



DIN EN ISO 9001:2000  
certified



**ADDI-DATA GmbH**  
**Airpark Business Center**  
**Airport Boulevard B210**  
**77836 Rheinmünster**  
**Germany**



**Technical support:**  
**+49 7229 1847- 0**

## **Function description**

**APCI-/CPCI-1710**

**Multifunction counter board**  
**- Digital I/O -**

Edition: 02.02 - 05/2008

## Product information

This manual contains the technical installation and important instructions for correct commissioning and usage, as well as production information according to the current status before printing. The content of this manual and the technical product data may be changed without prior notice. ADDI-DATA GmbH reserves the right to make changes to the technical data and the materials included herein.

## Warranty and liability

The user is not permitted to make changes to the product beyond the intended use, or to interfere with the product in any other way.

ADDI-DATA shall not be liable for obvious printing and phrasing errors. In addition, ADDI DATA, if legally permissible, shall not be liable for personal injury or damage to materials caused by improper installation and/or commissioning of the board by the user or improper use, for example, if the board is operated despite faulty safety and protection devices, or if notes in the operating instructions regarding transport, storage, installation, commissioning, operation, thresholds, etc. are not taken into consideration. Liability is further excluded if the operator changes the board or the source code files without authorisation and/or if the operator is guilty of not monitoring the permanent operational capability of working parts and this has led to damage.

## Copyright

This manual, which is intended for the operator and its staff only, is protected by copyright. Duplication of the information contained in the operating instructions and of any other product information, or disclosure of this information for use by third parties, is not permitted, unless this right has been granted by the product licence issued. Non-compliance with this could lead to civil and criminal proceedings.

## ADDI-DATA software product licence

Please read this licence carefully before using the standard software. The customer is only granted the right to use this software if he/she agrees with the conditions of this licence.

The software must only be used to set up the ADDI-DATA boards.

Reproduction of the software is forbidden (except for back-up and for exchange of faulty data carriers). Disassembly, decompilation, decryption and reverse engineering of the software are forbidden. This licence and the software may be transferred to a third party if this party has acquired a board by purchase, has agreed to all the conditions in this licence contract and the original owner does not keep any copies of the software.

## Trademarks

- ADDI-DATA is a registered trademark of ADDI-DATA GmbH.
- Turbo Pascal, Delphi, Borland C, Borland C++ are registered trademarks of Borland Insight Company.
- Microsoft C, Visual C++, Windows XP, 98, Windows 2000, Windows 95, Windows NT, EmbeddedNT and MS DOS are registered trademarks of Microsoft Corporation.
- LabVIEW, LabWindows/CVI, DasyLab, Diadem are registered trademarks of National Instruments Corp.
- CompactPCI is a registered trademark of PCI Industrial Computer Manufacturers Group.
- VxWorks is a registered trademark of Wind River Systems Inc.

# WARNING

The following risks result from improper implementation and from use of the board contrary to the regulations:



- ◆ Personal injury
- ◆ Damage to the board, PC and peripherals
- ◆ Pollution of the environment

◆ **Protect yourself, the others and the environment!**

◆ **Read carefully the safety precautions (yellow leaflet).**

If this leaflet is not with the documentation, please contact us and ask for it.

◆ **Observe the instructions of the manual.**

Make sure that you do not forget or skip any step. We are not liable for damages resulting from a wrong use of the board.

◆ **Used symbols:**



**IMPORTANT!**

designates hints and other useful information.



**WARNING!**

It designates a possibly dangerous situation.

If the instructions are ignored the board, PC and/or peripheral may be destroyed.

