ADDI-DATA®

Technischer Support:
+49 (0)7223 / 9493-0

Vorläufig

CE

Funktionsbeschreibung

ADDICOUNT APCI-1710

TTL Eingänge und Ausgänge für den PCI-Bus

2. Ausgabe 05/2001

Abbildungen

Tabellen

1 EINLEITUNG

1.1 Technische Dokumentation

Dieses Referenzhandbuch bezieht sich auf die Karte **APCI-1710** und beschreibt die Funktion "TTL I/O", die auf die Funktionsmodule der Karte konfiguriert werden kann. Bitte vergewissern Sie sich, daß Sie außerdem folgendes bekommen haben:

- das Handbuch **ADDICOUNT APCI-/CPCI-1710: Funktionsprogrammierbare Zählerkarte für den PCI-Bus**, das allgemeine Informationen für den Betrieb der Karte enthält,
- das gelbe Blatt mit den Sicherheitshinweisen.

1.2 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.

1.3 Schriftvereinbarungen

Die Signale auf dem 50poligen Stecker X6 über das Flachbandkabel sind alle auf ein Funktionsmodul bezogen. Bitte beachten Sie die folgenden Schriftvereinbarungen:

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

2 TTL I/O

2.1 Funktionsbeschreibung

Die **APCI-1710** stellt am Stecker **X6** digitale E/A, sowie GND und Vcc des PC zur Verfügung. Diese Signale müssen dem TTL Pegel entsprechen und sorgfältig behandelt werden, um die Karte nicht zu beschädigen, falls andere Signale angeschlossen werden.

Mit dem Kabel **FB8000** kann die Karte durch den Stecker X6 an die Peripherie angeschlossen werden.

Die Signale PA0 bis PA7, PB0 bis PB7 und PC0 bis PC7 sind an allen "Funktionsmodulen" angeschlossen. Sie werden entweder geteilt oder stehen nur für ein "Funktionsmodul" (z. B. "TTL I/O) zur Verfügung.

Beispiel: das Funktionsmodul Nr. 4 ist nur mit der Funktion TTL I/O programmiert. Die Signale können als Ein- oder Ausgänge (der zugewiesenen Funktion entsprechend) gesetzt werden.

Diese Funktion eignet sich besonders für Industrie-Anwendungen, für die Zuverlässigkeit und Robustheit verlangt werden.

Eigenschaften:

- Zur Vermeidung von Erdschleifen wird eine komplette galvanische Trennung durch Optokoppler für die Ein-/Ausgänge herangezogen
- Eingänge und Ausgänge können per Software invertiert werden
- Hardware- und Software-GATE möglich, rücklesbar

2.1.1 Blockdiagramm

2.1.2 Typische Anwendungen

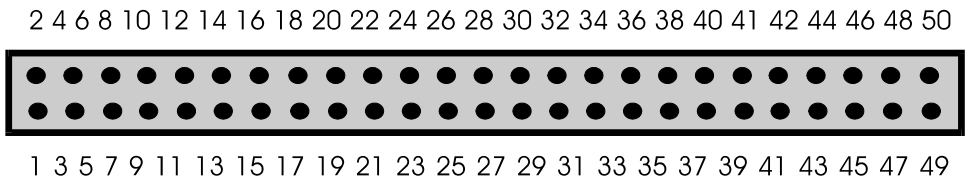
- Digitaler "one-shot"
- Echtzeituhr
- Ereigniszähler
- Programmierbarer Ratengenerator
- Frequenzgenerator
- Komplexer Signalgenerator

2.2 Benutzte Signale

Die Signale PA0 bis PA7, PB0 bis PB7 und PC0 bis PC7 sind an allen "Funktionsmodulen" angeschlossen. Sie werden entweder geteilt oder stehen nur für ein "Funktionsmodul" (z. B. "TTL I/O) zur Verfügung.

Pro "Funktionsmodul" stehen 2 TTL Ein- und Ausgänge (I und J) zur Verfügung. **Diese können nur in Verbindung mit dem Funktionsmodul TTL I/O genutzt werden.**

Fig 2-1: Pinbelegung des Flachbandsteckers X6



Pinnummer am Stecker	Name	Beschreibung	Pinnummer am FB8000
1	PC7*1	TTL, Ein- oder Ausgang; Nach Reset: Eingang	1
2	GND	PC GND, nicht isoliert	34
3	PC6	TTL, Ein- oder Ausgang; Nach Reset: Eingang	18
4	GND	PC GND, nicht isoliert	2
5	PC5	TTL, Ein- oder Ausgang; Nach Reset: Eingang	35
6	GND	PC GND, nicht isoliert	19
7	PC4	TTL, Ein- oder Ausgang; Nach Reset: Eingang	3
8	GND	PC GND, nicht isoliert	36
9	K1	TTL Ausgang; gleiches Signal wie H1 am Fronstecker, FM1	20
10	GND	PC GND, nicht isoliert	4
11	K2	TTL Ausgang; gleiches Signal wie H2 am Fronstecker, FM2	37
12	GND	PC GND, nicht isoliert	21
13	K3	TTL Ausgang; gleiches Signal wie H3 am Fronstecker, FM3	5
14	GND	PC GND, nicht isoliert	38
15	K4	TTL Ausgang; gleiches Signal wie H4 am Fronstecker, FM4	22
16	GND	PC GND, nicht isoliert	6
17	PA0	TTL, Ein- oder Ausgang; Nach Reset: Eingang	39
18	PA1	TTL, Ein- oder Ausgang; Nach Reset: Eingang	23
19	PA2	TTL, Ein- oder Ausgang; Nach Reset: Eingang	7
20	PA3	TTL, Ein- oder Ausgang; Nach Reset: Eingang	40
21	GND	PC GND, nicht isoliert	24
22	PA4	TTL, Ein- oder Ausgang; Nach Reset: Eingang	8
23	PA5	TTL, Ein- oder Ausgang; Nach Reset: Eingang	41

Pinnummer am Stecker	Name	Beschreibung	Pinnummer am FB8000
24	PA6	TTL, Ein- oder Ausgang; Nach Reset: Eingang	25
25	PA7	TTL, Ein- oder Ausgang; Nach Reset: Eingang	9
26	V. ext		42
27	PB0	TTL, Ein- oder Ausgang; Nach Reset: Eingang	26
28	PB1	TTL, Ein- oder Ausgang; Nach Reset: Eingang	10
29	PB2	TTL, Ein- oder Ausgang; Nach Reset: Eingang	43
30	PB3	TTL, Ein- oder Ausgang; Nach Reset: Eingang	27
31	GND		11
32	PB4	TTL, Ein- oder Ausgang; Nach Reset: Eingang	44
33	PB5	TTL, Ein- oder Ausgang; Nach Reset: Eingang	28
34	PB6	TTL, Ein- oder Ausgang; Nach Reset: Eingang	12
35	PB7	TTL, Ein- oder Ausgang; Nach Reset: Eingang	45
36	V. ext		29
37	PC0	TTL, Ein- oder Ausgang; Nach Reset: Eingang	13
38	PC1	TTL, Ein- oder Ausgang; Nach Reset: Eingang	46
39	PC2	TTL, Ein- oder Ausgang; Nach Reset: Eingang	30
40	PC3	TTL, Ein- oder Ausgang; Nach Reset: Eingang	14
41	GND		47
42	I4*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM4	31
43	J4*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM4	15
44	I3*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM3	48
45	J3*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM3	32
46	V. ext	PC +5 V Spannungsversorgung	16
47	I2*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM2	49
48	J2*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM2	33
49	I1*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM1	17
50	J1*2	TTL, Ein- oder Ausgang; Nach Reset: Ausgang, FM1	50

*1 PA, PB und PC Pullup auf 5 V

*2 Serienwiderstand 100 Ω, PD

PA, PB, PC und PD können nur über das Funktionsmodul "TTL I/O" genutzt werden.

2.3 E/A-Belegung der TTL I/Os

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

	Lesen				Schreiben			
	Modulnummer (Funktionsmodul 0)							
	D31...D24	D23...D16	D15.....D8	D7.....D0	D31...D24	D23...D16	D15.....D8	D7.....D0
BYTES	HIGHBYTE	MIDHIGHB	MIDLOWB	LOWBYTE	HIGHBYTE	MIDHIGHB	MIDLOWB	LOWBYTE
BASEx + 0	Port D J,I	Port C als Eingang	Port B als Eingang	Port A als Eingang	-	-	-	Ausgang I D0
BASEx + 4	-	-	-	-	-	-	-	Ausgang J D0
BASEx + 8	-	-	-	-	-	-	-	Port A als Ausgang
BASEx + 12	-	-	-	-	-	-	-	Port B als Ausgang
BASEx + 16	-	-	-	-	-	-	-	Port C als Ausgang
BASEx + 20	-	-	-	Control Word [D3..0]	-	-	-	Control Word D[3..0]
BASEx + 60	Modul-Identifikation „TL20“				-	-	-	-

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

Die Ports A, B, C sind nach dem Reset als Eingänge definiert, das Port D (I, J) ist nach dem Reset als Ausgang geschaltet

Die Ports A, B, C, D sind alle über 4K7 Pull Up Widerstände auf VCC gesetzt.

