



**DIN EN ISO 9001:2000
zertifiziert**



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

Funktionsbeschreibung

ADDICOUNT APCI-/CPCI-1710

TOR

2. 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 Wind River Systems Inc.

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	TOR	8
2.1	Funktionsbeschreibung	8
2.1.1	Blockdiagramm	8
2.1.2	Typische Anwendungen.....	8
2.2	Benutzte Signale.....	9
2.3	Steckerbelegung für alle Module mit TOR-Funktion.....	9
2.4	Anschlussbeispiel.....	10
2.5	E/A Adressbelegung.....	11
2.5.1	Schreibregister	11
2.5.2	Leseregister	11
2.6	Beschreibung der E/A-Funktionen	12
2.6.1	Funktionsbeschreibung	12
2.6.2	Timer1 REGISTER	12
2.6.3	Timer0 Register.....	12
2.6.4	TOR COMMANDO Register	13
2.6.5	TOR GATE Register	13
2.6.6	TOR SYNCHRONISATION GATE Register.....	14
2.6.7	TOR STATUS Register	14
2.6.8	TOR INTERRUPT STATUS Register.....	15
2.6.9	Versions-Register (Base +60)	15
2.7	Arbeiten mit der TOR Funktion	15
3	SOFTWARE-FUNKTIONEN	16
3.1	Einleitung.....	16
3.2	Interruptmaske	17

3.3	Initialisierung	19
1)	i_APCI1710_InitTorCounter (...)	19
2)	i_APCI1710_EnableTorCounter (...)	22
3)	i_APCI1710_DisableTorCounter (...)	24
4)	i_APCI1710_GetTorCounterInitialisation (...)	25
3.4	TOR Zähler lesen.....	27
1)	i_APCI1710_GetTorCounterProgressStatus (...)	27
2)	i_APCI1710_ReadTorCounterValue (...)	28
3.5	Interrupt-Kernelfunktionen für Windows NT/95	30
1)	i_APCI1710_KRNL_GetTorCounterProgressStatus (...)	30
2)	i_APCI1710_KRNL_ReadTorCounterValue (...)	32

Abbildungen

Abb. 2-1:	Blockdiagramm der TOR Funktion	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 (Schreibregister)	11
Tabelle 3-1:	Define-Wert.....	16
Tabelle 3-2:	Interruptmaske der Funktion TOR	17
Tabelle 3-3:	Rückgabe-Tabelle für den Zählerwert	18
Tabelle 3-4:	Basiszeitwert	20

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	DGI_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 TOR

2.1 Funktionsbeschreibung

Die "TOR" Funktion ist eine Zählerschnittstelle zum Zählen von Eingangssignalen innerhalb einer definierten Zeit.

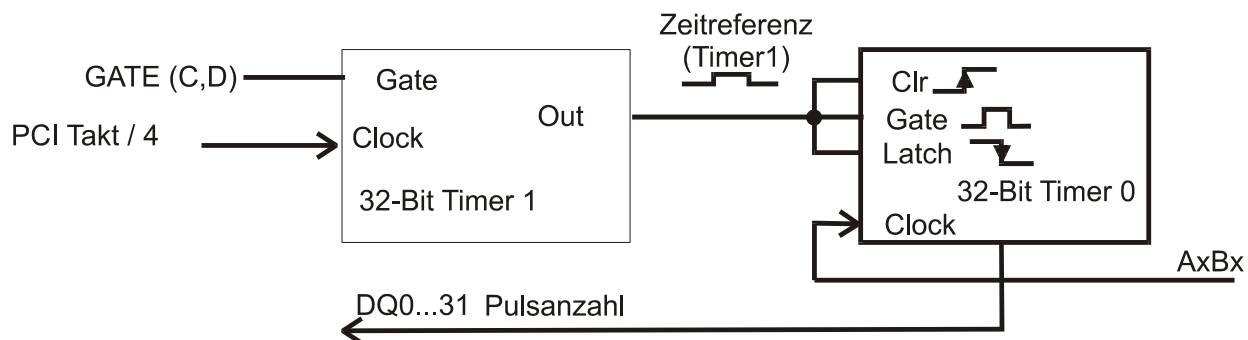
Auf einem Modul sind 2 TOR Zähler vorhanden. Jeder TOR Zähler enthält 2 x 32-Bit Timer.

Eigenschaften

- Zur Vermeidung von Erdschleifen wird eine komplette galvanische Trennung durch Optokoppler für die Ein-/Ausgänge herangezogen.
- Interruptstatus beim Messungsende
- Eingänge und Ausgänge können per Software invertiert werden.
- Software Gate

2.1.1 Blockdiagramm

Abb. 2-1: Blockdiagramm der TOR Funktion



2.1.2 Typische Anwendungen

- Frequenzmessung
- Impulszählung pro Zeitintervall

2.2 Benutzte Signale

Die Funktion "TOR" belegt **4 Eingänge (A bis D)** vom entsprechenden Funktionsmodul der **APCI-/CPCI-1710**.