<b>1</b>	<b>DEFINITION OF APPLICATION .....</b>	<b>6</b>
1.1	Intended use .....	6
1.2	Usage restrictions.....	6
1.3	Technical description .....	6
1.4	Function description .....	7
<b>2</b>	<b>DIGITAL INPUTS AND OUTPUTS.....</b>	<b>8</b>
2.1.1	Block diagram .....	8
2.1.2	Typical applications .....	9
<b>2.2</b>	<b>Used signals .....</b>	<b>9</b>
<b>2.3</b>	<b>Pin assignment of the front connector .....</b>	<b>10</b>
<b>2.4</b>	<b>Connection example .....</b>	<b>11</b>
<b>2.5</b>	<b>I/O mapping .....</b>	<b>12</b>
<b>2.6</b>	<b>Description of the I/O functions.....</b>	<b>12</b>
2.6.1	Select register .....	12
2.6.2	Output register .....	13
2.6.3	Input register .....	13
2.6.4	Recognition register (Base + 60).....	14
<b>2.7</b>	<b>Working with the “digital I/O function” .....</b>	<b>14</b>
<b>3</b>	<b>STANDARDSOFTWARE .....</b>	<b>15</b>
<b>3.1</b>	<b>Introduction .....</b>	<b>15</b>
<b>3.2</b>	<b>Software functions.....</b>	<b>16</b>
3.2.1	Initialisation.....	16
	1) i_APCI1710_InitDigitalIO (...) .....	16
3.2.2	Reading the digital I/O .....	17
	2) i_APCI1710_ReadDigitalIOChlValue (...) .....	17
	3) i_APCI1710_ReadDigitalIOPortValue (...) .....	18
3.2.3	Writing digital outputs.....	19
	4) i_APCI1710_SetDigitalIOMemoryOn (...) .....	19
	5) i_APCI1710_SetDigitalIOMemoryOff (...) .....	20
	6) i_APCI1710_SetDigitalIOChlOn (...) .....	21
	7) i_APCI1710_SetDigitalIOChlOff (...) .....	22
	8) i_APCI1710_SetDigitalIOPortOn (...) .....	23
	9) i_APCI1710_SetDigitalIOPortOff (...).....	24
3.2.4	Using the functions in the kernel mode .....	25
	Reading the inputs.....	25
	10) i_APCI1710_KRNL_ReadDigitalIOChlValue (...) .....	25
	11) i_APCI1710_KRNL_ReadDigitalIOPortValue (...).....	26
	Writing the outputs .....	27
	12) i_APCI1710_KRNL_SetDigitalIOChlOn (...) .....	27
	13) i_APCI1710_KRNL_SetDigitalIOPortOn (...) .....	28

**Figures**

Fig. 2-1: Block diagram of the function "digital I/O" ..... 8  
 Fig. 2-2: Pin assignment of the front connector ..... 10  
 Fig. 2-3: Connection example ..... 11

**Tables**

Table 1-1: Delivered manuals ..... 7  
 Table 2-1: Used signals..... 9  
 Table 2-2: I/O mapping of the function „digital I/O“ ..... 12  
 Table 3-1: Define Value..... 15

# 1 DEFINITION OF APPLICATION

## 1.1 Intended use

The board **APCI-1710** must be inserted in a PC with PCI 5V/32-bit slots, which is used as electrical equipment for measurement, control and laboratory pursuant to the norm IEC 61010-1.

The board **CPCI-1710** must be inserted in a CompactPCI system with PCI 5V/32-bit slots, which is used as electrical equipment for measurement, control and laboratory pursuant to the norm IEC 61010-1.

## 1.2 Usage restrictions

The board **APCI-/CPCI-1710** must not be used as safety related part for securing emergency stop functions

The board **APCI-/CPCI-1710** must not be used in potentially explosive atmospheres.

## 1.3 Technical description

This manual refers to the **APCI-1710** as well as to the **CPCI-1710** board. Make sure that you have received the following items:

- The CD 1 "Standard Software Drivers" with the ADDISET parameterize program and the required software drivers.
- The CD 2 "Technical Manuals". This CD contains the following:

- 1) The technical description **ADDICOUNT APCI-1710 / CPCI-1710: Function-programmable counter board for the PCI bus** (containing general information on the operation of the board)
- 2) A function description for each function which you want to program on the board
- 3) The yellow leaflet "Safety precautions"

According to the used function you will find the required assignment and programming functions in the different manuals for each function:

