ADDI-DATA®

☎ Technical support:
+49 (0)7223 / 9493-0

Technical description

PA 1508

Digital I/O board, optically isolated

Edition: 02.03 – 11/2006

Copyright

Guarantee and responsibility

Basically are effective our "general terms of delivery and payment". The manager receives them at the latest with the invoice. Claims for guarantee and responsibility in case of injuries and material damages are excluded, if they are due to one or some of the following causes:

- if the board has not been used for the intended purpose
- improper installation, operation and maintenance of the board
- if the board has been operated with defective safety devices or with not appropriate or nonfunctioning safety equipment
- nonobservance of the instructions concerning: transport, storage, inserting the board, use, limit values, maintenance, device drivers
- altering the board at the user's own initiative
- altering the source files at the user's own initiative
- not checking properly the parts which are subject to wear
- disasters caused by the intrusion of foreign bodies and by influence beyond the user's control.

Licence for ADDI-DATA software products

Read carefully this licence before using the standard software. The right for using this software is given to the customer, if he/she agrees to the conditions of this licence.

- this software can only be used for configuring ADDI-DATA boards.
- copying the software is forbidden (except for archiving/ saving data and for replacing defective data media).
- deassembling, decompiling, decoding and reverse engineering of the software are forbidden.
- this licence and the software can be transfered to a third party, so far as this party has purchased a board, declares to agree to all the clauses of this licence contract and the preceding owner has not kept copies of the software.

Trademarks

Borland C and Turbo Pascal are registered trademarks of Borland International, INC.
Burr-Brown is a registered trademark of Burr-Brown Corporation
Intel is a registered trademark of Intel Corporation
AT, IBM, ISA and XT are registered trademarks of International Business Machines Corporation
Microsoft, MS-DOS, Visual Basic and Windows are registered trademarks of Microsoft Corporation

The original version of this manual is in German. You can obtain it on request.

ADDI-DATA®

Declaration of Conformity

Document-Number/Month-Year: B-25809 / 06.1998

Manufacturer/Importer: ADDI-DATA GmbH
Dieselstraße 3
D-77833 OTTERSWEIER

Type: **PA 1508**

Product description: **Board to be inserted in an ISA slot of a PC**
8 digital inputs, 8 digital outputs
Watchdog function

The above named product complies with the following European directives:

Directive 72/23/EEC of 19 February 1973 on the harmonization of the laws of Member States relating to electrical equipment designed for use within certain voltage limits.

Directive 89/336/EEC of 3 May 1989 on the approximation of the laws of the Member States relating to electromagnetic compatibility.

The following norms have been applied:

IEC 61010-1 2002-08

IEC 61326-2 2004

2004/11/10
Date

Legally valid signature of the manufacturer

WARNING

In case of improper handling and if the board is not used for the purpose it is intended for:

people may be injured	**the board, PC and peripheral devices may be damaged**	**the environment may be polluted**

★★★ Protect yourself, others and the environment★★★

- **Read the yellow safety leaflet carefully !**
  If this leaflet is not with the documentation , please contact us.

- **Do observe the instructions in the manual !**
  Make sure that you have not skipped any step. We are not liable for damage resulting from the improper use of the board.

- **Symbols used**

⚠	**WARNING!** designates a possibly dangerous situation. If the instructions are ignored **the board, PC and/or peripheral devices may be damaged**.
i	**IMPORTANT!** designates hints and other useful information.

- **Do you have any questions?**
  Our technical support is always glad to help you.

1 INTENDED PURPOSE OF THE BOARD 1
1.1 Limits of use 2
2 USER 3
2.1 Qualification 3
2.2 Personal protection 3
2.3 In case of emergency 3
3 HANDLING THE BOARD 4
4 TECHNICAL DATA 5
4.1 Electromagnetic compatibility (EMC) 5
4.2 Physical set-up of the board 5
4.3 Limit values..6
4.4 Component scheme 9
5 INSTALLATION 10
5.1 Setting the base address through DIP switches 10
5.1.1 Examining the resources 10
5.1.2 Modifying the base address 11
5.2 Inserting the board 12
5.2.1 Opening the PC 12
5.2.2 Plugging the board into the slot 13
5.2.3 Closing the PC 13
5.3 Installing the software 14
5.3.1 Software installation under MS-DOS and Windows 3.11 14
5.3.2 Software installation under Windows NT / 95 14
5.4 Board configuration with ADDIREG 15
5.4.1 Program description 15
5.4.2 Registering a new board 18
5.4.3 Changing the registration of a board 19
5.4.4 Removing the ADDIREG program 19
5.4.5 Software downloads from the Internet 20
6 CONNECTION TO THE PERIPHERAL 21
6.1 Connector pin assignment 21
6.2 Connection principle 22
6.3 Connection examples 24
6.3.1 Connection to the screw terminal panel PX 9100-D 24

7 FUNCTIONS OF THE BOARD..25
7.1 Output channels..25
7.2 Diagnostic..25
7.3 Watchdog..26
7.4 Input channels..27
8 CORRECTION OF FAILURES..28
8.1 The PC can no longer be initialised. 28
8.2 The outputs do not react any more..28
8.3 The inputs deliver wrong values. 29
8.4 By operating the board, failures sometimes occur. 29
9 STANDARD SOFTWARE..30
9.1 Compatibility of the driver..30
9.2 Conventions..30
9.2 DIN 66001- Graphical symbols..32
9.3 Software functions (API)..33
9.3.1 Base address..33
1) i_PA1508_SetBoardInformation (...)..33
2) i_PA1508_CloseBoardHandle (..)..33
9.3.2 Digital inputs..34
1)i_PA1508_Read1DigitalInput (...)..34
2)i_PA1508_Read8DigitalInput (...)..34
9.3.3 Digital outputs..35
1)i_PA1508_SetOutputMemoryOn (...)..35
2)i_PA1508_SetOutputMemoryOff (...)..35
3)i_PA1508_Set1DigitalOutputOn (...)..35
4)i_PA1508_Set1DigitalOutputOff (...)..36
5)i_PA1508_Set8DigitalOutputOn (...)..36
6)i_PA1508_Set8DigitalOutputOff (...)..37
9.3.4 Watchdog..38
1) i_PA1508_EnableOutputsWatchdog (...)..38
2) i_PA1508_DisableOuputsWatchdog (..)..38
3) i_PA1508_RearmOuputsWatchdog..39
4) i_PA1508_ReadOuputsWatchdogStatus..39
9.4 Examples..40
9.4.1 Digital input channels..40
1) Determining the status of an input chanel..40
Example in C..40
2) Determining the state of 8 input channels..41
Example in C..41

9.4.2 Digital output channels........42
1) Setting a digital output channel and then resettting it (the other output channels are reset)........42
Example in C........42
2) Setting a digital output channel and then resetting it (the other outputs are not reset)........43
Example in C........43
3) Setting several digital output channel and then resettting them (The others are not modified)........44
Example in C........45
4) Setting simultaneously several digital outputs and then resetting them........46
Example in C........46
9.4.3 Watchdog management........47
Example in C........47

INDEX ....A

GLOSSARY........C

SYSTEM ADDRESS RANGES........E

Figures

Fig. 3-1: Handling the board 4
Fig. 4-1 Close delay relative to the load current 7
Fig. 4-2: Component scheme 9
Fig. 5-1: DIP switches 11
Fig. 5-2: types of slots 12
Fig. 5-4: Inserting the board 13
Fig. 5-5: Securing the board at the back cover 13
Fig. 5-6: ADDIREG registration program 15
Fig. 5-7: Configuring a new board 17
Fig. 6-1: Pin assignment of the 37-pin SUB-D male connector 21
Fig. 6-2: connection principle of the output channels 22
Fig. 6-2: connection principle of the output channels (continued) 23
Fig. 6-3: Connection principle of the input channels 23
Fig. 6-4: low ohmic power connection 24
Fig. 6-5: Economical solution for low currents 24

Tables

Table 5-1: Decoding table 11
Table 9-1: Type Declaration for Dos and Windows 3.1X 30
Table 9-2: Type Declaration for Windows 95/NT 31

1 INTENDED PURPOSE OF THE BOARD

The **PA 1508** board is the interface between the processing electronics of the PC and an industrial process. The board is to be used in a free PC ISA slot.

The PC is to comply with the EU directive 89/336/EEC and the specifications for EMC protection.

Products complying with these specifications bear the normed CE mark.

The **PA 1508** board integrated in a PC is used as an automation interface for any stored program control. An 8-bit data exchange occurs over the data bus through parallel data input/output. This data exchange is supported through the delivered driver commands.
All driver commands are intended to be used with an application program written especially for the respective application of the board **PA 1508**.

Please use the driver appropriate to your operating system and select the programming language you want to use. Please use the Debug-possibilities of your PC environment.

The commands of the **PA 1508** board are intended to be used as follows:

Outputs channels:
In case of overtemperature, undervoltage, overloading, time-out or voltage loss, the output channels switch off. The use without misfunction is ensured through an appropriate cabling and an adapted program control.

Watchdog:
The watchdog function is particularly recommended when the output channels execute control functions. Make sure with a control program that the output channels are updated within a period of 4.6 s minimum once.

Diagnostic:
The diagnostic is meaningful when the output channels are used for controlling. The diagnostic signals must especially be analysed in case of an increasing temperature of the site, of important charges or of charges with high starting currents. A fall in the external distribution voltage may also be noticed.

Input channels:
According to the intended purpose of the board, the input channels are used for the detection of signals.
When the output channels execute important control functions, we recommend
- using the input channels to read the outputs signal,
- setting at least 3 input channels as diagnostic control.

Please only use the board:
- in conditions providing absolute security
- in a closed housing which is adequately protected against environmental influences
- **with the accessories** we recommend.

The use of the board according to its intended purpose includes observing every advice given in this manual and in the safety leaflet.

Uses beyond these specifications are not allowed. The manufacturer is not liable for any damages which would result from the non-observance of this clause.

1.1 Limits of use

The board PA 1508 is not to be used for securing emergency stop functions.

No emergency stop functions **are to be controlled through the output channels.**

The board must not operate in a PC which does not provide for an adequate protection against environmental conditions (e.g.: liquids, dust, ...) and which is not equipped with an appropriate aeration system.

Make sure that your PC is equipped with a safety system and a shielded metal housing.
The installation of the board in sites lying under risk of explosion is excluded.
The board should not be submitted to vibrations without additional fixation.

WARNING!
The EMC tests have been carried out in a specific appliance configuration. We guarantee these limit values **only** in this configuration.

The tested appliance configuration is at your disposal on request.

The use of the board in a PC could change the PC features regarding noise emission and immunity. Increased noise emission or decreased noise immunity could result in the system not being conform anymore.

Please control that the PC-housing and the cable are effectively shielded before putting the device into operation.

Make sure that the board remains in its protective blister pack **until it is used**.

Do not remove or alter the identification numbers of the board. If you do, the guarantee expires.

2 USER

2.1 Qualification

Only a person trained in electronics is entitled to install the board in a PC, to put it into operation, to operate and to stop it.

The basic knowledge of a high-level programming language is sufficient for programming the board.
The knowledge of conversion between the different numerative systems (binary, decimal, hexadecimal) is necessary.
The basic knowledge of EMC is necessary to operate under the CE conformity.

2.2 Personal protection

Consider the country-specific regulations about

- the prevention of accidents
- electrical and mechanical installations
- radio interference suppression

Clothing: antistatic work-clothes (cotton-made clothes, shoes with conductive soles) and earthing strip on the wrist are strictly recommended before putting the board into operation.

Work conditions: Select a work place which protects you from electrostatic charges (conductive table, table layers out of cotton, conductive floor layers, ...)

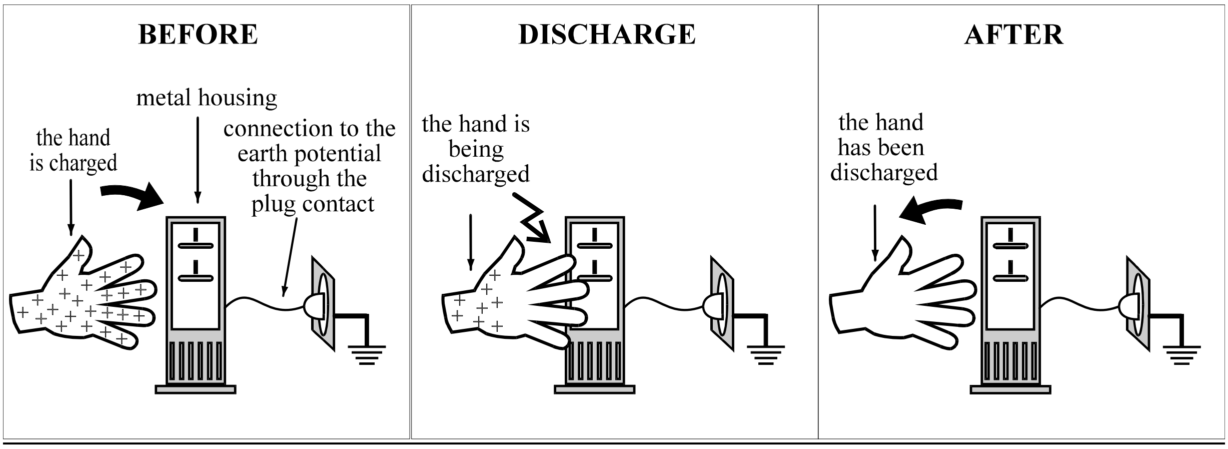
2.3 In case of emergency

System failure: Plug the connector and the cable out. Wait 10 min before removing the board from the PC.
Send the board back to us for inspection and/or repair.

