1. Introduction

About the PIO-32/32T(PCI) Board

PIO-32/32T(PCI) is a PCI bus-compatible add-on interface board designed for TTL level digital input/output. Plugged in a PCI bus expansion slot on the motherboard of a personal computer, the board can input and output up to 32 channels.

Features

PCI bus-compatible interface board is available to IBM PC compatibles PC in which PCI bus slots have been installed.

- TTL level input/output function, without any isolation, features a very quick signal response
- Up to 32 (8 signals x 4 groups) TTL level input signals
- Up to 32 (8 signals x 4 groups) TTL level output signals
- Four input signals can also generate interrupt requests
- Up to 30VDC, 40mA per signal, max. output
- In addition to its general-purpose input/output function, this board also supports:
    - Digital filter for input signals
    - Output data monitor
    - Handshake function for input signals of channels 0 to 15
    - Bit input/output function
    - Group input/output function

For details, refer to "Chapter 4 : I/O Ports and Registers"

Limited Three-Year Warranty

CONTEC Interface boards are warranted by CONTEC Co., LTD to be free from defects in material and workmanship for up to three years from the date of purchase by the original purchaser.

Repair will be free of charge only when this device is returned freight prepaid with a copy of the original invoice and a Return Merchandise Authorization to the distributor or the CONTEC group office, from which it was purchased.

This warranty is not applicable for scratches or normal wear, but only for the electronic circuitry and original boards. The warranty is not applicable if the device has been tampered with or damaged through abuse, mistreatment, neglect, or unreasonable use, or if the original invoice is not included, in which case repairs will be considered beyond the warranty policy.

How to Obtain Service

For replacement or repair, return the device freight prepaid, with a copy of the original invoice. Please obtain a Return Merchandise Authorization Number (RMA) from the CONTEC group office where you purchased before returning any product.

*** No product will be accepted by CONTEC group without the RMA number.**

Liability

The obligation of the warrantor is solely to repair or replace the product. In no event will the warrantor be liable for any incidental or consequential damages due to such defect or consequences that arise from inexperienced usage, misuse, or malfunction of this device.

About the Manual

This manual consists of the following chapters:

Chapter 1 Introduction

Chapter 2 Setup
This chapter describes the procedures for setting up the PIO-32/32T(PCI) board and setting its switches.

Chapter 3 External Connection
This chapter explains how to connect external devices to the board.

Chapter 4 I/O Ports and Registers
This chapter provides the assignment and definition of each I/O port bit used for the board.

Chapter 5 System Reference
This chapter summarizes hardware specifications of the board and provides circuit block diagrams.

Chapter 6 Troubleshooting
This chapter asks the set of questions you need to answer to trouble the board. It also includes a checklist to fill out before calling CONTEC group office for help.

Chapter 7 Index

2. Setup

Component Locations

Figure 2.1. shows the names of major parts on the PIO-32/32T(PCI) board.
Note that the switch setting shown below is the factory default.

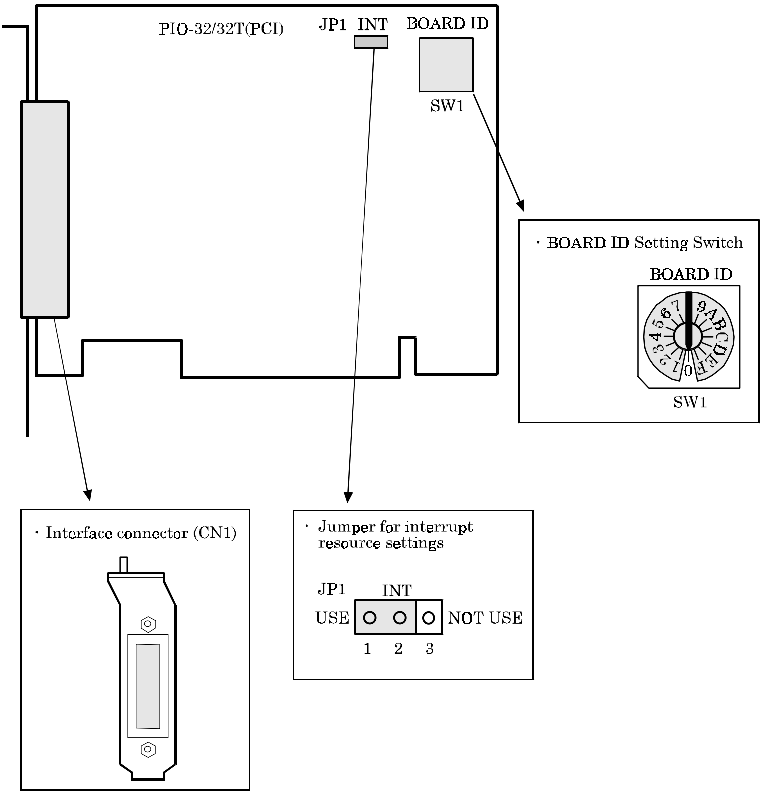

PIO-32/32T(PCI)
JP1 INT
BOARD ID
SW1
・BOARD ID Setting Switch
BOARD ID
0 1 2 3 4 5 6 7 8 9 A B C D E F
SW1
・Interface connector (CN1)
・Jumper for interrupt resource settings
JP1
INT
USE
NOT USE
1 2 3

Figure 2.1. Component Locations

Setting the Board ID

If you install two or more PIO-32/32T(PCI) boards on one personal computer, assign a different ID value to each of the boards to distinguish them.

The board IDs from 0 to F can be set to identify up to sixteen boards.

If only one board is used, the original factory setting (Board ID = 0) should be used.

Setting Procedure

To set the board ID, use the rotary switch on the board. Turn the SW1 knob to set the board ID as shown below.

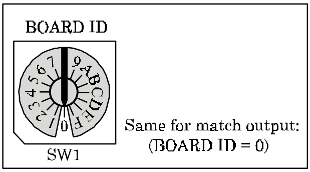


Figure 2.2.   Board ID Settings (SW1)

Setting the Interrupt Use

If you don't use the Interrupt function, you can let your PC recognize the PIO-32/32T(PCI) board as a no-interrupt function board in order to save the Interrupt resources of PC.

When the interrupt is used, this board is assigned the interrupt level from your PC.

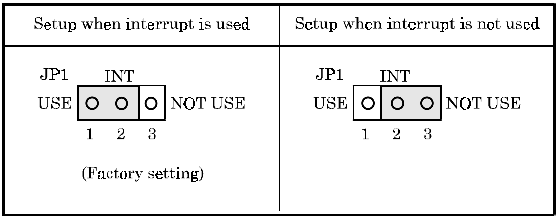

Setup when interrupt is used
Setup when interrupt is not used
JP1
INT
USE
NOT USE
1
2
3
(Factory setting)
JP1
INT
USE
NOT USE
1
2
3

Figure 2.3. Setting the Interrupt Use

Setting up the Board

The board setup method for the PIO-32/32T(PCI) board depends on the operating system being used. Set up the board using the method appropriate for the OS that you are using.

Installing for Windows 98

Installing the PIO-32/32T(PCI) board hardware

Before the PIO-32/32T(PCI) board can be used under Windows 98, the OS must recognize the I/O addresses and interrupt level (IRQ) to be used for the PIO-32/32T(PCI) board. Use the appropriate procedure to install the PIO-32/32T(PCI) board.

(1) Set the board ID.

(2) Be sure to check that the personal computer is off; then plug the board into a PCI bus slot in the system.

(3) Turn the personal computer ON to start up Windows 98.

(4) Windows 98 will come up with the [New Hardware] detection dialog box. In the [Add New Hardware Wizard] that appears next, check that "Multimedia Device" has been listed, then select [Next>].

(5) In the next dialog box, select a radio button of [Search for the best driver for your device. (Recommended).], then select [Next>].

(6) In the next dialog box, select both of the [Floppy disk drives (F)] and [Specify Location (L)] check boxes, then enter the drive name and the directory name, WIN95, into the [Location] field. (In the next dialog box, select two check box both of [Floppy disk drives] and [Specify Location:], then enter the drive name in the [Location] field.)
Insert the attached FD into the disk drive, then select [Next>].

(7) In the [Windows driver file search for the device] dialog box, check that "CONTEC Co., Ltd. - PIO-32/32T(PCI)" and "DIO_PI0.INF" in the [Location of driver] has been listed, then select [Next>].

(8) In the next dialog box, check the "Windows has finished installing the software that your new hardware device requires." message, then select [Finish]. After completed the board installation, be sure to check the assigned resources. The "Checking resources" part of this section explains you how to check the OS assigned resources of this board. (When board setup has been completed, be sure to check the assigned resources.)

Method of installing two or more PIO-32/32T(PCI) boards :

Follow the procedure below to install two PIO-32/32T(PCI) boards for use under Windows 98.

(1) Check the board ID of the first PIO-32/32T(PCI) board, plug it into a PCI bus slot, then start up Windows 98 to install the first board correctly.

(2) Check that the first PIO-32/32T(PCI) board has been set up correctly, as described in "Checking resources". Be sure to complete installation of the first board before attempting to install the second one.

(3) Exit Windows 98 and turn the personal computer off.