Tabelle 2-1: Benutzte Signale

AM STECKER	POLARITÄT	FUNKTION
A x +/-	Diff. / TTL	Digitaler Eingang 1 (TOR 1)
B x +/-	Diff. / TTL	Digitaler Eingang 2, (TOR2)
C x +/-	Diff. / TTL / Opt. 24V	Externer Gate (TOR1)
D x +/-	Diff. / TTL / Opt. 24V	Externer Gate (TOR2)

x: Nummer des Funktionsmoduls.

2.3 Steckerbelegung für alle Module mit TOR-Funktion

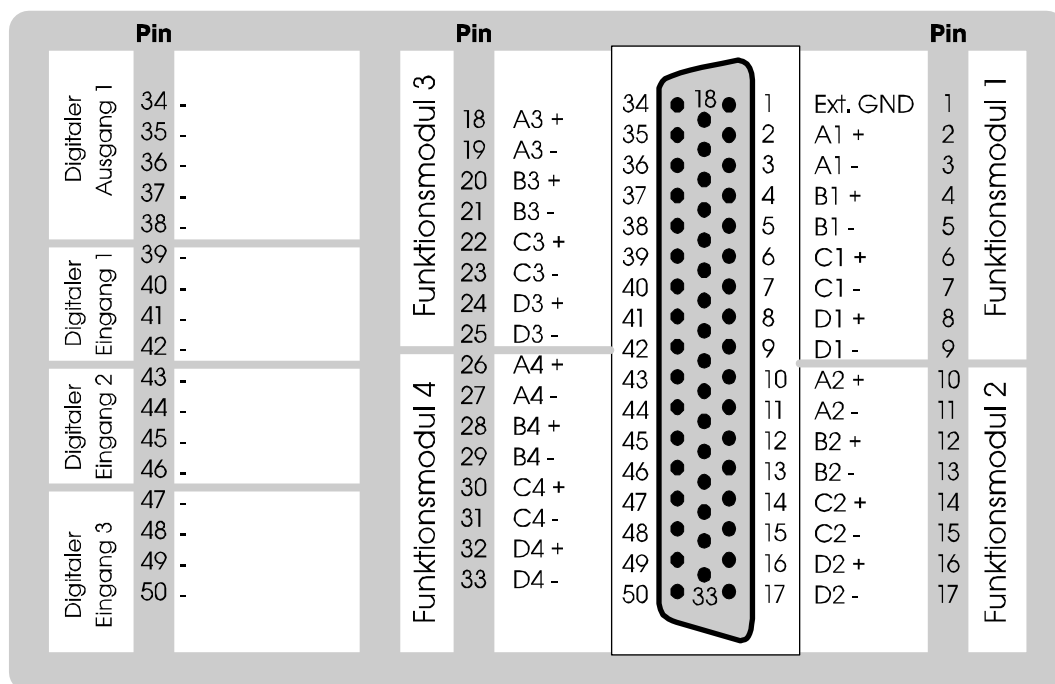
i

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.

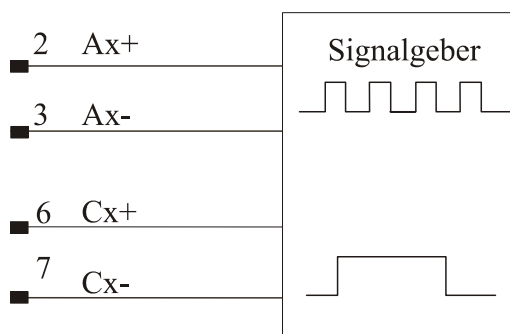
Abb. 2-2: Pinbelegung des 50-pol. SUB-D Steckers



-: Nicht belegt

2.4 Anschlussbeispiel

Abb. 2-3: Anschlussbeispiel



2.5 E/A Adressbelegung

2.5.1 Schreibregister

Tabelle 2-2: E/A-Belegung (Schreibregister)

		D31...D24	D23...D16	D15.....D8	D7.....D0
BYTES	Wr	HIGHBYTE	MIDHIGHBYTE	MIDLOWBYTE	LOWBYTE
BASEx + 0	<input checked="" type="checkbox"/>	TOR1 BASE_TIME			
BASEx + 4	<input checked="" type="checkbox"/>	-	-	-	TOR1 COMMANDO
BASEx + 8	<input checked="" type="checkbox"/>	-	-	-	TOR1 GATE
BASEx + 12	<input checked="" type="checkbox"/>	-	-	-	TOR1/TOR2 SYNCHRO_GATE
BASEx + 16		TOR2 BASE_TIME			
BASEx + 20	<input checked="" type="checkbox"/>	-	-	-	TOR2 COMMANDO
BASEx + 24	<input checked="" type="checkbox"/>	-	-	-	TOR2 GATE
BASEx + 28	<input checked="" type="checkbox"/>	-	-	-	TOR1/TOR2 SYNCHRO_GATE
BASEx + 60		-	-	-	-

