DIN EN ISO 9001:2000
certified

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

Technical description

ADDINUM PA 100

Digital input board

5th edition 09/2004

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

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 MSX-Box, 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.

1 DEFINITION OF APPLICATION

1.1 Intended use

The board **PA 100** must be inserted in a PC with ISA 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 **PA 100** must not to be used as safety related part for securing emergency stop functions.

The board must not be used in potentially explosive atmospheres.

1.3 General description of the board

The 24 V digital signals coming from the peripheral are connected to the 37-pin SUB-D pin connector of board **PA 100** through a shielded cable.

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. Check the PC and cable shielding capacity prior to putting the device into operation.

The connection with our standard cable ST010 complies with the specifications:
- metallized plastic hoods
- shielded cable
- cable shield folded back and firmly screwed to the connector housing.

The use of the board according to its intended purpose includes observing all advices given in this manual and 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.

2 FUNCTIONS

The **PA 100** board is intended for parallel data input over 32 lines while simultaneous separation of peripheral and system sides. The signal inputs are divided into 4 groups of 8 bits. All inputs are physically decoupled and filtered by optical couplers. Each input has a go line to the front connector. All inputs use the same return line.

The board does not require any software initialisation. It can be operated immediately after applying the operating voltage. Data transfer between CPU and peripheral occurs directly over a buffer.

The optical coupler inputs have a maximal load of 24 mA. To avoid damage to the optical coupler through pole reversal of input voltage, a diode is connected antiparallel to the LED of the optical coupler.

Through the optical isolation, disturbances from the peripheral side to the system bus side are avoided. In addition, all inputs are sieved with a RC filter, so that the effect of inductive and capacitive disturbances is reduced.

A high dynamic and switching threshold of the input signal is achieved by inserting a Z-diode in each input line.

All inputs comply with the industry norm: +24V for logical "1".

The base address on the board is freely selectable with a 10-pole block of DIP switches and jumpers. The adjusted address range needs 4 bytes within the 64 KB-I/O-address space of the CPU.

The board is equipped for direct use in an IBM or compatible PC with a 62-pin direct connector. The peripheral is connected over a 37-pin SUB-D male connector. Thus the following connection possibilities:

- over a cable directly to contact or signal transmitters
- over our standard cable to terminal board ADDIVARIOUS **PX900**.

The testadapter PX 910 is for the above mentioned possibilities a useful help for hardware and software set-ups.

3 OPERATION

The **PA 100** board needs no initialising. It can be used immediately after power has been turned on. The address decoding relates to the 64KB- I/O-address area. Both lower address bits are decoded for selection of 4 blocks of 8 inputs. The higher address bits are compared with the on-board set switches ; the board is enabled when they are identical. The control logic takes over coordination between board and microcomputer.

The peripheral signals are connected to the front connector and are carried to the optical couplers over the protection and filter circuitry.

4 MECHANICAL SET-UP

A 1,6 mm thick printed circuit is the mechanical and electrical connection (140 x 99 mm). The connection with the microcomputer bus system occurs over a 62 pole gold-plated direct connector. The board is plugged directly into the PC and is screwed onto the back pannel of the appliance with a hinge.

5 SIGNALS - DIRECT PLUGS

A0 - A15	: Address lines
D0 - D7	: Data lines
IOR/	: I/O Read
IOW/	: I/O Write
IO CH RDY	: I/O Channel Ready
AEN	: Address Enable
RESET DRV	: System Initialising
GND	: OV, Ground
+5V	: 5V supply
Board consumption	: 150 mA

6 PERIPHERAL SIGNALS - SOCKETS

The board is connected to the peripheral via a 37-pin SUB-D male connector.

6.1 Statical data for inputs

Input current: logical "1"	: > = 15 mA (at 24V)
Signal delay	: 65 µs, nominal
Maximal input frequency	: 5 kHz
Minimal input voltage at 10 mA	: 16V (up to 200Hz)
Upper limit of input voltage	: 28 V
Test voltage of individual inputs against PC side	: 0,5 kV

7 ADJUSTING THE BOARD

The base address of the board can be adjusted with switches. For each board is assigned a group of 4 addresses for data transfer with the CPU. The position of this address block within the I/O-address space is freely disposable.