In case of fire, dangerous vapours and gas can occur. Disconnect the PC from all external connections and plug the connector out. Only use non-residual carbon dioxide to extinguish the flames.
Consider the regulations of the professional association of your sector.

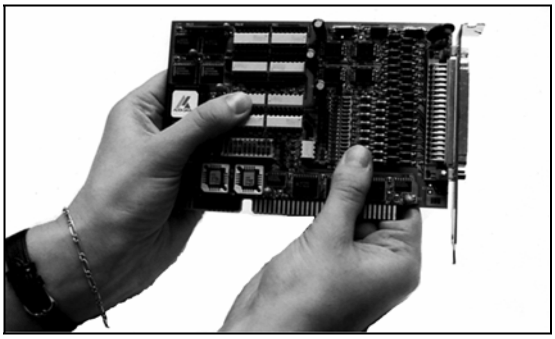
3 HANDLING THE BOARD

Fig. 3-1: Handling the board

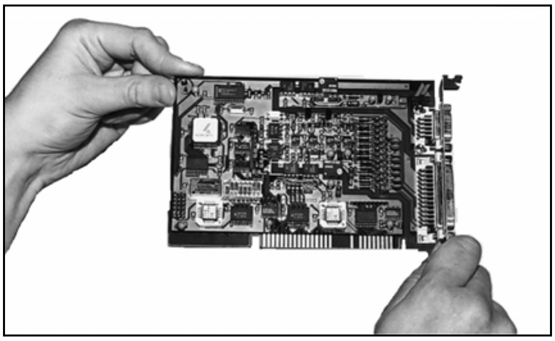


Discharge yourself; Wear an earthing strip.
Put your hand on an earthed, metallized surface (e.g.: radiator, tap, PC housing), before touching the board. If you are electrostatically charged, this could lead to an electric shock. Make sure your state of health allows it. Persons with cardiac problems or pacemaker act at their own risk.

Avoid the direct touching of the board



WRONG



RIGHT

4 TECHNICAL DATA

4.1 Electromagnetic compatibility (EMC)

The board has been subjected to EMC tests in an accredited laboratory.
In the tested appliance configuration, the emission limit values of the board complies with the norms EN55011 and EN55022.

The board complies with the limit values of the norm EN50082-2 as follows:

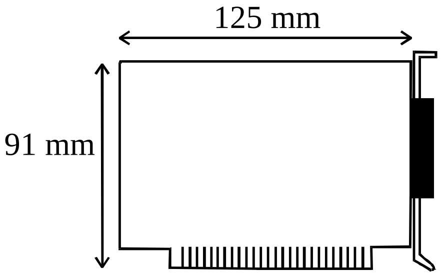
	Real value	Set value
ESD	4 kV	4 kV
Fields	10 V/m	10 V/m
Burst	4 kV	2 kV
Conducted radio interferences	10 V	10 V

WARNING!
The EMC tests have been carried out in a specific appliance configuration. We guarantee these limit values **only** in this configuration.

Consider the following aspects:
- your test program must be able to detect operation errors.
- your system must be set up so that you can find out what caused errors.

4.2 Physical set-up of the board

The board is assembled on a 2-layer printed circuit card.



Weight:	102 g
Installation in	XT or AT slot
Connection to the peripheral:	50-pin SUB-D male connector
Accessories:	Screw terminal panel **PX 9100** Cables: **ST010**, **ST010-S**, **ST011** See connection examples.

4.3 Limit values

Environmental conditions

Max. altitude:	up to 2000 m
Operating temperature:	0 to 60°C
Storage temperature:	-25 to 70°C
Relative humidity:	30 to 99% non condensing

Minimum PC requirements:

- Operating system:	MS DOS 3.3
- Slot types:	ISA (XT)

Energy requirements

Operating voltage Vcc of the PC:	5 V ± 5%
Current consumption Vcc (without load) typically:	20 mA ±10%
External operating voltage 0 V ext.:	10-30 V
Current consumption (0 V ext.):	according to the load

ISA-Bus interface

Bus speed:	8 MHz
Data bus access:	8-bit
Address decoding:	A1-A10 through DIP switches A11-A15 through jumper

Digital input channels

Number of input channels:	8 (separate grounds)
Nominal voltage:	24 VDC
Input current by nominal voltage:	6 mA
Logic input level:	U_H[1] max.: 30 V U_H min.: 17 V U_L[2] max.: 14 V U_L min.: 0 V
Switching threshold:	typical 16.5 V
Signal delay:	70 µs (at nominal voltage)
Max. input frequency:	5 kHz (at nominal voltage)

Digital output channels

Number of output channels:	8
Type of output channels:	High Side (load at ground)
Nominal voltage:	24 VDC
Distribution voltage range:	10 V up to 36 VDC min. 5 V
Max. current for the 8 outputs:	3 A typ. (with safety)

[1] U_H: Input voltage which corresponds to logic "1"

[2] U_L: Input voltage which corresponds to logic "0"

Digital output channels (continued)

Outputs 1 and 5[1]: .... max. output current: 1 A

Outputs 2, 3, 4, 6, 7, 8: .... max. output current: 500 mA

Short-circuit current by 24 V, R_{last} < 0,1 R: .... 1.5 A max.
(output channel switches off)

ON resistance of the output channel

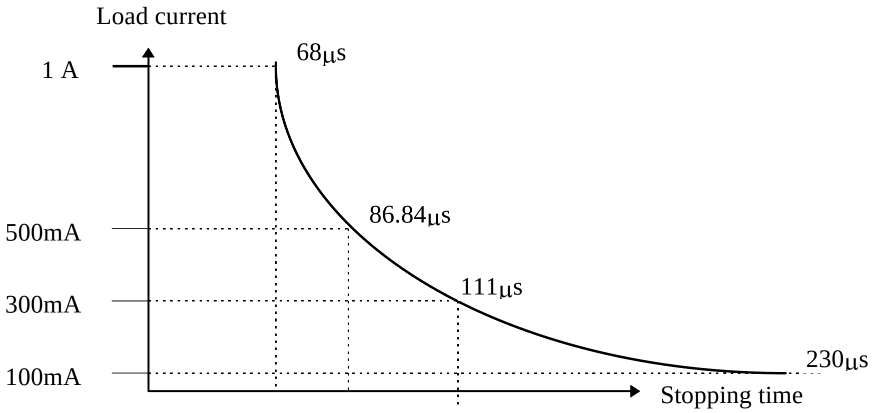
(R_{DS} ON switch): .... 0.4 R max.

Switching characteristics of the output channels
(0 V ext. = 24 V, T = 25°C, ohmic charge)

Start delay[2]: .... approx. 34 µs

Close delay: .... approx. 85 µs

Fig. 4-1 Close delay relative to the load current



Safety
Optical isolation
(DIN VDE 0411-100): .... 1000 V (from PC to peripheral)
Control voltage: .... 2000 V

Protection against overtemperature[3]

Activated at about: .... 150-170°C (chip temperature)

Deactivated (automatically) at about: .... 125-140°C (chip temperature)

Outputs (by overtemperature): .... output channels switch off

Diagnostic output channels: .... LOW level (within 100 ms)

Protection against overload (0 V ext.)

max. holding current (Ih): .... 3 A
max. switching current: .... 6 A

Typ. switching time with 5 x Ih: .... 7 s

[1] Outputs 1 and 5 are driven in the connector by 2 pins .

[2] Largely depending on the load

[3] In case of short-circuit or current limiting phase

Protection against overvoltage by Varistor (V ext)

Overload current Imax. 8/20 µs:	2500 A
Energy absorption W max. (2 ms):	15 J
Tension of the varistor ΔV (1 mA):	82 V by Tansorb diodes
Peak pulse Power:	1500 W @ 1 ms
Breakdown voltage, nominal:	36 V

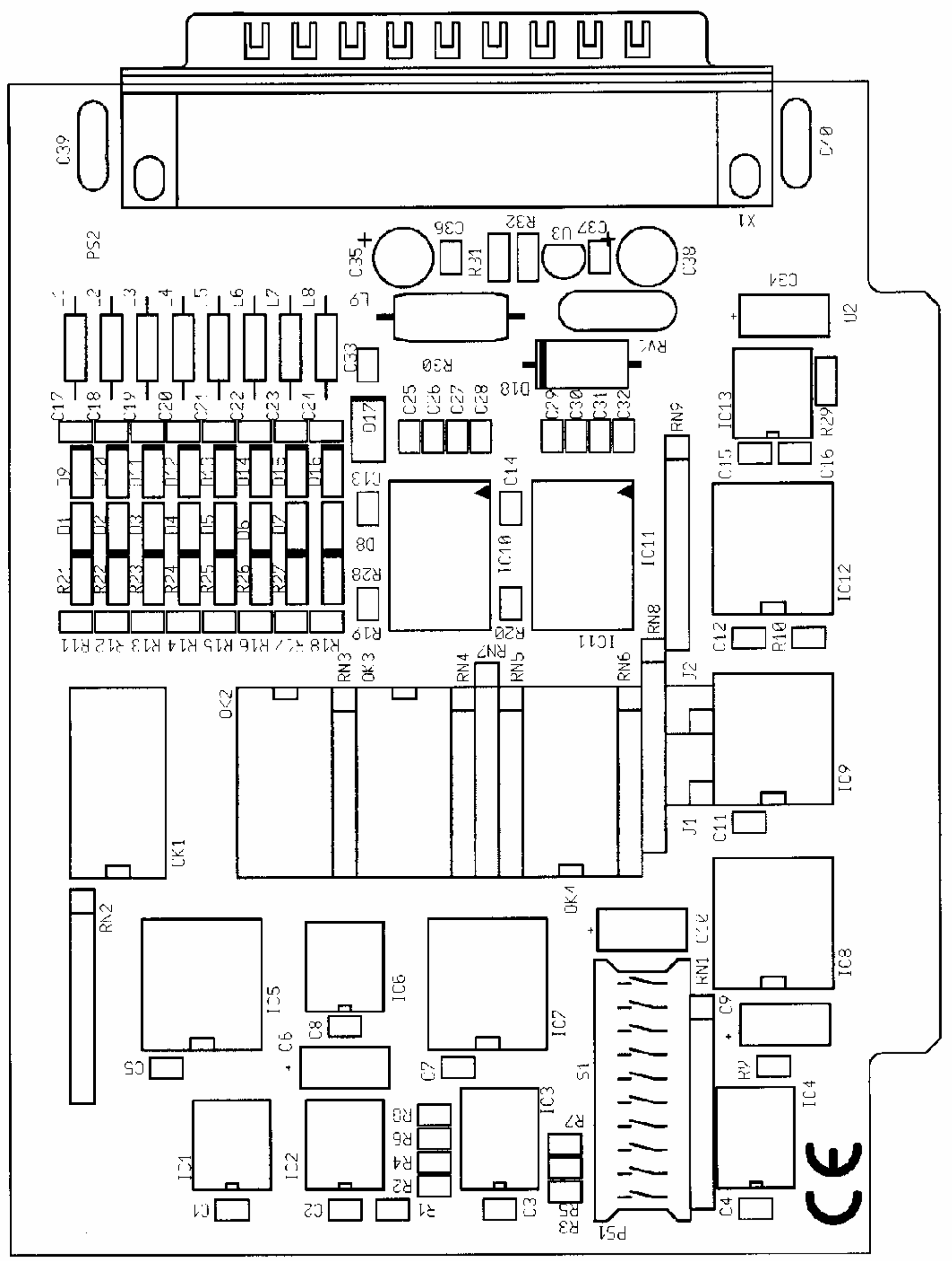
Protection against undervoltage (effective from 0 V ext. < 5V)

Outputs by undervoltage	all output channels switch off. (Shut down logic)
Diagnosis output channels by undervoltage	LOW level

Watchdog:	time: 4.6 s
Outputs after Timeout:	all output channels switch off.

4.4 Component scheme

Fig. 4-2: Component scheme



5 INSTALLATION

5.1 Setting the base address through DIP switches

WARNING!
If the base address set is wrong, the board and/or the PC may be damaged.

5.1.1 Examining the resources

Under Windows 95 and Windows NT, you will find the resources which are already used as follows:

Select under "Start":

- SETTINGS
- CONTROL PANEL
- ICON: SYSTEM
- DEVICE MANAGER
- REFRESH
- PROPERTIES
- RESOURCE DISPLAY: • INPUTS / OUTPUTS

For MS-DOS the MSD.EXE program (Microsoft diagnostic) returns the resource assignment. For the other operating systems you will find the system address range in the manual of your PC manufacturer.[4]

You have then to find out if additonal boards have been inserted in your PC.[5]

The predefined address range begins at 390Hex and ends at 391Hex. Select another address range by modifying the settings of the base address if the resources are already occupied by other PC components.