(4) Check the board ID of the second PIO-32/32T(PCI) board, then plug it into a PCI bus slot. Assign ID to the second board a board ID that is different from the ID assigned to the first board.

(5) Turn the personal computer on again to start up Windows 98.

(6) Windows 98 will come up with the [New Hardware] detection dialog box. In the [Add New Hardware Wizard] that appears next, check that "CONTEC Co., Ltd.- PIO-32/32T(PCI)" has been listed, then select [Next>].

(7) In the next dialog box, select a radio button of [Display a list of all the drivers in a specific location, so you can select the driver you want.], then select [Next>].

(8) In the next dialog box, select "CONTEC Co., Ltd. - PIO-32/32T(PCI) " from [Models], then select [Next>].

(9) In the [Windows driver file search for the device] dialog box, check that "CONTEC Co., Ltd. - PIO-32/32T(PCI)" and "CONTEC~*.INF" in the [Location of driver] has been listed, then select [Next>]. (* is a number which the OS assigned.)

(10) In the next dialog box, check the "Windows has finished installing the software that your new hardware device requires." message, then select [Finish]. This completes installation of the PIO-32/32T(PCI) board. After finishing installing the board, be sure to check the assigned resources again.

For installing the third board and any additional boards, follow the same steps as those for installing a second board. Before you can install a third board or additional boards, all PIO-32/32T(PCI) boards that are already installed must be in PCI bus slots.

Notes!

□ *The second PIO-32/32T(PCI) board cannot be properly installed unless the resources (I/O addresses and interrupt level) for the board can be allocated. Before attempting to install the second board, first determine what PC resources are free.*

□ *The resources used for each PIO-32/32T(PCI) board do not depend on the location of the PCI bus slot or the board itself. If you remove two or more PIO-32/32T(PCI) boards that have already been installed and then remount one of them on the computer, it is unknown which one of the sets of resources previously assigned to the two boards is assigned to the remounted board. In this case, re-check the resource settings.*

Checking resources

Follow the steps below to check the assigned resources managed by the OS.

(1) Double-click on the [System] option in [Control Panel] to open the [System Properties] property sheet. Select the [Device Manager] tab.

(2) Click on the [Multi-function adapters] folder.

(3) Click on the [CONTEC Co., Ltd. - PIO-32/32T(PCI)] folder to display its properties.

(4) Select the [Resources] tab to check the device type, resource settings, and the conflicting device list.

(5) If you want to change a resource setting, uncheck the [Use automatic settings] option in advance. To change the I/O address range (input/output Range), change the configuration name in the [Setting based on:] field. Since the interrupt level (Interrupt Request) cannot be changed, use the assigned IRQ.

Support software

CONTEC provides the following driver software for Windows 98.
API-PAC(W32) Ver. Dec. 1998 or later

This driver software supports up to sixteen PIO-32/32T(PCI) boards.

Note that when API-PAC(W32) is used, only 32 bit versions of development languages can be used. Neither driver can be supported by any language dedicated to 16 bit applications.

Installing for Windows 95

Installing the PIO-32/32T(PCI) board hardware

Before the PIO-32/32T(PCI) board can be used under the Windows 95 operation system (OS), the OS must recognize the assigned I/O address range and the interrupt level (IRQ) of this board and register these information into OS itself. Refer the following procedure to register the board information for Windows 95.

Checking the OS version

Note that the procedure for installing the PIO-32/32T(PCI) board depends on which version of Windows 95 you are using. Check the version of Windows 95 on your system as follows before installing the PIO-32/32T(PCI) board.

(1) Open [Control Panel] from [My Computer].

(2) Double-click on the [System] option to open the [System Properties] property sheet.

(3) Check the "System:" number displayed on the [General] page.
System : Microsoft Windows 95
4.00.950

The version numbers of Windows 95 include 4.00.950, 4.00,950a, 4.00.950B and 4.00.950C. The PIO-32/32T(PCI) board setup depends on the version number of Windows 95 that is being used.

Procedure for use under Windows 95 version 4.00.950 or 4.00.950a :

(1) Set the board ID.

(2) Be sure to check that the personal computer is off; then plug the board into a PCI bus slot in the system.

(3) Turn the personal computer ON to start up Windows 95.

(4) Windows 95 will come up with the [New Hardware] detection dialog box. Select [Multimedia Device: Select which driver you want to install for your new hardware.] and then [Driver from disk provided by hardware manufacturer].

(5) In the [Install From Disk] dialog box, the attached FD into the disk drive, enter the drive name and directory name in the [Copy Distributed File From] field, then click on [OK].
This completes installation of the PIO-32/32T(PCI) board.

(6) Follow the instructions on the screen to complete installation of the PIO-32/32T(PCI) board.
After completed the board installation, be sure to check the assigned resources. The "Checking resources" part of this section explains you how to check the OS assigned resources of this board.

Procedure for use under Windows 95 version 4.00.950B or 4.00.950C :

(1) Set the board ID.

(2) Be sure to check that the personal computer is off; then plug the board into a PCI bus slot in the system.

(3) Turn the personal computer on to start up Windows 95.

(4) Windows 95 will come up with the [New Hardware] detection dialog box. In the [Device Driver Wizard] that appears next, check that "CONTEC Co., Ltd. - Multimedia Device" has been listed, then select [Next>].

(5) In the next dialog box, select [Specify Location...]. Insert the attached FD in a drive, enter the drive name and directory name (WIN95) in the [Location] field, then click on [OK].

In the next dialog box, check the "Updated driver found for this device" message, then select [End]. This completes installation of the PIO-32/32T(PCI) board. After completed the board installation, be sure to check the assigned resources. The "Checking resources" part of this section explains you how to check the OS assigned resources of this board.

Method of installing two or more PIO-32/32T(PCI) boards (For use under Windows 95 version 4.00.950 or 4.00.950a) :

Follow the procedure below to install two PIO-32/32T(PCI) boards for use under Windows 95 version 4.00.950 or 4.00.950a.

(1) Check the board ID of the first PIO-32/32T(PCI) board, plug it into a PCI bus slot, then start up Windows 95 to install the first board correctly.

(2) Check that the first PIO-32/32T(PCI) board has been set up correctly, as described in "Checking resources". Be sure to complete installation of the first board before attempting to install the second one.

(3) Exit Windows 95 and turn the personal computer off.

(4) Check the board ID of the second PIO-32/32T(PCI) board, then plug it into a PCI bus slot. Assign ID to the second board a board ID that is different from the ID assigned to the first board.

(5) Turn the personal computer on again to start up Windows 95.

(6) Windows 95 will come up with the [New Hardware] detection dialog box. In [Multimedia Device: Select which driver you want to install for your new hardware.], select [Select from List].

(7) The [Select Hardware Type] dialog box will then appear.
In [Select Hardware Type to Install], select [Other Devices].

(8) In the [Select Device] dialog box that appears, select [CONTEC] from [Manufacturers] and select [CONTEC Co., Ltd. - PIO-32/32T(PCI)] from [Models].

(9) The [Change System Settings] dialog box appears. Follow the messages to restart the computer.

(10) When Windows 95 is restarted, installation of the second PIO-32/32T(PCI) board is completed. Check the assigned resources again.

For installing the third board and any additional boards, follow the same steps as those for installing a second board. Before you can install a third board or additional boards, all PIO-32/32T(PCI) boards that are already installed must be in PCI bus slots.

Notes!

- *The second PIO-32/32T(PCI) board cannot be properly installed unless the resources (I/O addresses and interrupt level) for the board can be allocated. Before attempting to install the second board, first determine what PC resources are free.*

- *The resources used for each PIO-32/32T(PCI) board do not depend on the location of the PCI bus slot or the board itself. If you remove two or more PIO-32/32T(PCI) boards that have already been installed and then re-mount one of them on the computer, it is unknown which one of the sets of resources previously assigned to the two boards is assigned to the re-mounted board. In this case, re-check the resource settings.*

Method of installing two or more PIO-32/32T(PCI) Boards (For use under Windows 95 version 4.00.950B or 4.00.950C) :

Follow the procedure below to install two PIO-32/32T(PCI) boards for use under Windows 95 version 4.00.950B or 4.00.950C.

(1) Check the board ID of the first PIO-32/32T(PCI) board. Then plug it into a PCI bus slot. Finally, start up Windows 95 to install the first board correctly.

(2) Check that the first PIO-32/32T(PCI) board has been set up correctly, as described in "Checking resources". Be sure to complete installation of the first board before attempting to install the second one.

(3) Exit Windows 95 and turn the personal computer OFF.

(4) Check the board ID of the second PIO-32/32T(PCI) board. Then plug it into a PCI bus slot. Assign to the second board a board ID different from that assigned to the first board.

(5) Turn the personal computer ON again to start up Windows 95.

(6) The OS will then automatically install the second board. When the installation has been completed, re-check the assigned resources.

For installing the third board and any additional boards, follow the same steps as those for installing a second board. Before you can install a third board or additional boards, all PIO-32/32T(PCI) boards that are already installed must be in PCI bus slots.

Notes!