The adjustment of the board occurs over 4 jumpers and 10 DIP switches. The jumpers are marked J1 to J4 and the switches are marked S1 to S10.

J1 to J4 can be installed or removed. S1 to S10 can be switched on or off.

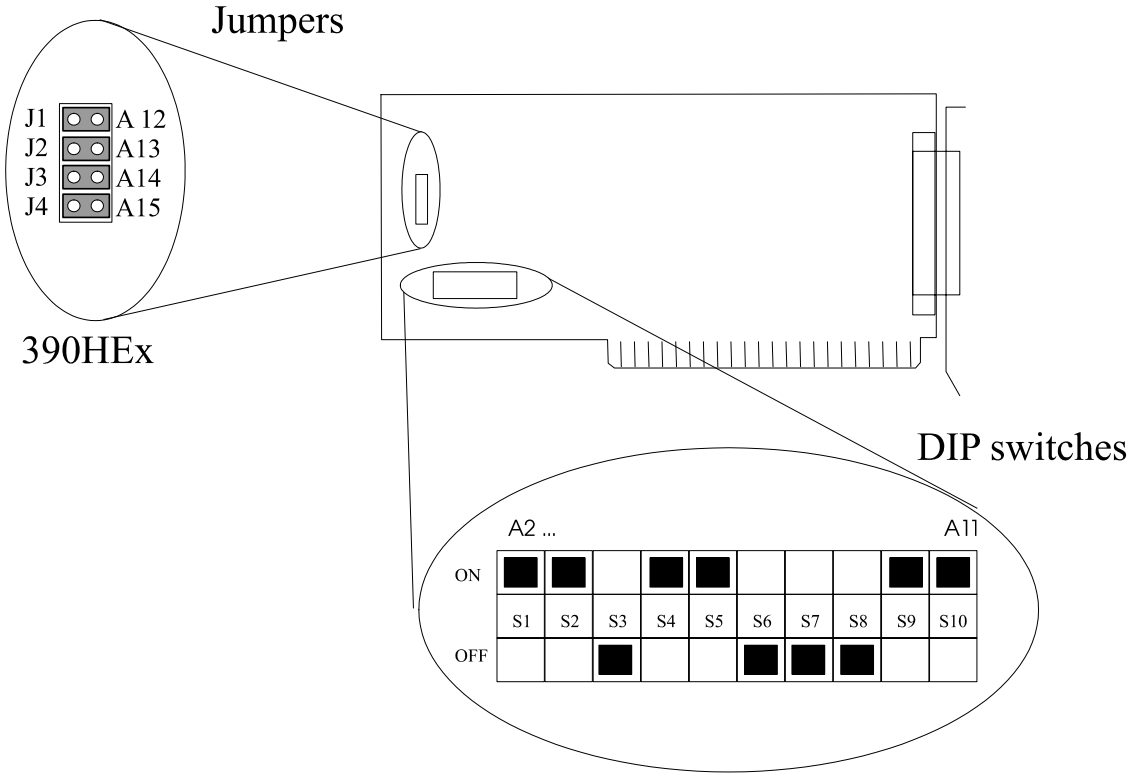
Caution ! - an installed jumper or a switch in "ON" position is equivalent to the logical 0
- a removed jumper or a switch in "OFF" position is equivalent to the logical 1

Switch:	J4	J3	J2	J1	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1
Adressbit:	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2

The board is delivered with the following configuration: **I/O-address H0390**

Switch:	J4	J3	J2	J1	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1
State:	0	0	0	0	0	0	1	1	1	0	0	1	0	0

Location of the jumper and the DIP switches on the board



8 INSTALLATION INTO THE SYSTEM

For installation, open the appliance following manufacturer instructions. Be sure to turn power off first. On the back pannel remove the covering plate of your free slot. Then press the board into the slot and fasten it to the back side with an hinge. The machine may now be closed and power may be turned on.

Afterwards system tests are carried out. The peripheral can be connected. We suggest to use the ST 010 cable, since this needs no inconvenient wiring.

A test adaptor is useful for simulating and testing input functions. With this aid you will have a quick and economical peripheral connection.