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

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

2.5.2 Leseregister

		D31...D24	D23...D16	D15.....D8	D7.....D0
BYTES	Rd	HIGHBYTE	MIDHIGHBYTE	MIDLOWBYTE	LOWBYTE
BASEx + 0	<input checked="" type="checkbox"/>	TOR1 COUNTER_VALUE			
BASEx + 4	<input checked="" type="checkbox"/>	-	-	-	TOR1 STATUS
BASEx + 8	<input checked="" type="checkbox"/>	-	-	-	TOR1 GATE
BASEx + 12	<input checked="" type="checkbox"/>	-	-	-	TOR1 INT_STATUS
BASEx + 16		TOR2 COUNTER_VALUE			
BASEx + 20	<input checked="" type="checkbox"/>	-	-	-	TOR2 STATUS
BASEx + 24	<input checked="" type="checkbox"/>	-	-	-	TOR2 GATE
BASEx + 28	<input checked="" type="checkbox"/>	-	-	-	TOR2 INT_STATUS
BASEx + 60	<input checked="" type="checkbox"/>	FUNKNBR2	FUNKNBR1	REVBYTE2	REVBYTE1

2.6 Beschreibung der E/A-Funktionen

2.6.1 Funktionsbeschreibung

Die Funktion "TOR" ist eine abgemagerte Version der Modulfunktion "Timer/Zähler". Das Pulssignal des Timers 1 gibt das Start- und Stoppsignal an Timer0. Timer 0 zählt die Eingangssignale. Nach dem Stoppsignal vom Timer 0 wird die Anzahl der Impulse gespeichert und kann durch E/A Befehle rückgelesen werden.

Der Timer 1 wird als Zeitreferenz-Generator verwendet.

Der Teilerfaktor wird in Timer 1 gesetzt und legt die Ausgangsfrequenz fest. Die Eingangsfrequenz wird gemäß dem PCI Takt eingestellt. Timer 0 wird mit dem Startevent synchronisiert.

Pulsmessung

Sobald das **Start**-Signal vom Timer 1 erteilt wird, wird der Timer 0 zurückgesetzt und zählt die Pulssignale vom Kanal **Ax(Bx)**.

Beim Ablaufen wird der Status-Bit "Counter in Progress" in Statusregister gesetzt.

Sobald der Timer ein **Stop**-Signal erzeugt hat, wird der Timer 0 gestoppt und der Status-Bit "Counter in Progress" zurückgesetzt.

Ein Interrupt kann auch erzeugt werden. Der Wert kann gelesen werden. Der zuletzt gemessene Wert wird im "Zählermessung" Register gelesen.

2.6.2 Timer1 REGISTER

Basisadresse + 0 für Tor1 und Basisadresse + 16 für Tor2

32-Bit Register: Der "Reload"-Wert für TIMER 1 wird geschrieben. Die Ausgangsfrequenz von Timer 1 wird durch den Teilerfaktor festgelegt.

2.6.3 Timer0 Register

Basisadresse + 0 für Tor1 und Basisadresse + 16 für Tor2

32-Bit Register zum Lesen des aktuellen Wertes der Pulsmessung (d.h. letztes Messungswertes).

2.6.4 TOR COMMANDO Register

Basisadresse + 4: TOR1

Basisadresse + 20: TOR2.

DQ0	0	Einfach-Modus
	1	Continuous-Modus
DQ1	0	Interruptgenerierung deaktiviert
	1	Interruptgenerierung aktiviert
DQ2:	0	Zählen beim "High" Pegel
	1	Invertiertes Bild vom digitalen Eingang, Zählen beim "Low" Pegel
DQ3	0	Externen Eingangsgate nicht verwendet
	1	Externer Gate startet den TOR Zähler

2.6.5 TOR GATE Register

Basisadresse + 8: TOR1

Basisadresse + 24: TOR2.

Schreiben:

DQ0	0	Deaktiviert den Zähler
	1	Aktiviert den Zähler
DQ4	0	Keine Schreibbefehl möglich
	1	Setzt die Initialisierung des TOR Zählers zurück

Lesen:

DQ0	0	Zähler deaktiviert
	1	Zähler aktiviert
DQ4	0	TOR Zähler nicht initialisiert
	1	TOR Zähler initialisiert

2.6.6 TOR SYNCHRONISATION GATE Register

Basisadresse +12: TOR1

Basisadresse +28: TOR2.

DQ0	0	Deaktiviert Zähler von Tor1 und Tor2
	1	TOR Zähler nicht initialisiert Zähler von TOR1 und TOR2
DQ4	0	Keine Befehl möglich
	1	Setzt Initialisierung von TOR1 und TOR2 zurück

2.6.7 TOR STATUS Register

Basisadresse + 4: TOR1

Basisadresse + 20: TOR2.

32-Bit Register: gibt die Statusinformation von "TOR" zurück.