- *A second PIO-32/32T(PCI) board cannot be properly installed unless the resources (I/O addresses and interrupt level) to be used for the board can be allocated. Before attempting to install a second board, first determine which PC resources are free.*

- *The resources used for each PIO-32/32T(PCI) board do not depend on the location of the PCI bus slot or the board itself. If you remove two or more PIO-32/32T(PCI) boards that have already been installed and then re-mount one of them on the computer, it is unknown which one of the sets of resources previously assigned to the two boards is assigned to the re-mounted board. In this case, check the resource settings again.*

Checking resources

Follow the steps below to check the assigned resources managed by the OS.

(1) Double-click on the [System] option in [Control Panel] to open the [System Properties] property sheet. Select the [Device Manager] tab.

(2) Click on the [Multi-function adapters] folder.

(3) Click on the [CONTEC Co., Ltd. - PIO-32/32T(PCI)] folder to display its properties.

(4) Select the [Resources] tab to check the device type, resource settings, and the conflicting device list.

(5) If you want to change a resource setting, uncheck the [Use automatic settings] option in advance. To change the I/O address range (Input/Output Range), change the configuration name in the [Setting based on:] field. Since the interrupt level (Interrupt Request) cannot be changed, use the assigned IRQ.

Support software

CONTEC provides the following driver software for Windows 95:
API-PAC(W32) Ver. Dec. 1998 or later
This driver software supports up to sixteen PIO-32/32T(PCI) boards.

Note that when API-PAC(W32) is used, only 32 bit versions of development languages can be used. Neither driver can be supported by any language dedicated to 16 bit applications.

Installing for Windows NT

Installing the PIO-32/32T(PCI) board requires separately priced CONTEC driver software. Follow the procedure below to install the board.

Verifying PC settings

Be sure that [PnP OS] is either [disabled] or set to [not to use] in the PC's BIOS setup. If this is set to [Windows 95], for example, the PIO-32/32T(PCI) might not be recognized properly.

Installing the PIO-32/32T(PCI) driver software

To use the PIO-32/32T(PCI) board under Windows NT operation system, you need an optional CONTEC installation driver program. Refer the following procedure to install this board to the
Windows NT OS.

(1) Set the board ID.

(2) Be sure that the personal computer power is off. Then plug the board into a PCI bus slot in the system.

(3) Start Windows NT with Administrator.

(4) Execute the optional driver program to install the board. Refer the driver program's manual or the driver program's help file for details.

After completed the board installation, be sure to check the assigned resources. The "Checking resources" part of this section explains you how to check the OS assigned resources of this board.

Checking resources

Follow the steps below to check the assigned resources managed by the OS.

(1) Open [Windows NT Diagnostic Program] from [Management Tools].

(2) Select [Resources] (IRQ/port settings). Check the types and settings of resources assigned to the relevant driver and the corresponding device list.

Support software

CONTEC provides the following driver software for Windows NT.
API-PAC(W32) Ver. Dec. 1998 or later
This driver software supports up to sixteen PIO-32/32T(PCI) boards.

Note that when API-PAC(W32) is used, only 32 bit versions of development languages can be used.

Installing for other OS System

For all the other operation systems, in addition of the Windows OS, we use MS-DOS as an example to show how to use the PIO-32/32T(PCI) board under the OS.

For all the other operation systems, refer the MS-DOS programs of the attached FD as an example.

For a PCI bus board, the system will automatically assign a usable resource/resources to the board. Refer the following procedure to copy the attached programs and to confirm the assigned resource/resources.

Procedure

(1) Set the board ID.

(2) Be sure that the personal computer is off; then plug the board into a PCI bus slot in the personal computer.

(3) Turn the personal computer ON to start up MS-DOS.

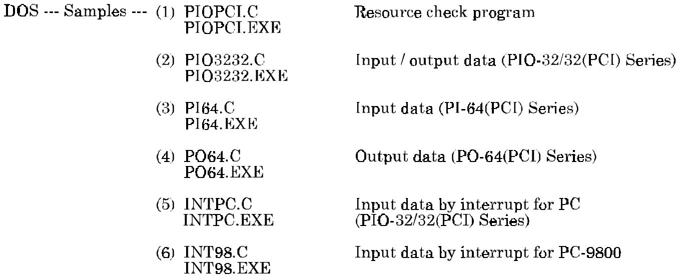
(4) Copy the programs that are under the DOS directory of the attached FD to a directory of your HDD.

(5) Execute the PIOPCI.EXE resource confirmation program.

(6) Check the I/O addresses and interrupt level (IRQ) displayed on the screen.

Sample programs for MS-DOS

The sample programs for using the PIO-32/32T(PCI) board under MS-DOS are written in Microsoft C.
The attached FD contains following sample programs.



DOS --- Samples ---	(1)	PIOPCI.C PIOPCI.EXE	Resource check program
	(2)	PIO3232.C PIO3232.EXE	Input / output data (PIO-32/32(PCI) Series)
	(3)	PI64.C PI64.EXE	Input data (PI-64(PCI) Series)
	(4)	PO64.C PO64.EXE	Output data (PO-64(PCI) Series)
	(5)	INTPC.C INTPC.EXE	Input data by interrupt for PC (PIO-32/32(PCI) Series)
	(6)	INT98.C INT98.EXE	Input data by interrupt for PC-9800

Figure 2.4. Sample Programs on FD

For details on I/O addresses, see Chapter 4 "I/O Ports and Registers."

3. External Connection

Interface Connector

Connecting the Interface Connector

To connect an external device to this board, plug the cable from the device into the interface connector (CN1).

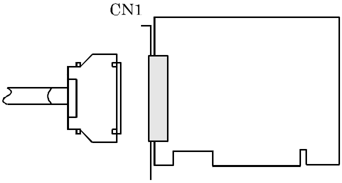


Figure 3.1. Connecting the Interface Connector

□ Connector used
PCR-E96LMD [mfd. by HONDA]

□ Applicable connector
PCR-E96FA [mfd. by HONDA] or equivalent

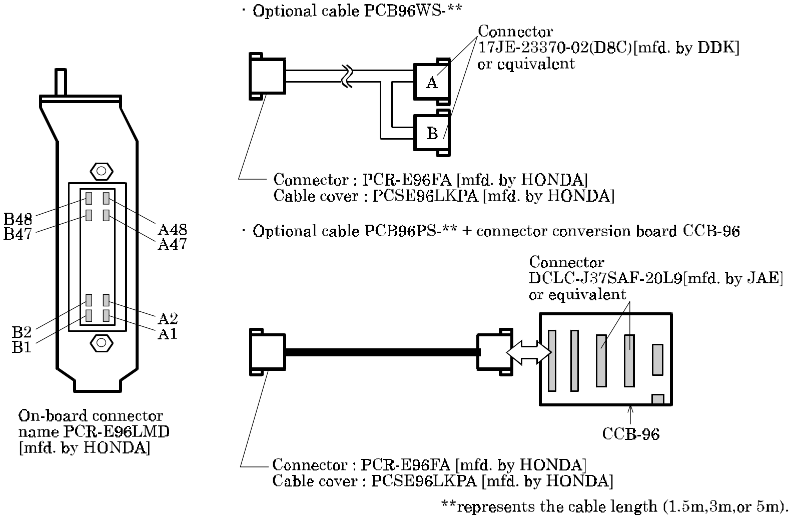
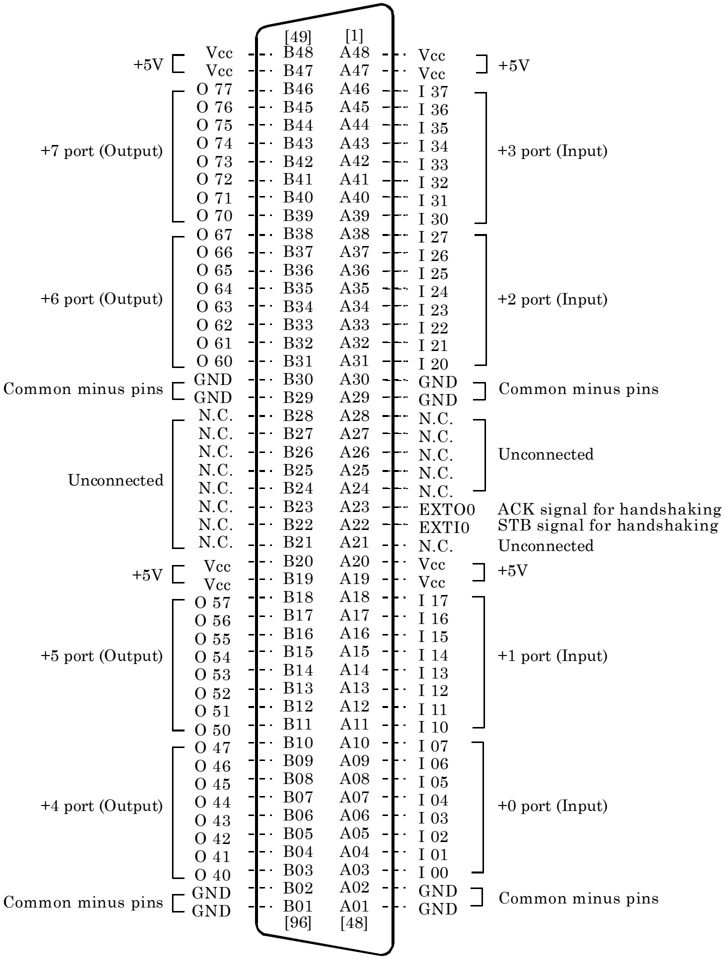

· Optional cable PCB96WS-**
Connector
17JE-23370-02(D8C)[mfd. by DDK]
or equivalent
A
B
Connector : PCR-E96FA [mfd. by HONDA]
Cable cover : PCSE96LKPA [mfd. by HONDA]
B48
B47
A48
A47
B2
B1
A2
A1
On-board connector
name PCR-E96LMD
[mfd. by HONDA]
· Optional cable PCB96PS-** + connector conversion board CCB-96
Connector
DCLC-J37SAF-20L9[mfd. by JAE]
or equivalent
CCB-96
Connector : PCR-E96FA [mfd. by HONDA]
Cable cover : PCSE96LKPA [mfd. by HONDA]
**represents the cable length (1.5m,3m,or 5m).