9 SOFTWARE

IMPORTANT!
Further information for installing and uninstalling the different drivers is to be found in the delivered description **"Installation instructions for the ISA bus".**

A link to the corresponding PDF file is available in the navigation pane (Bookmarks) of Acrobat Reader.

10 PROGRAMMING

For information exchange between the CPU and the board 4 addresses are assigned, which are differentiated through the 2 lowest address bits. The higher address bits A2 - A15 are compared with the DIP switch and jumper adjustment as described in chapter 7.

You will find programming examples in chapter 10.2. The board needs no software initialisation. It can be directly related to through its programmed address, for example with Basic command INP.

Suppose the address is 0390 hexadecimal. The 32 digital inputs are read with these commands:

```
A = INP (&H0390)
B = INP (&H0391)
C = INP (&H0392)
D = INP (&H0393)
```

Digital input 1 corresponds with bit 0 of INP (&H0390)
Digital input 32 corresponds with bit 7 of INP (&H0393).

In digital signal processing the state of a specified input must be checked often. For example digital input 4.

Following formulation is a simple way for checking whether an input is on logical "1" or "0"

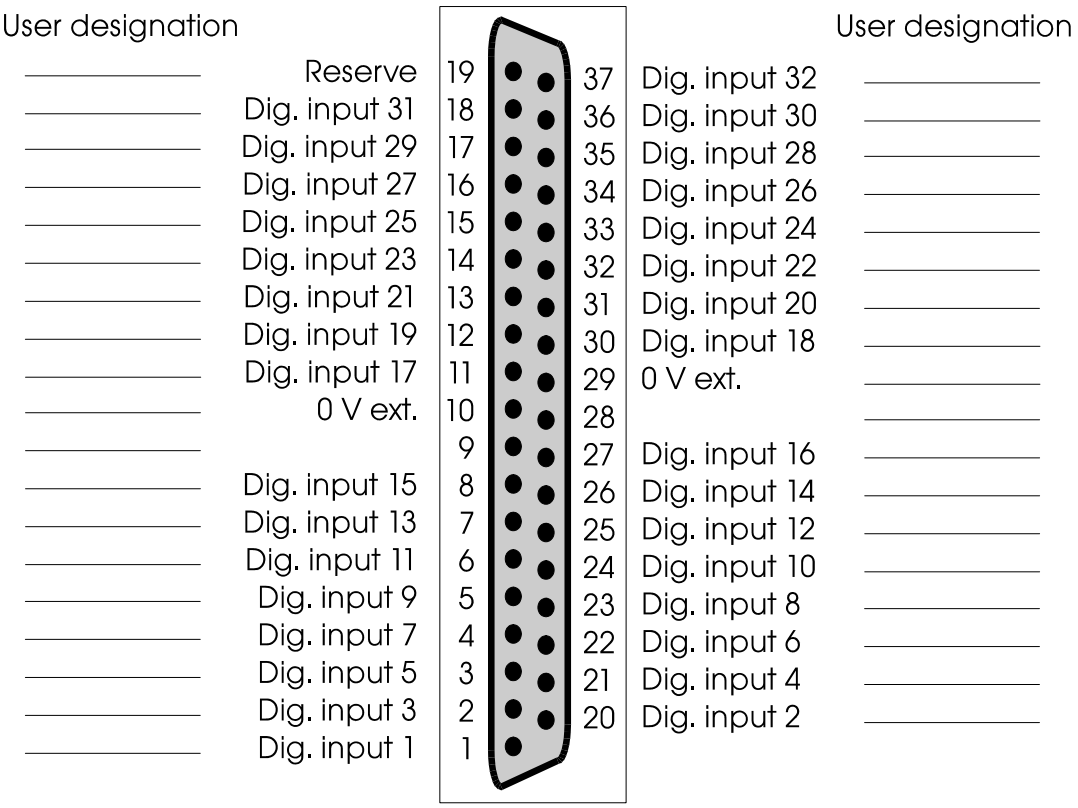
```
A = INP (&H0390)
IF (A AND &H08) = &H08 THEN PRINT "INPUT 4 = 1" ELSE
                            PRINT "INPUT 4 = 0"
```

Next formulation allows to verify if digital inputs 10 and 16 are set.

```
B = INP (&H0391)
IF (B AND &H82) = &H82 THEN PRINT "INPUT 10 and 16 = 1" ELSE

    PRINT "INPUT 10 and/or 16 = 0"
```

