. 1: Die Messung der ersten Flankenzeit startet nach dem nächsten Trigger-Signal.
BYTE	b_InterruptEnable	Aktiviert oder deaktiviert die Interruptfunktion. Ein Interrupt erfolgt nach dem Trigger APCI1710_DISABLE: Interrupt deaktiviert wurde.

-Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Aktiviert den ETM Zähler des ausgewählten Moduls (*b_ModulNbr*). Die Funktion "i_APCI1710_InitETM" soll als erste aufgerufen werden. Wird der Interrupt aktiviert, generiert der ETM Zähler einen Interrupt nach jedem Trigger-Signal. Siehe Funktion "i_APCI1710_SetBoardIntRoutineXX" und Tabelle 3-4.

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;

i_ReturnValue = i_APCI1710_EnableETM
                (b_BoardHandle,
                 0,
                 0,
                 0,
                 0,
                 APCI1710_DISABLE);
```

Return Wert:

0: Kein Fehler

- 1: Handle Parameter der Karte ist falsch
- 2: Die ausgewählte Modulnummer ist falsch.
- 3: Das ausgewählte Modul ist kein "ETM"-Modul.
- 4: Ausgewählter ETM Zähler ist falsch
- 5: ETM ist nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM"
- 6: Auswahl des Flankenpegels ist falsch
- 7: Auswahl des Trigger-Pegels ist falsch.
- 8: Mode-Auswahl ist falsch.
- 9: Mode-Auswahl für den ersten Trigger ist falsch.
- 10: Interrupt-Parameter ist falsch
- 11: Interrupt-Funktion ist nicht initialisiert.
Siehe Funktion "i_APCI1710_SetBoardIntRoutineXX".

3) i_APCI1710_DisableETM (...)**Syntax:**

```
<Return Wert> = i_APCI1710_DisableETM
                                     (BYTE      b_BoardHandle,
                                     BYTE      b_ModulNbr
                                     BYTE      b_ETM)
```

Parameter:**- Eingabe:**

BYTE	b_BoardHandle	Handle der APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_ETM	Auswahl des ETM Zähler (0 oder 1)

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Deaktiviert den ETM Zähler des ausgewählten Moduls (*b_ModulNbr*).

Funktionsaufruf:ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;

i_ReturnValue = i_APCI1710_DisableETM
               (b_BoardHandle,
                0,
                0);
```

Return-Wert

0: Kein Fehler

-1: Handle Parameter der Karte ist falsch

-2: Die ausgewählte Modulnummer ist falsch.

-3: Das ausgewählte Modul ist kein "ETM"-Modul.

-4: ETM nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM".

4) i_APCI1710_GetETMInitialisation (...)**Syntax:**

```

<Return Wert> = i_APCI1710_GetETMInitialisation
                                (BYTE      b_BoardHandle,
                                BYTE      b_ModulNbr,
                                BYTE      b_ETM,
                                PBYTE     pb_TimingUnit,
                                PULONG    pul_Timing,
                                PBYTE     pb_EdgeLevel,
                                PBYTE     pb_TriggerLevel,
                                PBYTE     pb_CycleMode,
                                PBYTE     pb_FirstTriggerMode,
                                PBYTE     pb_InterruptEnable,
                                PBYTE     pb_Enable)

```

Parameter:**- Eingabe:**

BYTE	b_BoardHandle	Handle der APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_ETM	Auswahl des ETM Zähler (0 oder 1)

- Ausgabe:

PBYTE	pb_TimingUnit	Einheit der Zeitbasis (0 bis 2) 0: ns 1: µs 2: ms
PULONG	pul_Timing	Wert der Zeitbasis. Siehe Tabelle "Wert des Zeitbasis"
PBYTE	pb_EdgeLevel	Flankenauswahl zu Zeitmessung 0: Misst die Low-Pegelzeit 1: Misst die High-Pegelzeit
PBYTE	pb_TriggerLevel	Auswahl des Trigger-Pegels 0: Trigger beim Low-Pegel 1: Trigger beim High-Pegel
PBYTE	pb_CycleMode	Modeauswahl 0: Einzel-Modus 1: Continuous Mode. Jeder Trigger stoppt die Messung and startet einen neuen Zyklus
PBYTE	pb_FirstTriggerMode	Modus des ersten Triggers 0: Die Messung der ersten Flankenzeit startet nachdem die Funktion "i_APCI1710_EnableETM" aufgerufen ist 1: Die Messung der ersten Flankenzeit startet nach dem nächsten Trigger-Signal
PBYTE	pb_InterruptEnable	Aktiviert oder deaktiviert die Interruptfunktion. APCI1710_ENABLE: Interrupt aktiviert. Ein Interrupt erfolgt nach dem Trigger APCI1710_DISABLE: Interrupt deaktiviert

PBYTE pb_Enable Gibt zurück, ob die ETM Funktion aktiviert ist
oder nicht
0: ETM nicht aktiviert
1: ETM aktiviert

Aufgabe:

Gibt die Information über die ETM (*b_ETM*) Initialisierung des ausgewählten Moduls (*b_ModulNbr*) zurück.