Figure 3.2. Connector Used

Interface Connector Pin Assignment

To connect an external device to this interface board, plug the device into the on-board 96-pin connector shown below in Figure 3.3.



		[49]	[1]		
+5V	Vcc	B48	A48	Vcc	+5V
	Vcc	B47	A47	Vcc	
+7 port (Output)	O 77	B46	A46	I 37	+3 port (Input)
	O 76	B45	A45	I 36	
	O 75	B44	A44	I 35	
	O 74	B43	A43	I 34	
	O 73	B42	A42	I 33	
	O 72	B41	A41	I 32	
	O 71	B40	A40	I 31	
	O 70	B39	A39	I 30	
+6 port (Output)	O 67	B38	A38	I 27	+2 port (Input)
	O 66	B37	A37	I 26	
	O 65	B36	A36	I 25	
	O 64	B35	A35	I 24	
	O 63	B34	A34	I 23	
	O 62	B33	A33	I 22	
	O 61	B32	A32	I 21	
	O 60	B31	A31	I 20	
Common minus pins	GND	B30	A30	GND	Common minus pins
	GND	B29	A29	GND	
Unconnected	N.C.	B28	A28	N.C.	Unconnected
	N.C.	B27	A27	N.C.	
	N.C.	B26	A26	N.C.	
	N.C.	B25	A25	N.C.	
	N.C.	B24	A24	N.C.	
	N.C.	B23	A23	EXTO0	ACK signal for handshaking
	N.C.	B22	A22	EXTI0	STB signal for handshaking
	N.C.	B21	A21	N.C.	Unconnected
+5V	Vcc	B20	A20	Vcc	+5V
	Vcc	B19	A19	Vcc	
+5 port (Output)	O 57	B18	A18	I 17	+1 port (Input)
	O 56	B17	A17	I 16	
	O 55	B16	A16	I 15	
	O 54	B15	A15	I 14	
	O 53	B14	A14	I 13	
	O 52	B13	A13	I 12	
	O 51	B12	A12	I 11	
	O 50	B11	A11	I 10	
+4 port (Output)	O 47	B10	A10	I 07	+0 port (Input)
	O 46	B09	A09	I 06	
	O 45	B08	A08	I 05	
	O 44	B07	A07	I 04	
	O 43	B06	A06	I 03	
	O 42	B05	A05	I 02	
	O 41	B04	A04	I 01	
	O 40	B03	A03	I 00	
Common minus pins	GND	B02	A02	GND	Common minus pins
	GND	B01	A01	GND	
		[96]	[48]		

Figure 3.3. Interface Connector Pin Assignment

Note!

The numbers in brackets are pin numbers defined by the connector manufacturer.

PCB96WS and CCB-96 Signal Assignments

Optional cable connectors and their corresponding signals are shown in Figure 3.4 below.

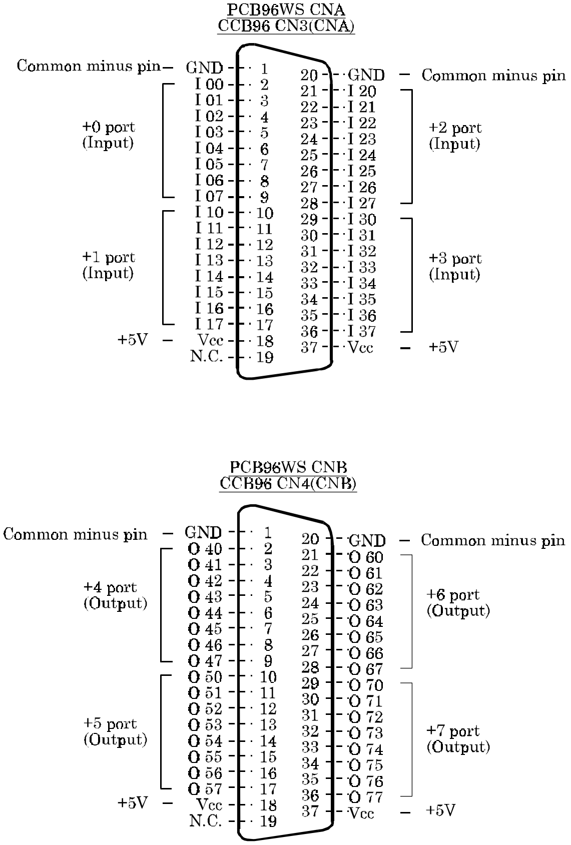


Figure 3.4. PCB96WS and CCB-96 Signal Assignments

Note!

Not support signals for handshaking.

Input Circuit and Output Circuit

Input Circuit

The input circuit of this board is illustrated in following Figure. The external digital signal, witch connected to the input logic, should be a TTL level signal. The computer will recognize this inputted signal by a negative logic. All the input signals are pull-upped on board therefore a relay switch signal or a semiconductor switch signal can be directly connected between the input signal and the minus common (GND).

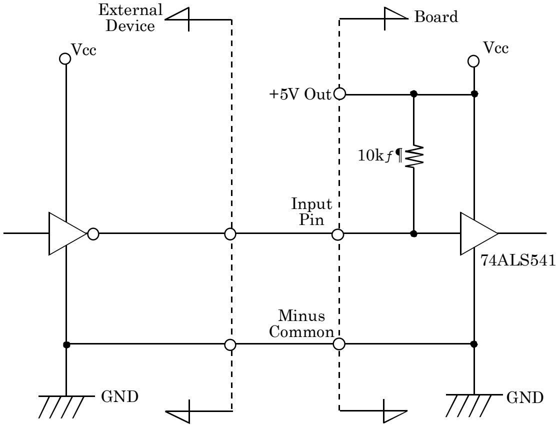


Figure 3.5. Input Circuit

Output Circuit

The output circuit of this board is illustrated in following Figure. The output signals are open-collector type signals. The board uses a negative logic to output the signals. All the output signals are NOT pull-upped on board therefore user must pull up these output signals on the application side.

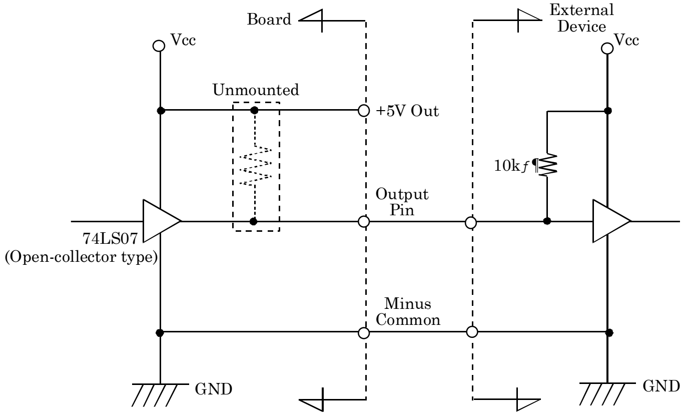


Figure 3.6. Output Circuit

STB, ACK Input Circuit and Output Circuit

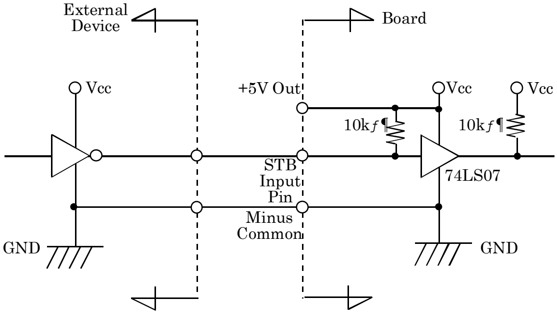


Figure 3.7. STB Input Circuit

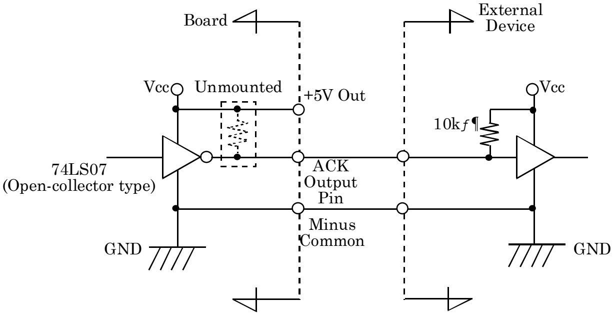


Figure 3.8. ACK Output Circuit

Connection Example

When you use I00 input and O40 output of this board, connecting method may be as shown in Figure 3.9.

