

PC-32

User's Manual

Low Cost, High Accuracy Analog Input and Output Boards
for IBM PC, PC/XT, PC/AT, PS/2 Computer Systems,
and Compatibles, 386, 486, ISA and EISA.

All right reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, by photocopying, recording, or otherwise without prior written permission.

First edition.

October 1987

October 1987 Printing

Information furnished in this manual is believed to be accurate and reliable; however no responsibility is assumed for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

IBM, IBM PC/XT/AT and IBM PS/2 are trademarks of International Business Machine Corporation.

BASIC is a trademark of Dartmouth College.

Microsoft is a trademark of Microsoft Corporation.

Table of Contents

Preface		i
Chapter 1 Introduction		1-1
Overview	1-1	
Features	1-2	
Software support	1-3	
Chapter 2 Architecture		2-1
D/A Subsystem	2-1	
A/D Subsystem	2-1	
Bus interface.	2-2	
Timing and control	2-3	
Chapter 3 Configuring the PC-32 card		3-1
Introduction	3-1	
Changing the jumper settings	3-2	
Bus Interface jumper settings.	3-2	
Clock Control.	3-4	
D/A jumper selections.	3-4	

A/D configuration.	3-5	
Chapter 4 Interconnections		4-1
Introduction	4-1	
Connections to the IBM backplane.	4-1	
User connection.	4-1	
Connection guidelines.	4-3	
Chapter 5 Programming the PC-32 card		5-1
Introduction	5-1	
Register structure.	5-1	
Chapter 6 USING THE PC-32 DRIVER SOFTWARE		6-1
Introduction.	6-1	
Language interface.	6-2	
Calling the PC-32 driver.	6-6	
Functions : Detailed Descriptions.	6-10	
Chapter 7 Calibration		7-1
Introduction	7-1	
A/D calibration.	7-1	
D/A calibration.	7-3	
Appendix A Hardware Specifications		A-1
Inputs	A-1	
Output	A-2	
Card Timing	A-2	
IBM PC Interface	A-2	

P r e f a c e

This manual is written for users of the PC-32¹ series of analog I/O cards. It provides all information necessary to successfully program and operate all cards in the series, as well as all information required to use the supplied driver software in conjunction with several languages.

This manual assumes :

- That you have a basic knowledge of electronic circuitry and measurement techniques.
- That you are familiar with the host PC which you are using.
- That you are capable of writing your own programs.

The manual contains the following sections.

Chapter 1 - Introduction.

- Chapter 1 contains an overview of the PC-32 series of cards, their capabilities and the capabilities of the supplied software.

Chapter 2 - Architecture.

- Chapter 2 discusses the basic operation of the PC-32 card.

Chapter 3 - Configuration.

- Chapter 3 discusses the selection of various board parameters and the configuration of the board for various operating requirements.

Chapter 4 - Interconnection.

- Chapter 4 describes the connection of the PC-32 series of boards to the host computer and to user inputs.

Chapter 5 - Programming the PC-32.

- Chapter 5 describes the register structure of the PC-32 series of cards and provides the information requires to program these registers.

Chapter 6 - Using the PC-32 driver software.

- Chapter 6 describes the use of the supplied software drivers.

1 For the rest of this manual, "PC-32" will be used to refer to all of the cards in the series. Where information is specific to a particular card, this will be stated in the text.

Chapter 7 - Calibration.

- Chapter 7 describes the procedures and equipment required to calibrate the PC-32 series of boards. Calibration software included with the PC-32 is also described.

Appendix A - Hardware Specifications.

- Appendix A provides complete electrical specifications for the PC-32 series of boards.

Appendix B - Software Specifications.

- Appendix B contains complete technical specification of the PC-32 driver software.

Appendix C - Commonly asked questions.

- Appendix C contains some commonly asked questions and their answers.

Chapter 1

Introduction

1.1. Overview

The PC-32 series of boards are full size, low cost high accuracy analog I/O boards for the IBM PC/XT/AT and compatible series of computers. The family consists of thirteen cards :

- PC-32. Contains an A/D converter with bipolar inputs and a D/A converter. The A/D gain is resistor programmable.
- PC-32/PGH Contains an A/D converter with bipolar inputs and a D/A converter. The A/D gain is software programmable to 1, 2, 4 or 8.
- PC-32/PGL. Contains an A/D converter with bipolar inputs and a D/A converter. The A/D gain is software programmable to 1, 10, 100 or 500.
- PC-32/U. Contains an A/D converter with unipolar inputs and a D/A converter. The A/D gain is resistor programmable.
- PC-32/U/PGH Contains an A/D converter with unipolar inputs and a D/A converter. The A/D gain is software programmable to 1, 2, 4 or 8.
- PC-32/U/PGL. Contains an A/D converter with unipolar inputs and a D/A converter. The A/D gain is software programmable to 1, 10, 100 or 500.
- PC-33. Contains only an A/D converter with bipolar inputs. The A/D gain is resistor programmable.
- PC-33/PGH Contains only an A/D converter with bipolar inputs. The A/D gain is software programmable to 1, 2, 4 or 8.
- PC-33/PGL. Contains only an A/D converter with bipolar inputs. The A/D gain is software programmable to 1, 10, 100 or 500.