Table 1-1: Delivered manuals

Function	PDF file (CD2 technical manuals)		Function description in SET1710	CFG file
	German	English		
Incremental counter	Inkr_zähler_d.pdf	Incr_counter_e.pdf	Incremental counter	inc_cpt.cfg
SSI	SSI_d.pdf	ssi_e.pdf	SSI	ssi.cfg
SSI monitor	SSI-Monitor_d	SSIMonitor_e.pdf	SSI_Monitor	ssi_mon.cfg
Chronos	chronos_d.pdf	chronos_e.pdf	Chronos	chronos.cfg
Counter/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_IO_d.pdf	TTL_IO_e.pdf	TTL I/O	tll_io.cfg
Digital I/O	dig_EA_d.pdf	dig_IO_e.pdf	Digital I/O	dig_IO.cfg
Pulse counter	Impulszähler_d.pdf	pulseCounter_e.pdf	Pulse counter	imp_cpt.cfg
ETM (Edge time measurement)	ETM_d.pdf	ETM_e.pdf	Edge time measurement	etm.cfg

**Please note:**

The board CPCI-1710 is compatible with the board APCI-1710 as far as the installation of the software is concerned. The ADDIREG and SET1710 programs make no difference between PCI and CompactPCI boards.

The API functions of the standard software are also identical.

## 1.4 Function description

Apart from a global description of the functions this manual contains:

- the pin assignment of the front connector
- a list of the used signals
- the I/O mapping
- a chapter about the API software functions of the standard software.

## 2 DIGITAL INPUTS AND OUTPUTS

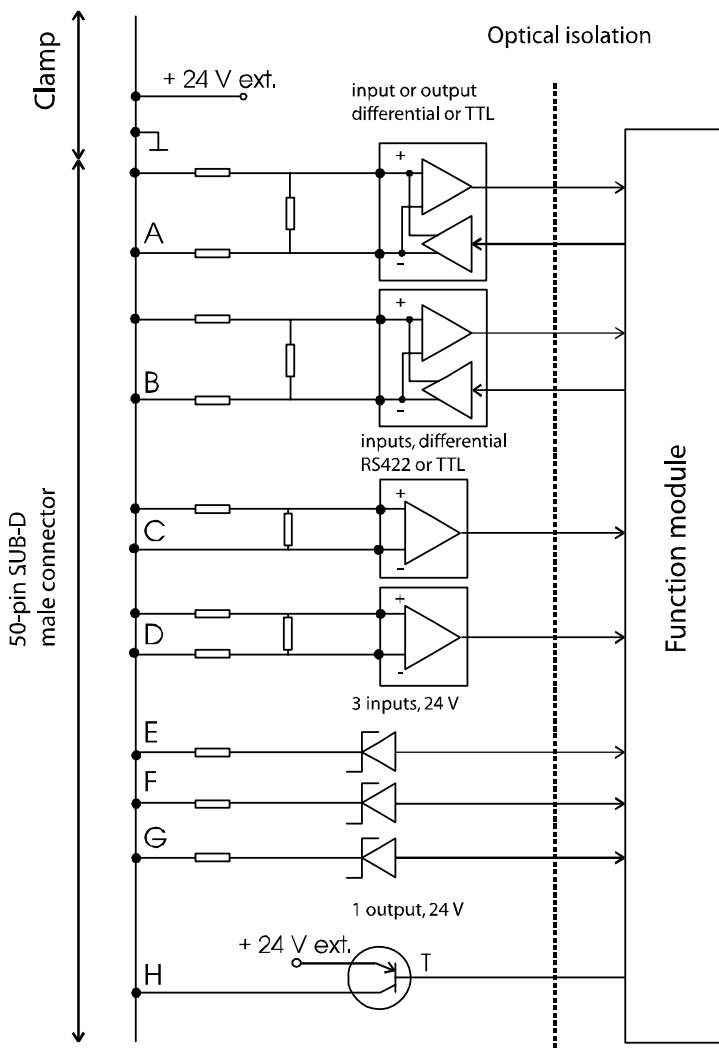
The function "Digital Inputs/Outputs" allows a comfortable control or monitoring of single differential, TTL or 24 V signals.