2.4 Beschreibung der E/A-Funktionen

Erkennungsregister

Beim Lesen der Adresse BASE + 60, steht die Erkennung der Funktion sowie die Revision der Funktion in ASCI Format.

BASE + 60 "T" "L" "2" "0"

Bedeutet: TTL I/O Revision 2.0

3 STANDARD-SOFTWARE

3.1 Einleitung

WICHTIG!
Merken Sie sich die folgenden Schriftweisen im Text:

Funktion: "i_APCI1710_SetBoardInformation"
Variable *ui_Address*

Bitte beachten: Die Karte **CPCI-1710** ist mit der **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.

Tabelle 3-1: Type Declaration für Dos und Windows 3.1X

	Borland C	Microsoft C	Borland Pascal	Microsoft Visual Basic Dos	Microsoft Visual Basic Windows
VOID	void	void	pointer		any
BYTE	unsigned char	unsigned char	byte	integer	integer
INT	int	int	integer	integer	integer
UINT	unsigned int	unsigned int	word	long	long
LONG	long	long	longint	long	long
PBYTE	unsigned char *	unsigned char *	var byte	integer	integer
PINT	int *	int *	var integer	integer	integer
PUINT	unsigned int *	unsigned int *	var word	long	long
PCHAR	char *	char *	var string	string	string

Tabelle 3-2: Type Declaration für Windows 95/NT

	Borland C	Microsoft C	Borland Pascal	Microsoft Visual Basic Dos	Microsoft Visual Basic Windows
VOID	void	void	pointer		any
BYTE	unsigned char	unsigned char	byte	integer	integer
INT	int	int	integer	integer	integer
UINT	unsigned int	unsigned int	long	long	long
LONG	long	long	longint	long	long
PBYTE	unsigned char *	unsigned char *	var byte	integer	integer
PINT	int *	int *	var integer	integer	integer
PUINT	unsigned int *	unsigned int *	var long	long	long
PCHAR	char *	char *	var string	string	string

Tabelle 3-3: Define-Wert

Define-Name	Dezimal-Wert	Hexadecimal-Wert
DLL_COMPILER_C	1	1
DLL_COMPILER_VB	2	2
DLL_COMPILER_PASCAL	3	3
DLL_LABVIEW	4	4

3.2 Software-Funktionen (API)

3.2.1 Initialisation

1) i_APCI1710_InitTTLIODirection (...)

Syntax:
<Return Wert> = i_APCI1710_InitTTLIODirection (BYTE b_BoardHandle,
BYTE b_ModulNbr,
BYTE b_PortAMode,
BYTE b_PortBMode,
BYTE b_PortCMode,
BYTE b_PortDMode)

Parameters:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte APCI-1710
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_PortAMode	Kanalkonfiguration für das TTL Port A 0: Als Eingangsport benutzt 1: Als Ausgangport benutzt
BYTE	b_PortBMode	Kanalkonfiguration für das TTL Port B 0: Als Eingangsport benutzt 1: Als Ausgangport benutzt
BYTE	b_PortCMode	Kanalkonfiguration für das TTL Port C 0: Als Eingangsport benutzt 1: Als Ausgangport benutzt
BYTE	b_PortDMode	Kanalkonfiguration für das TTL Port D 0: Als Eingangsport benutzt 1: Als Ausgangport benutzt

- Ausgabe:
Es erfolgt keine Ausgabe.

Aufgabe:
Konfiguriert den TTL I/O Betriebsmode für das ausgewählte Modul (*b_ModulNbr*).
Diese Funktion ist als Erste aufzurufen, bevor Sie eine andere Funktion aufrufen, die auf TTL I/O zugreift.

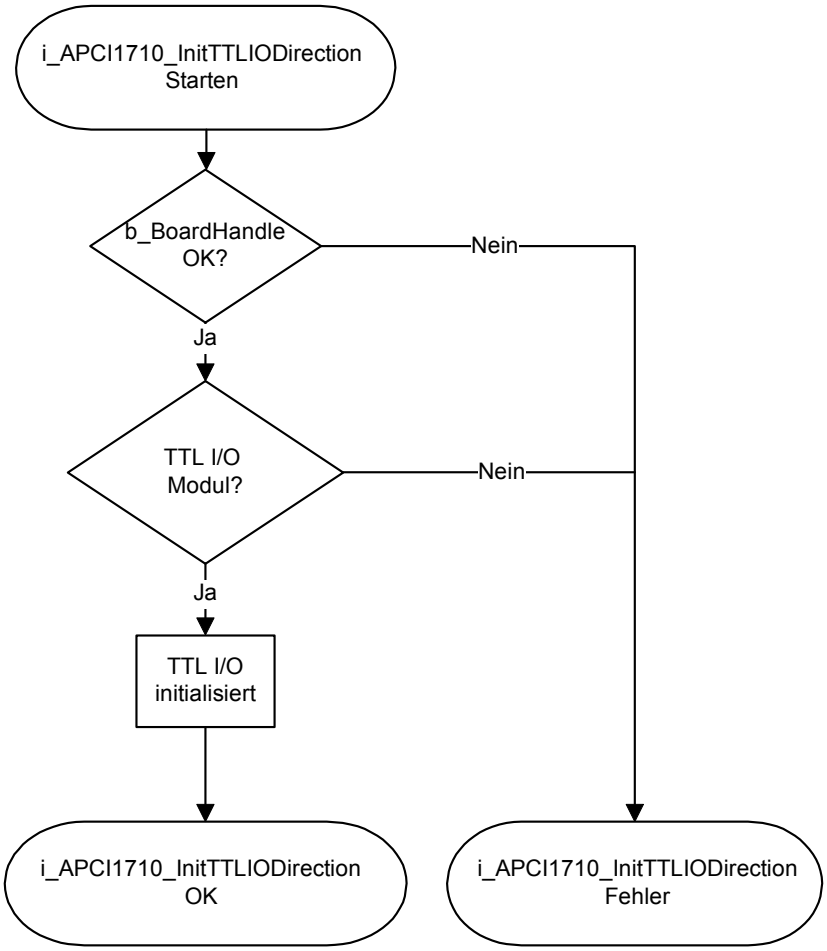
Funktionsaufruf:
ANSI C :

```
int             i_ReturnValue;
unsigned char   b_BoardHandle;

i_ReturnValue = i_APCI1710_InitTTLIODirection (b_BoardHandle,
                                               0,
                                               0,
                                               0,
                                               0,
                                               1);
```

Return Wert:
0: Kein Fehler
-1: Der Handle-Parameter der Karte ist falsch.
-2: Das ausgewählte Modul ist falsch.
-3: Das ausgewählte Modul ist kein TTL I/O Modul.
-4: Die Funktion ist nicht für diese Version verfügbar.
-5: Der ausgewählte Mode für das Port A ist falsch.
-6: Der ausgewählte Mode für das Port B ist falsch.
-7: Der ausgewählte Mode für das Port C ist falsch.
-8: Der ausgewählte Mode für das Port D ist falsch.

Eingabe	
b_BoardHandle	b_PortBMode
b_ModulNbr	b_PortCMode
b_PortAMode	b_PortDMode

Funktionsablauf



Ausgabe

<Return Wert>

3.2.2 TTL I/O lesen

1) i_APCI1710_ReadTTLIOChannelValue (...)

Syntax:
<Return Wert> = i_APCI1710_ReadTTLIOChannelValue
(BYTEb_BoardHandle,
BYTE b_ModulNbr,
BYTE b_SelectedPort,
BYTE b_InputChannel,
PBYTE pb_ChannelStatus)

Parameter:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte **APCI-1710**
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_SelectedPort	Auswahl des TTL I/O Ports (0 bis 3) 0: Port A 1: Port B 2: Port C 3: Port D
BYTE	b_InputChannel	Auswahl eines TTL Eingangskanals 0 bis 7 für Port A,B,C 0 oder 1 für Port D

- Ausgabe:

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

Aufgabe:
Gibt den Status des ausgewählten TTL Eingangs (*b_InputChannel*) für das entsprechende Modul (*b_ModulNbr*).