[4] For AT/368/468 please refer to the appendix: system address range

[5] When doubting check the ISA slots of your PC. Switch the PC off before opening it and looking inside.

5.1.2 Modifying the base address

Before installing the board

At delivery the base address is set to 0390H.

Check, that
- the base address is free
- the address range required by the board is not already used by the PC or by boards already installed in the PC.

If the base address or the address range **are wrong**
- **Select** another base address with the 10-pin block of DIP switches S1 and the jumper field J2 **.**

Decoding the base address
The base address is decoded in steps of each time 8 I/O addresses.

The base address can be selected between 0100H and 0FFFFH within the PC I/O address space.

In table 5-1 the address 0390H is decoded. (Settings at delivery).

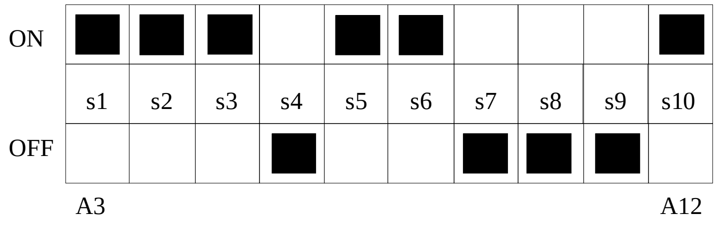
Table 5-1: Decoding table

Decoded address bus	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
Hex base address to be set	0				3				9				0		
Binary base address to be set	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0
DIP switches S1 Logic "0"= ON Logic "1" = OFF	*	*	*	*	*	s10 ON	s9 OFF	s8 OFF	s7 OFF	s6 ON	s5 ON	s4 OFF	s3 ON	s2 ON	s1 ON

* : decoded per Jumper on logic "0”

Fig. 5-1: DIP switches

IMPORTANT
You will find the switch **s1** on the **left** side of the DIP switches!



5.2 Inserting the board

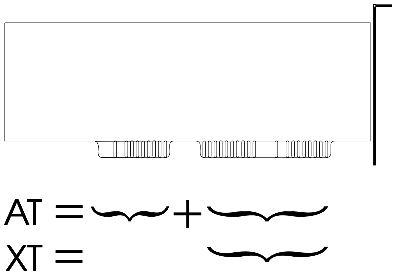
IMPORTANT!
Do observe the *safety instructions.*

5.2.1 Opening the PC

- Switch off your PC and all the units connected to the PC.
- Pull the PC mains plug from the socket.
- Open your PC as described in the manual of the PC manufacturer.

1. **Select a free ISA slot**

Fig. 5-2: types of slots



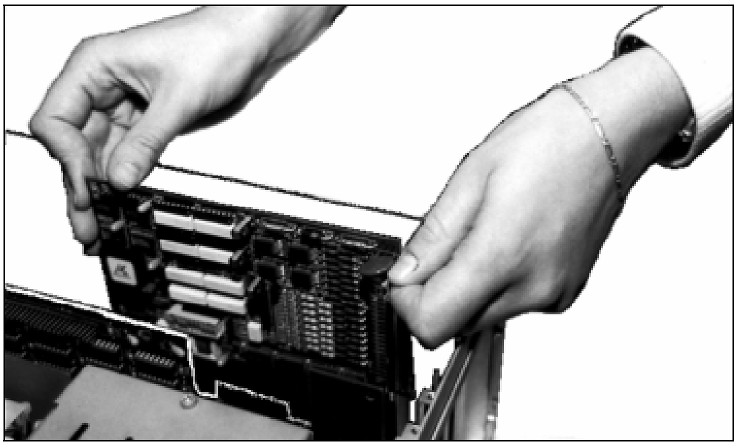
The board can be inserted either in a slot XT or AT. It can also be inserted in EISA slots with respect of certain conditions.

2. **Remove the back cover of the selected slot** according to the instructions of the PC manufacturer.
   Keep the back cover. You will need it if you remove the board.
3. **Discharge yourself from electrostatic charges**
4. **Take the board from its protective pack.**

5.2.2 Plugging the board into the slot

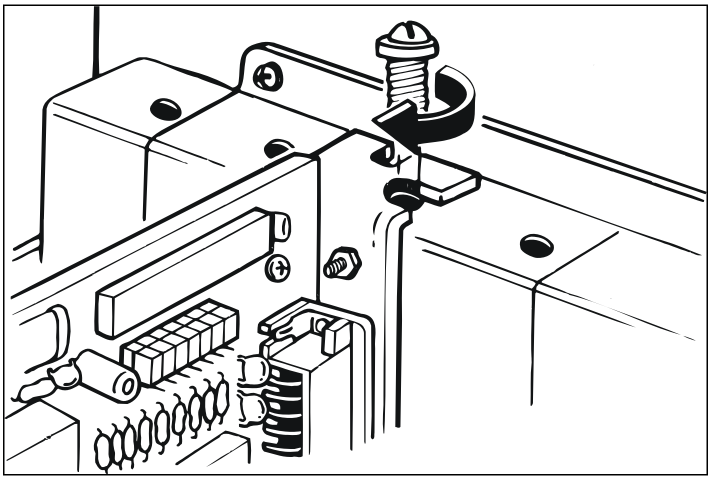
- **Discharge yourself from electrostatic charges**
- Insert the board **vertically into the chosen slot.**

Fig. 5-4: Inserting the board



- **Fasten the board** to the rear of the PC housing with the screw which was fixed on the back cover.

Fig. 5-5: Securing the board at the back cover



- **Tighten all loosen screws.**

5.2.3 Closing the PC

- Close your PC as described in the manual of the PC manufacturer.

5.3 Installing the software

The board is delivered with a CD-ROM which contains ADDIREG for Windows NT 4.0 and Windows 95.

You can download the latest version of the ADDIREG program from the Internet:
http://www.addi-data.de
http://www.addi-data.com

The CD also contains standard software for the ADDI-DATA boards:

- 16-bit for MS-DOS and Windows 3.11
- 32-bit for Windows NT/95.

5.3.1 Software installation under MS-DOS and Windows 3.11

- Copy the contents of PA1500\16bit on a disk.
  If several disks are to be used, the directory contents is stored in several sub-directories (Disk1, Disk2, Disk3...).
- Insert the (first) disk into a drive and change to this drive.
- Enter <INSTALL>.

The installation program gives you further instructions.

5.3.2 Software installation under Windows NT / 95

- Select the directory PA1500\32bit\Disk1 corresponding to the board.
- Start the setup program "setup.exe" (double click)
- Select one of the 3 parameters
  1- typical
  2- compact
  3- custom

Proceed as indicated on the screen and read the "Software License" and "Readme".
In "custom", you can select your operating system.

The installation program gives you further instructions.

5.4 Board configuration with ADDIREG

The ADDIREG registration program is a 32-bit program for Windows NT 4.0/ 95. The user can register all hardware information necessary to operate the ADDI-DATA PC boards.

IMPORTANT!
If you use one or several resources of the board, you cannot start the ADDIREG program.

5.4.1 Program description

IMPORTANT!
Insert the ADDI-DATA boards to be registered before starting the ADDIREG program.

If the board is not inserted, the user cannot test the registration.

Once the program is called up, the following dialog box appears.

Fig. 5-6: ADDIREG registration program

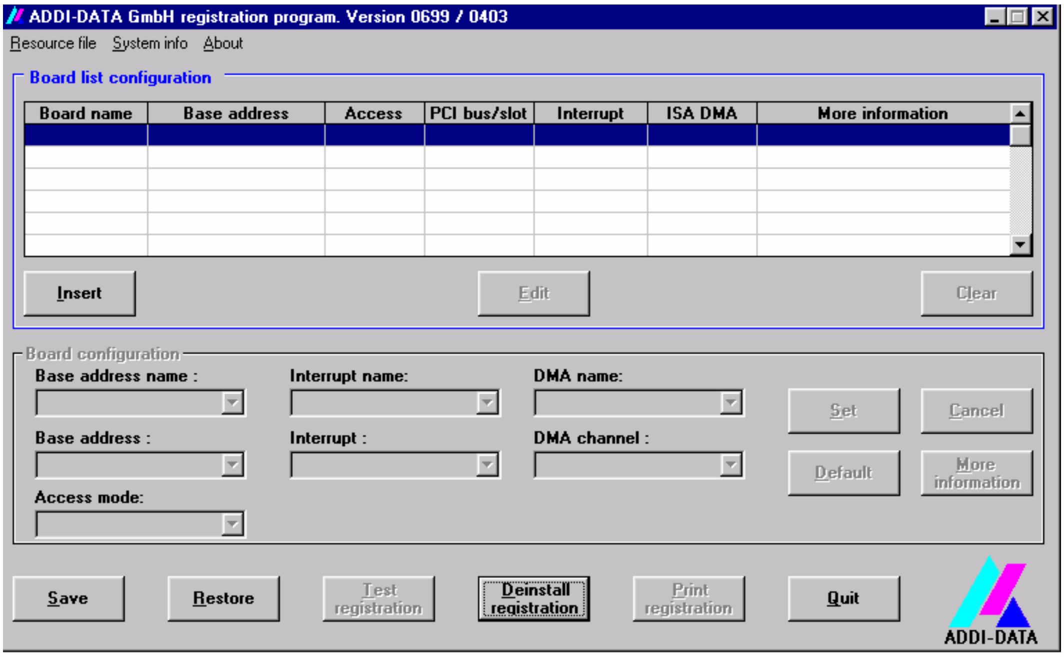


Table:
The table in the middle lists the registered boards and their respective parameters.

Board name:
Names of the different registered boards (eg.: APCI-3120).
When you start the program for the first time, no board is registered in this table.

Base address:
Selected base address of the board.

i

IMPORTANT!
The base address selected with the ADDIREG program must correspond to the one set through DIP-switches.

Access:
Selection of the access mode for the ADDI-DATA digital boards.
Access in 8-bit or 16-bit.

PCI bus / slot:
Used PCI slot. If the board is no PCI board, the message "NO" is displayed.

Interrupt:
Used interrupt of the board. If the board uses no interrupt, the message "Not available" is displayed.

i

IMPORTANT!
The interrupt selected with the ADDIREG program must correspond to the one set through DIP-switches.

ISA DMA:
Indicates the selected DMA channel or "Not available" if the board uses no DMA.

More information:
Additional information like the identifier string (eg.: PCI1500-50) or the installed COM interfaces.

Text boxes:

Under the table you will find 6 text boxes in which you can change the parameters of the board.

Base address name:
When the board operates with several base addresses (One for port 1, one for port 2, etc.) you can select which base address is to be changed.

Base address:
In this box you can select the base addresses of your PC board. The free base addresses are listed. The used base addresses do not appear in this box.

Interrupt name:
When the board must support different interrupt lines (common or single interrupts), you can select them in this box.

Interrupt:
Selection of the interrupt number which the board uses.

DMA name:
When the board supports 2 DMA channels, you can select which DMA channel is to be changed.

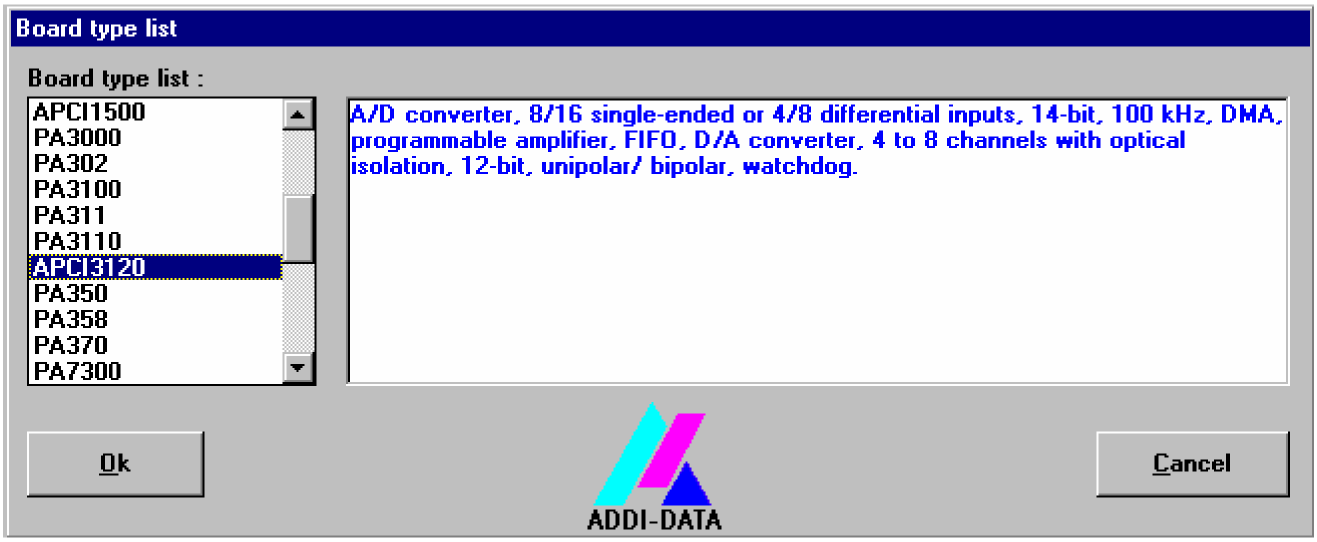
DMA channel:
Selection of the used DMA channel.