### Properties:

- 2 x differential RS422/RS485 inputs (24 V at the APCI-1710-24, or optional for the CPCI-1710 or CPCI-1711), channels C and D
- 2 x differential RS422/RS485 inputs/outputs (only for the APCI-1710 or CPCI-1710); selectable as input or output through software, channels A and B
- 3 x 24 V inputs, channels E, F, G (optional 5 V)
- 1 x 24 V output (H, optional 5 V)

### 2.1.1 Block diagram

Fig. 2-1: Block diagram of the function "digital I/O"



## 2.1.2 Typical applications

The function “digital inputs/outputs“ completes a complex counting application with additional 24 V, TTL or differential RS422 inputs and outputs signals.

- Comfortable monitoring of 24 V signals,RS422, TTL signals
- Comfortable control of 24 V, TTL, RS422, differential signals.

## 2.2 Used signals

The function „digital inputs/outputs“ occupies **5 to 7 inputs (A to G) and 1 to 3 outputs (A, B and H)** of the respecting function module of the **APCI-/CPCI-1710**.

**Table 2-1: Used signals**

<b>AT THE CONNECT OR</b>	<b>POLARITY</b>	<b>FUNCTION</b>
<b>A x +/-</b>	Diff./TTL/Opt. 24 V*	Digital input/output, switchable by software
<b>B x +/-</b>	Diff./TTL/ Opt. 24 V*	Digital input/output, switchable by software
<b>C x +/-</b>	Diff./TTL/Opt. 24 V	Digital input
<b>D x +/-</b>	Diff./TTL/Opt. 24 V	Digital input
<b>E x</b>	24 V/ Opt. 5V	Digital input
<b>F x</b>	24 V/Opt. 5V	Digital input
<b>G x</b>	24V/Opt. 5V	Digital input
<b>H x</b>	24 V/Opt. TTL	Digital output

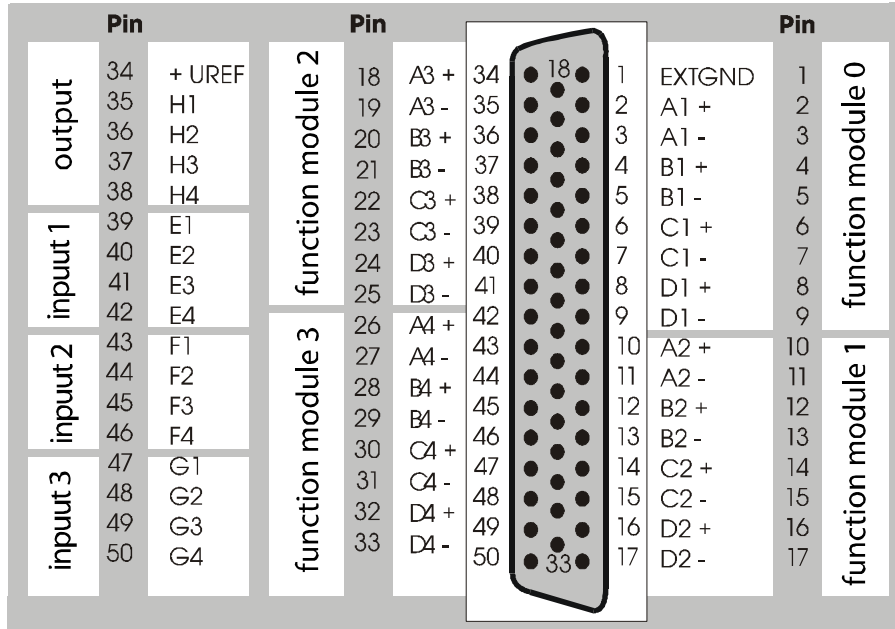
\*: only for the APCI-1710 or CPCI-1710/1711

x: Number of the function module.

## 2.3 Pin assignment of the front connector

The figure below is a connection example. The function “digital inputs/outputs” is implemented on all function modules.