Rufen Sie die Funktion "i_APCI1710_InitETM" zuerst auf, bevor Sie eine andere Funktion aufrufen.

Funktionsaufruf:

ANSI C:

```
unsigned char  b_TimingUnit;  
unsigned long  ul_Timing;  
unsigned char  b_EdgeLevel;  
unsigned char  b_TriggerMode;  
unsigned char  b_InterruptEnable;  
unsigned char  b_Enable;
```

```
i_ReturnValue = i_APCI1710_GetETMInitialisation  
                (b_BoardHandle,  
                 0,  
                 0,  
                 &b_TimingUnit,  
                 &ul_Timing,  
                 &b_EdgeLevel,  
                 &b_TriggerMode,  
                 &b_InterruptEnable,  
                 &b_Enable);
```

Return Wert:

- 0: Kein Fehler
- 1: Handle Parameter der Karte ist falsch
- 2: Die ausgewählte Modulnummer ist falsch.
- 3: Das ausgewählte Modul ist kein "ETM"-Modul.
- 4: Auswahl des ETM Zählers ist falsch.
- 5: ETM nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM".

3.4 ETM lesen

1) i_APCI1710_GetETMProgressStatus (...)

Syntax:

```
<Return Wert> = i_APCI1710_GetETMProgressStatus
                    (BYTE          b_BoardHandle,
                     BYTE          b_ModulNbr,
                     BYTE          b_ETM
                     PBYTE         pb_ETMStatus)
```

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle der APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_ETM	Auswahl des ETM Zähler (0 oder 1)

- Ausgabe:

PULONG	pb_ETMStatus	Rückgabe des ETM-Status. 0: Messung nicht gestartet. Kein Start-Trigger ist eingetroffen. 1: Messung gestartet. Ein Start-Trigger ist eingetroffen. 2: Messung gestoppt. Ein Stop-Trigger ist eingetroffen. Die Messung ist beendet. 3: Übergang ist erfolgt. Bitte die Zeitbasis mit der Funktion "i_APCI1710_InitETM" ändern
--------	--------------	--

Aufgabe:

Gibt den ETM Status (*pb_ETMStatus*) des ausgewählten Moduls (*b_ModulNbr*) zurück.

Funktionsaufruf:

ANSI C:

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

```
i_ReturnValue = i_APCI1710_GetETMProgressStatus
                (b_BoardHandle,
                 0,
                 0,
                 &pb_ETMStatus);
```

Return-Wert

0: Kein Fehler
-1: Handle Parameter der Karte ist falsch
-2: Die ausgewählte Modulnummer ist falsch.
-3: Das ausgewählte Modul ist kein "ETM"-Modul.
-4: Ausgewählter ETM Zähler ist falsch
-5: ETM nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM".

2) i_APCI1710_ReadETMValue (...)

Syntax:

<Return Wert> = i_APCI1710_ReadETMValue
 (BYTE b_BoardHandle,
 BYTE b_ModulNbr,
 BYTE b_ETM
 UINT ui_TimeOut,
 PBYTE pb_ETMStatus,
 PULONG pul_ETMValue)

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle der APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_ETM	Auswahl des ETM Zähler (0 oder 1)
UINT	ui_TimeOut	Auswahl des Timeouts (0 bis 65535) 0: Timeout nicht benutzt. Die Funktion gibt den ETM-Status zurück und misst den Zählerwert, wenn ein Stop-Signal eingetroffen ist. 1 bis 65535: legt das Timeout in ms fest. Die Funktion wird nach einem Timeout oder einem Stop-Signal gestoppt.

- Ausgabe:

PBYTE	pb_ETMStatus	Rückgabe des ETM-Status. 0: Messung nicht gestartet. Kein Start-Trigger ist eingetroffen. 1: Messung gestartet. Ein Start-Trigger ist eingetroffen. 2: Messung gestoppt. Ein Stop-Trigger ist eingetroffen. Die Messung ist beendet und <i>pul_ETMValue</i> gibt die ETM-Zeit zurück. 3: Übergang ist erfolgt. Bitte die Zeitbasis mit der Funktion "i_APCI1710_InitETM" ändern 4: Timeout ist erfolgt.
PULONG	pul_ETMValue	ETM-Zeitwert.

Aufgabe:

Rückgabe des ETM-Status (*pb_ETMStatus*) und des Zeitwerts (*pul_ETMValue*) nach einem Stop-Signal auf das ausgewählte Modul (*b_ModulNbr*). Diese Funktion ist nur verfügbar, wenn Sie die Interrupt-Funktion deaktiviert haben. Siehe Funktion "i_APCI1710_EnableETM" und Tabelle 3-4. Der ETM-Status kann mit der Funktion "i_APCI1710_GetETMProgressStatus" getestet werden.

