DIN EN ISO 9001:2000
zertifiziert

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

Funktionsbeschreibung

ADDICOUNT APCI-/CPCI-1710

Pulse Width Modulation PWM

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 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:**

i **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.

Abbildungen

Tabellen

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	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

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

2 PULSE WIDTH MODULATION (PWM)

2.1 Funktionsbeschreibung

Die Funktion "PWM" ist eine Schnittstelle zur Pulsbreitenmodulation. Sie erzeugt eine Frequenz und legt die Zeitdauer des Low und High Pegels fest.
Für diese Funktion stehen zur Verfügung:
- ein 32-Bit Frequenzgenerator für die Einstellung des LOW und HIGH Pegels,
- 2 digitale Eingänge als Start oder Stop-Trigger
- 2 digitale Ausgänge für die Frequenzausgabe.

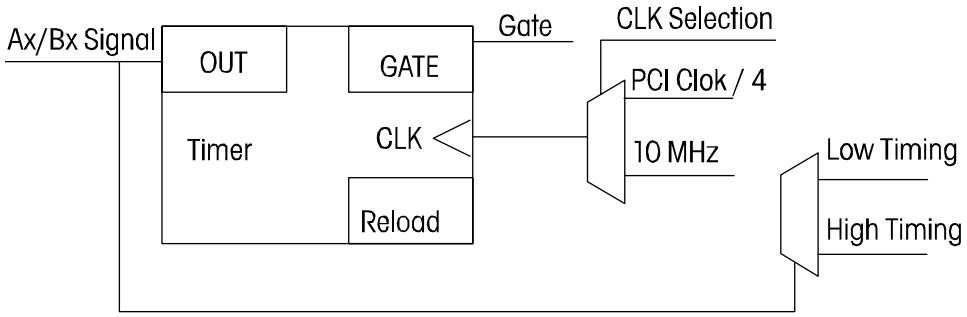
Eigenschaften:

Komplette galvanische Trennung der Ein- und Ausgänge durch Optokoppler zur Vermeidung von Erdschleifen.
- Interruptstatus nach Periodenende
- Signale bis zu 2,5 MHz können ausgegeben werden
- Auswahl des Start-Pegels
- Auswahl der Stop-Pegels
- Hardware-Gate
- Software-Gate

2.1.1 Blockschaltbid der PWM Funktion

Die Schnittstelle enthält:
- digitale Eingänge
- digitale Ausgänge

Abb. 2-1: Blockschaltbild der PWM Funktion



2.1.2 Typische Anwendungen

- Frequenzerzeugung
- Pulsebreitenmodulation
- Antriebstechnik

2.2 Benutzte Signale

Die Funktion "PWM" belegt **2 Eingänge und 2 Ausgänge** auf ein Funktionsmodul.

Tabelle 2-1: Benutzte Signale

Signale am Stecker	Polarität	Funktion
Ax +/ -	Diff. / TTL	Digitaler Ausgang (PWM0)
Bx +/ -	Diff. / TTL	Digitaler Ausgang (PWM1)
Cx + / -	Diff. / TTL/ (24V)	Externer Gate (PWM0)
Dx +/-	Diff. / TTL/ (24V)	Externer Gate (PWM0)

x: Nummer des Funktionsmoduls. (..) Option

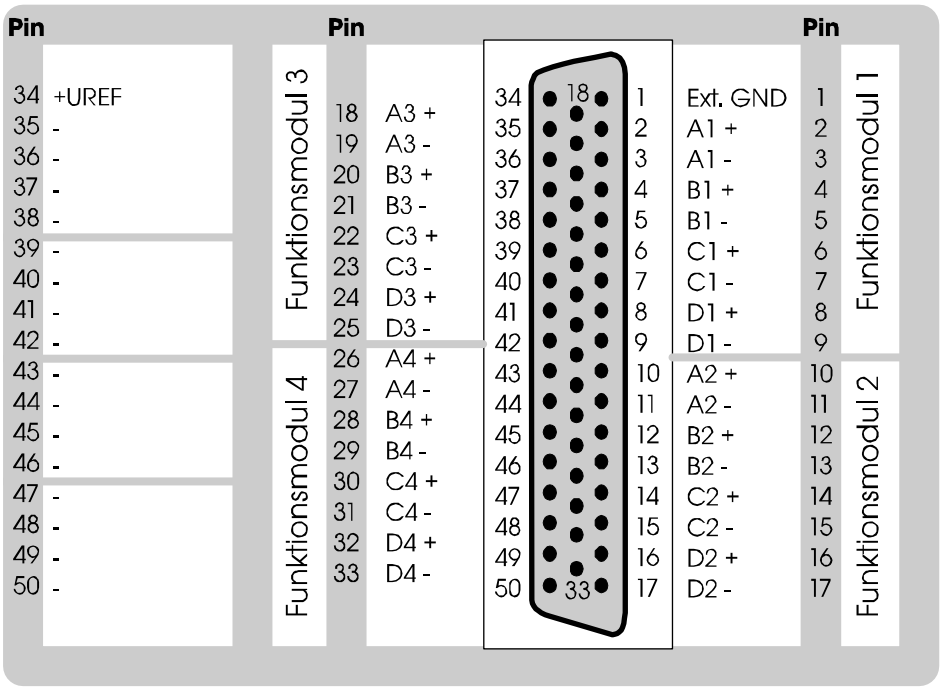
2.3 Pinbelegung der PWM

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.

Hierunter wird **der maximale Einsatz der PWM auf der Karte** abgebildet. Die Funktion ist auf allen Funktionsmodulen implementiert.

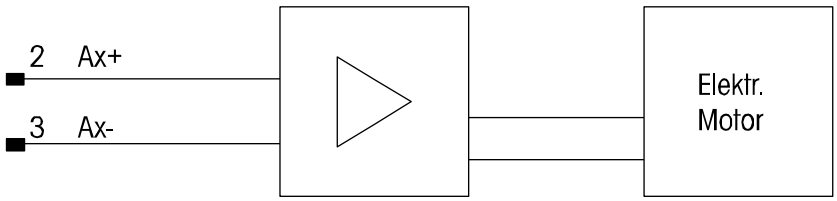
Abb. 2-2: Pinbelegung des Fronsteckers



-: nicht belegt

2.4 Anschlussbeispiel

Abb. 2-3: Anschlussbeispiel



2.5 E/A Adressbelegung der PWM Funktion

Tabelle 2-2: E/A Adressbelegung der PWM Funktion

	IOWR		IORD	
	D31.....D8	D7.....D0	D31.....D8	D7.....D0
BASEx + 0	PWM0 Timer (Low-Reloadwert)		PWM0 Low Base Timing	
BASEx + 4	PWM0 Timer (High-Reloadwert)		PWM0 High Base Timing	
BASEx + 8	-	PWM0 commando		PWM0 commando
BASEx + 12	-	PWM0/PWM1 synchro-gate		PWM0 status
BASEx + 16	-	Status register	-	PWM0 interrupt status
BASEx + 20	PWM1 Timer (Low-Reloadwert)		PWM1 Low Base Timing	
BASEx + 24	PWM1 Timer (High-Reloadwert)		PWM1 High Base Timing	
BASEx + 28	-	PWM1 commando		PWM1 commando
BASEx + 32	-	PWM1 gate		PWM1 status
BASEx + 36	-	PWM0/PWM1 synchro gate		PWM1 interrupt status
BASEx + 60	-		VERSION register	

-: keine Funktion

x: Nummer des Funktionsmoduls
Die Zugriffe werden immer in 32-Bit Breite geschrieben bzw. gelesen.