Fig. 2-2: Pin assignment of the front connector



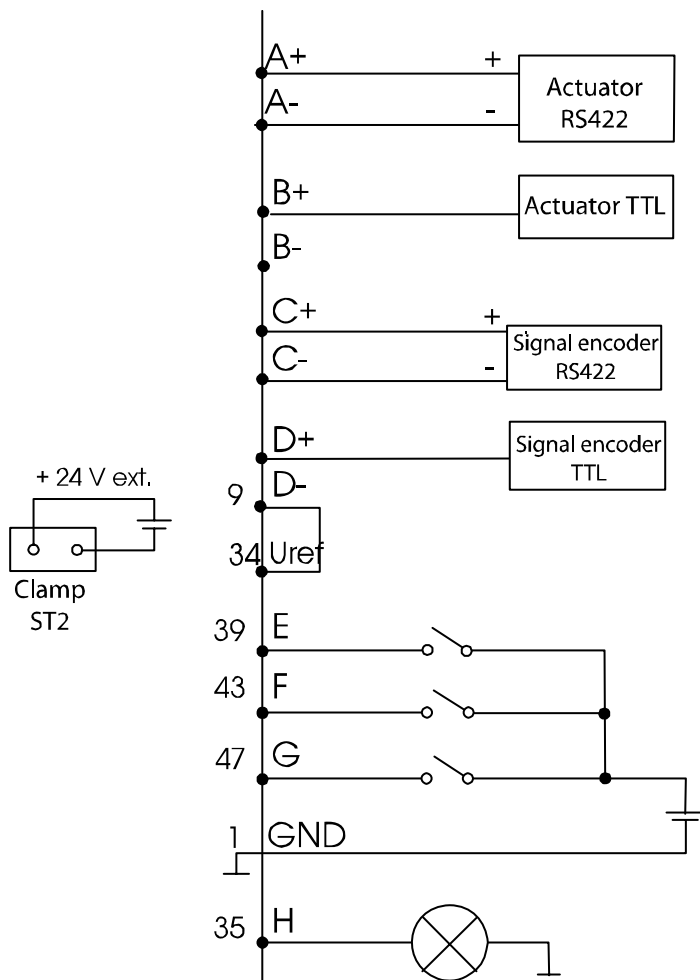
## 2.4 Connection example

Programming (module 1)

A: Output

B: Input

Fig. 2-3: Connection example



## 2.5 I/O mapping

Table 2-2: I/O mapping of the function „digital I/O“

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

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

-: No function; y: Data not relevant, x: Number of the function module.

The accesses are always read or written in 32-bit

## 2.6 Description of the I/O functions

### 2.6.1 Select register

8-bit register; sets the polarity of the input/output channels A and B.

This register can only be written. After a reset the value is set on "0", i.e. the line is set as input channel.

Bit	Logic value	Meaning
BIT D0	0	I/O lines A are set as input channel
	1	I/O lines A are set as output channel
BIT D1	0	I/O lines B are set as input channel
	1	I/O lines B are set as output channel

## 2.6.2 Output register

### Base address + 0:

8-bit register for controlling the digital outputs.

This register can only be written. After a reset the value is set to „0“, i.e. the outputs are set to „Low“.

Bit	Logic value	Meaning
BIT D0	0	Output H is set to "Low"
	1	Output H is set to "High"
BIT D1	0	Output A is set to "Low" (differential/digital)
	1	Output A is set to "High" (differential/digital)
BIT D2	0	Output B is set to "Low" (differential/digital)
	1	Output B is set to "High" (differential/digital)

## 2.6.3 Input register

### Base address + 0:

8-bit register for reading the digital inputs. This register can be only read and informs about the state of the digital inputs.

Bit	Logic value	Meaning
BIT D0	0	Input C is set to "Low" (diff.)
	1	Input C is set to "High" (diff.)
BIT D1	0	Input D is set to "Low" (diff.)
	1	Input D is set to "High" (diff.)
BIT D2	0	Input E is set to "High" (>17 V)
	1	Input E is set to "Low" or input voltage is <14 V
BIT D3	0	Input F is set to "High" (>17 V)
	1	Input F is set to "Low" or input voltage is <14 V
BIT D4	0	Input G is set to "High" (>17 V)
	1	Input G is set to "Low" or input voltage is <14 V
BIT D5	0	Input A is set to "Low" (diff.)
	1	Input A is set to "High" (diff.)
BIT D6	0	Input A is set to "Low" (diff.)
	1	Input A is set to "High" (diff.)