Funktionsaufruf:

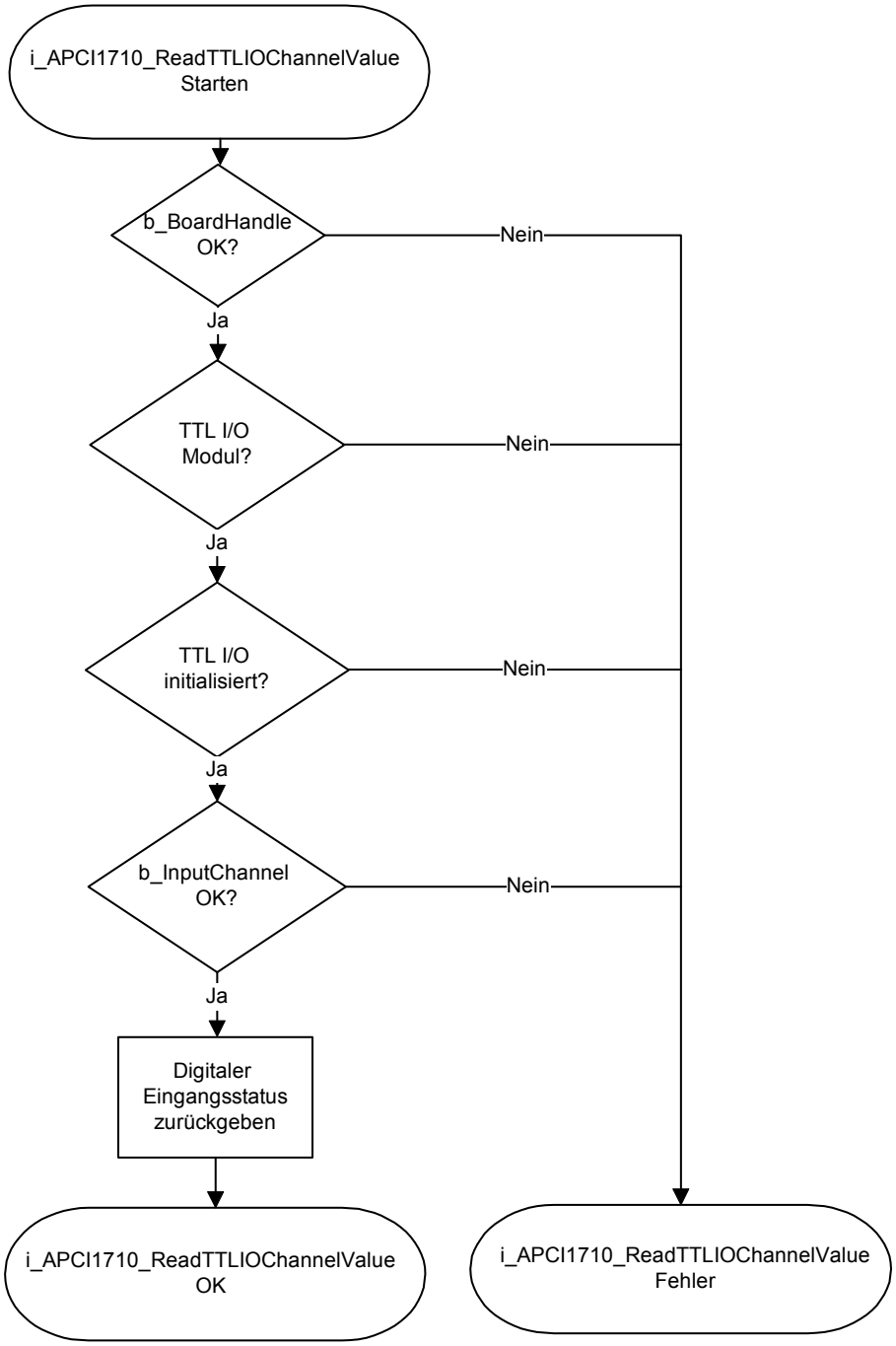
```
ANSI C :
int             i_ReturnValue;
unsigned char   b_BoardHandle;
unsigned char   b_ChannelStatus;

i_ReturnValue = i_APCI1710_ReadTTLIOChannelValue
                                        (b_BoardHandle,
                                         0, 0, 0,
                                        &b_ChannelStatus);
```

Return Wert:
0: Kein Fehler
-1: Der Handle-Parameter der Karte ist falsch.
-2: Das ausgewählte Modul ist falsch.
-3: Das ausgewählte Modul ist kein TTL I/O Modul.
-4: Das ausgewählte TTL Eingangsport ist falsch.
-5: Der ausgewählte TTL Eingang ist falsch.
-6: TTL I/O nicht initialisiert. Siehe Funktion "i_APCI1710_InitTTLIO"

Eingabe

b_BoardHandle	b_SelectedPort
b_ModulNbr	b_InputChannel

Funktionsablauf



Ausgabe

pb_ChannelStatus

<Return Wert>

2) i_APCI1710_ReadTTLIOPortValue (...)

Syntax:
<Return Wert> = i_APCI1710_ReadTTLIOPortValue
(BYTEb_BoardHandle,
BYTE b_ModulNbr,
BYTE b_SelectedPort,
PBYTE pb_PortValue)

Parameters:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte **APCI-1710**
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_SelectedPort	Selection of the TTL I/O port (0 bis 3) 0: Port A selection 1: Port B selection 2: Port C selection 3: Port D selection

- Ausgabe:

PBYTE	pb_PortValue	Status des digitalen TTL Eingangsports.

Aufgabe:
Gibt den Status des digitalen Eingangsports (*b_SelectedPort*) für das ausgewählte Modul (*b_ModulNbr*) zurück.

Funktionsaufruf:
ANSI C :

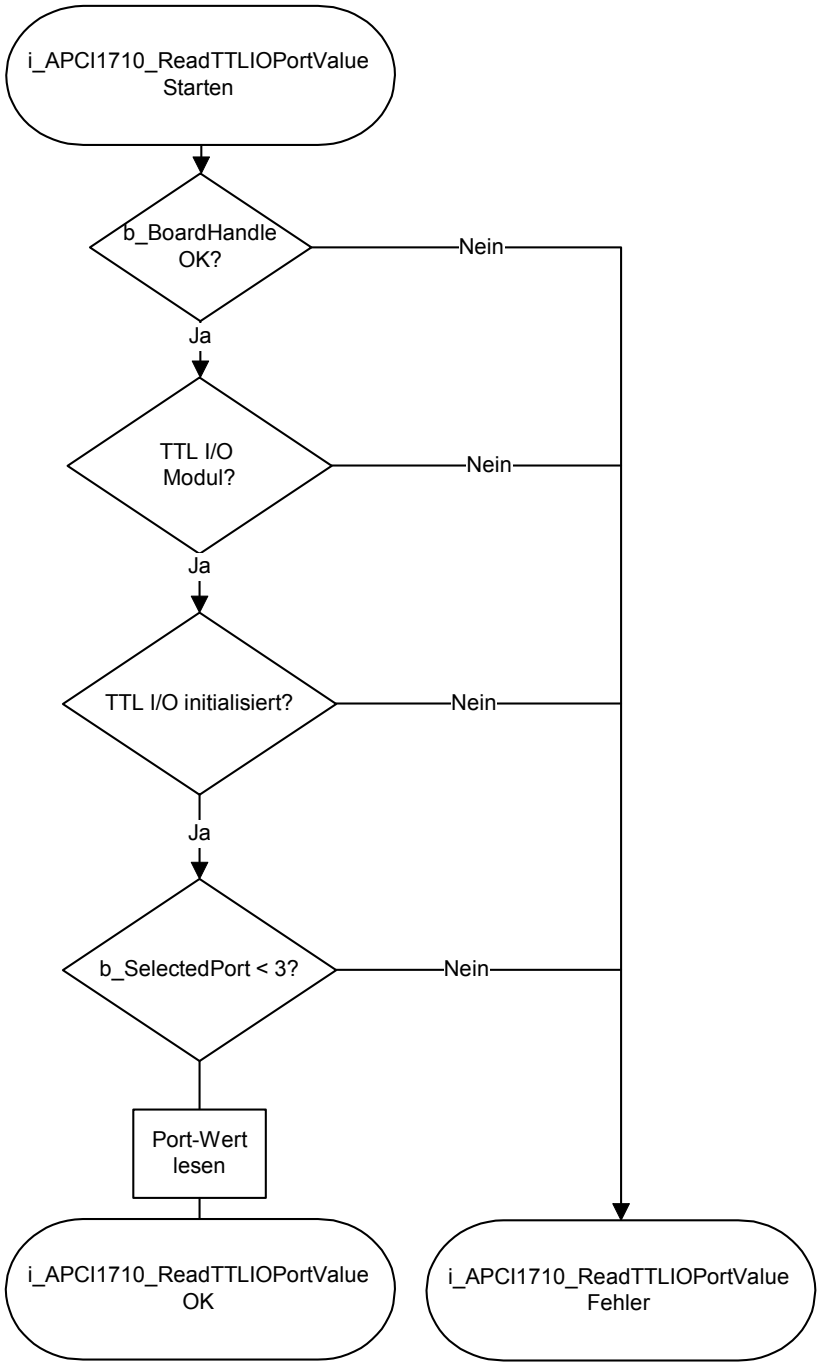
```
int             i_ReturnValue;
unsigned char   b_BoardHandle;
unsigned char   b_PortValue;

i_ReturnValue = i_APCI1710_ReadTTLIOPortValue (b_BoardHandle,
                                               0,
                                               0,
                                               &b_PortValue);
```