2.6 Beschreibung der E/A Funktionen

Die Funktion "PWM" ist eine abgemagerte Version der Funktion Zähler/Timer für die APCI-/CPCI-1710. Als Funktion kann der PCI Takt/4 -zeugt Rechtecksignale. Die Ausgangsimpulse aus dem Timer generieren die Pulsbreitenmodulation.

PWM Generator
Der Low/High Zeit-Teilerfaktor wird in den Timer geschrieben und legt die Ausgangsfrequenz fest. Als Eingangsfrequenz kann der PCI Takt/4 oder 40 MHz Quarz der Karte gesetzt werden.

2.6.1 Timer Register

Low Pegel

Der "Low" Reloadwert des Timers wird geschrieben.
Die Dauer des Low-Pegels für den Timer wird durch den Teilerfaktor festgelegt.

High Pegel

Der "High" Reloadwert des Timers wird geschrieben.
Die Dauer des High-Pegels für den Timer wird durch den Teilerfaktor festgelegt.

2.6.2 PWM commando register

DQ0:	0:	Die aktuelle Periode wird direkt nach dem Gate-Reset gestoppt
	1:	Die aktuelle Periode wird nach dem Gate-Reset und dem Periodensende gestoppt
DQ1:	0:	Das Ausgangssignal behält den Pegelzustand nach dem Stop-Signal
	1:	Das Ausgangssignal wird auf Low oder High nach dem Stop- Signal gesetzt.
DQ2:	0:	Das Ausgangssignal wird auf Low nach dem Stop-Signal gesetzt.
	1:	Das Ausgangssignal wird auf High nach dem Stop-Signal gesetzt.
DQ3:	0:	Interrupt-Generierung deaktiviert
	1:	Interrupt-Generierung aktiviert
DQ4:	0:	Externer Eingangsgate nicht benutzt
	1:	Externer Eingangsgate starte die PWM
DQ5:	0:	Die Periode startet mit einem Low-Pegel
	1:	Die Periode startet mit einem High-Pegel
DQ6:	nicht benutzt	
DQ7:	0:	PCI bus Takt /4 wird als Zeitbasis verwendet
	1:	10 MHz Takt wird als Zeitbasis verwendet.

2.6.3 PWM status register

Write:

DQ0:	0:	Deaktiviert die PWM
	1:	Aktiviert die PWM
DQ4:	0:	Kein Funktion
	1:	Setzt die PWM Initialisierung zurück

Read:

DQ0:	0:	PWM Software-Gate nicht gesetzt
	1:	PWM Software-Gate gesetzt
DQ4:	0:	PWM nicht initialisiert
	1:	PWM initialisiert
DQ5:	0:	PWM nicht gestartet
	1:	PWM im Laufen
DQ6:	0:	Externer Eingangsgate ist Low
	1:	Externer Eingangsgate ist High
DQ7:	0:	Ausgangssignal ist Low
	1:	Ausgangssignal ist High

2.6.4 PWM synchro gate/interrupt status register

Write:

DQ0:	0:	Deaktiviert PWM0 und PWM1
	1:	Aktiviert PWM0 und PWM1
DQ4:	0:	Keine Funktion
	1:	Setzt die Initialisierung von PWM0 und PWM1 zurück

Read:

DQ0:	1:	Interrupt bei Periodenende Die Interrupt-Abfrage wird durch das Lesen des Registers zurückgesetzt.
DQ31..1	Immer auf 0 gesetzt.	

2.6.5 Versions-Register

Basisadresse + 60
Enthält die Funktionbezeichnung und die Revision. (Lesebefehl, ASCI Format)

Beispiel

BASE + 60	"P"	"W"	"1"	"0"

bedeutet PWM; Revision 1.0

2.7 Grenzwerte

Maximale Ausgangsfrequenz: ..... 5 MHz

Teilerfaktor: ..... 4 bis 2^{32} - 1

2.8 Arbeiten mit PWM

1. Funktionsmodul auswählen.

2. Kanal auswählen.

3. Eingangsfrequenz auswählen.

4. Start/Stop-Pegel definieren.

5. Low-Zeit definieren

6. High-Zeit definieren.

7. Start-Bedingung auswählen (externes Gate oder Software-Start)

8. Starten der PWM Ausgabe.

3 STANDARDSOFTWARE

3.1 Define-Werte

Tabelle 3-1: Define-Tabelle

Define Name	Dezimal Wert	Hexadezimal Wert
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_30MHZ	30	1E
APCI1710_33MHZ	33	21
APCI1710_40MHZ	40	28

3.2 Interruptmaske

Jeder PWM Timer kann einen Interrupt generieren, nachdem der Periodenzyklus abgelaufen ist.
Um diesen Interrupt zu bekommen:

- sollen Sie den Interrupt mit "i_APCI1710_EnablePWM" freigeben und
- die Interruptroutine mit der Funktion "i_APCI1710_SetBoardIntRoutineX" gesetzt werden.

Tabelle 3-2: Interruptmaske der Funktion "PWM"

b_ModuleMask	ul_InterruptMask	Bedeutung
0000 000**1**	0**1**00 0000 0000 0000	PWM 0 Interrupt auf Modul 0 ausgelöst
0000 000**1**	**1**000 0000 0000 0000	PWM 1 Interrupt auf Modul 0 ausgelöst
0000 00**1**0	0**1**00 0000 0000 0000	PWM 0 Interrupt auf Modul 1 ausgelöst
0000 00**1**0	**1**000 0000 0000 0000	PWM 1 Interrupt auf Modul 1 ausgelöst
0000 0**1**00	0**1**00 0000 0000 0000	PWM 0 Interrupt auf Modul 2 ausgelöst
0000 0**1**00	**1**000 0000 0000 0000	PWM 1 Interrupt auf Modul 2 ausgelöst
0000 **1**000	0**1**00 0000 0000 0000	PWM 0 Interrupt auf Modul 3 ausgelöst
0000 **1**000	**1**000 0000 0000 0000	PWM 1 Interrupt auf Modul 3 ausgelöst

3.3 PWM-Funktionen

3.3.1 Initialisierung

1) i_APCI1710_InitPWM (...)

Syntax:

```
<Return Wert> =  i_APCI1710_InitPWM
                                    (BYTE       b_BoardHandle,
                                     BYTE       b_ModulNbr,
                                     BYTE       b_PWM,
                                     BYTE       b_ClockSelection,
                                    BYTE        b_TimingUnit,
                                     ULONG      ul_LowTiming,
                                     ULONG      ul_HighTiming,
                                     PULONG     pul_RealLowTiming,
                                     PULONG     pul_RealHighTiming)
```

Parameter

- Eingabe:

BYTE	b_BoardHandle	Handle der Karte xPCI-1710[1]
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1).
BYTE	b_ClockSelection	Auswahl des Takts-Signals - 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	Auswahl der Zeiteinheit (0 bis 4) 0: ns 1: µs 2: ms 3: s 4: mn
ULONG	ul_LowTiming	Zeitdauer des Low-Pegels. Siehe Tabelle 3-1
ULONG	ul_HighTiming	Zeitdauer des High-Pegels. Siehe Tabelle 3-1

- Ausgabe:

PULONG	pul_RealLowTiming	Reale Zeitdauer des Low-Pegels.
PULONG	pul_RealHighTiming	Reale Zeitdauer des High-Pegels.