- PC-33/U. Contains only an A/D converter with unipolar inputs. The A/D gain is resistor programmable.
- PC-33/U/PGH. Contains only an A/D converter with unipolar inputs. The A/D gain is software programmable to 1, 2, 4 or 8.
- PC-33/U/PGL. Contains only an A/D converter with unipolar inputs. The A/D gain is programmable to 1, 10, 100 or 500.
- PC-34. Contains only a D/A converter.

1.2. Features

These cards can be plugged into any of the fully bussed slots in the PC/XT/AT computer backplane.

1.2.1. A/D subsystem

The A/D subsystem's major component is a shielded module. This contains a multiplexer, high accuracy instrumentation amplifier and A/D converter. This module acquires either single ended or differential inputs and converts them into 16 bit digital codes.

The A/D gain may be either resistor programmable in the range of 1 to 2000, be set by software to 1, 2, 4 or 8 for the /PGH cards or be set by software to 1, 10, 100 or 500 for the /PGL cards.

The A/D may be operated in either single conversion or continuous conversion mode. In single conversion mode the card performs a single conversion on the selected input channel and stops on completion of this conversion. In continuous conversion mode conversions are performed at a set rate on a single channel, or on all channels sequentially.

A/D conversions may be monitored by either polled I/O or by interrupts. In polled I/O mode the software continuously polls the card's status register to check for completion of the current A/D conversion. In interrupt mode, the card automatically generates a hardware interrupt on completion of each conversion.

Key specifications

- A/D resolution : 16 Bits
- A/D linearity : 0.006%
- A/D full scale input ranges : Card may be ordered for unipolar (full scale range from +5mV to +10V) or bipolar (+-5mV to +-10V range).
- Number of A/D inputs : Switch selectable 16 single ended or 8 differential.
- A/D throughput rate : 2.5KHz for a gain of 1.

1.2.2. D/A Subsystem

The D/A subsystem's major component is a shielded module. This contains a high accuracy current mode D/A together with summing amplifier for voltage output operation. In addition to this the card contains a buffer register, which is used to hold data for the D/A until required.

The D/A may be operated in either single conversion or continuous conversion mode. In single conversion mode the card converts a digital word to an analog output, and stops on completion of this conversion. In continuous conversion mode conversions are performed continuously at a set rate.

Data for D/A conversions may be supplied either by polled I/O or by interrupts. In polled I/O mode the software continuously polls the card's status register to check whether the D/A buffer register is empty. In interrupt mode, the card automatically generates a hardware interrupt when the D/A buffer is empty.

Key specifications

- D/A resolution : 16 Bits
- D/A linearity : 0.002%
- Full scale output ranges : 0 to +2mA, -1 to +1 mA, 0 to -10V, -5V to +5V or -10V to +10V.
- D/A throughput rate : 10 KHz.

1.2.3. Interfacellogic

The PC-32 is accessed via I/O operations performed by the host processor. Of the 10 bit address received by the card, the most significant 6 bits select the card, and the least significant 4 bits select the register to be accessed.

The PC-32 occupies 16 byte locations. The base address of the card can be selected to be located anywhere between 200 (hex) and 3F8 (hex).

The PC-32 operates from the +5V line of the PC bus.

1.3. Software support

Supplied with the PC-32 card is a set of real time device drivers for use with a wide variety of software. These device drivers are written in assembly language, and are callable from the following languages :

- Microsoft C.
- MicrosoftFORTRAN.
- Microsoft PASCAL.
- IBM BASICA.
- Turbo C.
- Turbo Pascal.

All of these drivers support all modes of operation of the PC-32.

Chapter 2

Architecture

This chapter describes the architecture of the PC-32 series of cards. The block diagram in figure 2.1. highlights the major elements contained on the card, and their interrelationship. There are four major subsections. These are the following :

2.1. D/A Subsystem

The D/A subsystem contains a D/A converter and a buffer register. When a D/A strobe occurs, the data in the buffer register is transferred to the D/A, and hence to the analog output.

The D/A strobe is generated by the timing and control section, and also sets a D/A buffer empty bit in the board's status register.

Data may be transferred to the D/A buffer either by polled I/O or interrupts.

Note. Data in the buffer register cannot be transferred to the D/A by a write to the buffer register.