Buttons:

Edit [6]:
Selection of the highlighted board with the different parameters set in the text boxes. Click on "Edit" to activate the data or click twice on the selected board.

Insert:
When you want to insert a new board, click on "Insert". The following dialog window appears:

Fig. 5-7: Configuring a new board



A/D converter, 8/16 single-ended or 4/8 differential inputs, 14-bit, 100 kHz, DMA, programmable amplifier, FIFO, D/A converter, 4 to 8 channels with optical isolation, 12-bit, unipolar/ bipolar, watchdog.

All boards you can register are listed on the left. Select the wished board. (The corresponding line is highlighted).
On the right you can read technical information about the board(s).
Activate with "OK"; You come back to the former screen.

Clear:
You can delete the registration of a board. Select the board to be deleted and click on "Clear".

Set:
Sets the parametered board configuration. The configuration should be set before you save it.

Cancel:
Reactivates the former parameters of the saved configuration.

Default:
Sets the standard parameters of the board.

More information:
You can change the board specific parameters like the identfier string, the COM number, the operating mode of a communication board, etc...

[6] "x": Keyboard shortcuts; e.g. "Alt + e" for Edit

If your board does not support these information, you cannot activate this button.

Save:
Saves the parameters and registers the board.

Restore:
Reactivates the last saved parameters and registration.

Test registration:
Controls if there is a conflict between the board and other devices.
A message indicates the parameter which has generated the conflict. If there is no conflict, "OK" is displayed.

Deinstall registration:
Deinstalls the registrations of all board listed in the table.

Print registration:
Prints the registration parameter on your standard printer.

Quit:
Quits the ADDIREG program.

5.4.2 Registering a new board

i

IMPORTANT!
To register a new board, you must have administrator rights.
Only an administrator is allowed to register a new board or change a registration.

- Call up the ADDIREG program. The figure X-X is displayed on the screen.
  Click on "Insert". Select the wished board.

- Click on "OK". The default address, interrupt, and the other parameters are automatically set in the lower fields. The parameters are listed in the lower fields.
  If the parameters are not automatically set by the BIOS, you can change them.
  Click on the wished scroll function(s) and choose a new value.
  Activate your selection with a click.

- Once the wished configuration is set, click on "Set".

- Save the configuration with "Save".

- You can test if the registration is "OK".
  This test controls if the registration is right and if the board is present.
  If the test has been successfully completed you can quit the ADDIREG program.
  The board is initialised with the set parameters and can now be operated.

In case the registration data is to be modified, it is necessary to boot your PC again. A message asks you to do so. When it is not necessary you can quit the ADDIREG program and directly begin with your application.

5.4.3 Changing the registration of a board

IMPORTANT!
To change the registration of a board, you must have administrator rights. Only an administrator is allowed to register a new board or change a registration.

- Call up the ADDIREG program. Select the board to be changed.
  The board parameters (Base address, DMA channel, ..) are listed in the lower fields.

- Click on the parameter(s) you want to set and open the scroll function(s).

- Select a new value. Activate it with a click.
  Repeat the operation for each parameter to be modified.

- Once the wished configuration is set, click on "Set".

- Save the configuration with "Save".

- You can test if the registration is "OK".
  This test controls if the registration is right and if the board is present.
  If the test has been successfully completed you can quit the ADDIREG program.
  The board is initialised with the set parameters and can now be operated.

In case the registration data is to be modified, it is necessary to boot your PC again. A message asks you to do so. When it is not necessary you can quit the ADDIREG program and directly begin with your application.

5.4.4 Removing the ADDIREG program

The ADDI_UNINSTAL program is delivered on the CD-ROM.

- Install the ADDI_UNINSTAL program on your computer.
- Start the ADDIREG program and click on "Deinstall registration"
- Quit ADDIREG
- Start the ADDI_UNINSTAL program
- Proceed as indicated until the complete removing of ADDIREG.

You can also download the programm from Internet.

5.4.5 Software downloads from the Internet

You can download the latest version of the device driver for the PA1500 board**.**

http://www.addi-data.com

If you have any questions, do not hesitate to send us an e-mail:

info@addi-data.de or
hotline@addi-data.com

i

IMPORTANT!
Before using the board or in case of malfunction during operation, check if there is an update of the product (technical description, driver). The current version can be found on the internet or contact us directly.

6 CONNECTION TO THE PERIPHERAL

6.1 Connector pin assignment

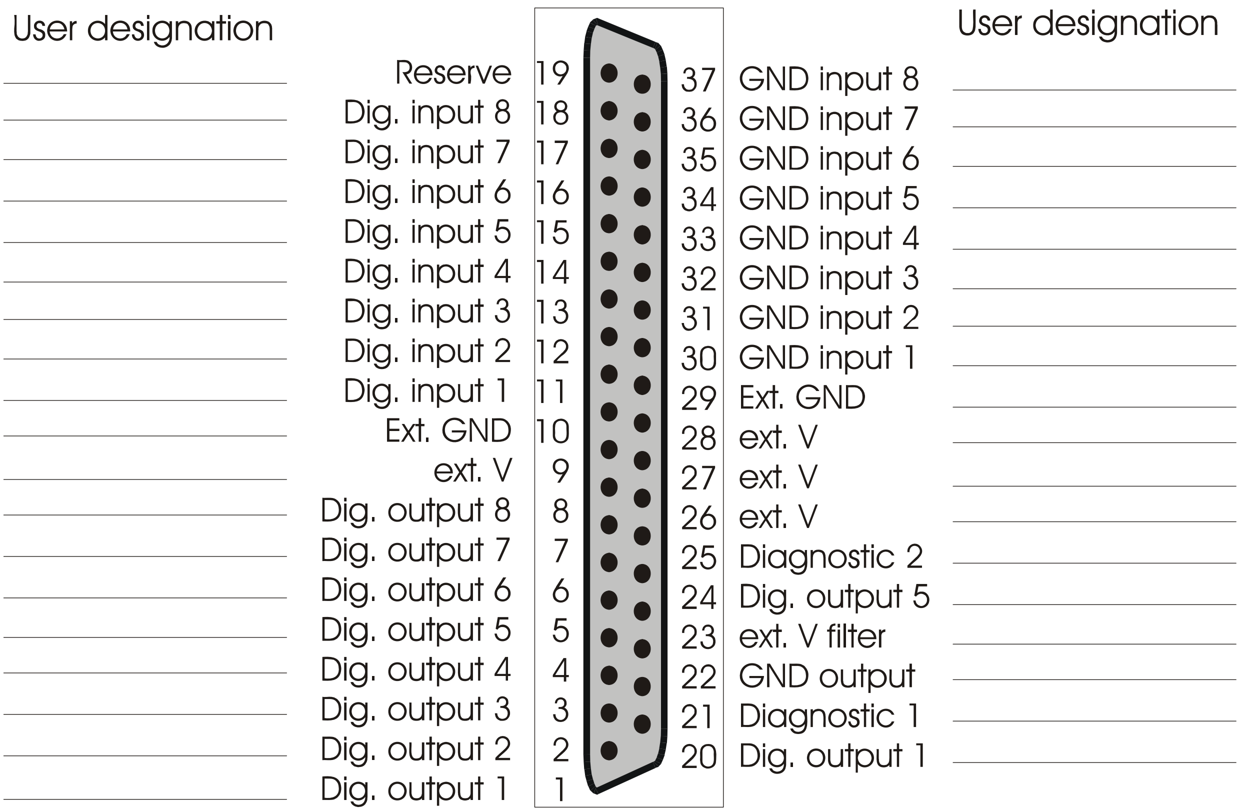
WARNING!
Interferences are emitted and coupled through the connection cable. A wrong cable would damage the operating and functions safeties of your system.

We recommend to use our standard connection cable.

Make sure by the installation of the connection cable that:
- it is installed within a sufficient distance from sensitive analog signals
- the distance from potential interference sources (eg.: frequency converters, supply circuits) is as long as possible.

If the output channels are operating with maximum load, you may install the connection cable with the appropriate cross section in a well-aerated room.

Fig. 6-1: Pin assignment of the 37-pin SUB-D male connector



User designation					User designation
	Reserve	19	37	GND input 8	
	Dig. input 8	18	36	GND input 7	
	Dig. input 7	17	35	GND input 6	
	Dig. input 6	16	34	GND input 5	
	Dig. input 5	15	33	GND input 4	
	Dig. input 4	14	32	GND input 3	
	Dig. input 3	13	31	GND input 2	
	Dig. input 2	12	30	GND input 1	
	Dig. input 1	11	29	Ext. GND	
	Ext. GND	10	28	ext. V	
	ext. V	9	27	ext. V	
	Dig. output 8	8	26	ext. V	
	Dig. output 7	7	25	Diagnostic 2	
	Dig. output 6	6	24	Dig. output 5	
	Dig. output 5	5	23	ext. V filter	
	Dig. output 4	4	22	GND output	
	Dig. output 3	3	21	Diagnostic 1	
	Dig. output 2	2	20	Dig. output 1	
	Dig. output 1	1			

6.2 Connection principle

you will find examples of connection in the chapter 6.3

Fig. 6-2: connection principle of the output channels

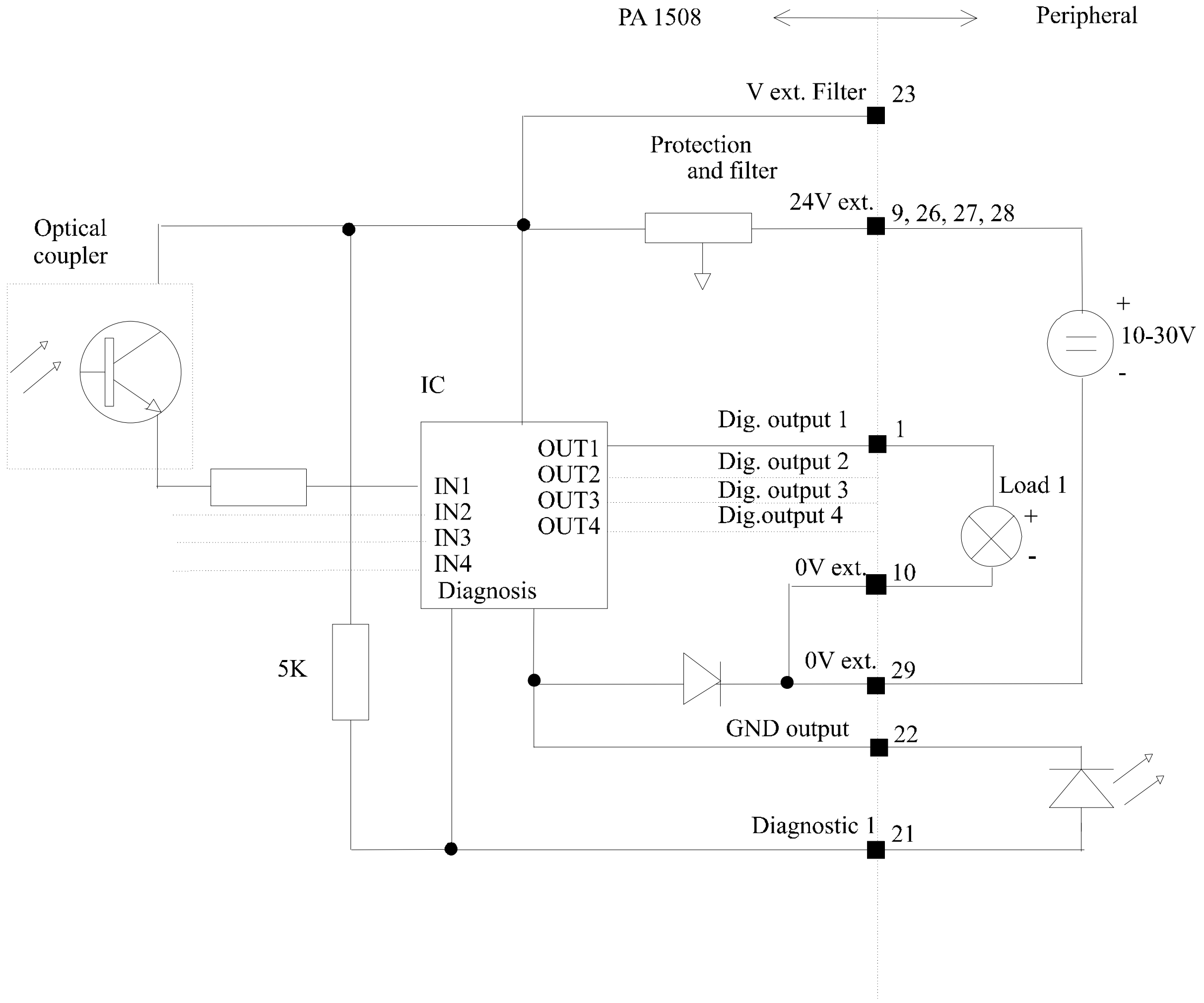


Fig. 6-2: connection principle of the output channels (continued)