DQ0	0	Messung läuft nicht
	1	Messung läuft. Rücksetzen beim Lesen der Adresse Base+0 für TOR1 und der Adresse Base+16 für TOR2
DQ1	0	Messung nicht beendet
	1	Messung beendet. Rücksetzen beim Lesen der Adresse Base+0 für TOR1 und der Adresse Base+16 für TOR2
DQ2	0	Messungszeit nicht abgelaufen
	1	Messungszeit ist abgelaufen. Rücksetzen beim Lesen der Adresse Base+0 für TOR1 und der Adresse Base+16 für TOR2
DQ4	0	Einfachen Mode ausgewählt
	1	Continuous Mode ausgewählt
DQ5	0	Interrupt-Mode deaktiviert
	1	Interrupt-Mode aktiviert
DQ6	0	Eingangssignal nicht invertiert
	1	Eingangssignal nicht invertiert
DQ7	0	Gate-Eingangssignal nicht verwendet
	1	Gate-Eingangssignal verwendet
Other bits		Beim Lesen auf 0 gesetzt

2.6.8 TOR INTERRUPT STATUS Register

Basisadresse + 12: TOR1

Basisadresse + 28: TOR2.

32-Bit Register: gibt die Interrupt-Status Information der TR Funktion zurück.

DQ0	=1	Interrupt wird beim Messungsende generiert. Die Interruptabfrage wird nach dem Lesen des Registers zurückgesetzt.
DQ31..1		Immer auf 0 gesetzt

2.6.9 Versions-Register (Base + 60)

Die Funktion und die Revision werden erkannt. (Lesebefehl, ASCII Format)

BASE + 60 "T" "O" "1" "3"

Bedeutet: TOR Revision 1.3

2.7 Arbeiten mit der TOR Funktion

1. Signalanschluss am Stecker
2. Initialisieren der TOR Funktion (Auswahl des TOR Zählers, Eingangsfrequenz,...)
3. Freigeben der Messung (Einzel- oder kontinuierliche Messung mit/ohne Interrupt)
4. Status des TOR Zählers lesen und auswerten
5. Messwert des TOR Zählers lesen.

3 SOFTWARE-FUNKTIONEN

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
DLL_COMPILER_VB5	5	5
APCI1710_SINGLE	0	0
APCI1710_CONTINUOUS	1	1
APCI1710_30MHZ	30	1E
APCI1710_33MHZ	33	21
APCI1710_40MHZ	40	28

3.2 Interruptmaske

Jeder TOR 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 TOR

b_ModuleMask	ul_InterruptMask	Bedeutung
0000 000 1	000 1 0000 0000 0000	Interrupt auf TOR Zähler 0 vom Modul 0 ausgelöst
0000 000 1	001 0 0000 0000 0000	Interrupt auf TOR Zähler 1 vom Modul 0 ausgelöst
0000 00 10	000 1 0000 0000 0000	Interrupt auf TOR Zähler 0 vom Modul 1 ausgelöst
0000 00 10	001 0 0000 0000 0000	Interrupt auf TOR Zähler 1 vom Modul 1 ausgelöst
0000 0 100	000 1 0000 0000 0000	Interrupt auf TOR Zähler 0 vom Modul 2 ausgelöst
0000 0 100	001 0 0000 0000 0000	Interrupt auf TOR Zähler 1 vom Modul 2 ausgelöst
0000 1000	000 1 0000 0000 0000	Interrupt auf TOR Zähler 0 vom Modul 3 ausgelöst
0000 1000	001 0 0000 0000 0000	Interrupt auf TOR Zähler 1 vom Modul 3 ausgelöst

Tabelle 3-3: Rückgabe-Tabelle für den Zählerwert

b_ModuleMask	ul_InterruptMask	Quelle	ul_CounterLatchValue
b_ModuleMask = 1	ul_InterruptMask = 4096	Zählzyklus vom Modul 0. TOR Zähler 0 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 0
b_ModuleMask = 1	ul_InterruptMask = 8192	Zählzyklus vom Modul 0. TOR Zähler 1 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 1
b_ModuleMask = 2	ul_InterruptMask = 4096	Zählzyklus vom Modul 1. TOR Zähler 0 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 0
b_ModuleMask = 2	ul_InterruptMask = 8192	Zählzyklus vom Modul 1. TOR Zähler 1 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 1
b_ModuleMask = 4	ul_InterruptMask = 4096	Zählzyklus vom Modul 2. TOR Zähler 0 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 0
b_ModuleMask = 4	ul_InterruptMask = 8192	Zählzyklus vom Modul 2. TOR Zähler 1 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 1
b_ModuleMask = 8	ul_InterruptMask = 4096	Zählzyklus vom Modul 3. TOR Zähler 0 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 0
b_ModuleMask = 8	ul_InterruptMask = 8192	Zählzyklus vom Modul 3. TOR Zähler 1 wird gestoppt. Die Messung wird gelatcht.	Zählerwert des TOR 1

3.3 Initialisierung

1) i_APCI1710_InitTorCounter (...)