[1] Gemeinsame Bezeichnung für APCI-1710 und CPCI-1710

Tabelle 3-3: Zeitwert-Tabelle

Clock selection	***b_TimingUnit***	***ul_TimingInterval* minimal value**	***ul_TimingInterval* maximal value**
APCI1710_30MHZ	ns (0)	266	4294967295
	µs (1)	1	571230650
	ms (2)	1	571230
	s (3)	1	571
	mn (4)	1	9
APCI1710_33MHZ	ns (0)	242	4294967295
	µs (1)	1	519691043
	ms (2)	1	519691
	s (3)	1	520
	mn (4)	1	8
APCI1710_40MHZ	ns (0)	200	4294967295
	µs (1)	1	429496729
	ms (2)	1	429496
	s (3)	1	429
	mn (4)	1	7

Aufgabe:
Konfiguriert den ausgwählten PWM Generator (*b_PWM*) auf das ausgewählte Modul (*b_ModulNbr*).
ul_LowTiming, *ul_HighTiming* und *ul_TimingUnit* legen die Zeitdauer des Low/High Pegels fest.
pul_RealLowTiming, *pul_RealHighTiming* geben den realen Zeitwert zurück.

Rufen Sie diese Funktion zuerst auf, bevor Sie auf andere PWM Funktionen zugreifen.

Funktionsaufruf:
ANSI C:

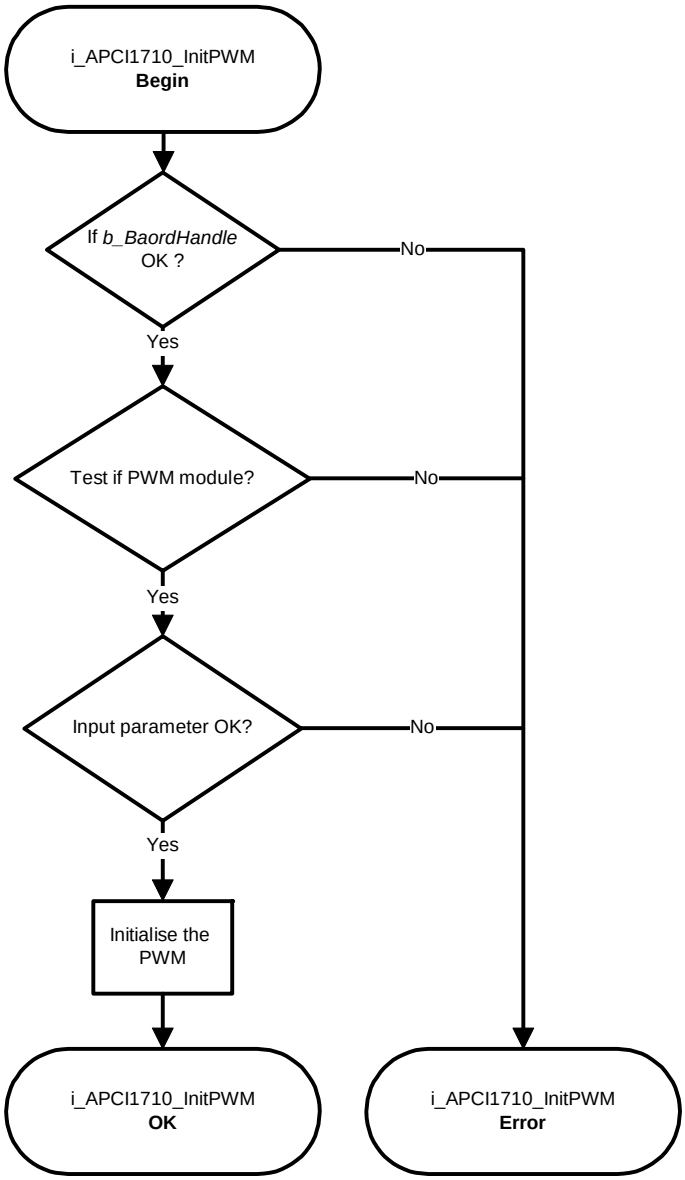
```
int             i_ReturnValue;
unsigned char   b_BoardHandle;
unsigned long   ul_RealLowTiming;
unsigned long   ul_RealHighTiming;

i_ReturnValue = i_APCI1710_InitPWM
                                    (b_BoardHandle,
                                     0,
                                     0,
                                     APCI1710_40MHZ,
                                     1,
                                     100,
                                     500,
                                     &ul_RealLowTiming,
                                     &ul_RealHighTiming);
```

Return Wert:
0: Kein Fehler
-1: Handle Parameter der Karte ist falsch
-2: Die ausgewählte Modulnummer ist falsch
-3: Das ausgewählte Modul ist kein "PWM"-Modul.
-4: PWM Auswahl ist falsch
-5: Der ausgewählte Eingangstakt ist falsch
-6: Die ausgewählte Zeiteinheit ist falsch
-7: Eingestellte "Low" Basiszeit ist falsch
-8: Eingestellte "High" Basiszeit ist falsch
-9: Der 40MHz Takt kann mit dieser Karte nicht verwendet können

Eingabe

b_BoardHandle	ul_LowTiming
b_ModulNbr	ul_HighTiming
b_PWM	pul_RealLowTiming
b_ClockSelection	pul_RealHighTiming
b_TimingUnit	

Funktion



Ausgabe

pul_RealLowTiming
pul_RealHighTiming

<Return Wert>

2) i_APCI1710_EnablePWM (...)

Syntax:

```
<Return Wert> =  i_APCI1710_EnablePWM
                                (BYTE    b_BoardHandle,
                                 BYTE    b_ModulNbr,
                                 BYTE    b_PWM,
                                 BYTE    b_StartLevel,
                                 BYTE    b_StopMode,
                                 BYTE    b_StopLevel,
                                 BYTE    b_ExternGate,
                                 BYTE    b_InterruptEnable)
```

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle der Karte APCI-/CPCI_1710
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1)
BYTE	b_StartLevel	Auswahl des Pegels vor dem Starten 0: Die Periode startet auf Low Pegel 1: Die Periode startet auf High Pegel
BYTE	b_StopMode	Auswahl des Stop-Modes 0: Die PWM wird driekt nach der Funktion "i_APCI1710_DisablePWM" gestoppt und die aktuelle Periode wird gestoppt 1: Nach der Funktion "i_APCI1710_DisablePWM",wird die PWM beim Periodensende gestoppt
BYTE	b_StopLevel	Stoppt die Auswahl des PWM-Pegels 0: Das Ausgangssignal hält seinen Pegel-Zustand nach Aufruf der Funktion "i_APCI1710_DisablePWM" 1: Das Ausgangssignal wird auf Low nach Aufruf der Funktion "i_APCI1710_DisablePWM" gesetzt 2: Das Ausgangssignal wird auf High nach Aufruf der Funktion "i_APCI1710_DisablePWM"
BYTE	b_ExternGate	Auswahl der externen Gate-Aktion 0: Externer Gate-Signal nicht benutzt. 1: Externer Gate-Signal benutzt
BYTE	b_InterruptEnable	Aktiviert or deaktiviert den PWM Interrupt. APCI1710_ENABLE: Aktiviert den PWM-Interrupt. Ein Interrupt erfolgt nach jeder Periode APCI1710_DISABLE: Deaktiviert den PWM-Interrupt

- Ausgabe:

Es erfolgt keine Ausgabe.