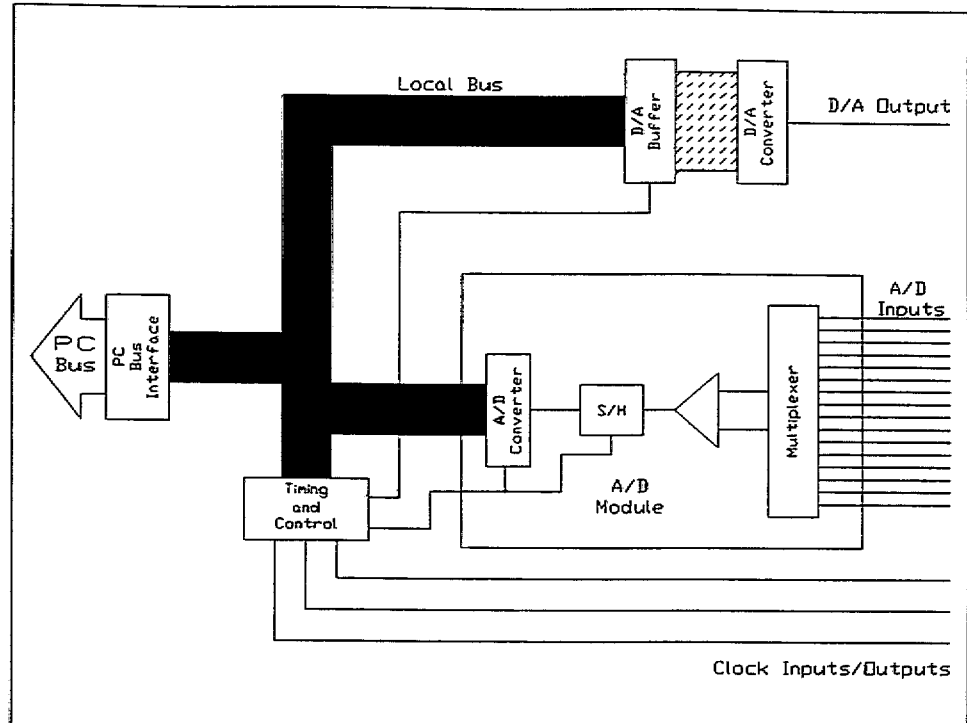
2.2. A/D Subsystem

The A/D subsystem contains several separate components :

- The input multiplexer. The multiplexer selects either one of eight differential input channels or one of sixteen single ended input channels. This channel is selected by a channel address, which is written to a channel address register. The board can be programmed to either increment this channel address on each successive A/D conversion, or to keep the channel address the same.

- The instrumentation amplifier. The instrumentation amplifier amplifies either the difference between two differential input, or the difference between a single ended input and ground. The degree of amplification can be selected either by a resistor or under software control (/PGH

Figure 2.1.
PC-32 Block
diagram.



and /PGL cards only).

- The sample and hold unit. The sample and hold unit holds the output of the instrumentation amplifier steady for the duration of the A/D converter's conversion process.
- The A/D converter performs the actual A/D conversion. An A/D conversion is begun by a A/D strobe. This is generated by the timing and control section.

Data may be transferred from A/D either by polled I/O or interrupts.

2.3. Bus interface.

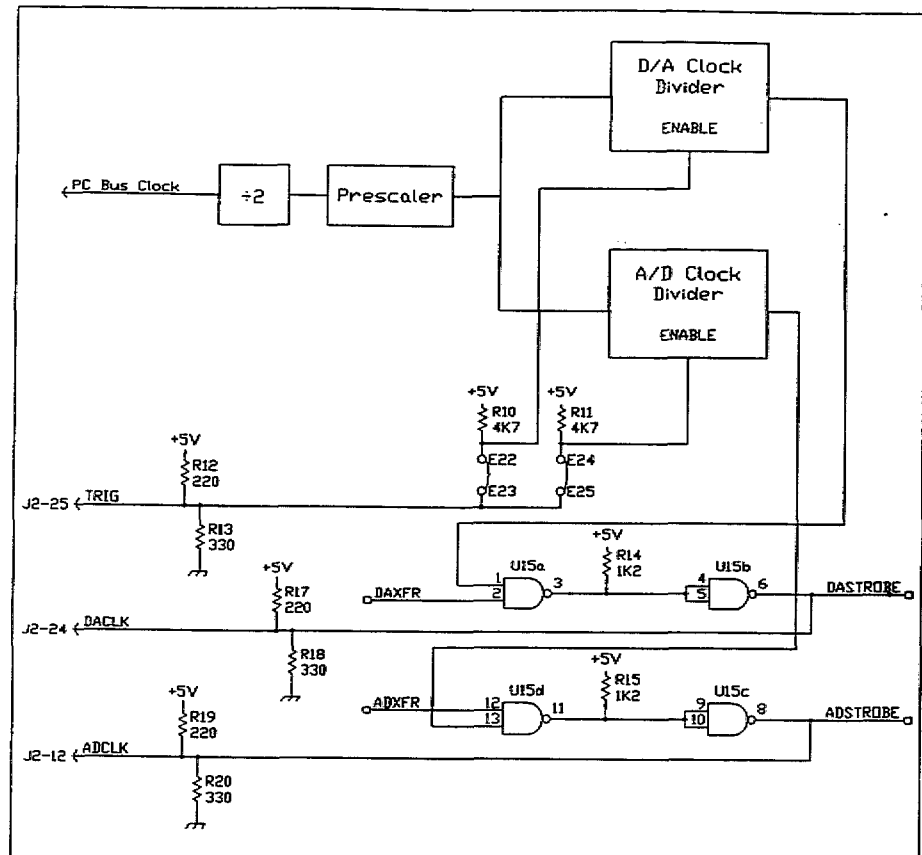
The bus interface is responsible for two functions :

- i. The decoding of the card's base address. The card's base address is set by jumpers.
- ii. The generation of interrupts. Interrupts can be generated under two conditions :
 - ii.i A/D interrupts are enabled and an A/D conversion cycle completes.
 - ii.ii D/A interrupts are enabled and the D/A buffer becomes empty.