Return Wert:
0: Kein Fehler
-1: Der Handle-Parameter der Karte ist falsch.
-2: Das ausgewählte Modul ist falsch.
-3: Das ausgewählte Modul ist kein TTL I/O Modul.
-4: Das ausgewählte TTL Eingangsport ist falsch.
-5: TTL I/O nicht initialisiert. Siehe Funktion "i_APCI1710_InitTTLIO"

Eingabe

b_BoardHandle b_SelectedPort
b_ModulNbr

Funktionsablauf



Ausgabe

pb_PortValue

<Return Wert>

3) i_APCI1710_ReadTTLIOAllPortValue (...)

Syntax:
<Return Wert> = i_APCI1710_ReadTTLIOAllPortValue
(BYTEb_BoardHandle,
BYTE b_ModulNbr,
PULONG pul_AllPortValue)

Parameters:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte **APCI-1710**
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)

- Ausgabe:

PULONG	pul_AllPortValue	Status der digitalen TTL Eingangsport A,B, C und D.

Aufgabe:
Gibt den Status aller Eingangsports (A, B, C und D) für das ausgewählte TTL I/O Modul (*b_ModulNbr)*.

Funktionsaufruf:
ANSI C :

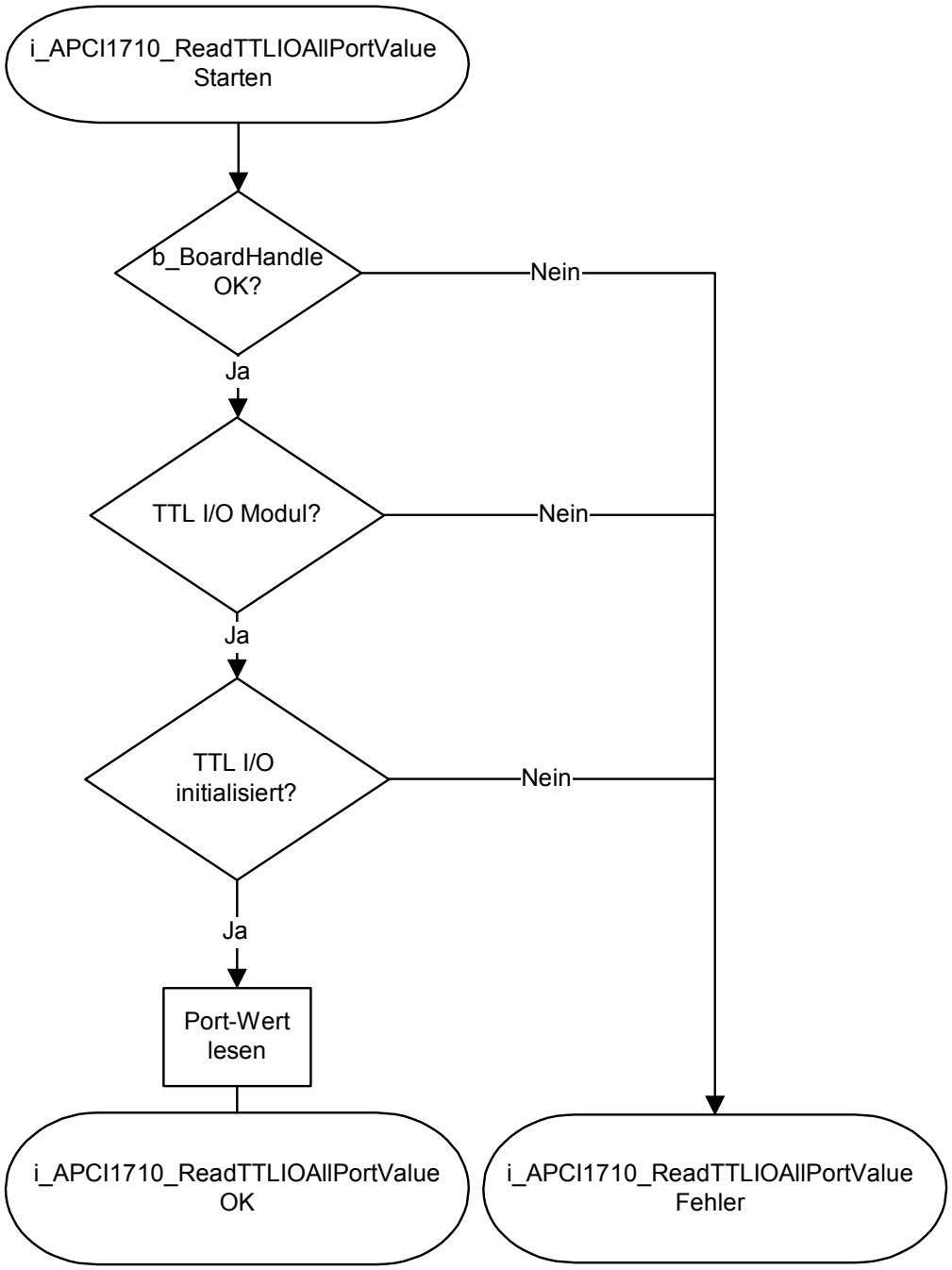
```
int             i_ReturnValue;
unsigned char   b_BoardHandle;
unsigned long   ul_AllPortValue;

i_ReturnValue = i_APCI1710_ReadTTLIOAllPortValue    (b_BoardHandle,
                                                     0,
                                                     & ul_AllPortValue);
```

Return Wert:
0: Kein Fehler
-1: Der Handle-Parameter der Karte ist falsch
-2: Das ausgewählte Modul ist falsch
-3: Das ausgewählte Modul ist kein TTL I/O Modul
-4: Die TTL I/O Funktion ist nicht initialisiert. Siehe Funktion "i_APCI1710_InitTTLIO"

Eingabe

b_BoardHandle
b_ModulNbr

Funktionsablauf



Ausgabe

PULONG pul_AllPortValue

<Return Wert>

3.2.3 TTL I/O schreiben

1) i_APCI1710_SetTTLIOChlOn (...)

Syntax:
<Return Wert> = i_APCI1710_SetTTLIOChlOn
(BYTEb_BoardHandle,
BYTE b_ModulNbr,
BYTE b_OutputChannel)

Parameters:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte **APCI-1710**
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_OutputChannel	Auswahl des digitalen Ausgangs (0 bis 25) 0: PD0 1: PD1 2 bis 9: PA0 bis PA7 10 bis 17: PB0 bis PB7 18 bis 25: PC0 bis PC7

- Ausgabe:
Es erfolgt keine Ausgabe.

Aufgabe:
Setzt den Ausgang *b_Channel*. Einen Ausgang setzten bedeutet auf High setzen.