Aufgabe:
Aktiviert den ausgewählten PWM Generator (*b_PWM*) des angegebenen Moduls (*b_ModulNbr*).

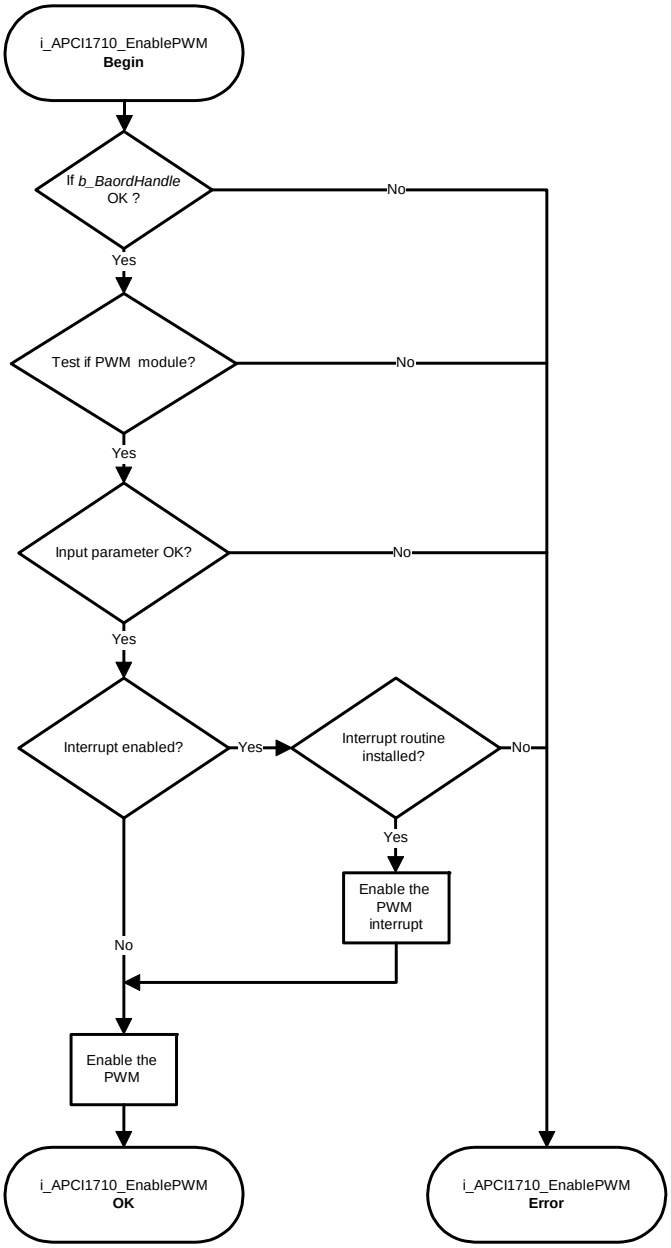
Rufen Sie die Funktion "i_APCI1710_InitPWM" auf, bevor Sie auf eine andere Funktion zugreifen.
Falls der PWM Interrupt aktiviert wird, generiert der PWM einen Interrup nach jeder Periode.
Siehe Funktion "i_APCI1710_SetBoardIntRoutineX" und Absatz 3.2 Interruptmaske.