2.4. Timing and control

The timing and control subsection is responsible for the generation of A/D and D/A strobes. A block diagram of this section is shown in figure 2.2.

Figure 2.2.
PC-32 Clock
system.



2.4.1. A/D and D/A clocks

The A/D and D/A clock generators are 16-bit counters capable of generating strobes to the A/D and D/A respectively. They may be programmed for any division ratio from 2 to 65536. The input to both generators comes from a prescaler which is driven from the PC Bus clock divided by two. This prescaler can also be programmed for a division ratio from 2 to 65536.

Either or both of the clock generators may be disabled by a low level on the external TRIG input. This allows the user to start and stop conversions under the control of external circuitry, but still retain the use of the clock generators.

2.4.2. Strobegeneration

A/D and D/A strobes may be generated in three ways:

- i. By accessing a particular register. This allows transfers from the D/A buffer register to the D/A or A/D conversions to be triggered under program control. Accessing this register results in a negative going pulse on the ADXFR or the DAXFR line respectively.

ii. From the A/D and D/A clock generator described above.

iii. From a an external input. Note that the ADCLK and DACLK lines are bidirectional, serving as either an input or an output. When strobes are generated by either of the first two methods, a negative going pulse will appear on these lines. However if the clock generators are disabled and the transfer registers not accessed, then the user may drive this line from a open-collectoroutput.

Chapter 3

Configuring the PC-32 card

3.1. Introduction

The PC-32 card can be configured in many different ways to suit each user's individual requirements. This configuration is set by the position of the various mini-jumps on the card. Four sets of mini-jumps are used, each set controlling a specific aspect of the operation of the card. These are as follows :

- i. Bus interface. Both the base address of the card and the interrupt level used by the card can be set. As supplied by the factory, the base address is set to 300H, and the interrupt level to 3. This allows operation in a standard PC/XT/AT which contains only conventional cards (Multifunction cards, disk controller cards, display cards etc), but may require modification if exotic cards (other scientific cards, certain backup systems etc) are installed. Note that the interrupt line is electronically disconnected from the PC bus unless specifically enabled by software, even if a interrupt level selection jumper is installed.
- ii. Clock control. The trigger line on the external interface connector of the PC-32 card may be used to gate or enable either the internal A/D clock or the internal D/A clock or both. This is selected by the position of two mini-jumps. As supplied by the factory, the trigger line enables both the A/D and the D/A clocks.
- iii. D/A operation. The output range and type of the D/A may be selected by mini-jumps. As supplied by the factory, the D/A output is set for voltage output, and the range from -10 to +10 Volts.
- iv. A/D operation. Two aspects of A/D operation may be configured. These are as follows :

iv.i The gain of the A/D may be set, by installation of two resistors and a capacitor.

iv.ii The A/D may be set for either differential or single ended operation, and the user installed gain setting components may be disconnected from the circuit, for calibration or diagnostics purposes. As supplied by the factory, the A/D is set for differential mode operation, and the user installed components are disconnected.

The above set of factory installed jumpers are designed to suit most applications, and are required for correct operation of the demonstration, diagnostics and calibration software supplied with the card.

3.2. Changing the jumper settings

In order to change the jumper settings follow the procedure below :

- i. Switch off the computer.
- ii. Remove the card.
- iii. Change the required jumpers.
- iv. Some A/D options require the DIP switches on the base of the A/D module to be changed. If this is the case, proceed as follows:
 - iv.i Remove the A/D module (the large silver module closest to the PC bus connector), by pulling gently in a direction parallel to the board. Ensure that the module comes out of the connector evenly (both sides should come out together).
 - iv.ii Change the required switches.
- v. Replace the module gently.
- vi. Replace the card in the PC.
- vii. Power up, and run a program (such as DEMO1), which executes the PC-32/33/34 diagnostics routines.

3.3. Bus Interface jumper settings.

3.3.1. Baseaddress

The base address setting is controlled by five jumpers, which short out the jumper locations numbered from 12 to 21. As supplied by the factory, the address is set to 300H, the address used by the IBM prototype card. The card occupies 16 consecutive locations. The following table describes which jumper should be installed for a particular base address.

Base location	Jumpers
200H	No jumpers in the block
210H	20-21
220H	18-19
230H	20-21, 18-19
240H	16-17
250H	20-21, 16-17
260H	18-19, 16-17
270H	20-21, 18-19, 16-17
280H	14-15
290H	20-21, 14-15
2A0H	18-19, 14-15
2B0H	20-21, 18-19, 14-15
2C0H	16-17, 14-15
2D0H	20-21, 16-17, 14-15
2E0H	18-19, 16-17, 14-15
2F0H	20-21, 18-19, 16-17, 14-15
300H	12-13
310H	20-21, 12-13
320H	18-19, 12-13
330H	20-21, 18-19, 12-13
340H	16-17, 12-13
350H	20-21, 16-17, 12-13
360H	18-19, 16-17, 12-13
370H	20-21, 18-19, 16-17, 12-13
380H	14-15, 12-13
390H	20-21, 14-15, 12-13
3A0H	18-19, 14-15, 12-13
3B0H	20-21, 18-19, 14-15, 12-13
3C0H	16-17, 14-15, 12-13
3D0H	20-21, 16-17, 14-15, 12-13
3E0H	18-19, 16-17, 14-15, 12-13
3F0H	20-21, 18-19, 16-17, 14-15, 12-13

3.3.2. Interruptlevel

The interrupt setting block consists of jumper locations 26 to 33. The interrupt level may be set to to level 3, 4, 5 or 7.