Funktionsaufruf:
ANSI C :

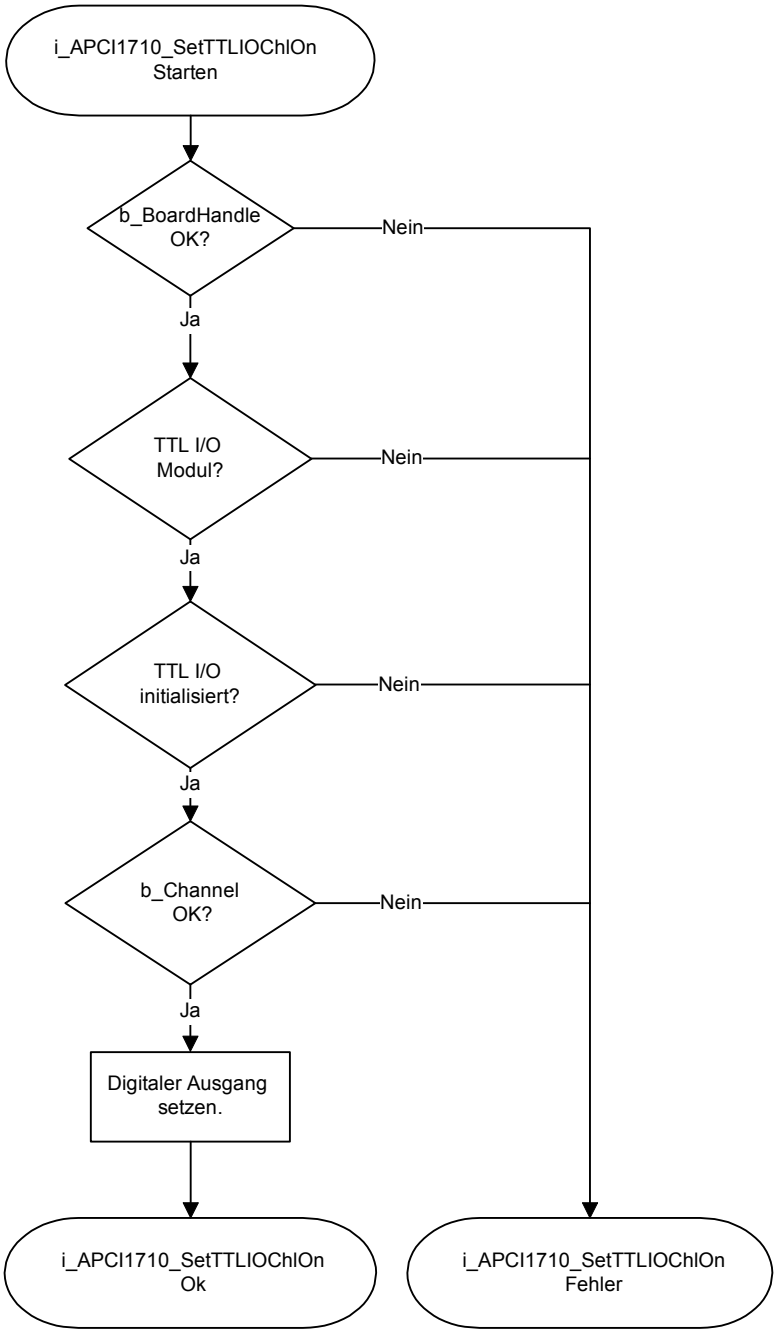
```
int             i_ReturnValue;
unsigned char   b_BoardHandle;

i_ReturnValue = i_APCI1710_SetTTLIOChlOn    (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 ausgewählte Modul ist kein TTL I/O Modul.
-4: Das ausgewählte TTL Ausgang ist falsch.
-5: Die TTL I/O Funktion ist nicht initialisiert.
Siehe Funktion "i_APCI1710_InitTTLIO"

Eingabe

b_BoardHandle b_OutputChannel
b_ModulNbr

Funktionsablauf



Ausgabe

<Return Wert>

2) i_APCI1710_SetTTLIOChlOff (...)

Syntax:
<Return Wert> = i_APCI1710_SetTTLIOChlOff
(BYTEb_BoardHandle,
BYTE b_ModulNbr,
BYTE b_OutputChannel)

Parameters:
- Eingabe:

BYTE	b_BoardHandle	Handle der Karte **APCI-1710**
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_OutputChannel	Auswahl des digitalen Ausgangs (0 bis 25) 0: PD0 1: PD1 2 bis 9: PA0 bis PA7 10 bis 17: PB0 bis PB7 18 bis 25: PC0 bis PC7

- Ausgabe:
Es erfolgt keine Ausgabe.

Aufgabe:
Setzt den Ausgang *b_Channel* zurück. Einen Ausgang zurücksetzen beduetet auf Low setzen.

Funktionsaufruf:
ANSI C :

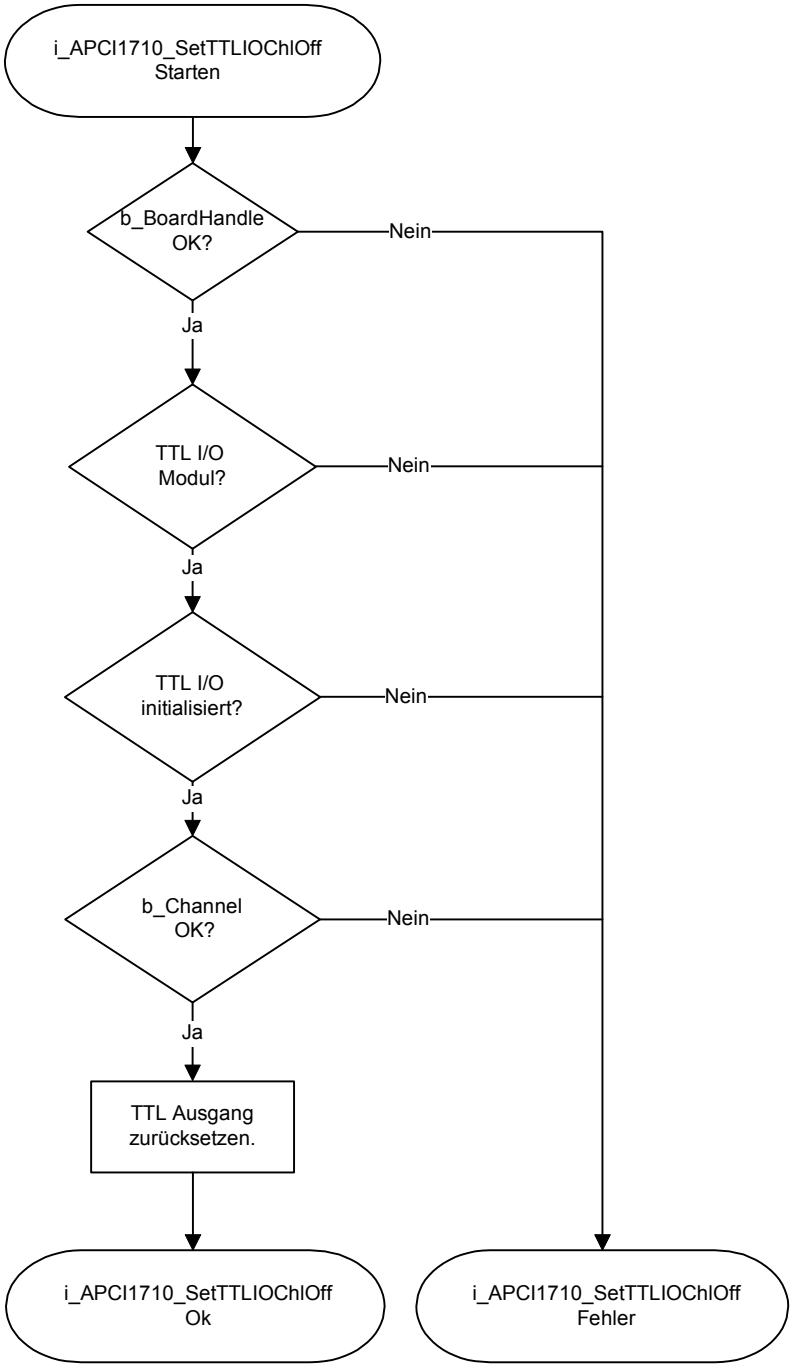
```
int             i_ReturnValue;
unsigned char   b_BoardHandle;

i_ReturnValue = i_APCI1710_SetTTLIOChlOff   (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 ausgewählte Modul ist kein TTL I/O Modul.
-4: Der ausgewählte TTL Ausgang ist falsch.
-5: Die TTL I/O Funktion ist nicht initialisiert.
Siehe Funktion "i_APCI1710_InitTTLIO"

Eingabe

b_BoardHandle b_OutputChannel
b_ModulNbr

Funktionsablauf



Ausgabe

<Return Wert>

3.2.4 Kernelfunktionen für Windows NT/95/98

IMPORTANT!
Diese Funktionen stehen nur für die Windows NT/95/98 Benutzer-Iinterruptroutine im synchronen Mode zur Verfügung.
Siehe Funktion "i_APCI1710_SetBoardIntRoutineWin32"

1) i_APCI1710_KRNL_ReadTTLIOChannelValue (...)