11 APPENDIX

11.1 Connector pin assignment



11.2 Programming examples: C, Pascal, Assembler, Basic

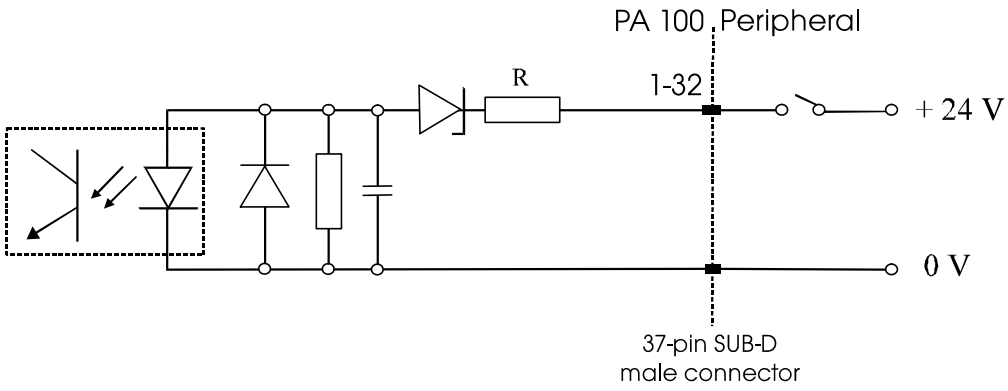
```
/* TURBO C
*/
/* Reads digital inputs of a PA100 board
*/
/* Used compiler: TURBO C version 2.0 mode SMALL
*/
#define Adr_PA100  0x390                  /* Base address of PA100
board    */
void main (void)                          /*   Main program
*/
   {
   char in_LSB1;
   char in_MSB1;
   char in_LSB2;
   char in_MSB2;
   int  in_WORD1;
   int  in_WORD2;

/* Example 1
*/
/* Digital inputs are read by four readings of 8 bits
*/

   in_LSB1 = inportb (Adr_PA100);         /* Bit 0 of byte in_LSB1
*/
                                          /* is digital input number 1
*/
   in_MSB1 = inportb (Adr_PA100 + 1);
   in_LSB2 = inportb (Adr_PA100 + 2);
   in_MSB2 = inportb (Adr_PA100 + 3);     /* Bit 7 of byte in_MSB2
*/
                                          /* is digital input number
32      */

/* Example 2
*/
/* Digital inputs are read by two readings of 16 bits
*/

   in_WORD1 = inport (Adr_PA100);         /* Bit 0 of word in_WORD1
*/
   in_WORD2 = inport (Adr_PA100 + 2);     /* -> digital input number 1
*/
                                          /* Bit 15 of word in_WORD1
*/
                                          /* -> digital input number
16      */
                                          /* Bit 0 of word in_WORD2
*/
                                          /* -> digital input number
17      */
                                          /* Bit 15 of word in_WORD2
*/
                                          /* -> digital input number
32      */
   }

(* TURBO PASCAL
*)
(* Reads digital inputs of a PA100 board
*)
(* Used compiler: TURBO PASCAL version 4.0
*)
PROGRAM  PA100;
USES DOS;
CONST
    Adr_PA100 = $390;                      (* Base address of PA100
board   *)
VAR in_1 : BYTE;
```