3.3.2.1. Jumper settings

The interrupts jumper settings are described in the table below :

Interrupt level	Jumper
level 3	26-27
level 4	28-29
level 5	30-31
level 7	32-33

NOTE. Only one of the jumpers in the above block may be installed at any one time.

3.3.2.2. Selection of interrupt level

In a standard PC, interrupts are allocated as follows :

level 3	Used by COM2: (if installed)
level 4	Used by COM1: (if installed)
level 5	Used by fixed disks (XT and AT)
level 7	Used by LPT1: (if installed)

An interrupt level which is not already used must be selected. As level 3 is least often used, this is normally selected. Note that unless the interrupts are specifically enabled by software, the interrupt output from the card is tri-stated (does not have any effect on the PC bus).

3.4. Clock Control.

In the section describing external connections to the card, the trigger input was described. This input gates either the internal D/A clock, the internal A/D clock or both. The jumper settings are as follows :

Jumper	Effect
22-23	D/A internal clock gated.
24-25	A/D internal clock gated.

Note that both jumpers can be installed simultaneously, as can neither jumper.

3.5. D/A jumper selections.

The D/A may be jumpered for several modes of operation. The D/A jumpers are installed in the locations numbered 1 to 11 as follows :

Mode	Jumpers.
Unipolar current output (0 to 20 mA)	2-3, 6-7
Bipolar current output (-1 to +1 mA)	1-2, 6-7

Unipolar Voltage output (0 to -10 V)	2-3, 4-5, 7-8, 9-10
Bipolar Voltage output (-10 to +10 V)	1-2, 4-5, 7-8, 10-
Bipolar Voltage output (-5 to +5 V)	1-2, 4-5, 7-8, 9-10

3.6. A/D configuration.

3.6.1. A/D gainsetting

The gain of the A/D may be set in the range 1 to 2000. This gain is set by the installation of two resistors, R23 and R24, of which one is fixed, and one variable for adjustment purposes. R25 can be used if no adjusting resistor is needed. R23 and R24 are connected in series, and their combined value may be calculated as follows:

$$R = 100000 / (\text{Gain} - 1)$$

where R is in Ohms.

At higher gains, the A/D's instrumentation amplifier requires additional settling time. This is achieved by inserting an additional capacitor, C6, into the PCB. The value of this capacitor is dependant on gain, and may be found from the following table :

Gain Range	Capacitor (uF)	A/D Throughput
1 - 10	None Required.	2.5 KHz
10 - 100	0.022	800Hz
100 - 200	0.039	400Hz
200 - 400	0.082	200Hz
400 - 1000	0.22	80Hz
1000 - 2000	0.47	40Hz

Note that the accuracy of the system is dependant on the gain, as follows :

Gain	System Accuracy
1	0.0075%
2	0.009%
4	0.01%
10	0.01%
100	0.01%
200	0.02%
400	0.04%
1000	0.06%
2000	0.13%

3.6.2. A/D jumper settings

The following aspects of A/D operation may be controlled :

i. Digitally controlled gain. For cards with digital gain control, the following jumpers must be installed. Note that in this case no gain setting components may be inserted. Note also that these jumpers may NOT be installed if the card does not have the programmable gain option. These jumpers are not installed at the factory.

Mode	Jumpers.
Digital gain	34-35, 36-37

ii. User installed gain component activation. For the user installed gain components to be effective, the following jumpers must be installed. Deactivating these components is useful for diagnostics and calibration purposes. These jumpers are not installed at the factory.

Mode	Jumpers.
Components activated	38-39, 40-41

iii. A/D output coding. The A/D output coding may be set to either offset binary or complimentary binary. This done by a switch on the base of the A/D module. This switch is set to offset binary at the factory.

iv. The A/D may be configured for either 16 channel singled ended operation, or for 8 channel differential operation. This done by both a switch setting on the base of the A/D module AND jumper settings. Note that if the jumper setting and the switch setting are not both changed to reflect the mode change, permanent damage may result to the A/D module. The jumper settings corresponding to the two switch settings are given below :

A/D Switch Setting	Jumpers.
Single ended	42-43, 44-45
Differential	42-44

For digital gain mode, jumpers 40-41 must be installed as well as jumpers 34-35 and 36-37. This applies only to PGH and PGL units.

Chapter 4

Interconnections

4.1. Introduction

The PC-32 family of boards plug into IBM PC/XT/AT or compatible expansion slots at connector J1, and connects to the user's circuitry at D-type connector J2. This chapter describes these two connectors.

4.2. Connections to the IBM backplane.

The PC-32 series of boards can be plugged into any slot of the IBM backplane, with the exception of the J8 slot of the XT. This particular slot requires the -CARDSLCT signal, which is not used on other slots. All communication to and from the host processor is carried out via this connector.