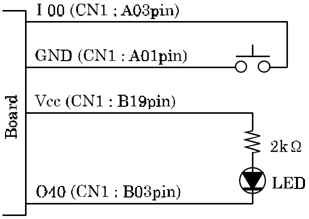


Figure 3.9. Connection Example

Surge Protection

When connecting to digital outputs a load that may generate a voltage surge or current, for example an inductive load such as a relay coil or incandescent lamp, suitable protection measures are required to prevent damage to the output stage or malfunction owing to noise. The instantaneous interruption of current flowing through a coil, including a relay, results in the sudden generation of a high-voltage pulse. If the voltage exceeds the withstand voltage of the transistor, the transistor performance may be degraded or the transistor may be damaged. To prevent this, be sure to connect a surge absorption element when driving an inductive load including a relay coil. Example of measures against voltage surge are shown in Figure 3.10 below.

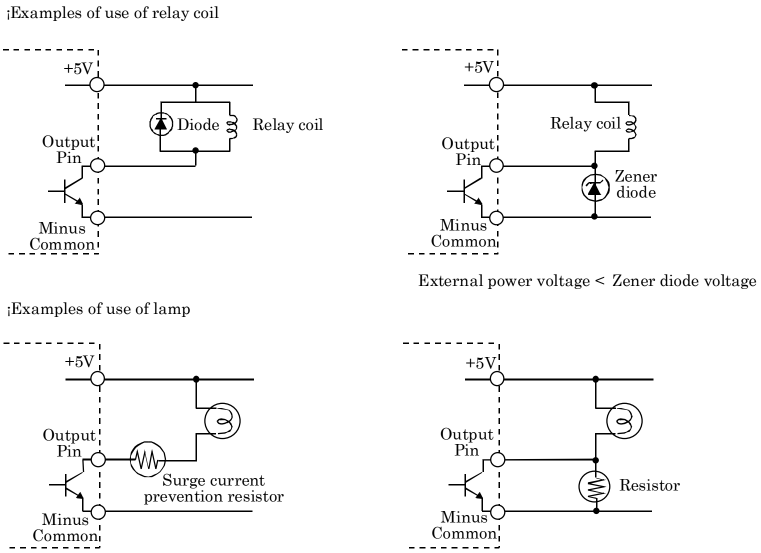


Figure 3.10. Samples of Voltage Surge Protection

Note!

The protection circuit must be installed less than 50 cm from the load and contact to provide effective protection.

A Protection Function of the +5V Outputs

A protection function, which prevents excessive current flow from the +5V outputs, is attached to this board. In case of accidental short of the +5V output and GND, for example, the function works, and the board operation may become impossible temporarily.
In such a case, you should turn the PC off and wait for several minutes before you use the board again.

4. I/O Ports and Registers

I/O Address Map

Board I/O Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
+0H	Input Group 0							
	I07 [7]	I06 [6]	I05 [5]	I04 [4]	I03 [3]	I02 [2]	I01 [1]	I00 [0]
+1H	Input Group 1							
	I17 [15]	I16 [14]	I15 [13]	I14 [12]	I13 [11]	I12 [10]	I11 [9]	I10 [8]
+2H	Input Group 2							
	I27 [23]	I26 [22]	I25 [21]	I24 [20]	I23 [19]	I22 [18]	I21 [17]	I20 [16]
+3H	Input Group 3							
	I37 [31]	I36 [30]	I35 [29]	I34 [28]	I33 [27]	I32 [26]	I31 [25]	I30 [24]
+4H	Input Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Input Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Input Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Input Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]
+8H	Handshake Status Input							
	X	X	X	X	INT	IBF	STB	ACK
+9H	Not Allowed							
+AH	Bit Data Input							
	0	0	0	0	0	0	0	BDT
+BH	Not Allowed							
+CH	Group Data Input							
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
+DH	Not Allowed							
+EH	Interrupt Setting							
	X	X	IN1	IN0	X	IR	IS1	IS0
+FH	Digital Filter Setting							
	X	X	X	ST4	ST3	ST2	ST1	ST0
+10H	Not Allowed							
+11H	Interrupt Status							
	0	0	0	0	INTS3	INTS2	INTS1	INTS0
+12H , +1FH	Not Allowed							

Ixx represents an input signal;
Oxx represents an output signal
that will be read back here;
Numbers in brackets [ ] represent
input bit numbers.

Note ¥ All access except to input group 0 to 7 (port +0 to +7) should be byte access.
¥ Input by word access to input group 0 to 7 should be to I/O addresses that are multiples of 2 (+0,+2,+4,+6).
¥ Input by double word access to input group 0 to 7 should be to I/O addresses that are multiples of 4 (+0, +4).

Figure 4.1. Input Port Assignments

Board I/O Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
+0H	Output Group 0							
	Not Allowed							
+1H	Output Group 1							
	Not Allowed							
+2H	Output Group 2							
	Not Allowed							
+3H	Output Group 3							
	Not Allowed							
+4H	Output Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Output Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Output Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Output Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]
+8H	ACK Signal Setting							
	0	0	0	0	0	0	0	ACK
+9H	Not Allowed							
+AH	Bit Data							
	0	0	0	0	0	0	0	BDT
+BH	Bit Select							
	0	0	BS5	BS4	BS3	BS2	BS1	BS0
+CH	Group Data							
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
+DH	Group Select							
	0	0	0	0	0	PS2	PS1	PS0
+EH	Interrupt Setting							
	0	0	IN1	IN0	0	IR	IS1	IS0
+FH	Digital Filter Setting Register							
	0	0	0	ST4	ST3	ST2	ST1	ST0
+10H	Interrupt Musk Register							
	0	0	0	0	INTM3	INTM2	INTM1	INTM0
+11H	Interrupt Clear Register							
	0	0	0	0	INTC3	INTC2	INTC1	INTC0
+12H ~ +1FH	Not Allowed							

Oxx represents an output signal;
Numbers in brackets [ ] represent
output bit numbers.

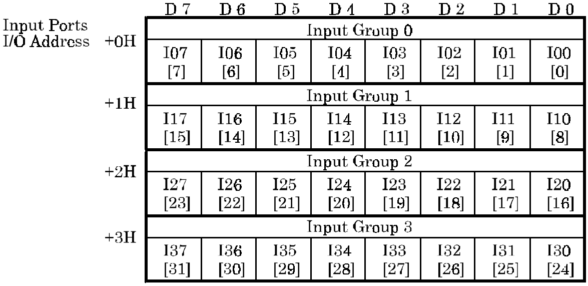
Note ¥ All access except to output group 0 to 7 (port +0 to +7) should be byte access.
¥ Output by word access from output group 4 - 7 should be from I/O addresses that are multiples of 2 (+4,+6).
¥ Output by double word access from output group 4 - 7 should be from I/O addresses that are multiples of 4 (+4).

Figure 4.2. Output Port Assignments

Input/Output Data by Direct Access to I/O Ports

Data Input

I/O address + 0H to + 3H input ports are used to read input channel data. The following table shows input channels and their corresponding input ports. If an input channel is "ON" (low level), the corresponding bit of the input port contains "1". For example, if channel I07 is ON, bit D7 of the I/O address + 0 port is read "1". If an input channel is "OFF" (high level), the corresponding bit of the input port is read "0".



Input Ports I/O Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
+0H	Input Group 0							
	I07 [7]	I06 [6]	I05 [5]	I04 [4]	I03 [3]	I02 [2]	I01 [1]	I00 [0]
+1H	Input Group 1							
	I17 [15]	I16 [14]	I15 [13]	I14 [12]	I13 [11]	I12 [10]	I11 [9]	I10 [8]
+2H	Input Group 2							
	I27 [23]	I26 [22]	I25 [21]	I24 [20]	I23 [19]	I22 [18]	I21 [17]	I20 [16]
+3H	Input Group 3							
	I37 [31]	I36 [30]	I35 [29]	I34 [28]	I33 [27]	I32 [26]	I31 [25]	I30 [24]

Ixx represents an input signal of CN1 connector

Figure 4.3. Input Port " I/O Address +0H ~ 3H "

Programming examples

The following programming examples check the input channel I07. If this channel is "ON" then program will continue. The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

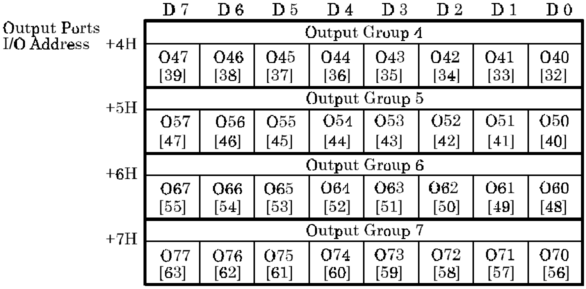
```
DAT% = INPUT(PORT%)
IF(DAT% AND &H80) = &H80 THEN
     □
     □
     □
```

Microsoft C or C++ (MS-DOS version)

```
data_in = inp(port);
while(data_in & 0x80)
     □
     □
     □
```

Data Output

I/O address + 4H to + 7H output ports are used to output data.
The following table shows the relationship of output channels and output ports. Setting an output bit of output ports to "1" will switch the corresponding output transistor to "ON" (low level). Resetting an output bit of output ports to "0" will switch the corresponding output transistor to "OFF" (high level).