### 2.6.4 Recognition register (Base + 60)

Function description and revision (read command, ASCII format)

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

Meaning: Digital I/O revision 1.1

## 2.7 Working with the "digital I/O function"

1. Connecting the signals
2. Initialising A and B channels as input or output
3. Controlling the outputs
4. Reading the inputs

## 3 STANDARDSOFTWARE

### 3.1 Introduction



#### **IMPORTANT!**

Note the following style conventions in the text:

Function: "i\_APCI1710\_SetBoardInformation"

Variable *ui\_Address*

**Table 3-1: Define Value**

<b>Define name</b>	<b>Decimal value</b>	<b>Hexadecimal value</b>
DLL_COMPILER_C	1	1
DLL_COMPILER_VB	2	2
DLL_COMPILER_PASCAL	3	3
DLL_LABVIEW	4	4

## 3.2 Software functions

### 3.2.1 Initialisation

#### 1) i\_APCI1710\_InitDigitalIO (...)

##### Syntax:

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

##### Parameter

##### -Input:

BYTE	b_BoardHandle	Handle parameter of the APCI-1710
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_ChannelAMode	Mode of channel A 0: Channel that is used as digital input 1: Channel is used as digital output.
BYTE	b_ChannelBMode	Mode of channel B 0: Channel is used as digital input 1: Channel is used as digital output

##### -Output:

There is no output.

##### Task:

Configures the operation mode of the digital input/output channels of the selected module (*b\_ModulNbr*).

The function has to be called before calling another function that accesses the I/O channels.

##### Calling convention:

ANSI C:

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

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

##### Return value:

0: No error  
-1: The handle parameter of the board is wrong  
-2: Module selection is wrong  
-3: The selected module is no "digital I/O" module.  
-4: Configuration of channel A is wrong  
-5: Configuration of channel B is wrong

### 3.2.2 Reading the digital I/O

#### 2) `i_APCI1710_ReadDigitalIOChIValue (...)`

**Syntax:**

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

**Parameter:**

**-Input:**

BYTE	b_BoardHandle	Handle parameter of the APCI-1710
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_InputChannel	Selection of the digital input (0 to 6)
		0: Channel C
		1: Channel D
		2: Channel E
		3: Channel F
		4: Channel G
		5: Channel A
		6: Channel B

**-Output:**

PBYTE	pb_ChannelStatus	Status of the digital input
		0: Channel is disabled
		1: Channel is enabled

**Task:**

Shows the status of the selected digital input (*b\_InputChannel*) of the selected module (*b\_ModulNbr*).

**Calling convention:**

ANSI C:

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

**Return value:**

0: No error  
-1: Handle parameter of the board is wrong  
-2: Module selection is wrong  
-3: The selected module is no "Digital I/O" module.  
-4: The selected digital input is wrong  
-5: Digital I/O function not initialised. See function "i\_APCI1710\_InitDigitalIO".  
-6: Channel A is switched as digital output  
-7: Channel B is switched as digital output.

### 3) i\_APCI1710\_ReadDigitalIOPortValue (...)

**Syntax:**

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

**Parameter****-Input:**

BYTE	b_BoardHandle	Handle parameter of the APCI-1710
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)

**-Output:**

PBYTE	pb_PortValue	Status of the digital input port
-------	--------------	----------------------------------

**Task:**

Shows the status of the digital input port of the selected module (*b\_ModulNbr*).

**Calling convention:**

ANSI C:

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

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

**Return value:**

- 0: No error
- 1: Handle parameter of the board is wrong
- 2: Module selection is wrong
- 3: The selected module is no "Digital I/O" module.
- 4: Digital I/O function not initialised. See function "i\_APCI1710\_InitDigitalIO".

### 3.2.3 Writing digital outputs

#### 4) i\_APCI1710\_SetDigitalIOMemoryOn (...)

##### Syntax:

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

##### Parameter

##### -Input:

BYTE	b_BoardHandle	Handle parameter of the <b>APCI1710</b>
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)

##### -Output:

There is no output.