Syntax:

```
<Return Wert> = i _APCI1710_KRNL _ReadTTLIOChannelValue
                                (UINT    ui_BaseAddress,
                                 BYTE    b_ModulNbr,
                                 BYTE    b_SelectedPort,
                                 BYTE    b_InputChannel,
                                 PBYTE   pb_ChannelStatus)
```

Parameters:

- Eingabe:

UINT	ui_BaseAddress	Basisadresse der APCI-1710 Karte
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_SelectedPort	Auswahl des TTL I/O Ports (0 bis 3) 0: Port A 1: Port B 2: Port C 3: Port D
BYTE	b_InputChannel	Auswahl eines TTL Eingangskanals 0 bis 7 für Port A,B,C 0 oder 1 für Port D

- Ausgabe:

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

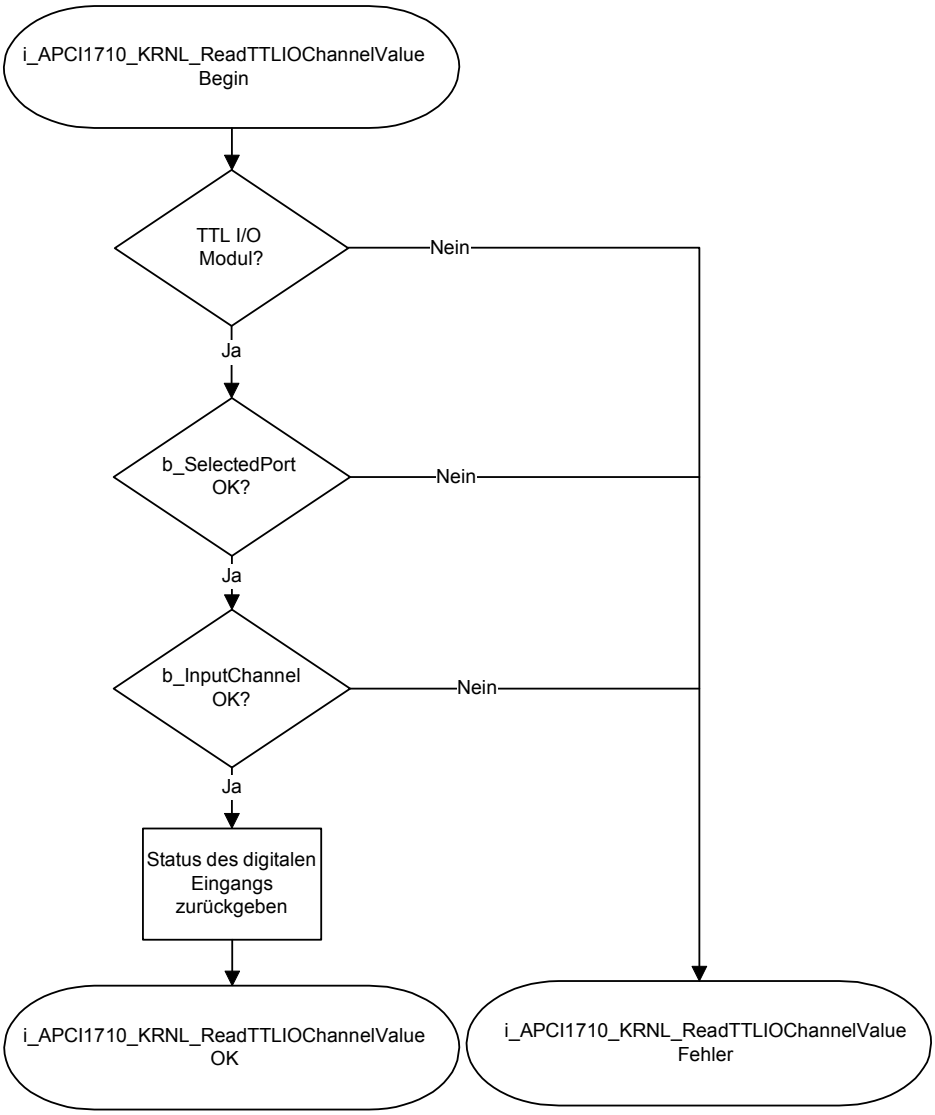
Aufgabe:
Gibt den Status des ausgewählten TTL Eingangs (*b_InputChannel*) für das entsprechende Modul (*b_ModulNbr*) zurück.

Funktionsaufruf:
ANSI C :

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

i_ReturnValue = i_APCI1710_KRNL _ReadTTLIOChannelValue
                                                (ui_BaseAddress,
                                                 0,
                                                 0,
                                                 0,
                                                &b_ChannelStatus);
```

Return Wert:
0: Kein Fehler
-1: Das ausgewählte Modul ist falsch
-2: Das ausgewählte Modul ist kein TTL I/O Modul.
-3: Das ausgewählte TTL Eingangsport ist falsch.
-4: Der ausgewählte TTL Eingang ist falsch.

Eingabe	
ui_BaseAddress b_ModulNbr	b_SelectedPort b_InputChannel
Funktionsablauf	



Ausgabe
pb_ChannelStatus
<Return Wert>

2) i_APCI1710_KRNL_ReadTTLIOPortValue (...)

Syntax:
<Return Wert> = i_APCI1710_KRNL_ReadTTLIOPortValue
(UINT ui_BaseAddress,
BYTE b_ModulNbr,
BYTE b_SelectedPort,
PBYTE pb_PortValue)

Parameters:
- Eingabe:

UINT	ui_BaseAddress	Basisadresse der APCI-1710 Karte
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_SelectedPort	Selection of the TTL I/O port (0 bis 3) 0: Port A selection 1: Port B selection 2: Port C selection 3: Port D selection

- Ausgabe:

PBYTE	pb_PortValue	Status des digitalen TTL Eingansports.

Aufgabe:
Gibt den Status des digitalen Eingangsports (*b_SelectedPort*) für das ausgewählte Modul (*b_ModulNbr*) zurück.

Funktionsaufruf:
ANSI C :

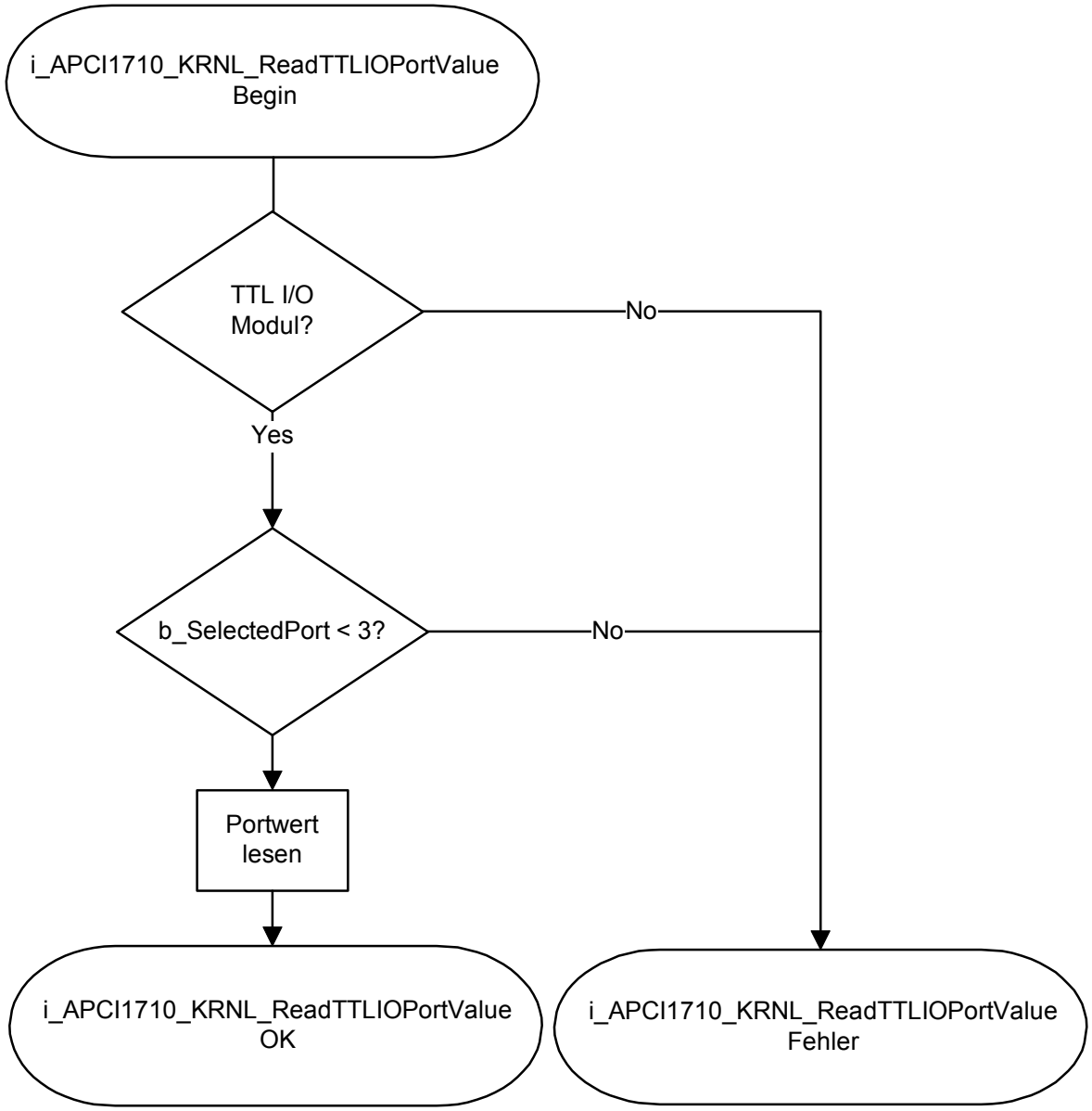
```
int             i_ReturnValue;
unsigned int    ui_BaseAddress;
unsigned char   b_PortValue;