Output Ports I/O Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
+4H	Output Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Output Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Output Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Output Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]

Oxx represents an output signal of CN1 connector

Figure 4.4. Output Port " I/O Address +4H ~ 7H "

Programming examples

The following programming examples turn the O47 output transistor to "ON". The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

```
    OUT PORT% + &H04, &H80
```

Microsoft C or C++ (MS-DOS version)

```
    outp(port + 0x04, 0x80) ;
```

Note!

When the PC is turned ON, all output ports are reset to "0".

Input/Output Data by Bit Number

PIO-32/32T(PCI) supports bit input/output function. Under this function, you can specify a bit number and then input this bit status or output to this bit. The following figure shows the relations of bit numbers and their corresponding input ports.

Input Ports I/O Address	D7	D6	D5	D4	D3	D2	D1	D0
+0H	Input Group 0							
	I07 [7]	I06 [6]	I05 [5]	I04 [4]	I03 [3]	I02 [2]	I01 [1]	I00 [0]
+1H	Input Group 1							
	I17 [15]	I16 [14]	I15 [13]	I14 [12]	I13 [11]	I12 [10]	I11 [9]	I10 [8]
+2H	Input Group 2							
	I27 [23]	I26 [22]	I25 [21]	I24 [20]	I23 [19]	I22 [18]	I21 [17]	I20 [16]
+3H	Input Group 3							
	I37 [31]	I36 [30]	I35 [29]	I34 [28]	I33 [27]	I32 [26]	I31 [25]	I30 [24]
+4H	Input Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Input Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Input Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Input Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]

+AH	Bit Data							
	0	0	0	0	0	0	0	BDT

Ixx represents an input signal; Oxx represents an output signal that will be read back here; Numbers in brackets [ ] represent input bit numbers; BDT is input data.

Figure 4.5. Input Port " I/O Address +0H ~ 7H " and " I/O Address +AH "

The following figure shows bit numbers and their corresponding output ports.

Output Ports I/O Address	D7	D6	D5	D4	D3	D2	D1	D0
+0H	Output Group 0							
	Not Used							
+1H	Output Group 1							
	Not Used							
+2H	Output Group 2							
	Not Used							
+3H	Output Group 3							
	Not Used							
+4H	Output Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Output Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Output Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Output Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]
+AH	Bit Data							
	0	0	0	0	0	0	0	BDT
+BH	Bit Select							
	0	0	BS5	BS4	BS3	BS2	BS1	BS0

Oxx represents an output signal; Numbers in brackets [ ] represent output bit numbers; BDT is output data; BS0 to BS5 specify an input or output bit number.

Figure 4.6. Output Port " I/O Address +0H ~ 7H "and " I/O Address +AH ~ BH "

Input a Bit Data

(1) Select the bit number that you are going to input by outputting this bit number to an I/O address + BH output port, the Bit Select port. The bit numbers are from 0 to 63 (3FH).

(2) Input this bit data from I/O address + AH input port, the Bit Data port. The BDT indicates the status of the specified bit signal.

Programming examples

The following programming examples input the bit [31] (1FH). The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

```
OUT PORT% + &H0B, &H1F
BDT% = INP(PORT% + &H0A)
```

Microsoft C or C++ (MS-DOS version)

```
outp(port + 0x0b, 0x1f) ;
bit_data = inp(port + 0x0a) ;
```

Output a Bit Data

(1) Select the bit number that you are going to output by outputting this bit number to an I/O address + BH output port, the Bit Select port. The bit numbers are from 32 (20H) to 63 (3FH).

(2) Output this bit data to an I/O address + AH output port, the Bit Data port. The BDT is the output data of the specified bit signal.

Programming examples

The following programming examples output "1" to bit [63] (3FH). The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

```
OUT PORT% + &H0B, &H3F
OUT PORT% + &H0A, &H01
```

Microsoft C or C++ (MS-DOS version)

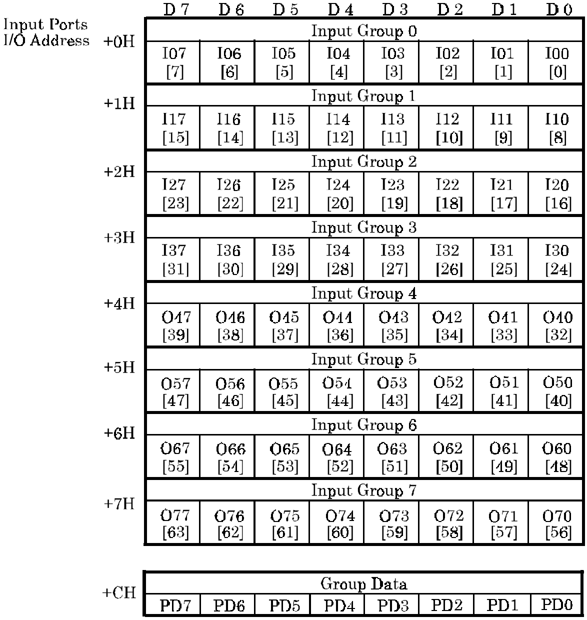
```
outp(port + 0x0b, 0x3f) ;
outp(port + 0x0a, 0x01) ;
```

Note!

After the PC is turned ON, all output bits are reset to "0".

Input/Output Data by Group Number

PIO-32/32T(PCI) supports group input/output function. Under this function, you can specify a group number instead of the I/O port and then input from this port or output to this port. The following figure shows group numbers and their corresponding input ports.



Input Ports I/O Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
+0H	Input Group 0							
	I07 [7]	I06 [6]	I05 [5]	I04 [4]	I03 [3]	I02 [2]	I01 [1]	I00 [0]
+1H	Input Group 1							
	I17 [15]	I16 [14]	I15 [13]	I14 [12]	I13 [11]	I12 [10]	I11 [9]	I10 [8]
+2H	Input Group 2							
	I27 [23]	I26 [22]	I25 [21]	I24 [20]	I23 [19]	I22 [18]	I21 [17]	I20 [16]
+3H	Input Group 3							
	I37 [31]	I36 [30]	I35 [29]	I34 [28]	I33 [27]	I32 [26]	I31 [25]	I30 [24]
+4H	Input Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Input Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Input Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Input Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]
+CH	Group Data							
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

Ixx represents an input signal; Oxx represents an output signal that will be read back here.

Figure 4.7. Input Port " I/O Address +0H ~ 7H " and " I/O Address + CH "

The following figure shows group numbers and their corresponding output ports.

Output Ports I/O Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
+0H	Output Group 0							
	Not Used							
+1H	Output Group 1							
	Not Used							
+2H	Output Group 2							
	Not Used							
+3H	Output Group 3							
	Not Used							
+4H	Output Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Output Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Output Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Output Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]

+CH	Group Data							
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
+DH	Group Select							
	0	0	0	0	0	PS2	PS1	PS0

Oxx represents an output signal; PS0 to PS2 specify an input or output group number.

Figure 4.8. Output Port " I/O Address +0H ~ 7H " and " I/O Address +CH ~ DH "

Input a Group Data

(1) Select the group number that you are going to input by outputting this group number to an I/O address + DH output port, the Group Select port. The group numbers are from 0 to 7.

(2) Input this group data from an I/O address + CH input port, the Group Data port. The PD0 to PD7 represent the status of the specified group signals.

Programming examples

The following programming examples input data from input group 0. The "PORT%" and "port" are variables of I/O addresses.

BASIC (MS-DOS version)

```
OUT PORT% + &H0D, &H00
GDT% = INP(PORT% + &H0C)
```

Microsoft C or C++ (MS-DOS version)

```
outp(port + 0x0d, 0x00) ;
group_data = inp(port + 0x0c) ;
```

Output a Group Data

(1) Select the group number that you are going to output by outputting this group number to an I/O address + DH output port, the Group Select port. The group numbers are from 4 to 7.

(2) Output this group data to an I/O address + CH output port, the Group Data port. The PD0 to PD7 are output data of the specified group, which represent the related output channels.

Programming examples

The following programming examples output "FFH" to group 4. The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

```
OUT PORT% + &H0D, &H04
OUT PORT% + &H0C, &HFF
```

Microsoft C or C++ (MS-DOS version)

```
outp(port + 0x0d, 0x04) ;
outp(port + 0x0c, 0xff) ;
```

Note!

After the PC is turned ON, all output bits are reset to "0".

Digital Filter Function of Input Channels

PIO-32/32T(PCI) is equipped with a digital filter function for input signals. This function provides all input channels with a means of cutting chattering and noise.

Digital Filter Function Principle

To use this function, you must set a filter time. The function circuits will then check all input signals synchronizing with the PCI bus clock. A signal level, low or high, will be input only if the signal level remains stable during the filter set time. Therefore, if a signal level changes faster than the filter set time, this level change will be ignored and the previous signal level will be input.