Syntax:

<Return Wert> = i_APCI1710_InitTorCounter
 (BYTE b_BoardHandle,
 BYTE b_ModulNbr,
 BYTE b_TorCounter,
 BYTE b_PCIIInputClock,
 BYTE b_TimingUnit,
 ULONG ul_TimingInterval,
 PULONG pul_RealTimingInterval)

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle der Karte APCI-/CPCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1).
BYTE	b_PCIIInputClock	Auswahl des PCI-Bus Takts - APCI1710_30MHZ: Der PC hat einen PCI-Bus Takt von 30 MHz - APCI1710_33MHZ: Der PC hat einen PCI-Bus Takt von 33 MHz - APCI1710_40MHZ: Die XPCI-1710 Karte hat einen integrierten Takt von 40MHz
BYTE	b_TimingUnit	Basiszeiteinheit (0 bis 4) 0: ns 1: µs 2: ms 3: s 4: mn
ULONG	ul_TimingInterval	Basiszeitwert. Siehe Tabelle 3-6: Basiszeitwert

- Ausgabe:

PULONG pul_RealTimingInterval Realer Zeitwert.

Tabelle 3-4: Basiszeitwert

PCI-Bus Takt	b_TimingUnit	ul_TimingInterval Mindestwert	ul_TimingInterval Höchstwert
APCI1710_30MHz	ns (0)	133	4294967295
	µs (1)	1	571230650
	ms (2)	1	571230
	s (3)	1	571
	mn (4)	1	9
APCI1710_33MHz	ns (0)	121	4294967295
	µs (1)	1	519691043
	ms (2)	1	51969
	s (3)	1	520
	mn (4)	1	8
APCI1710_40MHZ	ns (0)	100	4294967295
	µs (1)	1	429496729
	ms (2)	1	429496
	s (3)	1	429
	mn (4)	1	7

Aufgabe:

Konfiguriert den selektierten TOR Zähler (*b_TorCounter*) vom angegebenen Modul (*b_ModulNbr*). *ul_TimingInterval* und *ul_TimingUnit* legen die Zeitbasis für die Messung fest. *pul_RealTimingInterval* gibt den realen Zeitwert zurück. Diese Funktion ist als erste aufzurufen bevor eine andere Funktion aufgerufen wird, welche auf den TOR Zähler zugreift.

Funktionsaufruf

ANSI C :

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

```
i_ReturnValue = i_APCI1710_InitTorCounter
                (b_BoardHandle,
                 0,
                 0,
                 APCI1710_33MHZ,
                 1,
                 100,
                 &ul_RealTimingInterval);
```

Return-Wert

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Das ausgewählte Modul ist falsch

-3: Das Modul ist kein TOR Modul

-4: Selektierter TOR Zähler ist falsch

-5: Der ausgewählte PCI Eingangstakt ist falsch

-6: Selektierte Zeiteinheit ist falsch

-7: Selektierte Basiszeit ist falsch

-8: 40MHz Takt kann nicht mit dieser Karte verwendet werden

-9: 40MHz Takt kann nicht mit dieser TOR Version verwendet werden

2) i_APCI1710_EnableTorCounter (...)**Syntax:**

<Return Wert> = i_APCI1710_EnableTorCounter
 (BYTE b_BoardHandle,
 BYTE b_ModulNbr,
 BYTE b_TorCounter,
 BYTE b_InputMode,
 BYTE b_ExternGate,
 BYTE b_CycleMode,
 BYTE b_InterruptEnable)

Parameter:**- Eingabe:**

BYTE	b_BoardHandle	Handle der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1).
BYTE	b_InputMode	Auswahl des Eingangssignal-Pegels 0: TOR zählt bei jedem "Low"-Pegel 1: TOR zählt bei jedem "High"-Pegel 2: APCI1710_TOR_SIMPLE_MODE: Funktion wie 4fach-MODE, es werden jedoch nur eine der insgesamt vier Flanken pro Periode ausgewertet. 3: APCI1710_TOR_DOUBLE_MODE Funktion wie 4fach-MODE, es werden jedoch nur zwei der insgesamt vier Flanken pro Periode ausgewertet. 4: APCI1710_TOR_QUADRUPLE_MODE: Die Flankenauswerteschaltung generiert im 4fach-MODE aus jeder Flanke zweier zueinander phasenverschobener Signale einen Zählimpuls.
BYTE	b_ExternGate	Selektierte Aktion für den externen Gate 0: Externes Gate-Signal nicht verwendet 1: Externes Gate-Signal verwendet. Falls der Einzelmodus (single mode) eingestellt ist, wird jedes High-Pegel-Signal den Zähler starten. Falls der Continuous-Modus eingestellt ist, wird das erste High-Pegel-Signal den TOR Zähler starten Der Gate-Eingang wird als Signal B verwendet.
BYTE	b_CycleMode	Erfassungsmodus des TOR Zählers
BYTE	b_InterruptEnable	TOR Zähler Interrupt APCI1710_ENABLE: aktiviert den TOR Zähler Interrupt APCI1710_DISABLE: deaktiviert den TOR Zähler Interrupt

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:

Gibt den TOR Zähler (*b_TorCounter*) vom angegebenen Modul (*b_ModulNbr*) frei. Die Funktion "i_APCI1710_InitTorCounter" soll vor dieser Funktion aufgerufen werden. Falls der TOR Zähler Interrupt freigegeben wird, generiert der TOR Zähler einen Interrupt nachdem das Zeitzyklus abgelaufen ist. Siehe Funktion "i_APCI1710_SetBoardIntRoutineX" und die Beschreibung der Interruptmaske im Absatz 3.2. Der *b_CycleMode* Parameter legt fest, ob die Messung ein Zyklus lang oder mehrere dauert.