##### Task:

Enables the digital output memory. After calling the function, the outputs that were set with the function "i\_APCI1710\_SetDigitalIOOutputXOn" will not be reset. You can reset them with the function "\_APCI1710\_SetDigitalIOOutputXOff".

##### Calling convention:

##### ANSI C:

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

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

##### Return value:

- 0: No error.
- 1: Handle parameter of the board is wrong.
- 2: Module selection is wrong
- 3: The selected module is no "Digital I/O" module.
- 4: Digital I/O function is not initialised. See function "i\_APCI1710\_InitDigitalIO".

## 5) i\_APCI1710\_SetDigitalIOMemoryOff (...)

### Syntax:

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

### Parameter

#### -Input:

BYTE	b_BoardHandle	Handle parameter of the <b>APCI1710</b>
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)

#### -Output:

There is no output.

### Task:

Disables the digital output memory.

### Calling convention:

ANSI C:

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

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

### Return value:

- 0: No error.
- 1: Handle parameter of the board is wrong.
- 2: Module selection is wrong
- 3: The selected module is no "Digital I/O" module.
- 4: Digital I/O function not initialised. See function "i\_APCI1710\_InitDigitalIO".

## 6) i\_APCI1710\_SetDigitalIOChlOn (...)

### Syntax:

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

### Parameter

#### -Input:

BYTE	b_BoardHandle	Handle parameter of the APCI-1710
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_OutputChannel	Selection of the digital output (0 to 2) 0: Channel H 1: Channel A 2: Channel B

#### -Output:

There is no output.

### Task:

Sets the output that is entered in the parameter *b\_Channel*.  
Setting an output means setting the output to „High“.

### Calling convention:

#### ANSI C:

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

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

### Return value:

0: No error  
-1: Handle parameter of the board is wrong  
-2: Module selection is wrong.  
-3: The selected module is no "Digital I/O" module.  
-4: The selected digital output is wrong.  
-5: Digital I/O function not initialised. See function "i\_APCI1710\_InitDigitalIO".  
-6: Digital channel A is used as input.  
-7: Digital channel B is used as input.

## 7) i\_APCI1710\_SetDigitalIOChlOff (...)

### Syntax:

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

### Parameter

#### -Input:

BYTE	b_BoardHandle	Handle parameter of the APCI-1710
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_OutputChannel	Selection of the digital output channel (0 to 2)
		0: Channel H
		1: Channel A
		2: Channel B

#### -Output:

There is no output.

### Task:

Resets the output that is inserted in the parameter *b\_Channel*. Resetting means setting the output to LOW.

### Calling convention:

#### ANSI C:

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

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

### Return value:

0: No error  
-1: Handle parameter of the board is wrong  
-2: Module selection is wrong  
-3: The selected module is no "Digital I/O" module.  
-4: The selected digital output is wrong  
-5: Digital I/O function not initialised. See function "i\_APCI1710\_InitDigitalIO".  
-6: Digital channel A is used as input.  
-7: Digital channel B is used as input.  
-8: Digital output memory is switched off. Firstly call the function "i\_APCI1710\_SetDigitalIOMemoryOn".

**8) i\_APCI1710\_SetDigitalIOPortOn (...)****Syntax:**

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

**Parameter****-Input:**

BYTE	b_BoardHandle	Handle parameter of the APCI-1710
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_PortValue	Output value (0 to 7)

**-Output:**

There is no output.

**Task:**

Sets one or various output ports. Setting an output means setting an output on "High". When you have switched off the digital output memory ("OFF"), all other channels will be set to "0".

**Calling convention:**ANSI C:

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

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

**Return value :**

- 0: No error
- 1: Handle parameter of the board is wrong
- 2: Module selection is wrong
- 3: The selected module is no "Digital I/O" module.
- 4: Output value is wrong
- 5: Digital I/O function not initialised. See function "i\_APCI1710\_InitDigitalIO".
- 6: Digital channel A is used as input.
- 7: Digital channel B is used as input.