4.3. User connection.

The PC-32 is connected to the user interface via a female 25 way D-type connector. This connector accommodates the following signals :

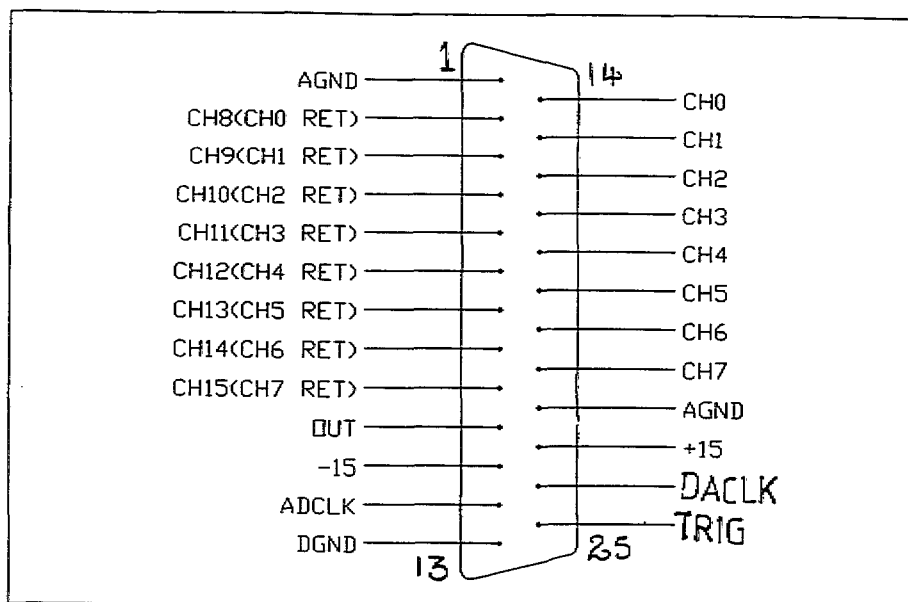
- 16 lines of analog input.
- 1 line of analog output.
- 3 clock lines.
- 5 power supply lines.

Figure 4.1. shows these connections, together with their pin assignments.

4.3.1. Signaldefinitions.

i. CH0 - CH15. These are the analog input lines. Note that in differential mode, CH8 functions as the return line for CH0 etc.

Figure 4.1.
PC-32
Connector.



ii. OUT. This is the analog output line.

iii. AGND. Two analog ground lines are provided. The analog input and output lines are measured relative to AGND.

iv. +15. This line provides a +15 V power supply to the user's interface. Maximum permissible current draw is 20 mA. The return line for this current is AGND.

v. -15. This line provides a -15 V power supply to the user's interface. Maximum permissible current draw is 20 mA. The return line for this current is AGND.

vi. TRIG. This line may be used to enable or disable the board's internal clock sources. The function of this line is dependant on the position of jumpers on the board, as described in chapter 3. The line is terminated in a 220/330 Ohm TTL termination.

vii. DACLK and ADCLK. These lines are the sample clocks to the D/A and A/D respectively. These line are bidirectional, and are terminated in 220/330 Ohm TTL terminations. Both lines may be driven from open-collector TTL drivers. Note that the internal clock source must be disabled by software for predictable results when driving these lines.

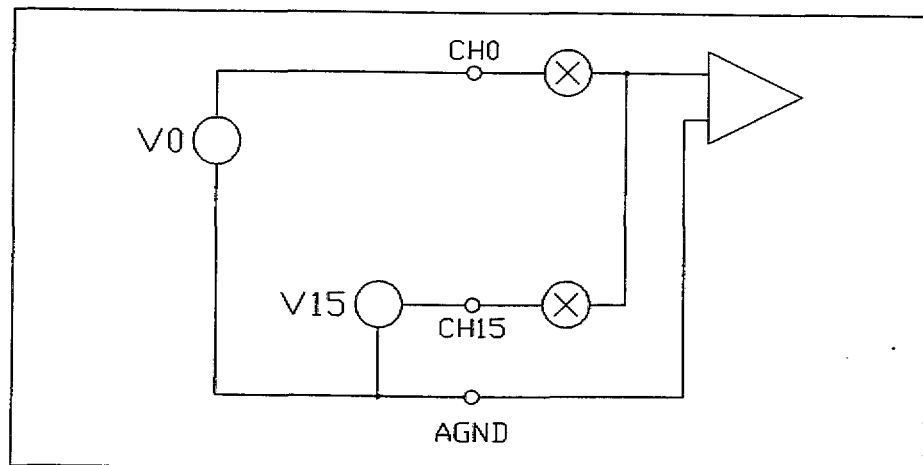
viii. DGND. This is the digital ground. Any digital circuitry tied to the three digital clock lines should be referenced to this line.

4.3.2. Recommendedanaloginputschemes.

Analog signals can be input into the PC-32 in two possible ways - single ended or differential.

4.3.2.1. Single ended inputs.

Figure 4.2.
Single
ended inputs.



In single ended connections, input signals share a common low side. This is analog ground. This input mode has the advantage of giving the maximum number of inputs. Its major disadvantage is the loss of common mode rejection obtainable from differential mode. Single ended inputs are very sensitive to noise. Their use is not recommended with lead lengths of greater than 18 inches, or gains of greater than 5. Single ended input connections are shown in figure 4.2.