Funktionsaufruf:

ANSI C :

```
int          i_ReturnValue;
unsigned char b_BoardHandle;

i_ReturnValue = i_APCI1710_EnableTorCounter
                (b_BoardHandle,
                 0,
                 0,
                 1,
                 0
                 APCI1710_SINGLE,
                 APCI1710_DISABLE);
```

Return-Wert:

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Das ausgewählte Modul ist falsch

-3: Das Modul ist kein TOR Modul

-4: Der ausgewählte TOR Zähler ist falsch

-5: TOR Zähler nicht initialisiert. Siehe Funktion "i_APCI1710_InitTorCounter"

-6: Das ausgewählte TOR Eingangssignal ist falsch

-7: Das ausgewählte Mode für das externe Gate-Signal ist falsch

-8: Der ausgewählte Erfassungszyklus des TOR Zählers ist falsch

-9: Interrupt-Parameter ist falsch

-10: Interruptfunktion nicht initialisiert.

Siehe Funktion "i_APCI1710_SetBoardIntRoutineX"

3) i_APCI1710_DisableTorCounter (...)

Syntax:

<Return Wert> = i_APCI1710_DisableTorCounter
(BYTE b_BoardHandle,
BYTE b_ModulNbr,
BYTE b_TorCounter)

Parameters:**- Eingabe:**

BYTE	b_BoardHandle	Handle der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1).

- Ausgabe:

Es erfolgt keine Ausgabe.

Task:

Deaktiviert den TOR Zähler (*b_TorCounter*) vom angegebenen Modul (*b_ModulNbr*). Falls der TOR Zähler deaktiviert wird, nachdem ein Zyklus gestartet hat und falls der TOR Zähler mit der Funktion "i_APCI1710_EnableTorCounter" wieder startet, wird der Status-Register gelöscht.

Funktionsaufruf:

ANSI C :

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

```
i_ReturnValue = i_APCI1710_DisableTorCounter  
                (b_BoardHandle,  
                 0,  
                 0);
```

Return-Wert:

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Das ausgewählte Modul ist falsch

-3: Das Modul ist kein TOR Modul

-4: Der ausgewählte TOR Zähler ist falsch

-5: TOR Zähler nicht initialisiert. Siehe Funktion "i_APCI1710_InitTorCounter"

-6: TOR Zähler nicht freigegeben.

Siehe Funktion "i_APCI1710_EnableTorCounter"

4) i_APCI1710_GetTorCounterInitialisation (...)**Syntax:**

```
<Return Wert> = i_APCI1710_GetTorCounterInitialisation
                (BYTE    b_BoardHandle,
                 BYTE    b_ModulNbr,
                 BYTE    b_TorCounter,
                 PBYTE   pb_TimingUnit,
                 PULONG  pul_TimingInterval,
                 PBYTE   pb_InputMode,
                 PBYTE   pb_ExternGate,
                 PBYTE   pb_CycleMode,
                 PBYTE   pb_Enable,
                 PBYTE   pb_InterruptEnable)
```

Parameter:**- Eingabe:**

BYTE	b_BoardHandle	Handle der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1)

- Ausgabe:

PBYTE	pb_TimingUnit	Base time unit (0 to 4) 0: ns 1: µs 2: ms 3: s 4: mn
PULONG	pul_TimingInterval	Basiszeitwert. Siehe Tabelle 3-4: Basiszeitwert
PBYTE	pb_InputMode	Auswahl des Eingangssignal-Pegels 0: TOR zählt bei jedem "Low"-Pegel 1: TOR zählt bei jedem "High"-Pegel
PBYTE	pb_ExternGate	Selektierte Aktion für den externen Gate 0: Externes Gate-Signal nicht verwendet 1: Externes Gate-Signal verwendet.
PBYTE	pb_CycleMode	Zyklusart der TOR Erfassung. APCI1710_SINGLE: Einzelzyklus-MODE. Der TOR Zähler zählt nur einen Zyklus. Die Funktion ist bei jedem Zyklus aufzurufen. APCI1710_CONTINUOUS: Jeder Zeitzyklus startet einen neuen Zählablauf.
PBYTE	pb_Enable	Zeigt, ob der TOR Zähler freigegeben ist oder nicht 0: TOR Zähler nicht aktiviert 1: TOR Zähler aktiviert
PBYTE	pb_InterruptEnable	Auswahl des TOR Interrupts. APCI1710_ENABLE: TOR Interrupt aktiviert APCI1710_DISABLE: TOR Interrupt deaktiviert

Aufgabe:

Returns the TOR (*b_TorCounter*) initialisation of the selected module (*b_ModulNbr*). You must call up the "i_APCI1710_InitTorCounter" Funktion before calling up this Funktion.