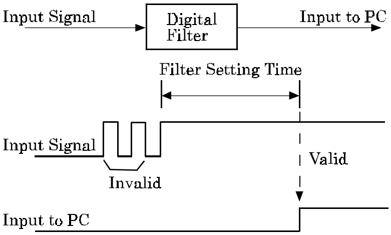


Figure 4.9. Digital Filter Operation Principle

		D7	D6	D5	D4	D3	D2	D1	D0
Input Ports I/O Address	+FH	Digital Filter Setting							
		x	x	x	ST4	ST3	ST2	ST1	ST0

		D7	D6	D5	D4	D3	D2	D1	D0
Output Ports I/O Address	+FH	Digital Filter Setting							
		0	0	0	ST4	ST3	ST2	ST1	ST0

Figure 4.10. I/O Ports of Digital Filter Function

Set Digital Filter Time

To set the digital filter time, write a five-digit setting data to the "I/O address + 0FH" output port, the Digital Filter Setting port. The setting data should be: 0 to 20(14H).

Writing a "0" to this port will disable the digital filter function.

Note!

□ *After power on, this port is reset to "0".*

□ *Writing a setting data other than a "0" will apply this filter function to all input channels. This function cannot be applied to particular pins only but is applied to all input channels.*

□ *Do not set a data more than 20(14H). Doing so may cause the board malfunctioning.*

The following table shows digital filter times and their corresponding setting data.

Digital Filter Time[sec.] = 2^n / (16 x 10^6)

n: = setting data(0 to 20)

Digital filter time	n	Digital filter time	n
The filter function is not used.	0	64ƒÊsec	10
		128ƒÊsec	11
0.125ƒÊsec	1	256ƒÊsec	12
0.25ƒÊsec	2	512ƒÊsec	13
0.5ƒÊsec	3	1.024msec	14
1ƒÊsec	4	2.048msec	15
2ƒÊsec	5	4.096msec	16
4ƒÊsec	6	8.192msec	17
8ƒÊsec	7	16.384msec	18
16ƒÊsec	8	32.768msec	19
32ƒÊsec	9	65.536msec	20

Figure 4.11. Digital Filter Time and Setting Data

For example, to set a 1 msec filter time, the setting data should be 14(0EH) because 0.001 = (2^{14} / 16000000)

Programming examples

The following programming examples set the filter time for 1 msec. The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

```
OUT PORT% + &H0F, &H0E.
```

Microsoft C or C++ (MS-DOS version)

```
outp(port + 0x0f, 0x0e) ;
```

Confirm the Digital Filter Setting Data

Read the "I/O address + 0FH" input port, the Digital Filter Setting port, will get the filter setting data that you have set.

The following programming examples input the filter setting data. The "PORT%" and "port" are sample I/O addresses.

Programming examples

BASIC (MS-DOS version)

```
TIM% = INP(PORT% + &H0F)
```

Microsoft C or C++ (MS-DOS version)

```
set_time = inp(port + 0x0f) ;
```

Output Data Monitor Function

This function allows you to read the last output data without affecting that data.

Input Ports I/O Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
+4H	Input Group 4							
	O47 [39]	O46 [38]	O45 [37]	O44 [36]	O43 [35]	O42 [34]	O41 [33]	O40 [32]
+5H	Input Group 5							
	O57 [47]	O56 [46]	O55 [45]	O54 [44]	O53 [43]	O52 [42]	O51 [41]	O50 [40]
+6H	Input Group 6							
	O67 [55]	O66 [54]	O65 [53]	O64 [52]	O63 [51]	O62 [50]	O61 [49]	O60 [48]
+7H	Input Group 7							
	O77 [63]	O76 [62]	O75 [61]	O74 [60]	O73 [59]	O72 [58]	O71 [57]	O70 [56]

Figure 4.12. Input Port " I/O Address + 4H ~ 7H "

Input from an I/O Port

To monitor output data, you can read it as it is being output from the output port.

Programming examples

The following programming examples output data AAH to I/O address + 4H output port and then read it from I/O address + 4H input port to confirm the output data. The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

```
    OUT PORT% + &H04, &HAA
    MDT% = INP(PORT% + &H04)
```

Microsoft C or C++ (MS-DOS version)

```
    outp(port + 0x04, 0xaa) ;
    m_data = inp(port + 0x04) ;
```

Confirm an Output Bit

Refer to the section "Input/Output Data by Bit Number."

Monitor Output Data by Port Number

Refer to the section "Input/Output Data by Group Number."

Handshake of Input Channel 0 to 15

Under this function, an external STB signal will latch channel 0 to 15 input data into input registers. This STB signal can also issue an interrupt request signal if you set the interrupt event for it. An ACK signal can be issued through software commands. By using these STB and ACK signals, you can then handshake the input actions with external circuits.

Note!

Outputting a hexadecimal data F0H to I/O address + 8H port will reset (initial) the handshake function.

Input Ports I/O Address		D7	D6	D5	D4	D3	D2	D1	D0
	+8H	Handshake Status Input							
		x	x	x	x	INT	IBF	STB	ACK

Output Ports I/O Address		D7	D6	D5	D4	D3	D2	D1	D0
	+8H	ACK Signal Setting							
		0	0	0	0	0	0	0	ACK

Figure 4.13. I/O Port Assignment of Handshake Function

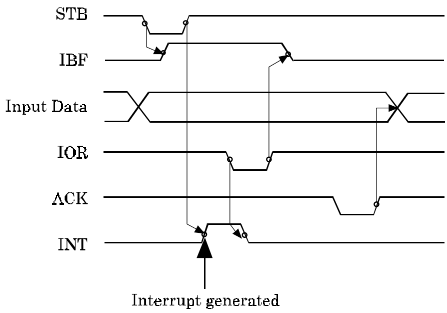


Figure 4.14. Timing of Handshake Function

Programming examples

The following programming examples input a latched data by checking the INT status of the handshake status register. The "PORT%" and "port" are sample I/O addresses.

BASIC (MS-DOS version)

```
OUT PORT% + &H08, &HF0
                      'INITIALIZE HANDSHAKEFUNCTION
WHILE((INP(PORT% + &H08) AND &H08) = 0)
                         'INPUT STATUSREGISTER
WEND                     'CHECK INT STATUS
DAT% = INP(PORT%)        'INPUT DATA
OUT PORT% + &H08, 0      'OUTPUT ACK SIGNAL
OUT PORT% + &H08, 1
```

Microsoft C or C++ (MS-DOS version)

```
#include<stdio.h >

void main (void)
{
  int dat;

  outp(port + 0x08, 0xf0) ;
                   /* initialize handshake function */
  while(!(inp(port + 0x08) & 0x08)) ;
                             /* Cheching INT status */
  data = inp(port + 0x00);             /* input data */
  outp(port + 0x08, 0);        /* output ACK signal */
  outp(port + 0x08, 1);
}
```

Interrupt Control Function

If the No.1-2 pins of JP1 is jumped as "Using Interrupt", up to 4 interrupt events, either I00 to I03 input signals or three of the four input signals and the handshake signal (STB signal), can generate an interrupt request signal.

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
Input Ports	Interrupt Setting							
I/O Address +EH	x	x	IN1	IN0	x	IR	IS1	IS0

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
	Interrupt Status							
I/O Address +11H	0	0	0	0	INTS3	INTS2	INTS1	INTS0

Figure 4.15. Input Port Assignment of Interrupt Function

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
Output Ports	Interrupt Setting							
I/O Address +EH	0	0	IN1	IN0	0	IR	IS1	IS0

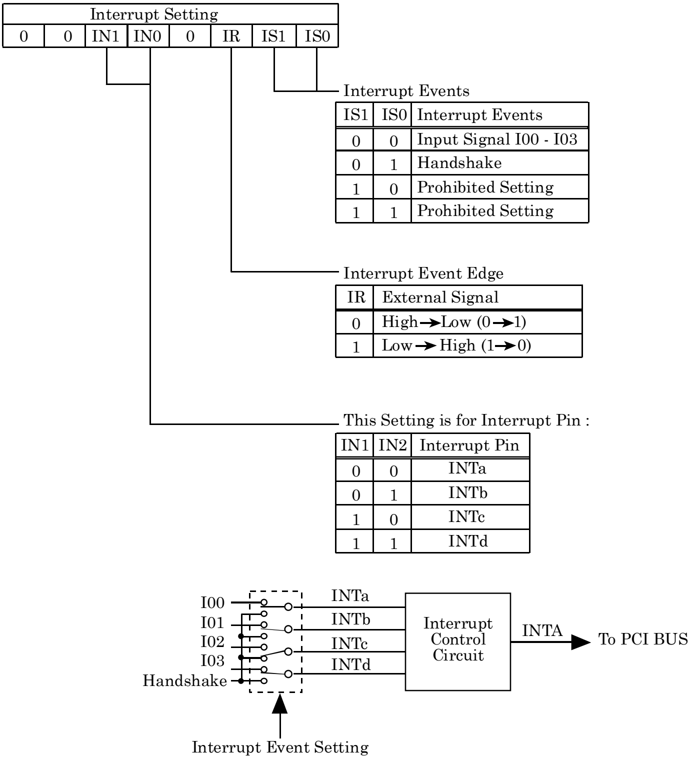
	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
	Interrupt Mask Resister							
I/O Address +10H	0	0	0	0	INTM3	INTM2	INTM1	INTM0

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
	Interrupt Clear Resister							
I/O Address +11H	0	0	0	0	INTC3	INTC2	INTC1	INTC0

Figure 4.16. Output Port Assignment of Interrupt Function

Set Interrupt Events

You use the Interrupt Setting output port to set interrupt events and the active logic of these events.
When you set the interrupt events,
I00 input signal can be connected only to internal signal INTa;
I01 input signal can be connected only to internal signal INTb;
I02 input signal can be connected only to internal signal INTc;
I03 input signal can be connected to internal signal INTd only. The handshake event, however, can be connected to any of the internal signals INTa to INTd.