## 9) i\_APCI1710\_SetDigitalIOPortOff (...)

### Syntax:

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

### Parameter

#### -Input:

BYTE	b_BoardHandle	Handle parameter of the APCI-1710
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_PortValue	Output value (0 to 7)

#### -Output:

There is no output.

### Task:

Resets one or various output ports (to LOW).

### Calling convention:

ANSI C:

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

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

### Return value:

0: No error  
-1: Handle parameter of the board is wrong.  
-2: Module selection is wrong.  
-3: The selected module is no "Digital I/O" module.  
-4: Output value is wrong.  
-5: Digital I/O function not initialised. See function "i\_APCI1710\_InitDigitalIO".  
-6: The digital channel A is used as input.  
-7: The digital input B is used as input.  
-8: Digital output memory is switched off. First, call the function "i\_APCI1710\_SetDigitalIOMemoryOn".

### 3.2.4 Using the functions in the kernel mode



#### IMPORTANT!

These functions are only available for the user interrupt routine under Windows NT and Windows 95/98 in the synchronous mode.

See function "i\_APCI1710\_SetBoardIntRoutineWin32".

#### Reading the inputs

#### 10) i\_APCI1710\_KRNL\_ReadDigitalIOChIValue (...)

##### Syntax:

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

##### Parameter

##### -Input:

UINT	ui_BaseAddress	Base address of the board
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_InputChannel	Selection of the digital input (0 to 6)

##### -Output:

PBYTE	pb_ChannelStatus	Status of the digital input 0: Channel disabled 1: Channel enabled.
-------	------------------	---

##### Task:

Returns the status of the digital input channel (*b\_InputChannel*) for the indicated module (*b\_ModulNbr*).

##### Calling convention:

##### ANSI C:

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

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

##### Return value:

0: No error  
-1: Module selection is wrong  
-2: The selected module is no "Digital I/O" module.  
-3: The selected digital input channel is wrong  
-4: Channel A is switched as output  
-5: Channel B is switched as digital output.

**11) i\_APCI1710\_KRNL\_ReadDigitalIOPortValue (...)****Syntax:**

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

**Parameter****-Input:**

UINT	ui_BaseAddress	Base address of the board
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)

**-Output:**

PBYTE	pb_PortValue	Status of the digital input port.
-------	--------------	-----------------------------------

**Task:**

Returns the status of the selected digital input port for the indicated module (*b\_ModulNbr*).

**Calling convention:**ANSI C:

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

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

**Return value:**

0: No error  
-1: Module selection is wrong  
-2: The selected module is no "Digital I/O" module.

Writing the outputs

12) `i_APCI1710_KRNL_SetDigitalIOChlOn (...)`

**Syntax:**

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

**Parameter**

**Input:**

UINT	ui_BaseAddress	Base address of the board
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_OutputChannel	Selection of the digital output channel (0 to 2) 0: Channel H 1: Channel A 2: Channel B

**Output:**

There is no output.

**Task:**

Sets the output that is entered in the parameter *b\_Channel* .  
Setting an output means setting an output on "High".

**Calling convention:**

ANSI C:

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

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

**Return Value:**

- 0: No error
- 1: Module selection is wrong
- 2: The selected module is no "Digital I/O" module.
- 3: The selected digital output is wrong.
- 4: Digital channel A is used as input.
- 5: Digital channel B is used as input.

**13) i\_APCI1710\_KRNL\_SetDigitalIOPortOn (...)****Syntax:**

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

**Parameter****-Input:**

UINT	ui_BaseAddress	Base address of the board
BYTE	b_ModulNbr	Number of the module to be configured (0 to 3)
BYTE	b_PortValue	Output value (0 to 7)

**-Output:**

There is no output.

**Task:**

Sets one or various output ports. Setting means setting an output to „High“. All other outputs are set to “0”.

**Calling convention:**ANSI C:

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

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

**Return value:**

- 0: No error
- 1: Module selection is wrong
- 2: The selected module is no "Digital I/O" module.
- 3: Output value is wrong.
- 4: Digital channel A is used as input.
- 5: Digital channel B is used as input.