Funktionsaufruf:

ANSI C :

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
unsigned char b_TimingUnit;
unsigned char b_InputMode;
unsigned char b_CycleMode;
unsigned char b_InterruptEnable;
unsigned long ul_TimingInterval;
```

```
i_ReturnValue = i_APCI1710_GetTorCounterInitialisation
                (b_BoardHandle,
                 0,
                 0,
                 &b_TimingUnit,
                 &ul_TimingInterval,
                 &b_InputMode,
                 &b_CycleMode,
                 &b_InterruptEnable);
```

Return-Wert:

0: Kein Fehler

-1: Der Handle-Parameter der Karte ist falsch

-2: Das ausgewählte Modul ist falsch

-3: Das Modul ist kein TOR Modul

-4: Der ausgewählte TOR Zähler ist falsch

-5: TOR nicht initialisiert. Siehe Funktion "i_APCI1710_InitTorCounter"

3.4 TOR Zähler lesen

1) i_APCI1710_GetTorCounterProgressStatus (...)

Syntax:

<Return Wert> = i_APCI1710_GetTorCounterProgressStatus
 (BYTE b_BoardHandle,
 BYTE b_ModulNbr,
 BYTE b_TorCounter,
 PBYTE pb_TorCounterStatus)

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1)

- Ausgabe:

PBYTE	pb_TorCounterStatus	Gibt den TOR Status zurück. 0: Zählzyklus nicht gestartet. Software-Gate nicht gesetzt 1: Zählzyklus gestartet. Software-Gate gesetzt 2: Zählzyklus gestoppt und beendet. 3: Zeit ist abgelaufen. Basiszeit muss mit der Funktion "i_APCI1710_InitTorCounter" geändert werden.
-------	---------------------	--

Aufgabe:

Gibt den Status des selektierten TOR Zählers (*b_TorCounter*) im angegebenen Modul (*b_ModulNbr*).

Funktionsaufruf:

ANSI C :

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

```
i_ReturnValue = i_APCI1710_GetTorCounterProgressStatus
                (b_BoardHandle,
                 0,
                 0,
                 &pb_TorCounterStatus);
```

Return-Wert:

0: Kein Fehler
 -1: Der Handle-Parameter der Karte ist falsch
 -2: Das ausgewählte Modul ist falsch
 -3: Das Modul ist kein TOR Modul
 -4: Der ausgewählte TOR Zähler ist falsch
 -5: TOR nicht initialisiert. Siehe Funktion "i_APCI1710_InitTorCounter"
 -6: TOR nicht freigegeben. Siehe Funktion "i_APCI1710_EnableTorCounter"

2) i_APCI1710_ReadTorCounterValue (...)**Syntax:**

<Return Wert> = i_APCI1710_ReadTorCounterValue
 (BYTE b_BoardHandle,
 b_ModulNbr,
 b_TorCounter,
 ui_TimeOut,
 pb_TorCounterStatus,
 pul_TorCounterValue)

Parameter:**- Eingabe:**

BYTE	b_BoardHandle	Handle der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1)
UINT	ui_TimeOut	Auswahl der Ablaufzeit (0 bis 65535) 0: Kein Zeitablauf verwendet. Die Funktion gibt den TOR Status zurück und prüft ob der Zählzyklus den gemessenen Zählwert gestoppt hat. 1 bis 65535: legt den Zeitablauf in ms fest Die Funktion wird nach Timeout oder Zählzyklusstop rückgegeben.

- Ausgabe:

PBYTE	pb_TorCounterStatus	Gibt den TOR Status zurück 0: Zählzyklus nicht gestartet. Software-Gate nicht gesetzt 1: Zählzyklus gestartet. Software-Gate gesetzt. 2: Zählzyklus gestoppt. The Zählzyklus wird beendet. 3: Es hat sich ein Überlauf ergeben. Die Zeitbasis muss durch die Funktion "i_APCI1710_InitTorCounter" geändert werden 4: Timeout ist aufgetreten
PULONG	pul_TorCounterValue	TOR Zählwert.

Aufgabe:

Gibt den Status (*pb_TorCounterStatus*) des TOR Zählers (*b_TorCounter*) und den Zählwert (*pul_TorCounterValue*) vom angegebenen Modul (*b_ModulNbr*) zurück, nachdem ein Zählzyklus gestoppt wird.

Funktionsaufruf:

ANSI C :