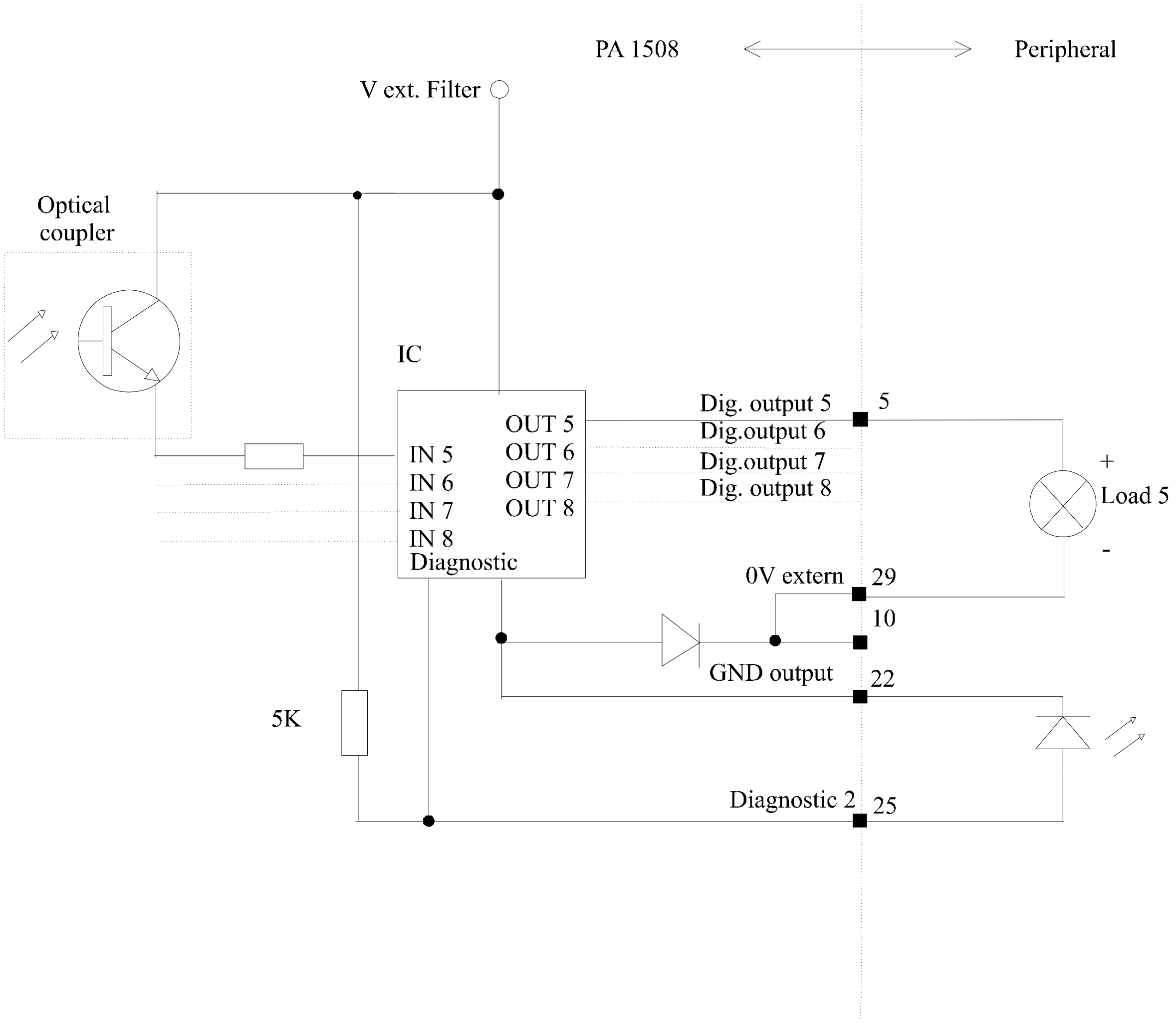
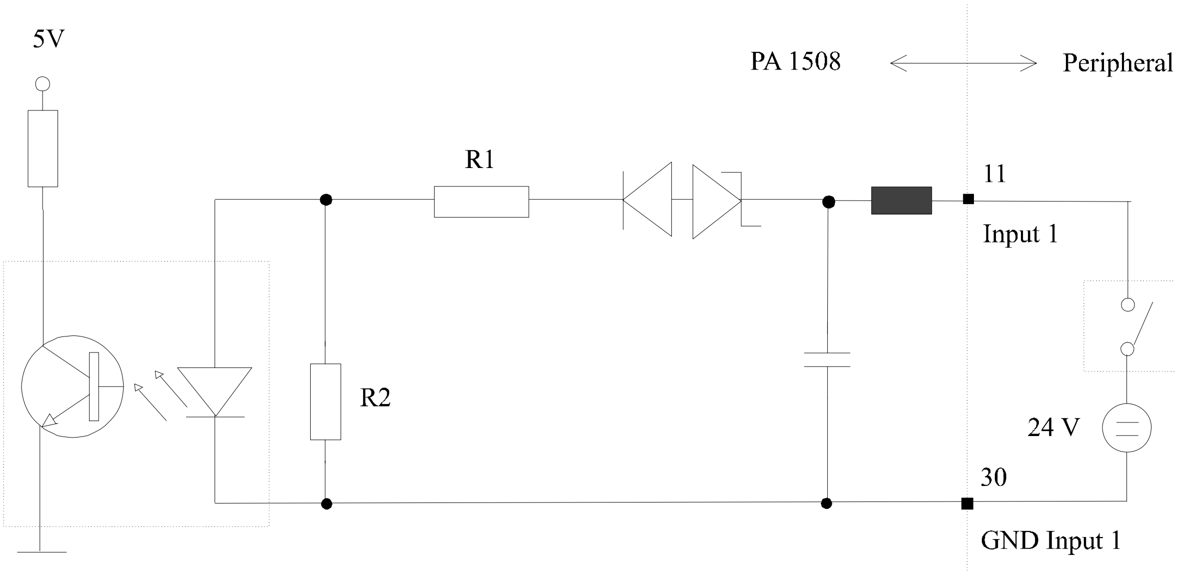


Fig. 6-3: Connection principle of the input channels



6.3 Connection examples

6.3.1 Connection to the screw terminal panel PX 9100-D

Fig. 6-4: low ohmic power connection

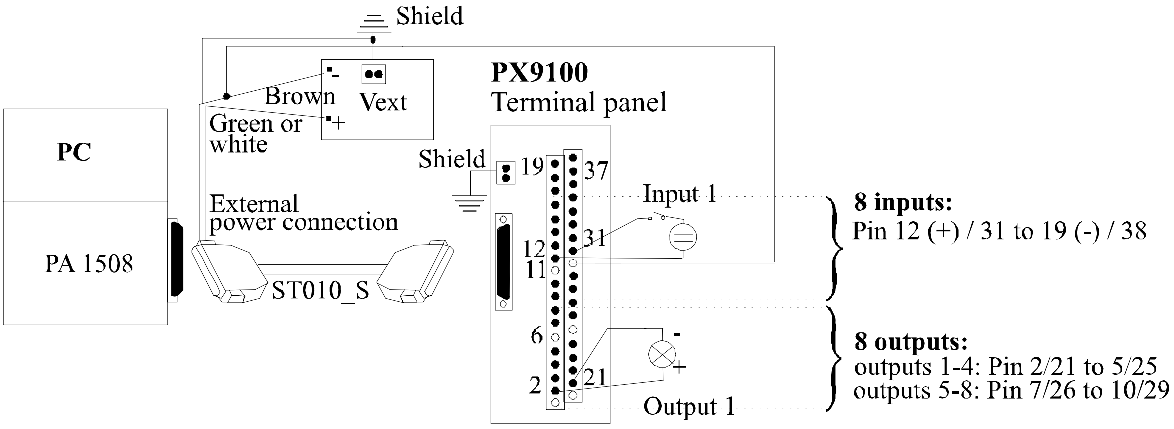
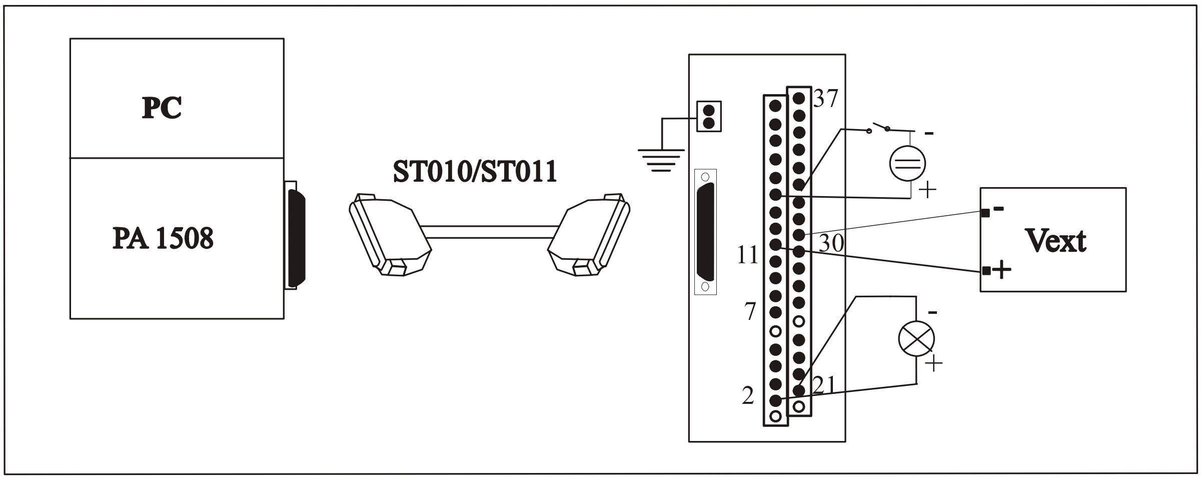


Fig. 6-5: Economical solution for low currents



7 FUNCTIONS OF THE BOARD

7.1 Output channels

The output channels can change their status according to the programmed data bus information. The necessary data bus information is transmitted to the corresponding driver function as a counter value (logic "1" or "0").

Logic "1"
means that the external supply voltage Vextern is connected to the output channels on the 37-pin front connector.

Logic "0"
means that the external voltage is no longer supplied to the output channels through the front connector.

Through the software functions the switching status of the outputs 1-8 can parallelly be modified, as long as the watchdog is not in alarm status.

Function safety:
for an increased protection of the functions the user can enable the watchdog or the diagnostic.

Safety through redundancy:
The control current circuitries of the outputs 1-4 and 5-8 are redundant: they are controlled by the different operating drivers.
You detect if one of both drivers has run down or switched off:
- when the outputs 1-4 and 5-8 are so connected that a control function is only enabled when both signals correspond.

Reset: After reset the outputs are separated from the supply voltage.

Driver: The driver supports the following output functions:
- set one output and reset the others
- set one output without affecting the others
- reset one output
- set or reset parallelly and independently all outputs.

7.2 Diagnostic

Both diagnostic signals are independent from one another and control each 4 outputs.

Diagnostic signal 1:	Analysis of the outputs 1 to 4
Diagnostic signal 2:	Analysis of the outputs 5 to 8

The diagnostic signal on the diagnostic pin of the front connector is an active LOW signal. In normal operation the signal is set to high (0V ext.)

By overtemperature of a driver the corresponding channel switches off. The signal on the diagnostic output is set to low within 100 ms.
Protection against overtemperature resets automatically when the chip temperature falls under 125-140°C.

7.3 Watchdog

The watchdog function allows controlling the board and its misfunctions during the application.

The update of the output triggers the watchdog.

If the watchdog has been enabled by software and if it is not triggered within 4.6 s, the outputs are switched off and cannot be operated.

The watchdog can have 3 different status which are readable per software:

"OFF" The watchdog is switched off. the outputs are not affected by it.

"ON" The watchdog is switched on by driver command. It controls the program course and affects the output status per timeout (> 4.6 s)

"Alarm" The watchdog indicates an alarm by setting all outputs in the rest state. The driver commands cannot affect the outputs. Through continuous polling of the parameters for the watchdog status (pb_WatchdogStatus) you can see with the function ReadOutputswatchdogStatus if the watchdog is in alarm status.

The following table should help the user to see in which order the different status are completed:

DRIVER COMMAND	WATCHDOG STATUS
By "High boot"	"OFF": The watchdog is off.
i_PA1508_SetOutputMemoryOn + i_PA1508_EnableOutputsWatchdog	"ON": The watchdog is switched on by driver command. It controls the program course and affects the outputs by timeout (> 4,6 s).
i_PA1508_DisableOutputsWatchdog	"OFF": The watchdog is off.
i_PA1508_RearmOutputsWatchdog	"ON": The watchdog is switched on by driver command. It controls the program course and affects the outputs by timeout (> 4,6 s).