Funktionsaufruf:
ANSI C:

```
int             i_ReturnValue;
unsigned char b_BoardHandle;

i_ReturnValue = i_APCI1710_EnablePWM
                                    (b_BoardHandle,
                                     0,
                                     0,
                                     0,
                                     1,
                                     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 "PWM"-Modul.
-4: PWM Auswahl ist falsch
-5: PWM nicht initialisert, siehe Funktion "i_APCI1710_InitPWM"
-6: Auswahl des PWM Start-Pegels ist falsch
-7: Auswahl des PWM Stop-Modus ist falsch
-8: Auswahl des PWM Stop-Pegels ist falsch
-9: Auswahl des externen Gate-Signals ist falsch
-10: Interruptparameter falsch
-11: Interruptfunktion nicht initialisiert. Siehe Funktion "i_APCI1710_SetBoardIntRoutineX"

Eingabe	
b_BoardHandle	b_StopMode
b_ModulNbr	b_StopLevel
b_PWM	b_ExternGate
b_StartLevel	b_InterruptEnable
Funktion	



Ausgabe
<Return Wert>

3) i_APCI1710_SetNewPWMTiming (...)

Syntax:
<Return Wert> = i_APCI1710_InitPWM
(BYTE b_BoardHandle,
BYTE b_ModulNbr,
BYTE b_PWM,
BYTE b_TimingUnit,
ULONG ul_LowTiming,
ULONG ul_HighTiming)

Parameter:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte APCI-/CPCI_1710
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1).
BYTE	b_TimingUnit	Auswahl der Zeiteinheit (0 bis 4) 0: ns 1: µs 2: ms 3: s 4: mn
ULONG	ul_LowTiming	Zeitdauer des Low-Pegels. Siehe Tabelle 3-1
ULONG	ul_HighTiming	Zeitdauer des High-Pegels. Siehe Tabelle 3-1

- Ausgabe:
Es erfolgt keine Ausgabe

Aufgabe:
Setzt eine neue Zeitperiode. *ul_LowTiming*, *ul_HighTiming* und *ul_TimingUnit* legen die LOW/High Zeitbasis für die Periode fest.

Funktionsaufruf:
ANSI C:

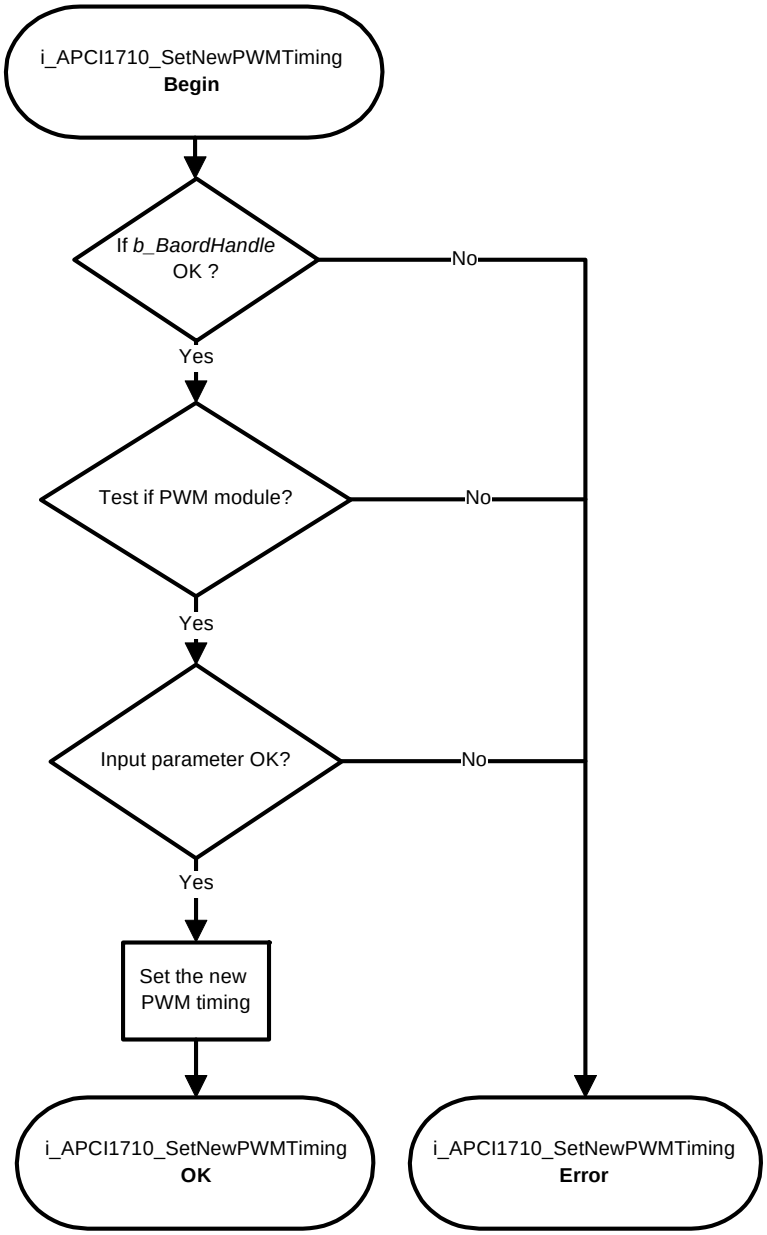
```
int             i_ReturnValue;
unsigned char   b_BoardHandle;
i_ReturnValue = i_APCI1710_InitPWM
                (b_BoardHandle,
                 0,
                 0,
                 1,
                 100,
                 500);
```

Return Wert:
0: Kein Fehler
-1: Handle Parameter der Karte ist falsch
-2: Die ausgewählte Modulnummer ist falsch
-3: Das ausgewählte Modul ist kein "PWM"-Modul.
-4: PWM Auswahl ist falsch

-5: PWM nicht initialisiert. Siehe Funktion "i_APCI1710_InitPWM"
-6: Die ausgewählte Zeiteinheit ist falsch
-7: Eingestellte "Low" Basiszeit ist falsch
-8: Eingestellte "High" Basiszeit ist falsch

Eingabe	
b_BoardHandle	b_TimingUnit
b_ModulNbr	ul_LowTiming
b_PWM	ul_HighTiming

Funktion



Ausgabe

<Return Wert>

4) i_APCI1710_DisablePWM (...)

Syntax:
<Return Wert> = i_APCI1710_DisablePWM
(BYTE b_BoardHandle,
BYTE b_ModulNbr,
BYTE b_PWM)

Parameter:
- Eingabe**:**

BYTE	b_BoardHandle	Handle der Karte APCI-/CPCI_1710
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1)

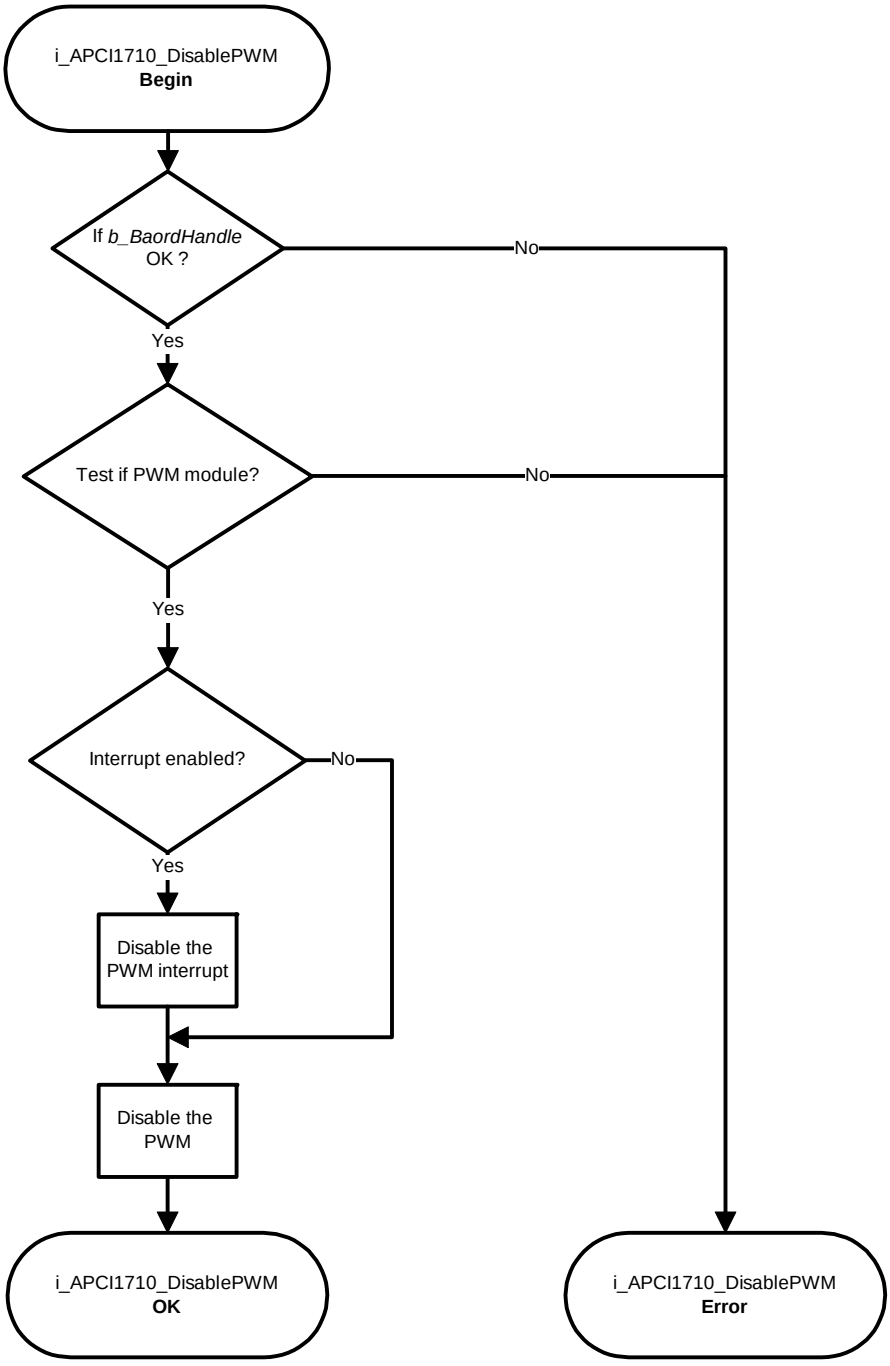
- Ausgabe:
Es erfolgt keine Ausgabe

Aufgabe:
Deaktiviert the selected PWM (*b_PWM*) of the selected module (*b_ModulNbr*). The output signal level depends on the initialisation with "i_APCI1710_EnablePWM".
See the parameters *b_StartLevel*, *b_StopMode* and *b_StopLevel* of this function.

Funktionsaufruf:
ANSI C:
```
int           i_ReturnValue;
unsigned char b_BoardHandle;

i_ReturnValue = i_APCI1710_DisablePWM
                (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 "PWM"-Modul.
-4: PWM Auswahl ist falsch
-5: PWM nicht initialisiert. Siehe Funktion"i_APCI1710_InitPWM"
-6: Auswahl des PWM Start-Pegels ist falsch
-7: Auswahl des PWM Stop-Modus ist falsch
-8: Auswahl des PWM Stop-Pegels ist falsch
-9: Auswahl des externen Gate-Signals ist falsch
-10: Interruptparameter falsch
-11: Interruptfunktion nicht initialisiert.