Der durch *pul_ETMValue* zurückgegebene Wert ist nicht der richtige Zeitwert. Benutzen Sie die "i_APCI1710_ConvertETMValue" Funktion.

Sonst gilt das folgende Formel, um den richtigen Zeitwert auszurechnen:

$Zeitwert = pul_ETMValue \times pul_RealTiming.$

pul_RealTiming ist der rückgegebene Wert von "i_APCI1710_InitETM". Die Zeiteinheit ist durch die Variable *b_TimingUnit* von der Funktion "i_APCI1710_InitETM".

Funktionsaufruf:ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
unsigned char b_ETMStatus;
unsigned long ul_ETMValue;

i_ReturnValue = i_APCI1710_ReadETMValue
                (b_BoardHandle,
                 0,
                 0,
                 0
                 &pb_ETMStatus,
                 &ul_ETMValue);
```

Return-Wert:

0: Kein Fehler

-1: Handle Parameter der Karte ist falsch

-2: Die ausgewählte Modulnummer ist falsch.

-3: Das ausgewählte Modul ist kein "ETM"-Modul.

-4: Ausgewählter ETM Zähler ist falsch

-5: ETM nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM".

-6: Der Timeout Parameter ist falsch (0 bis 65535).

-7: Interruptroutine installiert. Die gemessene ETM-Zeit kann nicht direkt gelesen werden.

3) i_APCI1710_ReadETMTotalTime

Syntax:

<Return Wert> = i_APCI1710_ReadETMTotalTime

(BYTE_ b_BoardHandle,
 BYTE_ b_ModulNbr,
 BYTE_ b_ETM,
 UINT_ ui_TimeOut,
 PBYTE_ pb_ETMtatus,
 PULONG_ pul_ETMValue)

Parameter:

-Eingabe:

BYTE	b_BoardHandle	:Handle der APCI-1710
BYTE	b_ModulNbr	:Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_ETM	:Auswahl des ETM Zähler (0 oder 1)
UINT	ui_TimeOut	:Auswahl des Timeouts (0 bis 65535)

Wenn Sie diese Funktion der Interruptfunktion verwenden, wird dieser Parameter nicht verwendet.

0: Timeout nicht benutzt. Die Funktion gibt den ETM-Status zurück und misst den Zählerwert, wenn ein Stoppsignal eingetroffen ist.

1 bis 65535:
 Legt das Timeout in ms fest. Die Funktion kehrt nach einem Timeout oder nach einem Stoppsignal zurück

-Ausgabe:

PULONG pul_ETMValue : ETM Gesamtzeitwert.

Aufgabe:

Gibt den ETM-Status (pb_ETMtatus) und den Gesamtzeitwert value (pul_ETMValue) zurück nachdem ein Stoppsignal auf dem ausgewählten ETM-Modul (b_ModulNbr) eingetroffen ist. Diese Funktion ist nur verfügbar, wenn Sie die Interruptfunktion deaktiviert haben. Siehe Funktion "i_APCI1710_EnableETM" und Tabelle 3-1. Sie können den ETM-Status mit "i_APCI1710_GetETMProgressStatus" testen. Der von pul_ETMValue zurückgegebene Wert entspricht nicht dem tatsächlich gemessenen Zeitwert. Sie müssen die Funktion "i_APCI1710_ConvertETMValue" verwenden oder diesen Vorgang durchführen, um den Zeitwert zu berechnen:

Zeitwert = pul_ETMValue x pul_RealTiming.
 pul_RealTiming ist der von "i_APCI1710_InitETM" zurückgegebene Parameter und die Zeiteinheit ist b_TimingUnit von der Funktion "i_APCI1710_InitETM".

Return-Wert:

- 0: Kein Fehler
- 1: Handle Parameter der Karte ist falsch
- 2: Ausgewählte Modulnummer ist falsch
- 3: Ausgewähltes Modul ist kein ETM-Modul
- 4: Ausgewählter ETM Zähler ist falsch
- 5: ETM nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM"
- 6: Timeout Parameter ist falsch

4) i_APCI1710_ConvertETMValue (...)**Syntax:**