Application:

Write a program which within periods < 4.6 s modifies or updates the outputs. (For example with the functions Set1DigitalOutputOn, Set8DigitalOutputOn, Set1DigitalOutputOff, Set8DigitalOutputOff).

Example:
In the main program a program loop is executed and calls up the function Set1DigitalOutputOn in a cyclic mode.
The value to be transmitted is only changed when the switching status of the output is to be modified.
The simple cycle of the program loop can excess 4.6 s only in case of misfunction.
The watchdog is hence always triggered during normal operation, once the driver command "Set1DigitalOutputOn" has been executed.
If the driver command Set1DigitalOutputOn is no longer called up because of an error, the watchdog trigger runs down. After 4.6 s the board is set to the alarm status.

The watchdog is now in the alarm status.
To activate it again the user has two possibilities. He can
- either call up the function i_PA1508_RearmsOutputsWatchdog (...)
- or call up the 2 functions i_PA1508_DisableOutputsWatchdog and i_PA1508_EnableOutputsWatchdog. The watchdog is on.

Reset:
After switching on the PC, the watchdog is off. The watchdog time is 4.6 s.

Driver:
The driver supports the following watchdog functions:
- set the watchdog ON
- set the watchdog OFF

7.4 Input channels

The inputs collect external signal status since any input information is loaded by drivers functions as a value in a memory unit of the PC.

The user has to determine if signals higher than 17 V can be collected by the inputs or not.

Logic "1"
means that the input voltage range on the 37-pin connector is > 17 V.

Logic "0"
means that the input voltage range on the 37-pin connector is < 15 V.

Driver:
The driver supports the following input functions:
- read one input
- read all inputs in parallel through a binary code
- set one input
- set all inputs through binary code.

8 CORRECTION OF FAILURES

In this chapter you will find the common failures you can encounter during the operation of the board **PA 1508**. The following tables should help you to find a quick solution. Yet this table is not complete. In case of problem our technical support is at your disposal for any question about the board.

8.1 The PC can no longer be initialised.

Possible cause:	**Solution:**
The board is not correctly installed. It is not vertically inserted in the slot and causes short-circuits.	Switch the PC off. Release the board and insert it again. Control the position of the plug contacts.
Conflict between the base addressses. The address range of the board is already used by (an)other board(s).	Switch the PC off. Release the board, change the base address and insert it again.
With the insertion of the board, a cable in the PC housing has been released or a screw is not fixed and creates a short-circuit	Switch the PC off. Release the screw and reconnect the cable with the housing.
Network cable has not been plugged in again.	Plug the network cable.

8.2 The outputs do not react any more.

Possible cause:	**Solution:**
Watchdog has run down	Switch the watchdog off per software.
External supply voltage is not connected	Connect the external supply voltage.
External supply voltage is off or has fallen under 8 V. The outputs switch off between 5V-8V	Check the voltage distribution.
The supply voltage poles have reversed.	Set the outputs to "0". Correct the voltage reversal.
The output drivers are off because of overtemperature.	Let the board cool down. Improve the aeration or reduce the load.
The protection against overloading is defective and the internal supply voltage is separated from the external supply voltage. Between pin 22 and 23 there is no voltage available.	Send back the board for repair.

8.3 The inputs deliver wrong values.

Possible cause:	Solution:
The connected signal transmitter does not syppply enough current. 6 mA are necessary by 24 V.	Connect a stronger signal transmitter.
Connection cable too long: ohmic losses reduce the signal voltage so that it lies under 16.5 V.	Reduce the length of the connection cable or increase it cross section.
GND connection misses. Each input includes its own GND-supply.	Check all GND connections.
Voltage reversal on one of the inputs	Compare the pin assignment with the connections and correct the voltage reversal.
The inputs read the information too late or too early.	Check the synchronisation with the signal transmitter. If necessary program delay cycles or read the program several times.
The programmed base address does not correspond to the set base address	Check the base address settings and test the inputs

8.4 By operating the board, failures sometimes occur.

Possible cause:	Solution:
The connection cable is not correctly screwed or has loose connections	Screw correctly the cable or change it.
Mass loops create interferences.	Conduct separatly the GND lines of the inputs.Wire the GND in a star shape.
The environment creates strong interferences.	Lead the input and output connections twisted in pairs with the corresponding GND potential and earth the cable shield on both ends. Change the cabling so that the distance from potential interference sources is as long as possible or even remove these interferences sources.

9 STANDARD SOFTWARE

9.1 Compatibility of the driver

When you operate several boards with the same software, some problems of compatibility can occur, especially when the software of these boards operate in different versions.
Compare the versions of the standard software and control in the update file (Update.txt) which modifications have been completed and if your application is concerned by these modifications.

9.2 Conventions

IMPORTANT!
Note the following conventions in the text:

Function: "i_PA1508_SetBoardInformation"
Variable *ui_Address*

Table 9-1: Type Declaration for Dos and 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

Table 9-2: Type Declaration for Windows 95/NT

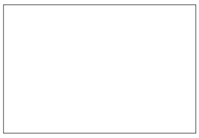
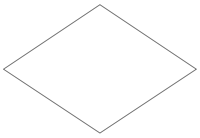
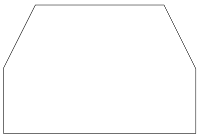
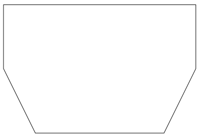
	Borland C	Microsoft C	Borland Pascal	Microsoft Visual Basic Dos	Microsoft Visual Basic Windows
VOID	void	void	pointer		any
BYTE	unsigned char	unsigned char	byte	integer	integer
INT	int	int	integer	integer	integer
UINT	unsigned int	unsigned int	long	long	long
LONG	long	long	longint	long	long
PBYTE	unsigned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

9.3 DIN 66001- Graphical symbols

This chapter describes all the software functions (API) necessary for the operation of the **PA 1508** board.

To illustrate these functions, we designed flow charts with the graphical symbols listed below.

It gives the user a quick overview of the hierarchy between the different functions.

	**Process,** general (including inputs and outputs)
	**Decision** **Selection unit** (eg.: switch)
	**Loop limit** Beginning
	**Loop limit** End

9.4 Software functions (API)

9.4.1 Base address

1) i_PA1508_SetBoardInformation (...)

Syntax:

```
<Return value> =  i_PA1508_SetBoardInformation
                                      (UINT      ui_Address,
                                       PBYTE     pb_BoardHandle)
```

Parameters:

UINT	ui_Address	Base address of the board **PA 1508**
PBYTE	pb_BoardHandle	Handle[1] of board **PA 1508** to use the functions

Task:
Verifies if board **PA 1508** is present and stores the base address.
A handle is returned to the user which allows to use the next functions.
Handles allow to operate several boards.

IMPORTANT!
When you operate under Windows 95 and Windows NT, the necessary resources are automatically stored from the first callup of the function.

The user can ask for the resources in the system settings.

Return value:
0: No error
-1: Base address already used
-2: No handle is available for the board (up to 10 handles can be used)

2) i_PA1508_CloseBoardHandle (..)

IMPORTANT!
Call up this function each time you want to quit the user program!

Syntax:

```
<Return value> =  i_PA1508_CloseBoardHandle     (BYTE    b_BoardHandle)
```

Parameter:

BYTE	b_BoardHandle	Handle of board **PA 1508**

Task:
Releases the board handle. Blocks the access to the board.

Return value:
0: No error
-1: The handle parameter of the board is wrong

[1] Identification number of the board

9.4.2 Digital inputs

1) i_PA1508_Read1DigitalInput (...)

Syntax :

```
<Return value> =  i_PA1508_Read1DigitalInput  (BYTE      b_BoardHandle,
                                               BYTE      b_Channel,
                                               PBYTE     pb_ChannelValue)
```

Parameters:

BYTE	b_BoardHandle	Handle of board **PA 1508**
BYTE	b_Channel	The number of the input to be read (1 to 8)
PBYTE	pb_ChannelValue	State of the digital input: 0 -> No AC voltage on the input 1 -> AC voltage on the input

Task:

Indicates the state of an input. The variable *b_Channel* passes the input to be read (1 to 8). A value is returned with the variable *pb_ChannelValue* :
"0": No AC voltage on the input
"1": AC voltage on the input

Return value:

0: No error
-1: The handle parameter of the board is wrong
-2: The input number is wrong

2) i_PA1508_Read8DigitalInput (...)

Syntax:

```
<Return value> =  i_PA1508_Read8DigitalInput  (BYTE   b_BoardHandle,
                                               PBYTE pb_PortValue)
```

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**
PBYTE	pb_PortValue	State of the digital input port (0 to 255)

Task:

Indicates the state of the 8 digital inputs.
A value is returned with the variable *pb_PortValue* .
"0": No AC voltage on the input
"1": AC voltage on the input

Example:

pb_PortValue = 245 (= F5 Hex)

Input number	1	2	3	4	5	6	7	8

Hex value	5				F			

Binary value / State	1	0	1	0	1	1	1	1

There is AC voltage on inputs 1, 3, 5, 6, 7, 8
Theres is no AC voltag on the inputs 2 and 4.

Return value:

0: No error
-1: Handle parameter of the board is wrong

9.4.3 Digital outputs

1) i_PA1508_SetOutputMemoryOn (...)

Syntax:
<Return value> = i_PA1508_SetOutputMemoryOn (BYTE b_BoardHandle)

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**

Task:
Activates the digital output memory. After calling up this function, the outputs you have previously activated with the function i_PA1508_SetXDigitalOutputOn are not reset. You can reset them with the function i_PA1508_SetXDigitalOutputOff.

Return value:
0: No error
-1: Handle parameter of the board is wrong
-2: There is no output on this board

2) i_PA1508_SetOutputMemoryOff (...)

Syntax:
<Return value> = i_PA1508_SetOutputMemoryOff (BYTE b_BoardHandle)

Parameters:

BYTE	b_BoardHandle	Handle of the PA 1508

Task:
Deactivates the digital output memory.

Return value:
0: No error
-1: Handle parameter of the board is wrong
-2: There is no output on this board

3) i_PA1508_Set1DigitalOutputOn (...)

Syntax:
<Return value> = i_PA1508_Set1DigitalOutputOn
(BYTE b_BoardHandle,
BYTE b_Channel)

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**
BYTE	b_Channel	Number of the output you want to set (1 to 8)

Task:
Sets the output which has been passed with *b_Channel* to V ext. All other outputs are separated from the V ext.

Exception:
If you have switched on with the function "SetOutputMemoryOn" the same function will be executed, but the switchng status of the outputs does not change.

Return value:
0: No error
-1: Handle parameter of the board is wrong
-2: There is no output on this board
-3: Input number is not between 1 and 8

IMPORTANT!
If you want to set several outputs at the same time to 0 V ext., use the function "Set8DigitalOuputOn"

4) i_PA1508_Set1DigitalOutputOff (...)

Syntax :
<Return value> = i_PA1508_Set1DigitalOutputOff
(BYTE b_BoardHandle,
BYTE b_Channel)

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**
BYTE	b_Channel	Number of the output you want to reset (1 to 8)

Task:
Separates the output you have passed with *b_Channel* from 0 V ext.

IMPORTANT!
You can use this function only if the digital output memory is ON. See function i_PA1508_SetOutputMemoryOn (..).

Return value:
0: No error
-1: The handle parameter of the board is wrong
-2: The input number is not between 1 and 8
-3: Digital output memory OFF.
First use the function "i_PA1508_SetDigitalOutputMemoryOn"

IMPORTANT!
If you want to separate several outputs at the same time from 0 V ext., use the function "Set8DigitalOuputOff"

5) i_PA1508_Set8DigitalOutputOn (...)

Syntax:
<Return value> = i_PA1508_Set8DigitalOutputOn
(BYTE b_BoardHandle,
BYTE b_Value)

Parameters:

BYTE	b_BoardHandle	Handle of the board **PA 1508**
BYTE	b_Value	Output value (0 to 255)

Task:
Sets all outputs to the switching position coded in the parameter "b_value"
"1": Sets the outputs to 0 V ext.
"0": separates the outputs from 0 V ext.

Exception:
When the output memory is active, (function i_PA1508_SetOutputMemoryON), the binary code changes.
"1": Sets the outputs to 0 V ext.
"0": The switching position of the outputs does not change

Example:

b_Port = 1
b_Value = 245 (= F5 Hex)

Output number	1	2	3	4	5	6	7	8
Binary value / State	1	0	1	0	1	1	1	1
Hex value	5				F			

The outputs 1, 3, 5, 6, 7, 8 are set to 0 V ext.
The outputs 2 and 4 are separated from 0 V ext.

Return value:
0: No error
-1: The handle parameter of the board is wrong

6) i_PA1508_Set8DigitalOutputOff (...)