Interrupt Setting

0	0	IN1	IN0	0	IR	IS1	IS0

Interrupt Events

IS1	IS0	Interrupt Events
0	0	Input Signal I00 - I03
0	1	Handshake
1	0	Prohibited Setting
1	1	Prohibited Setting

Interrupt Event Edge

IR	External Signal
0	High→Low (0→1)
1	Low→High (1→0)

This Setting is for Interrupt Pin :

IN1	IN2	Interrupt Pin
0	0	INTa
0	1	INTb
1	0	INTc
1	1	INTd

Figure 4.17. Explanation of Interrupt Setting Register

Interrupt Event Masking

PIO-32/32T(PCI) uses I/O address + 10H output port for masking. Setting a bit to "1" will mask the related event from generating an unwanted interrupt request. For example, if you set the I01 input signal as an interrupt event and you set INTM1 to "1" from I/O address + 10H, then I01 input signal will not generate an interrupt request, but it will still set the INTS1 of interrupt status register. The relationship of interrupt events "INTa to INTd" and interrupt masking bits is as following:

INTa : INTM0
INTb : INTM1
INTc : INTM2
INTd : INTM3

Note!

After power on, this + 10H output port is reset as all "1".

Interrupt Status Register

PIO-32/32T(PCI) connects only one interrupt request signal to the PCI bus, the INTA bus signal. The board combines these four interrupt requests into one signal. Before setting more than two interrupt events for generating interrupt requests in your interrupt handler program, you must first determine which event has requested this interrupt service. The Interrupt Status input port is designed for this purpose. For example, if you have set the I00 to I03 as interrupt events and the active logic is from low to high, a change of I00 from low to high will set the interrupt status INTS0 to "1" and generate an interrupt request. Interrupt events and their corresponding interrupt status are shown below:

INTa : INTS0
INTb : INTS1
INTc : INTS2
INTd : INTS3

Check Interrupt Setting

You can check what you have set for the interrupt setting register by reading the I/O address + 0EH input port.

Clear Interrupt Request Signal

The interrupt of the PCI bus uses a level trigger instead of an edge trigger, which is used by the ISA bus. Therefore, the arrival of an interrupt request means that the INTA has been assigned to low. If you do not clear this request signal, the CPU will respond to this interrupt request repeatedly. To avoid this kind of malfunctioning in your interrupt handler program, you must clear the interrupt request signal before you enable the interrupt.
For PIO-32/32T(PCI), you can use the Interrupt Clear output port for this purpose. For example, if the INTS2 is "1" in the Interrupt Status register of your interrupt handler program, INTc will generate an interrupt request. Setting the INTC2 to "1" by output to the Interrupt Clear port will clear this INTS2 interrupt request to enable the next interrupt.

INTa	INTC0
INTb	INTC1
INTc	INTC2
INTd	INTC3

Programming example

The following sample program is part of a sample interrupt handler program. The "port" is sample I/O address.

Microsoft C or C++ (MS-DOS version)

```
_disable() ;
n = inp(port + 0x11) & 0x0f ;
                /*Find out which event(s)
                     has/have requested interrupt.*/
outp(port + 0x11, n)   ;
                /*Clear interrupt request signal.*/
      .
      .         /* Service the interrupt request.*/
      .
_enable() ;
```

5. System Reference

Block Diagram

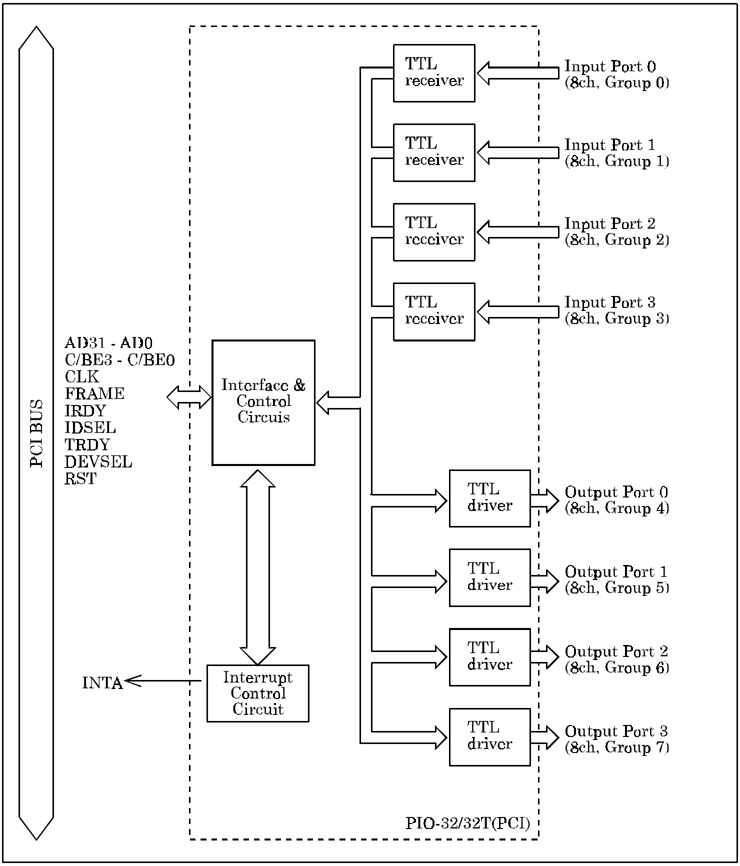

PCI BUS
AD31 - AD0
C/BE3 - C/BE0
CLK
FRAME
IRDY
IDSEL
TRDY
DEVSEL
RST
Interface & Control Circuis
INTA
Interrupt Control Circuit
TTL receiver
Input Port 0 (8ch, Group 0)
TTL receiver
Input Port 1 (8ch, Group 1)
TTL receiver
Input Port 2 (8ch, Group 2)
TTL receiver
Input Port 3 (8ch, Group 3)
TTL driver
Output Port 0 (8ch, Group 4)
TTL driver
Output Port 1 (8ch, Group 5)
TTL driver
Output Port 2 (8ch, Group 6)
TTL driver
Output Port 3 (8ch, Group 7)
PIO-32/32T(PCI)

Figure 5.1. Block Diagram

Specifications

Table 5.1. lists the chief specifications of this board.

Table 5.1. Specifications

Item		Specification
Input	Type	TTL level input (Negative logic)
	Resistor	10kƒ ¶(1TTL load)
	Number of channels	32 channels (4 of these 32 can be used as interrupt signal)
	Interrupt	Combine four interrupt signals to one interrupt request signal as the INTA. Either rising edge or falling edge of input signal can generate interrupt.
	Response time	200ns or less
Output	Type	Open collector output (Negative logic)
	Voltage	30VDC (Max.)
	Current	40mA max. per channel
	Number of channels	32 channels
	Response time	200ns or less (Typical, depends on pull up resistor)
I/O Address		Any 32-byte boundary
Boards in one system		Maximum of 16 boards can be install in a same system.
Power consumption		5VDC 500mA (Max.)
Operating condition		0 to 50 Ž20% to 90% (not condensing)
Connecting distance		1.5m (Typical, depends on wiring environment)
Dimension		120.0 × 107.0 × 18.5mm
Weight		100g

6. Troubleshooting

If you are having trouble with your board or program, first answer the following questions to see if you can find the problem.

QUESTIONS:

STEPS TO TAKE:

Is the pilot light on?

The system must be turned on. Check the main power switch.

Did the system boot up?

If the board is installed and the system did not boot up, check the following:

□ Make sure that the board is plugged in firmly.

□ Can you boot up if you chose not to use the interrupt?

Does your program work?

If the system booted and your program does not work, check the following:

□ Try one of the programs provided on disk and see if it works.

□ Check the board address of your program.

□ Check your interrupt handler program.

Does the sample program work?

Check the following:

□ Try it with only the PIO-32/32T(PCI) board installed.

□ Make sure that input signals are connected to the right pins.

□ Make sure that output signals are recalled back correctly.

□ Pin connections on the I/O cables are secure.

Still have a problem?

Contact CONTEC group office where you purchased as shown in the last page of this manual.

Note!

Before you call, please make a list of the following information. Our technical representatives will need the following information to help you.

1. Your name, company, and phone number.

2. The brand and type of computer you are using

3. OS and Version (e.g. DOS Ver.6.2)

4. Name of the CONTEC board that you are using.

5. Names of other boards in the computer.

6. The programming language that you are using (and the version number).

7. Are you using your own program or a CONTEC sample program?

8. List AUTOEXEC.BAT.

9. List CONFIG.SYS.

7. Index