```
<Return Wert> = i_APCI1710_ConvertETMValue
                                (BYTE      b_BoardHandle,
                                BYTE      b_ModulNbr,
                                ULONG     ul_ETMValue,
                                PULONG    pul_Hour,
                                PBYTE     pb_Minute,
                                PBYTE     pb_Second,
                                PUINT     pui_MilliSecond,
                                PUINT     pui_MicroSecond,
                                PUINT     pui_NanoSecond)
```

Parameter:**- Eingabe:**

BYTE	b_BoardHandle	Handle der APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
ULONG	ul_ETMValue	ETM-Zeitwert. Siehe "i_APCI1710_ReadETMValue"

- Ausgabe:

PULONG	pul_Hour	Zeitmessung in Stunden
PBYTE	pb_Minute	Zeitmessung in Minuten
PBYTE	pb_Second	Zeitmessung in Sekunden
PUINT	pui_MilliSecond	Zeitmessung in Millisekunden
PUINT	pui_MicroSecond	Zeitmessung in Mikrosekunden
PUINT	pui_NanoSecond	Zeitmessung in Nanosekunden

Aufgabe:

Konvertierung der gemessenen ETM Zeit (*ul_ETMValue*) in h, mn, s, ms, μ s und ns.

Funktionsaufruf:ANSI C:

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
unsigned int  ui_MilliSecond;
unsigned int  ui_MicroSecond;
unsigned int  ui_NanoSecond;
unsigned char b_Second;
unsigned char b_Minute;
i_ReturnValue = i_APCI1710_ConvertETMValue
                (b_BoardHandle,
                0,
                0,
                &b_Minute,
                &b_Second,
                &ui_MilliSecond,
                &ui_MicroSecond,
                &ui_NanoSecond);
```

Return-Wert

0: Kein Fehler

-1: Handle Parameter der Karte ist falsch

-2: Die ausgewählte Modulnummer ist falsch.

-3: Das ausgewählte Modul ist kein "ETM"-Modul.

-4: ETM nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM".

3.5 Funktionen in Kernel-Mode

i

WICHTIG!

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

1) i_APCI1710_KRNL_ReadETMValue (...)

Syntax:

```
<Return Wert> = i_APCI1710_KRNL_ReadETMValue
                                (UINT          ui_BaseAddress,
                                BYTE           b_ModulNbr,
                                BYTE           b_ETM
                                PBYTE         pb_ETMStatus,
                                PULONG        pul_ETMValue)
```

Parameter:

- Eingabe:

UINT	ui_BaseAddress	Basisadresse der APCI-/CPCI-1710 Karte
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_ETM	Auswahl des ETM Zähler (0 oder 1)

- Ausgabe:

PBYTE	pb_ETMStatus	Rückgabe des ETM-Status. 0: Messung nicht gestartet. Kein Start-Trigger ist eingetroffen. 1: Messung gestartet. Ein Start-Trigger ist eingetroffen. 2: Messung gestoppt. Ein Stop-Trigger ist eingetroffen. Die Messung ist beendet und <i>pul_ETMValue</i> gibt die ETM-Zeit zurück. 3: Übergang ist erfolgt. Bitte die Zeitbasis mit der Funktion "i_APCI1710_InitETM" ändern
PULONG	pul_ETMValue	ETM-Zeitwert.

Aufgabe:

Rückgabe des ETM-Status (*pb_ETMStatus*) und des Zeitwerts (*pul_ETMValue*) nach einem Stop-Signal auf das ausgewählte Modul (*b_ModulNbr*). Diese Funktion ist nur verfügbar, wenn Sie die Interrupt-Funktion deaktiviert haben. Siehe Funktion "i_APCI1710_EnableETM" und Tabelle 3-4. Der ETM-Status kann mit der "i_APCI1710_KRNL_GetETMProgressStatus" Funktion getestet werden.

Der durch *pul_ETMValue* zurückgegebene Wert ist nicht der richtige Zeitwert. Benutzen Sie die "i_APCI1710_ConvertETMValue" Funktion. Sonst gilt die folgende Formel, um den richtigen Zeitwert auszurechnen:

$$\text{Zeitwert} = \text{pul_ETMValue} \times \text{pul_RealTiming}.$$
pul_RealTiming ist der rückgegebene Wert von "i_APCI1710_InitETM". die Zeiteinheit ist durch die Variable *b_TimingUnit* von der Funktion "i_APCI1710_InitETM".

Funktionsaufruf:

ANSI C:

```
int          i_ReturnValue;
unsigned int ui_BaseAddress;
unsigned char b_ETMStatus;
unsigned long ul_ETMValue;
```

```
i_ReturnValue = i_APCI1710_KRNL_ReadETMValue
                (ui_BaseAddress,
                 0,
                 &pb_ETMStatus,
                 &ul_ETMValue);
```

Return-Wert

0: Kein Fehler

-1: Die ausgewählte Modulnummer ist falsch.

-2: Das ausgewählte Modul ist kein "ETM"-Modul.

-3: Ausgewählter ETM Zähler ist falsch

-4: ETM nicht initialisiert. Siehe Funktion "i_APCI1710_InitETM".

-5: Interruptroutine installiert. Die gemessene ETM-Zeit kann nicht direkt gelesen werden.