Syntax:
<Return value> = i_PA1508_Set8DigitalOutputOff (BYTE b_BoardHandle,
BYTE b_Value)

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**
BYTE	b_Value	Output value (0 to 255)

Task:
Sets all outputs to the switching position coded in the parameter "b_value"
"1": Separates the outputs from 0 V ext.
"0": The switching position of the outputs does not change

Example:

b_Port = 1
b_Value = 245 (= F5 Hex)

Output number	1	2	3	4	5	6	7	8
Binary value / State	1	0	1	0	1	1	1	1
Hex value	5				F			

The outputs 1, 3, 5, 6, 7, 8 are set to 0 V ext.
The switching position of the outputs 2 and 4 does not change.

IMPORTANT!
You can use this function only if the digital output memory is ON. See function i_PA1508_SetOutputMemoryOn (..).

Return value:

0: No error
-1: The handle parameter of the board is wrong
-2: The output memory is switched off.
Please call up the function "i_PA1508_SetOutputMemoryOn"

9.4.4 Watchdog

1) i_PA1508_EnableOutputsWatchdog (...)

Syntax:
<Return value> = i_PA1508_ EnableOutputsWatchdog
(BYTE b_BoardHandle)

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**

Task:
Starts the watchdog. The watchdog sets the outputs to 0 after 4.6 s.

i

IMPORTANT!
The function i_PA1508_SetOutputMemoryON is to be called up before calling up i_PA1508_EnableOutputsWatchdog.

Return value:
0: No error
-1: Handle parameter of the board is wrong
-2: There is no output on the board

2) i_PA1508_DisableOuputsWatchdog (..)

Syntax:
<Return value> = i_PA1508_ DisableOutputsWatchdog
(BYTE b_BoardHandle)

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**

Task:
Disables the watchdog.

Return value:
0: No error
-1: Handle parameter of the board is wrong
-2: Watchdog not initialised. See function "i_PA1508_EnableOutputsWatchdog"

3) i_PA1508_RearmOuputsWatchdog

Syntax:

```
<Return value> =  i_PA1508_ RearmOutputsWatchdog
                                        (BYTE b_BoardHandle)
```

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**

Task:

After the watchdog time has run down, you can call up this function to quit the alarm state resulting from a timeout without switching the watchdog off. This function is used to replace both following functions: "DisableOutputsWatchdog" and "EnableOutputsWatchdog"

Return value:

0: No error
-1: Handle parameter of the board is wrong
-2: There is no output on the board
-3: Watchdog not initialised. See function "i_PA1508_EnableOutputsWatchdog"

4) i_PA1508_ReadOuputsWatchdogStatus

Syntax:

```
<Return value> =  i_PA1508_ ReadOutputsWatchdogStatus
                                        (BYTE b_BoardHandle
                                         PBYTE pb_WatchdogStatus)
```

Parameters:

BYTE	b_BoardHandle	Handle of the **PA 1508**
PBYTE	pb_WatchdogStatus	0: Watchdog has not run down 1: Watchdog has run down

Task:

Reads the status of the watchdog.

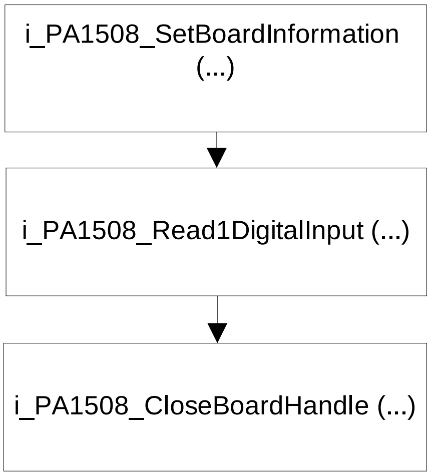
Return value:

0: No error
-1: Handle parameter of the board is wrong
-2: There is no output on the board
-3: Watchdog not initialised. See function "i_PA1508_EnableOutputsWatchdog"

9.5 Examples

9.5.1 Digital input channels

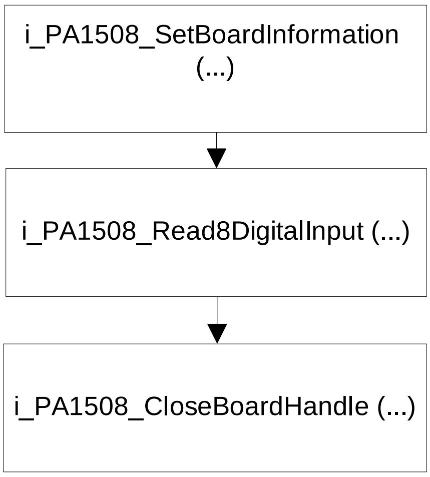
1) Determining the status of an input chanel


i_PA1508_SetBoardInformation (...)
i_PA1508_Read1DigitalInput (...)
i_PA1508_CloseBoardHandle (...)

Example in C

```
#include "pa1508.h"
#include <stdio.h>
#include <conio.h>

void main   (VOID)
{
UINT ui_Address = 0x390; //Base address of the board, fixed to 390Hex
BYTE  b_BoardHandle      = 0;  //Handle of the board
BYTE  b_Channel          = 1;  //Number of the channel to be read
BYTE  b_ChannelValue     = 0;  //Value of the channel

/***********************************************************/
/* The function SetBoardInformation checks the board       */
/* parameters and returns the handle of the board.         */
/***********************************************************/

      i_PA1508_SetBoardInformation (ui_Address,&b_BoardHandle);

/****************************************************************/
/* The function Read1DigitalInput reads the value on the        */
/* channel 1 and sets the value in the parameter b_ChannelValue */
/****************************************************************/

           i_PA1508_Read1DigitalInput  ( b_BoardHandle,
                                         b_Channel,
                                         &b_ChannelValue);
/*******************************************************/
/* The function CloseBoardHandle blocks the access     */
/* to the board and closes the handle of the board     */
/*******************************************************/

             i_PA1508_CloseBoardHandle (b_BoardHandle);}
```

2) Determining the state of 8 input channels

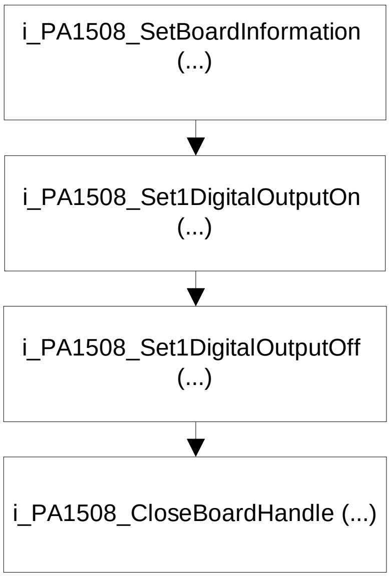

i_PA1508_SetBoardInformation (...)
i_PA1508_Read8DigitalInput (...)
i_PA1508_CloseBoardHandle (...)

Example in C

```
#include "pa1508.h"
#include <stdio.h>
#include <conio.h>

void main   (VOID)
{
UINT ui_Address = 0x390;//Base address of the board, set to 390Hex
BYTE  b_BoardHandle      = 0; //Handle of the board
BYTE  b_Channel          = 1; //Number of the channel to be read
BYTE  b_PortValue   = 0; //Value of the channel

/**********************************************************/
/* The function SetBoardInformation checks the parameters */
/* of the board and returns the handle of the board       */
/**********************************************************/

i_PA1508_SetBoardInformation (ui_Address,&b_BoardHandle);

/*********************************************************/
/* The function Read8DigitalInput reads the value        */
/* of the 8 channels and set the value in b_PortValue    */
/*********************************************************/

           i_PA1508_Read8DigitalInput  ( b_BoardHandle,
                                         b_Channel,
                                         &b_PortValue);

/***************************************************/
/* The function CloseBoardHandle blocks the access */
/* to the board and closes the board handle        */
/***************************************************/

i_PA1508_CloseBoardHandle (b_BoardHandle);
}
```

9.5.2 Digital output channels

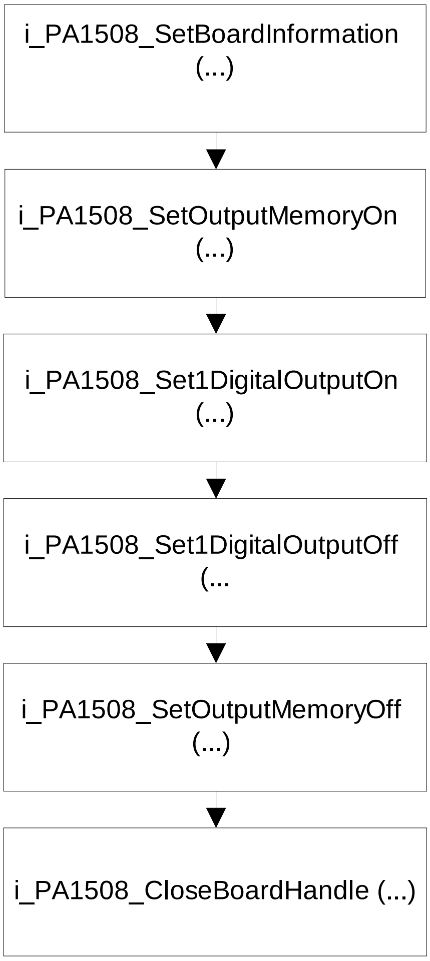
1) Setting a digital output channel and then resettting it (The other output channels are reset)


i_PA1508_SetBoardInformation (...)
i_PA1508_Set1DigitalOutputOn (...)
i_PA1508_Set1DigitalOutputOff (...)
i_PA1508_CloseBoardHandle (...)

Example in C

```
#include "pa1508.h"
void main (VOID)
{
UINT  ui_Address = 0x390; //Base address of the board, set to 390Hex
BYTE  b_BoardHandle = 0;  //Handle of the board
BYTE  b_Channel = 1;      //Number of the channel to be read

/**********************************************************/
/* The function SetBoardInformation checks the parameters */
/* of the board and returns the handle of the board       */
/**********************************************************/

i_PA1508_SetBoardInformation  (ui_Address, &b_BoardHandle);

/***********************************************/
/* The function activates the output channel   */
/* which was given with the variable b_Channel */
/***********************************************/

i_PA1508_Set1DigitalOutputOn  (b_BoardHandle, b_Channel);

/*******************************************************/
/* The function Set1DigitalOutputOn deactivates one    */
/* channel which was given with the variable b_Channel */
/*******************************************************/

i_PA1508_Set1DigitalOutputOff (b_BoardHandle, b_Channel);

/***************************************************/
/* The function CloseBoardHandle blocks the access */
/* to the board and closes the board handle        */
/***************************************************/

i_PA1508_CloseBoardHandle  (b_BoardHandle);}
```

2) Setting a digital output channel and then resetting it (the other outputs are not reset)

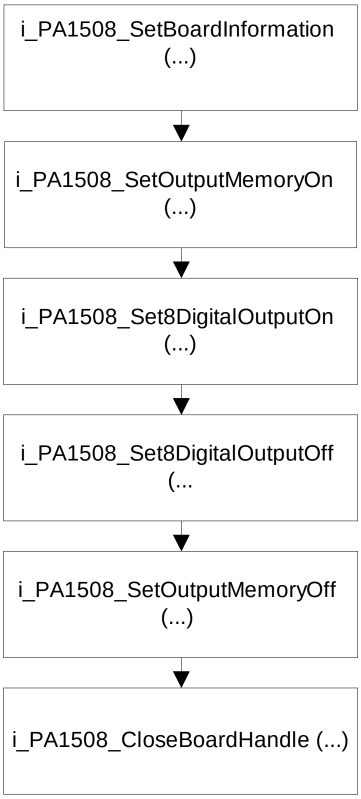

i_PA1508_SetBoardInformation
(...)
i_PA1508_SetOutputMemoryOn
(...)
i_PA1508_Set1DigitalOutputOn
(...)
i_PA1508_Set1DigitalOutputOff
(...
i_PA1508_SetOutputMemoryOff
(...)
i_PA1508_CloseBoardHandle (...)