i_ReturnValue = i_APCI1710_KRNL _ReadTTLIOPortValue
                                        (ui_BaseAddress,
                                         0,
                                         0,
                                         &b_PortValue);
```

Return Wert:
0: Kein Fehler
-1: Das ausgewählte Modul ist falsch
-2: Das ausgewählte Modul ist kein TTL I/O Modul.
-3: Das ausgewählte TTL Eingangsport ist falsch.

Eingabe

ui_BaseAddress b_SelectedPort
b_ModulNbr

Funktionsablauf



Ausgabe

pb_PortValue

<Return Wert>

3) i_APCI1710_KRNL_ReadTTLIOAllPortValue (...)

Syntax:
<Return Wert> = i_APCI1710_KRNL_ReadTTLIOAllPortValue
(UINT ui_BaseAddress,
BYTE b_ModulNbr,
PULONG pul_AllPortValue)

Parameters:
- Eingabe:

UINT	ui_BaseAddress	Basisadresse der APCI-1710 Karte
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)

- Ausgabe:

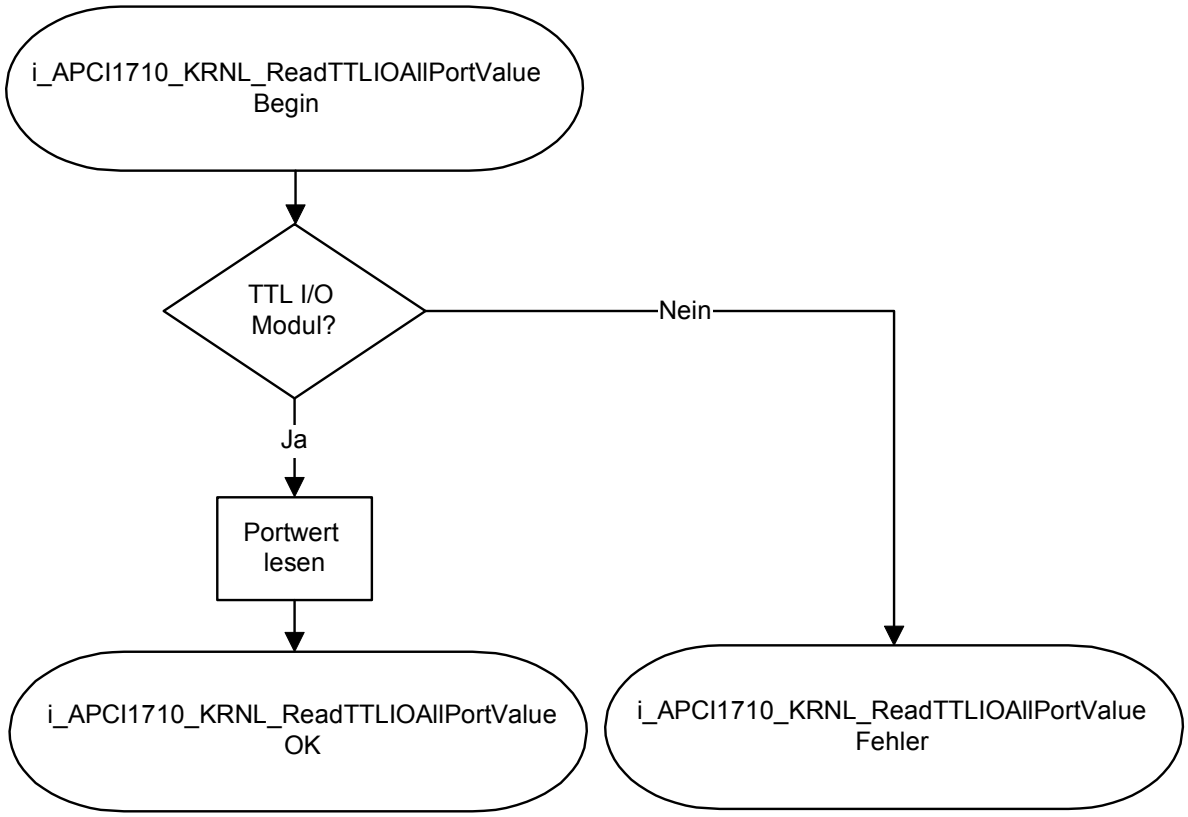
PULONG	pul_AllPortValue	Status der digitalen TTL Eingangsport A,B, C und D.

Aufgabe:
Gibt den Status aller Eingangsports (A, B, C und D) für das ausgewählte TTL I/O Modul (*b_ModulNbr*) zurück.

Funktionsaufruf:
ANSI C :

```
int             i_ReturnValue;
unsigned int    ui_BaseAddress;
unsigned long   ul_AllPortValue;

i_ReturnValue = i_APCI1710_KRNL _ReadTTLIOAllPortValue
                                        (ui_BaseAddress,
                                         0,
                                         &ul_AllPortValue);
```

Return Wert:
0: Kein Fehler
-1: Das ausgewählte Modul ist falsch
-2: Das ausgewählte Modul ist kein TTL I/O Modul.

Eingabe
ui_BaseAddress b_ModulNbr
Funktionsablauf

Ausgabe
pul_AllPortValue
<Return Wert>

4) i_APCI1710_KRNL_SetTTLIOChlOn (...)

Syntax:
<Return Wert> = i_APCI1710_KRNL_SetTTLIOChlOn
(UINT ui_BaseAddress,
BYTE b_ModulNbr,
BYTE b_OutputChannel)

Parameters:
- Eingabe:

UINT	ui_BaseAddress	Basisadresse der APCI-1710 Karte
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_OutputChannel	Auswahl des digitalen Ausgangs (0 bis 25) 0: PD0 1: PD1 2 bis 9: PA0 bis PA7 10 bis 17: PB0 bis PB7 18 bis 25: PC0 bis PC7

- Ausgabe:
Es erfolgt keine Ausgabe.

Aufgabe:
Setzt den Ausgang *b_Channel*. Einen Ausgang setzten bedeutet auf High setzen.

Funktionsaufruf:
ANSI C :

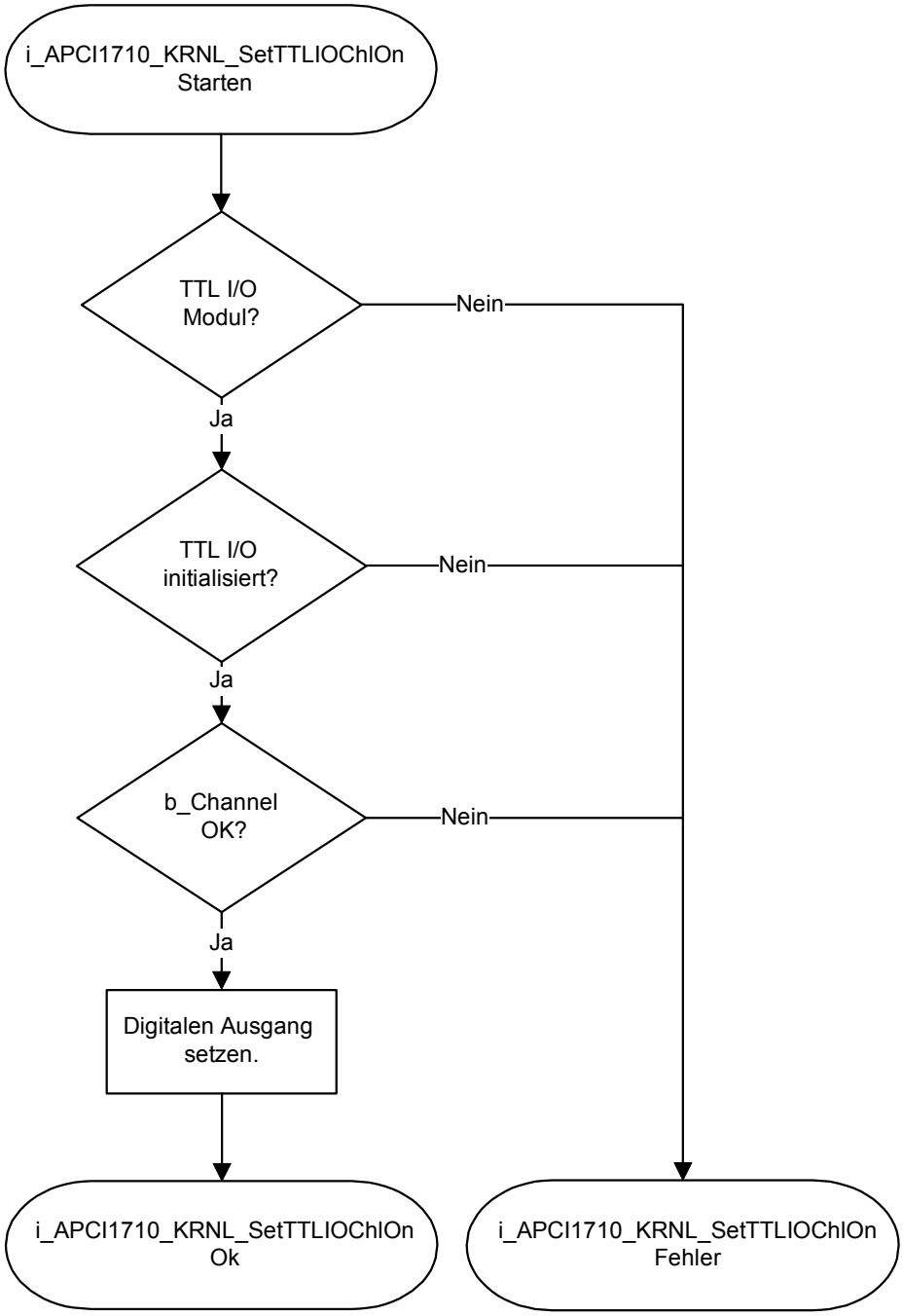
```
int             i_ReturnValue;
unsigned int    ui_BaseAddress;