4.3.2.2. Differential inputs.

In a differential input mode, two multiplexer switches are used per channel. This means that only 8 input channels are available. The advantage however is greatly improved noise performance. Single ended input connections are shown in figure 4.3.

NOTE. All signal inputs must be referred to analog ground. This can be done by connecting a 1 to 10 kilohm resistor from the low end of each input to analog ground.

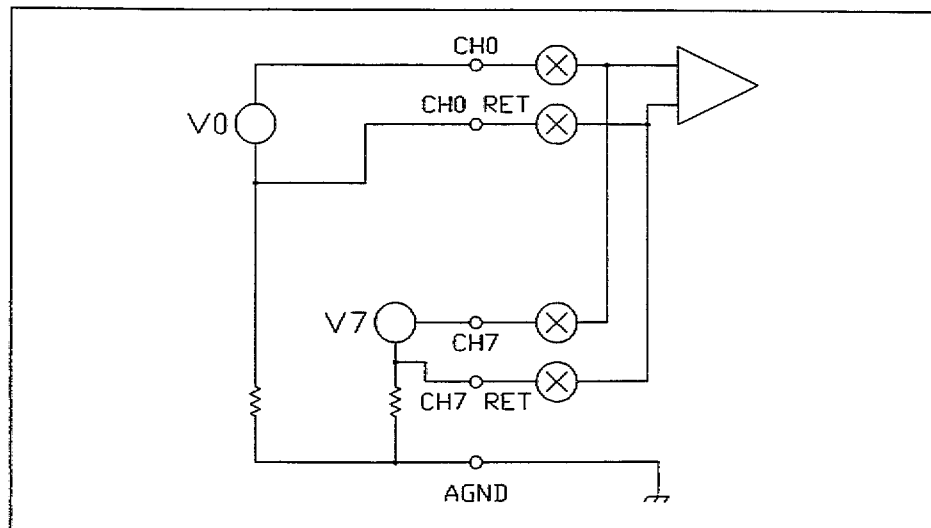
4.3.3. Analog output

The analog output is referenced to analog ground. It may be jumpered to give either current or voltage output.

4.4. Connection guidelines.

The PC-32 is a very high performance I/O subsystem. Its performance may however be severely affected by incorrect connection techniques. This especially true of noise levels.

Figure 4.3.
Differential
Inputs.



4.4.1. Twisted pair input lines.

When using differential inputs, noise levels may be significantly reduced by twisting the low and high input leads of each channel together. This should NOT be done for single ended inputs.

4.4.2. Shielded input lines.

Wherever possible, leads should be shielded. Optimally, each twisted pair in a differential input system should be individually shielded. The shield should be tied to analog ground at the instrument end of the connection only.

4.4.3. Grounding.

If user circuitry is connected to the PC-32 it is of utmost importance to keep the digital and analog ground separate. Although these grounds are tied together on the PC-32, this MUST NOT be done in the user's circuitry.

Chapter 5

Programming the PC-32 card

5.1. Introduction

At the lowest level, the PC-32 can be programmed using I/O input and output instructions. This chapter contains the information required to do this. Although not difficult, this is time consuming, and requires detailed knowledge of the PC-32, as well as the operation of the host PC and its operating system. In order to simplify this process, a set of device drivers is provided along with the card. The use of these allows access to all card functions. These drivers are described in the next chapter.

5.2. Register structure.

The PC-32 uses 16 consecutive address locations in I/O space. Each location has a separate read and write function, giving a total of 32 possible registers. Of these, 13 are used :

Offset	Read	Write
from	function	function
Base		
0	-	Prescale
1	-	D/A clock
2	-	A/D clock

Register structure.

3	-	Clock mode
4	Status	Interrupt
5	-	-
6	-	-
7	-	-
8	A/D LSB	D/A LSB
9	A/D MSB	D/A MSB
10	-	A/D channel
11	A/D start	D/A transfer
12	-	-
13	-	-
14	-	-
15	-	-

Note that all registers marked "-" are reserved, either for manufacturing test or for future expansion. DO NOT either write to or read from these registers.

Note also that the addresses above are given as offsets from the base address of the card. This base address is jumper selected as described in the previous chapter.

Each register will now be described in detail.

5.2.1. Writeregisters.

5.2.1.1. Clock Mode (offset 3)

This register is used to set the mode of the internal D/A and A/D clocks. Five possible commands may be used :

- i. Initialize. (00110100b, 34H). This command initializes the clock system, and must always be written to the clock mode port prior to any other operation.
- ii. D/A disable. (01111000b, 78H). Disable internal D/A clock. This command disables the internal D/A clock source, and must be given if either an external D/A clock is to be used, or software controlled transfer is to be used.
- iii. D/A enable. (01110100b, 74H). If this command is given, then the internal clock source generates clock pulses to the D/A.
- iv. A/D disable. (10111000b, B8H). Disable internal A/D clock. This command disables the internal A/D clock source, and must be given if either an external A/D clock is to be used, or software controlled transfer is to be used.
- v. A/D enable. (10110100b, B4H). If this command is given, then the internal clock source generates clock pulses to the A/D.

Note that either a disable or enable command must be given to both the A/D and D/A clock sources prior to any data conversions. There is no power up default.