Example in C

```
#include "pa1508.h"

void main (VOID)
{
UINT  ui_Address = 0x390; //Base address of the board, set to 390Hex
BYTE  b_BoardHandle = 0;  //Handle of the board
BYTE  b_Channel = 1;      //Number of the channel to read

/**********************************************************/
/* The function SetBoardInformation checks the parameters */
/* of the board and returns the handle of the board       */
/**********************************************************/

i_PA1508_SetBoardInformation  (ui_Address, &b_BoardHandle);

/***********************************************************/
/* The function SetOutputMemoryOn stores the current state */
/* of the output. This is neccessary for the function      */
/* Set1DigitalOutputOff                                    */
/***********************************************************/

i_PA1508_SetOutputMemoryOn  (b_BoardHandle);
```

```
/**************************************************************/
/* The function Set1DigitalOutputOn activates one channel     */
/* wich was given with the variable b_Channel                 */
/**************************************************************/

i_PA1508_Set1DigitalOutputOn  (b_BoardHandle, b_Channel);

/*******************************************************/
/* The function Set1DigitalOutputOff deactivates one   */
/* channel which was given with the variable b_channel */
/*******************************************************/

i_PA1508_Set1DigitalOutputOff (b_BoardHandle, b_Channel);

/******************************************************/
/* The function CloseBoardHandle blocks the           */
/* access to the board and closes the board handle    */
/******************************************************/

i_PA1508_CloseBoardHandle  (b_BoardHandle);
```

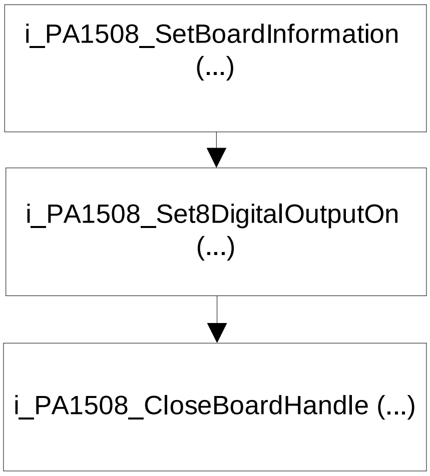
3) Setting several digital output channel and then resettting them (The others are not modified)


i_PA1508_SetBoardInformation (...)
i_PA1508_SetOutputMemoryOn (...)
i_PA1508_Set8DigitalOutputOn (...)
i_PA1508_Set8DigitalOutputOff (...
i_PA1508_SetOutputMemoryOff (...)
i_PA1508_CloseBoardHandle (...)

Example in C

```
#include "pa1508.h"

void main (VOID)
{
UINT  ui_Address = 0x390; //Base address of the board, set to 390Hex
BYTE  b_BoardHandle = 0;  //Handle of the board
BYTE  b_Channel = 0xF5;   //Number of channels to be set

/**********************************************************/
/* The function SetBoardInformation checks the parameters */
/* of the board and returns the handle of the board       */
/**********************************************************/

i_PA1508_SetBoardInformation  (ui_Address, &b_BoardHandle);

/********************************************************************/
/* The function SetOutputMemoryOn stores the current state of the   */
/* output. This is neccessary for the function Set1DigitalOutputOff */
/********************************************************************/

i_PA1508_SetOutputMemoryOn  (b_BoardHandle);

/*********************************************************/
/* The function Set8DigitalOutputOn activates            */
/* the following outputs: 0,2,4,5,6,7, (b_Channel = 0xF5)*/
/*********************************************************/

i_PA1508_Set8DigitalOutputOn  (b_BoardHandle, b_Channel);

/**********************************************************/
/* The function Set8DigitalOutputOff deactivates          */
/* the following outputs: 0,2,4,5,6,7, (b_Channel = 0xF5) */
/**********************************************************/

i_PA1508_Set8DigitalOutputOff (b_BoardHandle, b_Channel);

/******************************************************/
/* The function CloseBoardHandle blocks the           */
/* access to the board and closes the board handle    */
/******************************************************/

i_PA1508_CloseBoardHandle  (b_BoardHandle);
}
```

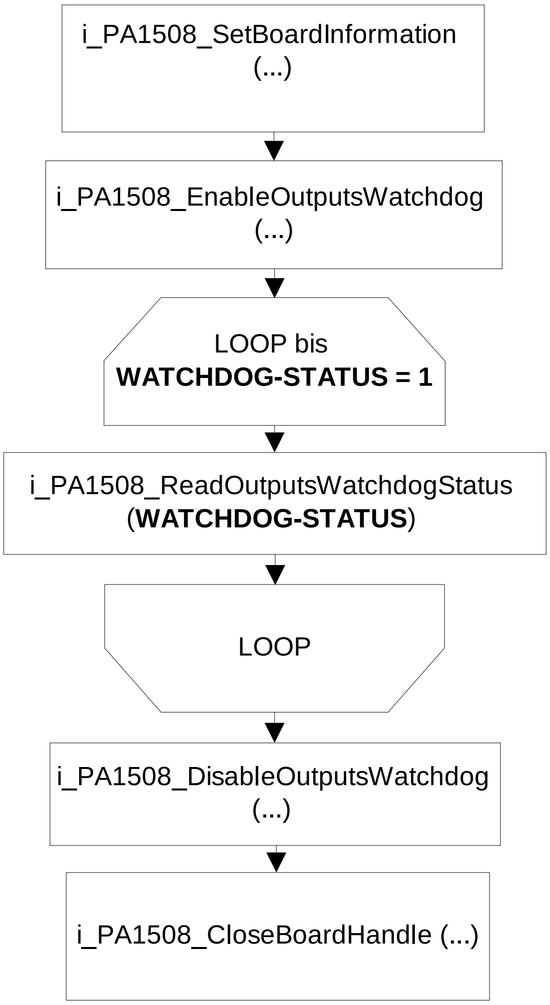
4) Setting simultaneously several digital outputs and then resetting them.


i_PA1508_SetBoardInformation
(...)
i_PA1508_Set8DigitalOutputOn
(...)
i_PA1508_CloseBoardHandle (...)

Example in C

```
#include "pa1508.h"

void main (VOID)
{
UINT  ui_Address = 0x390; //Base address of the board, set to 390Hex
BYTE  b_BoardHandle = 0;  //Variable for the board handle
BYTE  b_Channel = 0xF5;    //Number of channels to be set

/**********************************************************/
/* The function SetBoardInformation checks the parameters */
/* of the board and returns the handle of the board       */
/**********************************************************/

i_PA1508_SetBoardInformation  (ui_Address, &b_BoardHandle);

/******************************************************/
/* The function Set8DigitalOutputOn activates the     */
/* following outputs: 0,2,4,5,6,7, (b_Channel = 0xF5) */
/* and deactivates the outputs 1,3.                   */
/******************************************************/

i_PA1508_Set8DigitalOutputOn  (b_BoardHandle, b_Channel);

/******************************************************/
/* The function CloseBoardHandle blocks the           */
/* access to the board and closes the board handle    */
/******************************************************/

i_PA1508_CloseBoardHandle  (b_BoardHandle);

}
```

9.5.3 Watchdog management


i_PA1508_SetBoardInformation (...)
i_PA1508_EnableOutputsWatchdog (...)
LOOP bis
WATCHDOG-STATUS = 1
i_PA1508_ReadOutputsWatchdogStatus (**WATCHDOG-STATUS**)
LOOP
i_PA1508_DisableOutputsWatchdog (...)
i_PA1508_CloseBoardHandle (...)

Example in C

```
#include "pa1508.h"

void main (VOID)
{
UINT  ui_Address = 0x390; //Base address of the board, set to 390Hex
BYTE  b_BoardHandle = 0;  //Handle of the board
BYTE  b_WatchdogStatus = 0;  //Variable for Watchdog Status


/*********************************************************/
/* The function SetBoardInformation checks the parameters */
/* of the board and returns the handle of the board       */
/*********************************************************/

i_PA1508_SetBoardInformation  (ui_Address, &b_BoardHandle);
/**************************************/
/* The function EnableOutputsWatchdog */
/* activates the watchdog             */
/**************************************/

i_PA1508_EnableOutputsWatchdog (b_BoardHandle);

/****************************************/
/* Wait until the watchdog has run down */
/****************************************/

while (b_WatchdogStatus == 0)
   {
   i_PA1508_ReadOutputsWatchdogStatus    (b_BoardHandle,
                                   & b_WatchdogStatus);
   }

/**************************************/
/* This function RearmOutputsWatchdog */
/* starts the watchdog again          */
/**************************************/

i_PA1508_RearmOutputsWatchdog (b_BoardHandle);

/****************************************/
/* Wait until the watchdog has run down */
/****************************************/

while (b_WatchdogStatus == 0)
   {
   i_PA1508_ReadOutputsWatchdogStatus    (b_BoardHandle,
                                   & b_WatchdogStatus);
   }


/***************************************************/
/* The function DisableOutputsWatchdog deactivates */
/* the watchdog                                    */
/***************************************************/

i_PA1508_DisableOutputsWatchdog (b_BoardHandle);


/******************************************************/
/* The function CloseBoardHandle blocks the           */
/* access to the board and closes the board handle    */
/******************************************************/

i_PA1508_CloseBoardHandle  (b_BoardHandle);
          }
```

INDEX

ADDIREG
changing the configuration 19
removing 19

board
plugging 13

Board
handling 4
Limits of use 2
Physical set-up 5

Component scheme 9

Connection principle 22–23
of the inputs 23
of the outputs 22–23

Connector pin assignement 21

Diagnosis 1

DIP switches 11

Electromagnetic compatibility see EMC

EMC
Specifications 1
Tested appliance configuration 2, 5

Inputs 1, 6

Installation 10–48

Intended purpose of the board 1

Internet
error analysis 20

Limit values 6

Limits of use 2

Outputs 1, 6

PC
Minimmum requirements 6
opening 12

Personal protection
Clothing 3
Work conditons 3

Security 2

Slot
Types 12

Technical data 5–9

Watchdog 1

GLOSSARY

Active LOW signal It signals an active state by logic „0“. (Negative logic)

Address range Place in the memory of the PC in which the board operates.
The base address determines the smallest address range which the board needs to operate.

Base address The base address is the first I/O-Address of the programmable register of the board

Binary value „1“ and „0“ are binary values.

Bit Smallest information unit. State "0" or "1".

Board Layer with components for the extension of the PC applications. It represents the electronic adaptation and connection between the PC and other devices.

Bus A system connection which electrically connects all parts to enable a communication.

Byte A group of 8 bits

Cycle time Minimum time period between 2 updates

Data bus The DAQ boards of the PC exchange data with the central unit through the data bus number.

Decoding Memory places in the PC can be addressed through the 16 address bits (A0-A15). By a complete decoding of the address (up to A15), the board is only active when it is accessed through the set base address.

Debug: Programmable high languages supplying with debug functions to find errors.

DIP switches Component made of 10 different switch position. Position in component scheme: S1

Driver Group of all commands which are used to program the board.

EMC Electromagnetic compatibility

Front connector 37-pin SUB-D male connector for the connection of peripheral signals which are processed by the board

Handle parameter Identification number of the board. When several boards with the same type are inserted in the PC, the handle parameter of the driver command determines on which board the driver command is to apply.

High Signal Signal which is interpreted as logic „1“ by passing a definite switching threshold.

ISA Industrial Standard Architecture; Norm for the ISA bus

Input Unit to convert the signals present on the front connector in a digital data bus information which is then transmitted to the central unit of the PC.

Jumper Short circuit device

Load current Current which flows through the load

Low signal Signal which is interpreted as logic „0“ by passing a definite switching threshold.

Output memory The output memory is an intermediate memory which loads the counter values transmitted to the outputs

Output Unit to convert the digital data bus information in signals specified by the board. These signals are then to be transmitted from the front connector to the peripheral.

Ouput number The outputs are numbered from one upwards. The ouput number does not correspond to the pin

Pin number The front connector contains numbered connection which are described as pins.

Polling Continuous questionning of a status per Software

Reset By switching the PC on a reset signal is generated to initialise all logical components.

Resources The resources of a board are all addresses, interrupt and DMA channels with which the board communicates to the system.

Return value Value returned to the calling program after calling or operating a function of the driver commands

Switch OFF delay	Time period between the board access and the switching off of the outputs
Switch-ON delay	Time period between the board access and the switching on of the outputs.
Software	Control program for the harware operation
SPS	Programmable automatisation device set to control the system. (Stored Program System)
Timeout	Passing of the watchdog time
Update	Actualisation of the analog outputs
Write access	To program the outputs the data information must be loaded on the data bus. The data bus information can then be transmitted to the output through a cyclic write access

SYSTEM ADDRESS RANGES

Address	Function
000..01F	1st DMA controller
020..03F	1st interrupt controller
040..05F	Timer 82C54
060..06F	Keyboard controller
070..07F	Real time clock & NMI Mask
080..09F	DMA-Page Register
0A0..0BF	2nd interrupt controller
0C0..0DF	2nd DMA controller
0F0..0FF	Coprocessor
0100..01EF	Free
01F0..01F7	Hard disk controller
01F8..01FF	Free
0200..020F	Game port
0210..021F	Extension unit
0220..025F	Reserved
0260..0277	Free
0278..027F	LPT2
0280..02E7	Reserved
02E8..02EF	COM4

Address	Function
02F0..02F7	Reserved
02F8..02FF	COM2
0300...031F	Prototyp board (generally free)
0320...032F	Hard disk controller (only PC)
0330...035F	Free
0360...036F	Network boards
0370...0377	2nd floppy disk drive controller
0378...037F	LPT1
0380...038F	SDLC / BSC
0390...039F	Free
03A0...03AF	SDLC / BSC
03B0...03BF	Monochrom graphic board
03C0...03CF	EGA graphic board
03D0...03DF	CGA graphic board
03E0...03E7	Free
03E8...03EF	COM3
03F0...03F7	Diskette drive controller
03F8...03FF	COM1
0400...0FFFF	Free bzw. Redundant addressing

This indications refer to AT/386/486 Systems