i_ReturnValue = i_APCI1710_KRNL_SetTTLIOChlOn   (ui_BaseAddress,
                                                 0,
                                                 0);
```

Return Wert:
0: Kein Fehler
-1: Das ausgewählte Modul ist falsch
-2: Das ausgewählte Modul ist kein TTL I/O Modul.
-3: Das ausgewählte TTL Ausgang ist falsch.

Eingabe

ui_BaseAddress b_OutputChannel
b_ModulNbr

Funktionsablauf



Ausgabe

<Return Wert>

5) i_APCI1710_KRNL_SetTTLIOChlOff (...)

Syntax:
<Return Wert> = i_APCI1710_KRNL_SetTTLIOChlOff
(UINT ui_BaseAddress,
BYTE b_ModulNbr,
BYTE b_OutputChannel)

Parameters:
- Eingabe:

UINT	ui_BaseAddress	Basisadresse der APCI-1710 Karte
BYTE	b_ModulNbr	Nummer des zu konfigurierenden Moduls (0 bis 3)
BYTE	b_OutputChannel	Auswahl des digitalen Ausgangs (0 bis 25) 0: PD0 1: PD1 2 bis 9: PA0 bis PA7 10 bis 17: PB0 bis PB7 18 bis 25: PC0 bis PC7

- Ausgabe:
Es erfolgt keine Ausgabe.

Aufgabe:
Setzt den Ausgang *b_Channel* zurück. Einen Ausgang zurücksetzen beduetet auf Low setzen.

Funktionsaufruf:
ANSI C :

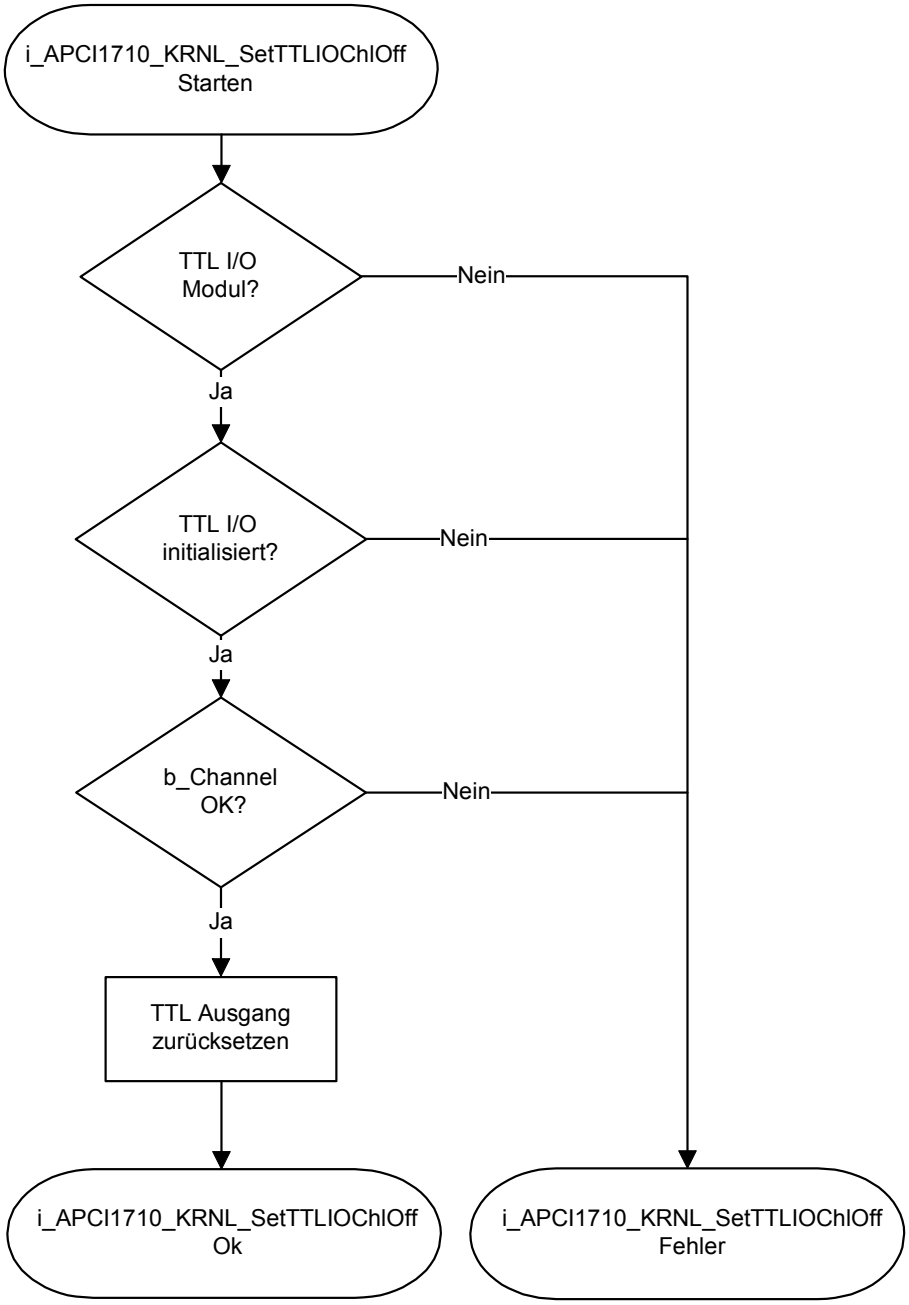
```
int             i_ReturnValue;
unsigned int    ui_BaseAddress;

i_ReturnValue = i_APCI1710_KRNL_SetTTLIOChlOff  (ui_BaseAddress,
                                                 0,
                                                 0);
```

Return Wert:
0: Kein Fehler
-1: Das ausgewählte Modul ist falsch
-2: Das ausgewählte Modul ist kein TTL I/O Modul.
-3: Der ausgewählte TTL Ausgang ist falsch.

Eingabe

ui_BaseAddress b_OutputChannel
b_ModulNbr

Funktionsablauf



Ausgabe

<Return Wert>