Eingabe	
b_BoardHandle b_ModulNbr	b_PWM
Funktion	



Ausgabe
<Return Wert>

5) i_APCI1710_GetPWMInitialisation (...)

Syntax:

```
<Return Wert> =  i_APCI1710_GetPWMInitialisation
                                (BYTE    b_BoardHandle,
                                 BYTE    b_ModulNbr,
                                 BYTE    b_PWM,
                                 PBYTE   pb_TimingUnit,
                                 PULONG  pul_LowTiming,
                                 PULONG  pul_HighTiming,
                                 PBYTE   pb_StartLevel,
                                 PBYTE   pb_StopMode,
                                 PBYTE   pb_StopLevel,
                                 PBYTE   pb_ExternGate,
                                 PBYTE   pb_InterruptEnable,
                                 PBYTE   pb_Enable)
```

Parameter:

- Eingabe:

BYTE	b_BoardHandle	Handle der Karte APCI-/CPCI_1710
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1)

- Ausgabe:

PBYTE	pb_TimingUnit	Base timing Unit (0 to 4) 0: ns 1: µs 2: ms 3: s 4: mn
PULONG	pul_LowTiming	Low base timing value. See base timing value description table
PULONG	pul_HighTiming	High base timing value. See base timing value description table
PBYTE	pb_StartLevel	Start period level selection 0: The period starts with a low level 1: The period starts with a high level
PBYTE	pb_StopMode	Stop mode selection 0: The PWM is stopped directly after the function "i_APCI1710_DisablePWM" and stops the current period 1: After the function "i_APCI1710_DisablePWM" the PWM is stopped at the end of the current period
PBYTE	pb_StopLevel	Stop PWM level selection 0: The output signal keeps the level after the function "i_APCI1710_DisablePWM" 1: The output signal is set to low after the function "i_APCI1710_DisablePWM" 2: The output signal is set to high after the function "i_APCI1710_DisablePWM"

PBYTE	pb_ExternGate	External gate action selection 0: External gate signal not used. 1: External gate signal used.
PBYTE	pb_InterruptEnable	Aktiviert or Deaktiviert the PWM interrupt. APCI1710_ENABLE: Aktiviert the PWM interrupt. An interrupt occurs after each period APCI1710_DISABLE: Deaktiviert the PWM interrupt
PBYTE	pb_Enable	Indicate if the PWM is enabled or not 0: PWM not enabled 1: PWM enabled

Aufgabe:
Return the PWM (*b_PWM*) initialisation of the selected module (*b_ModulNbr*).
First call the function " i_APCI1710_InitPWM" before you call this function.

Funktionsaufruf:
ANSI C:

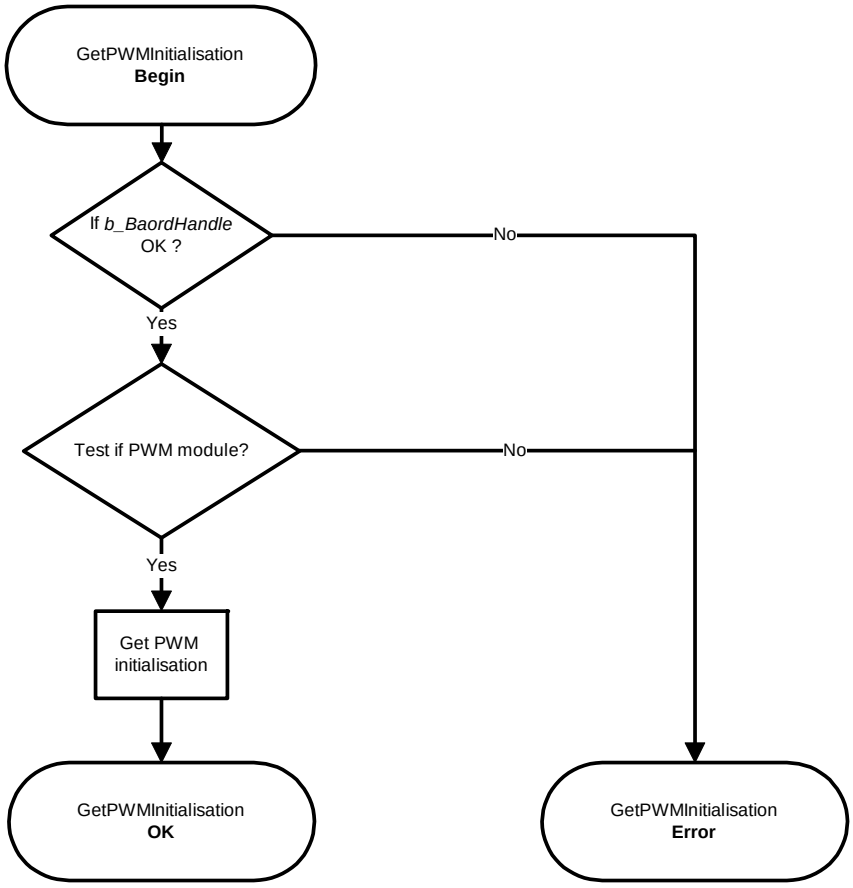
```
unsigned char     b_TimingUnit;
unsigned long     ul_LowTiming;
unsigned long     ul_HighTiming;
unsigned char     b_StartLevel;
unsigned char     b_StopMode;
unsigned char     b_StopLevel;
unsigned char     b_ExternGate;
unsigned char     b_InterruptEnable;
unsigned char     b_Enable;

i_ReturnValue = i_APCI1710_GetPWMInitialisation
                                    (b_BoardHandle,
                                     0,
                                     0,
                                     &b_TimingUnit,
                                     &ul_LowTiming,
                                     &ul_HighTiming,
                                     &b_StartLevel,
                                     &b_StopMode,
                                     &b_StopLevel,
                                     &b_ExternGate,
                                     &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 "PWM"-Modul.
-4: PWM Auswahl ist falsch
-5: PWM nicht initialisiert. Siehe Funktion "i_APCI1710_InitPWM"

Eingabe

b_BoardHandle	pb_StartLevel
b_ModulNbr	pb_StopMode
b_PWM	pb_StopLevel
pb_TimingUnit	pb_ExternGate
pul_LowTiming	pb_InterruptEnable
pul_HighTiming	pb_Enable

Funktion



Ausgabe

pb_TimingUnit	pul_LowTiming
pul_HighTiming	pb_StartLevel
pb_StopMode	pb_StopLevel
pb_ExternGate	pb_InterruptEnable
pb_Enable	

<Return Wert>

3.3.2 PWM Status

1) i_APCI1710_GetPWMStatus (...)

Syntax:
<Return Wert> = i_APCI1710_GetPWMStatus
(BYTE b_BoardHandle,
BYTE b_ModulNbr,
BYTE b_PWM,
PBYTE pb_PWMOutputStatus,
PBYTE pb_ExternGateStatus)

Parameter:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte APCI-/CPCI_1710
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1)
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)

- Ausgabe:

PBYTE	pb_PWMOutputStatus	Status des PWM Ausgangs. 0: PWM Ausgangspegel ist Low 1: PWM Ausgangspegel ist High.
PBYTE	pb_ExternGateStatus	Status des externen Gate. 0: Externer Gate ist Low. 1: Externer Gate ist High.

Aufgabe:
Gibt den Status des ausgewählten PWM (*b_PWM*) vom angegebenen Modul (*b_ModulNbr*) zurück.

Funktionsaufruf:
ANSI C:

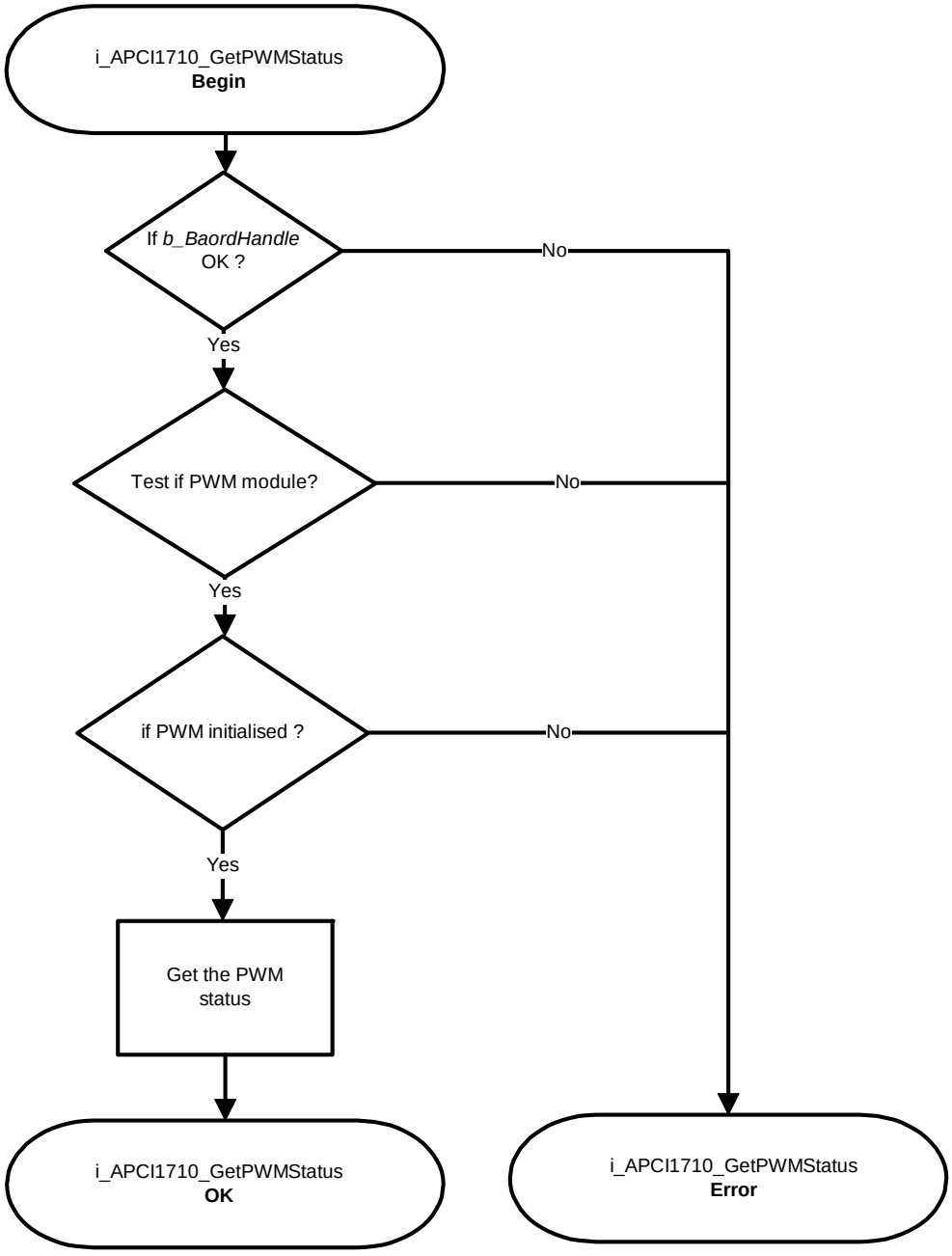
```
int               i_ReturnValue;
unsigned char b_BoardHandle;
unsigned char b_PWMOutputStatus;
unsigned char b_ExternGateStatus;
i_ReturnValue = i_APCI1710_GetPWMStatus
                                    (b_BoardHandle,
                                     0,
                                     0,
                                     &b_PWMOutputStatus,
                                     &b_ExternGateStatus);
```

Return Wert:
0: Kein Fehler
-1: Handle Parameter der Karte ist falsch
-2: Die ausgewählte Modulnummer ist falsch
-3: Das ausgewählte Modul ist kein "PWM"-Modul.
-4: PWM Auswahl ist falsch
-5: PWM nicht initialisiert. Siehe Funktion "i_APCI1710_InitPWM"
-6: PWM nicht aktiviert. Siehe Funktion "i_APCI1710_EnablePWM"

Eingabe

b_BoardHandle	pb_PWMOutputStatus
b_ModulNbr	pb_ExternGateStatus
b_PWM	

Funktion



Ausgabe

pb_PWMOutputStatus
pb_ExternGateStatus

<Return Wert>

3.3.3 Windows NT/95 interrupt kernel functions

IMPORTANT!
These functions are only available for the Windows NT and Windows 95/98 user interrupt routine in the synchronous mode. See function "i_APCI1710_SetBoardIntRoutineWin32"

1) i_APCI1710_KRNL_GetPWMStatus (...)

Syntax:
<Return Wert> = i_APCI1710_KRNL_GetPWMStatus
(UINT ui_BaseAddress,
BYTE b_ModulNbr,
BYTE b_PWM,
PBYTE pb_PWMOutputStatus,
PBYTE pb_ExternGateStatus)

Parameter:
- Eingabe:

UINT	ui_BaseAddress	Basisaddresse der **APCI-/CPCI-1710**
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1)

- Ausgabe:

PBYTE	pb_PWMOutputStatus	Status des PWM Ausgangs. 0: PWM Ausgangspegel ist Low 1: PWM Ausgangspegel ist High.
PBYTE	pb_ExternGateStatus	Status des externen Gate. 0: Externer Gate ist Low. 1: Externer Gate ist High.

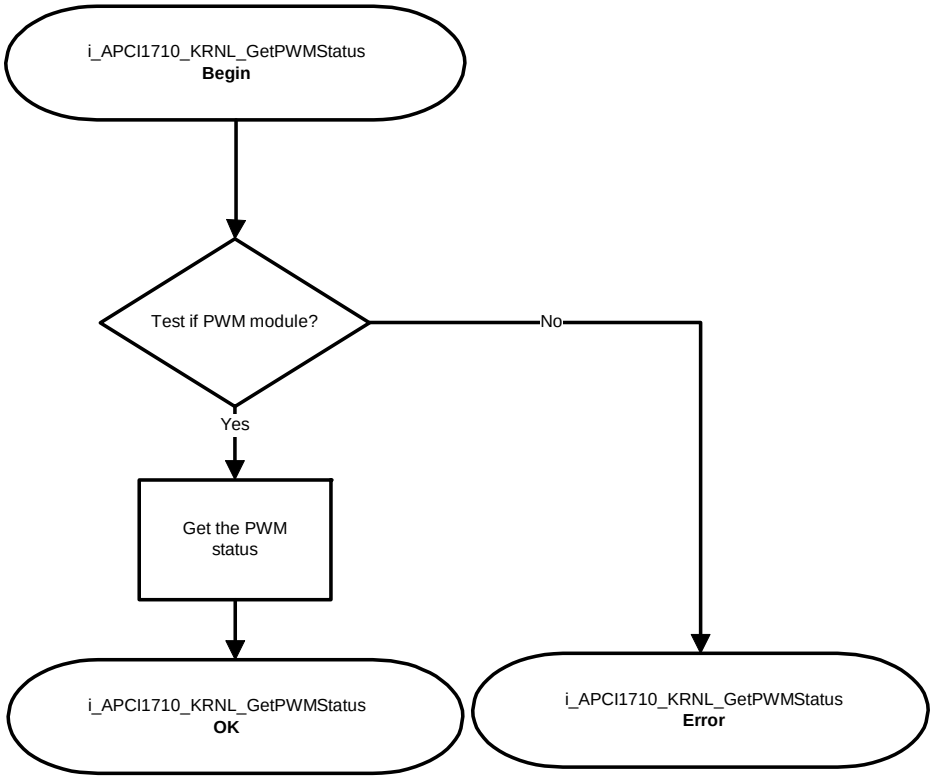
Aufgabe:
Gibt den Status des ausgewählten PWM (*b_PWM*) vom angegebenen Modul (*b_ModulNbr*) zurück.

Funktionsaufruf:
ANSI C:

```
int                i_ReturnValue;
unsigned int       ui_BaseAddress;
unsigned char      b_PWMOutputStatus;
unsigned char      b_ExternGateStatus;
i_ReturnValue = i_APCI1710_KRNL_GetPWMStatus
                                   (ui_BaseAddress,
                                    0,
                                    0,
                                    &b_PWMOutputStatus,
                                    &b_ExternGateStatus);
```

Return Wert:
0: Kein Fehler
-1: Module selection wrong
-2: The module is not a PWM module
-3: PWM Auswahl ist falsch
-4: PWM nicht initialisiert. Siehe Funktion_APCI1710_InitPWM"
-5: PWM nicht aktiviert. Siehe Funktion "i_APCI1710_EnablePWM"

Eingabe	
ui_BaseAddress b_ModulNbr b_PWM	pb_PWMOutputStatus pb_ExternGateStatus
Funktion	



Ausgabe
pb_PWMOutputStatus pb_ExternGateStatus
<Return Wert>

2) i_APCI1710_KRNL_SetNewPWMTiming (...)

Syntax:

```
<Return Wert> =  i_APCI1710_KRBL_InitPWM
                                (UINT     ui_BaseAddress,
                                 BYTE     b_ModulNbr,
                                 BYTE     b_PWM,
                                 BYTE     b_ClockSelection,
                                 BYTE     b_TimingUnit,
                                 ULONG    ul_LowTiming,
                                 ULONG    ul_HighTiming)
```

Parameter:

- Eingabe:

UINT	ui_BaseAddress	Basisadresse der **APCI-/CPCI-1710**
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PWM	Ausgewählter PWM Generator (0 oder 1).
BYTE	b_ClockSelection	Auswahl des Takts-Signals - 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	Auswahl der Zeiteinheit (0 bis 4) 0: ns 1: µs 2: ms 3: s 4: mn
ULONG	ul_LowTiming	Zeitdauer des Low-Pegels. Siehe Tabelle 3-1
ULONG	ul_HighTiming	Zeitdauer des High-Pegels. Siehe Tabelle 3-1

- Ausgabe:

Es erfolgt keine Ausgabe

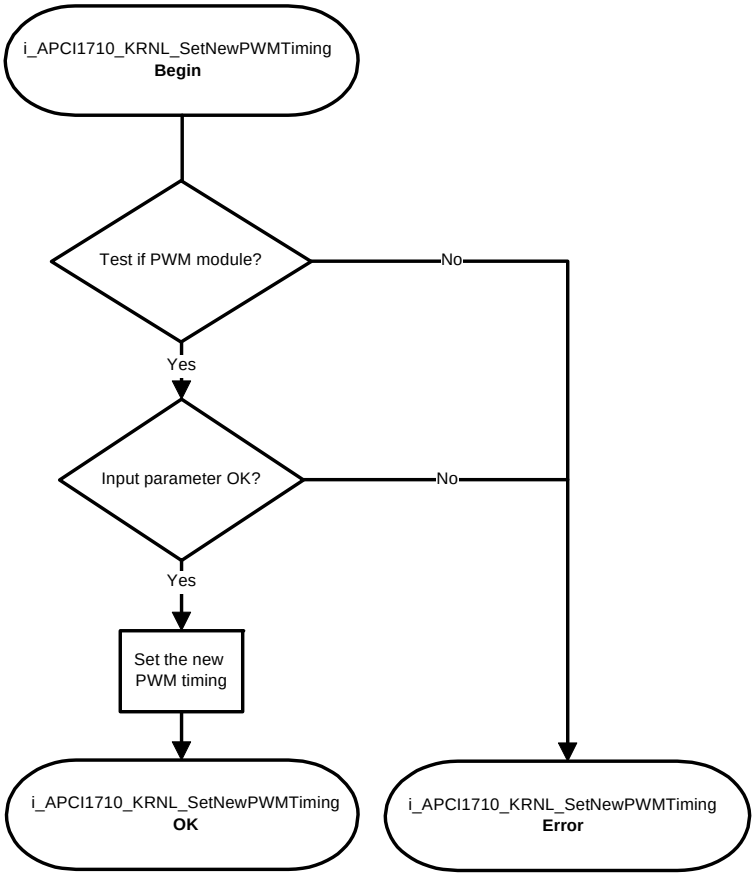
Aufgabe:

Setzt eine neue Zeitperiode. *ul_LowTiming*, *ul_HighTiming* und *ul_TimingUnit* legen die LOW/High Zeitbasis für die Periode fest.

Funktionsaufruf:
ANSI C:

```
int             i_ReturnValue;
unsigned int    ui_BaseAddress;
i_ReturnValue = i_APCI1710_InitPWM
                                    (ui_BaseAddress,
                                     0,
                                     0,
                                     APCI1710_40MHZ,
                                     1,
                                     100,
                                     500);
```

Return Wert:
0: Kein Fehler
-1: Handle Parameter der Karte ist falsch
-2: Die ausgewählte Modulnummer ist falsch
-3: Das ausgewählte Modul ist kein "PWM"-Modul.
-4: PWM Auswahl ist falsch
-5: PWM nicht initialisiert. Siehe Funktion_APCI1710_InitPWM"
-6: Die ausgewählte Zeiteinheit ist falsch
-7: Eingestellte "Low" Basiszeit ist falsch
-8: Eingestellte "High" Basiszeit ist falsch

Eingabe	
ui_BaseAddress b_ModulNbr b_PWM b_ClockSelection	b_TimingUnit ul_LowTiming ul_HighTiming
Funktion	
	
Ausgabe	
<Return Wert>	