```
int          i_ReturnValue;
unsigned char b_BoardHandle;
unsigned char b_TorCounterStatus;
unsigned long ul_TorCounterValue;

i_ReturnValue = i_APCI1710_ReadTorCounterValue
                (b_BoardHandle,
                 0,
                 0,
                 0,
                 &pb_TorCounterStatus,
                 &ul_TorCounterValue);
```

Return-Wert:

- 0: Kein Fehler
- 1: Der Handle-Parameter der Karte ist falsch
- 2: Das ausgewählte Modul ist falsch
- 3: Das Modul ist kein TOR Modul
- 4: Der ausgewählte TOR Zähler ist falsch
- 5: TOR nicht initialisiert. Siehe Funktion "i_APCI1710_InitTorCounter"
- 6: TOR nicht freigegeben. Siehe Funktion "i_APCI1710_EnableTorCounter"
- 7: Timeout-Parameter ist falsch (0 bis 65535)

3.5 Interrupt-Kernelfunktionen für Windows NT/95



WICHTIG!

Diese Funktionen sind nur für Windows NT und Windows 95 Benutzer-Interruptroutine im synchronen Mode vorhanden. Siehe Funktion "i_APCI1710_SetBoardIntRoutineWin32"

1) i_APCI1710_KRNL_GetTorCounterProgressStatus (...)

Syntax:

```
<Return Wert> = i_APCI1710_KRNL_GetTorCounterProgressStatus
                    (UINT          ui_BaseAddress,
                     BYTE          b_ModulNbr,
                     BYTE          b_TorCounter,
                     PBYTE         pb_TorCounterStatus)
```

Parameter:

- Eingabe:

UINT	ui_BaseAddress	Basisadresse der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1)

- Ausgabe:

PBYTE	pb_TorCounterStatus	Gibt den TOR Status zurück
		0: Zählzyklus nicht gestartet. Software-Gate nicht gesetzt
		1: Zählzyklus gestartet. Software-Gate gesetzt.
		2: Zählzyklus gestoppt. The Zählzyklus wird beendet.
		3: Es hat sich ein Überlauf ergeben. Die Zeitbasis muss durch die Funktion "i_APCI1710_InitTorCounter" geändert werden
		4: Timeout ist aufgetreten

Task:

Returns the TOR status (*pb_TorCounterStatus*) of the selected TOR module (*b_ModulNbr*).

Funktionsaufruf:

ANSI C :

```
int          i_ReturnValue;
unsigned int  ui_BaseAddress;
unsigned char b_TorCounterStatus;
i_ReturnValue = i_APCI1710_KRNL_GetTorCounterProgressStatus
                (ui_BaseAddress,
                 0,
                 0,
                 &pb_TorCounterStatus);
```

Return Wert:

0: Kein Fehler

-1: Das ausgewählte Modul ist falsch

-2: Das Modul ist kein TOR Modul

-3: Der ausgewählte TOR Zähler ist falsch

-4: TOR nicht initialisiert. Siehe Funktion "i_APCI1710_InitTorCounter"

-5: TOR nicht freigegeben. Siehe Funktion "i_APCI1710_EnableTorCounter"

2) i_APCI1710_KRNL_ReadTorCounterValue (...)**Syntax:**

```
<Return Wert> = i_APCI1710_KRNL_ReadTorCounterValue
                                (UINT      ui_BaseAddress,
                                BYTE       b_ModulNbr,
                                BYTE       b_TorCounter,
                                PBYTE      pb_TorCounterStatus,
                                PULONG     pul_TorCounterValue)
```

Parameter:**- Eingabe:**

UINT	ui_BaseAddress	Basisadresse der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des Moduls zu konfigurieren (0 bis 3)
BYTE	b_TorCounter	Selektierter TOR Zähler (0 oder 1)

- Ausgabe:

PBYTE	pb_TorCounterStatus	Gibt den TOR Status zurück 0: Zählzyklus nicht gestartet. Software-Gate nicht gesetzt 1: Zählzyklus gestartet. Software-Gate gesetzt. 2: Zählzyklus gestoppt. The Zählzyklus wird beendet. 3: Es hat sich ein Überlauf ergeben. Die Zeitbasis muss durch die Funktion "i_APCI1710_InitTorCounter" geändert werden 4: Timeout ist aufgetreten
PULONG	pul_TorCounterValue	TOR Zählwert.

Task:

Returns the TOR status (*pb_TorCounterStatus*) and the TOR counter value (*pul_TorCounterValue*) of the selected TOR module (*b_ModulNbr*) after a counting cycle stopped.

Funktionsaufruf:

ANSI C :

```
int          i_ReturnValue;
unsigned int  ui_BaseAddress;
unsigned char b_TorCounterStatus;
unsigned long ul_TorCounterValue;
i_ReturnValue = i_APCI1710_KRNL_ReadTorCounterValue
                (ui_BaseAddress,
                0,
                0,
                &pb_TorCounterStatus,
                &ul_TorCounterValue);
```


Return Wert:

0: Kein Fehler

-1: Das ausgewählte Modul ist falsch

-2: Das Modul ist kein TOR Modul

-3: Der ausgewählte TOR Zähler ist falsch

-4: TOR nicht initialisiert. Siehe Funktion "i_APCI1710_InitTorCounter"

-5: TOR nicht freigegeben. Siehe Funktion "i_APCI1710_EnableTorCounter"