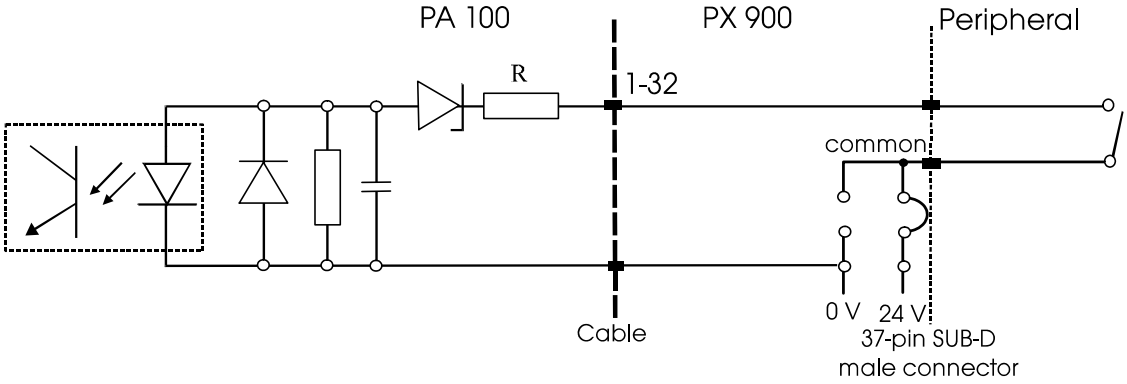
```
    in_2 : BYTE;
    in_3 : BYTE;
    in_4 : BYTE;
    in_5 : INTEGER;
    in_6 : INTEGER;
                                                (*    ***** Main program
*****    *)
BEGIN
(* Example 1 :
*)
(* Digital inputs are read by four readings of 8 bits
*)

   in_1 := PORT [Adr_PA100];                (* Bit 0 of byte in_1 is
*)
   in_2 := PORT [Adr_PA100 + 1];            (* digital input number 1
*)
   in_3 := PORT [Adr_PA100 + 2];            (* Bit 7 of byte in_4 is
*)
   in_4 := PORT [Adr_PA100 + 3];            (* digital input number 32
*)
```

```
(* Example 2 :
*)
(* Digital inputs are read by two readings of 16 bits
*)

   in_5 := PORTW [Adr_PA100];              (* Bit 0 of in_5
*)
   in_6 := PORTW [Adr_PA100 + 2];          (* -> digital input number
1     *)
                                           (* Bit 15 of in_5
*)
                                           (* -> digital input number
16     *)
                                           (* Bit 0 of in_6
*)
                                           (* -> digital input number
17     *)
                                           (* Bit 15 of in_6
*)
                                           (* -> digital input number
32     *)
END.



; TASM
; Reads digital inputs of a PA100 board
; Used compiler: TASM version 1.0 mode SMALL

Basic_Address_PA100  = 390h     ; Base address of PA100 board

   DOSSEG
   .MODEL SMALL
   .STACK 100h
   .DATA
in_1   db  ?
in_2   db  ?
in_3   db  ?
in_4   db  ?
in_5   dw  ?
in_6   dw  ?

   .CODE
   MOV  AX,@DATA
   MOV  DS,AX                   ; Initializes data segment

; Example 1 :
; Digital inputs are read by four readings of 8 bits

   MOV  DX,Basic_Address_PA100

   IN   AL,DX
   MOV  [in_1],A1               ; Bit 0 of byte in_1 is
   INC  DX                      ; digital input number 1
   IN   AL,DX
   MOV  [in_2],AL
   INC  DX
   IN   AL,DX
   MOV  [IN_3],AL
   INC  DX
   IN   AL,DX                   ; Bit 7 of byte in_4 is
   MOV  [in_4],A1               ; digital input number 32

; Example 2:
; Digital inputs are read by two readings of 16 bits
   MOV  DX,Basic_Address_PA100

   IN   AX,DX                   ; Bit 0 of in_5 -> digital input number
1
   MOV  [in_5],AX               ; Bit 15 of in_5 -> digital input
number 16
   ADD  DX,2                    ; Bit 0 of in_6 -> digital input number
17
```

```
   IN   AX,DX                    ; Bit 15 ofin_6 ->digital input number
32
   MOV  [in_6],AX
   MOV  AH,4CH                   ; Back to DOS
   INT  21H
   END



' TURBO BASIC
' Reads digital inputs of a PA100 board
' Used compiler: TURBO BASIC version 1.00

 Adr_PA100% = &H390                  ' Base address of PA100 board

' Digital inputs are read by four readings of 8 bits

   in_1% = inp (Adr_PA100%)          ' Bit 0 of word in_1 is digital
input
                                     ' number 1
   in_2% = inp (Adr_PA100% + 1)
   in_3% = inp (Adr_PA100% + 2)
   in_4% = inp (Adr_PA100% + 3)      ' Bit 7 of word
                                     'in_4 is digital input number 32
```

11.3 Circuit configuration





11.4 Connection to screw terminal boards