5.2.1.2. Prescale (offset 0)

As discussed in the introduction, the frequency presented to the prescaler is the PC's clock frequency divided by two. This frequency is then divided by the value programmed into the prescaler register. The resulting prescaler output frequency is then used as the input to the D/A and A/D clock dividers. The prescaler value is a 16 bit word, ranging from 2 to 65535. Both bytes are written to the same register, LSB first.

5.2.1.3. D/A clock (offset 1)

The actual internal D/A clock frequency is the prescaler output frequency divided by the value programmed into the D/A clock register. The D/A clock value is a 16 bit word, ranging from 2 to 65535. Both bytes are written to the same register, LSB first.

5.2.1.4. A/D clock (offset 2)

The actual internal A/D clock frequency is the prescaler output frequency divided by the value programmed into the A/D clock register. The A/D clock value is a 16 bit word, ranging from 2 to 65535. Both bytes are written to the same register, LSB first.

5.2.1.5. Configuration (offset 4)

The configuration register consists of the following bits :

i. Bit 0 (LSB). D/A Interrupt enable.

When set interrupts from the D/A are routed to the PC's interrupt system. Note that bit 3, interrupt enable must also be set for interrupt operation.

ii. Bit 1. A/D Interrupt enable.

When set interrupts from the A/D are routed to the PC's interrupt system. Note that bit 3, interrupt enable must also be set for interrupt operation.

iii. Bit 2. Address scan enable.

When this bit is set, then successive A/D strobes increment the A/D channel address. When not set, the channel address does not increment, and hence remains the same as set in the A/D channel register.

iv. Bit 3. Interrupt enable.

This bit enables the interrupt system. For interrupt to occur, both this bit, and either the A/D or D/A interrupt enable bit must be set. NOTE : The interrupt enable bit must be set prior to the A/D or D/A bit (two separate output operations must be done, the first setting this bit, and the second setting this bit as well as either the A/D or D/A enable bit).

- v. Bit 4. Reserved.
- vi. Bit 5. Reserved.
- vii. Bit 6. Reserved.
- viii. Bit 7 (MSB). Reserved.

5.2.1.6. D/A LSB (offset 8)

The least significant byte of the D/A output value is written to this port. Coding is offset binary. Note that the D/A value held in a buffer until a D/A strobe occurs. Only then is the data in the buffer transferred to the output.

5.2.1.7. D/A MSB (offset 9)

The most significant byte of the D/A output value is written to this port. Coding is offset binary. Note that the D/A value held in a buffer until a D/A strobe occurs. Only then is the data in the buffer transferred to the output.

5.2.1.8. D/A Transfer (offset 11)

Any write to this location generates a D/A transfer strobe, so transferring the data in the D/A buffer to the D/A output. The value used in the write operation to the D/A transfer port is ignored.

5.2.1.9. A/D Channel (offset 10)

This register consists of the following bits :

- i. Bit 0 (LSB). AD0

This is the LSB of the A/D channel address. The next A/D conversion will be on this channel. Subsequent conversions will be on the same channel if the scan enable bit in the configuration register is not set, and on subsequent channels if it is set. Note that the channel address wraps round to zero. The actual channel address may be read from the CAD0-CAD3 bits of the status register.

- ii. Bit 1. AD1

Part of the A/D channel address.

- iii. Bit 2AD2

Part of the A/D channel address.

- iv. Bit 3. AD3

MSB of the A/D channel address.

- v. Bit 4. G0

G0 is the LSB of the gain setting value. The gain setting value applies to all subsequent A/D conversions. Note that this only applies to cards with the programmable gain option. A table a gain versus gain setting is given below.

vi. Bit 5. G1

MSB of the gain setting value.

vii. Bit 6. Reserved.

viii. Bit 7. (MSB). Reserved.

Gain setting table :

Cards with a PGH suffix (gain setting in binary):

Gain setting	Gain
00	1
01	2
10	4
11	8

Cards with a PGL suffix (gain setting in binary):

Gain setting	Gain
00	1
01	10
10	100
11	500

5.2.2. Readregisters.

5.2.2.1. Status (offset 4)

This register contains the following bits :

i. Bit 0 (LSB). CAD0

This is the LSB of the A/D channel from which the most recent A/D data was read. This is valid only if the A/D ready bit is set.

ii. Bit 1. CAD1

Part of the A/D channel address.

iii. Bit 2. CAD2

Part of the A/D channel address.

iv. Bit 3. CAD3

MSB of the A/D channel address.

v. Bit 4. D/A ready.

This bit is set if the D/A buffer register is empty. It is set by a D/A strobe (either from the internal clock, external clock or software generated), and reset by a write to the D/A LSB register.

vi. Bit 5. A/D ready.

This bit is set if the A/D buffer register is empty. It is set when the A/D completes a conversion, and is reset by a read from the A/D MSB register.

vii. Bit 6. Reserved.

viii. Bit 7 (MSB). Reserved.

5.2.2.2. A/D LSB (offset 8)

The LSB of the result of the most recent A/D conversion may be read from this register. Data format is offset binary.

5.2.2.3. A/D MSB (offset 9)

The MSB of the result of the most recent A/D conversion may be read from this register. Data format is offset binary.

5.2.2.4. A/D Start (offset 11)

A read from this register generates an A/D strobe, so starting the A/D converter. The returned data is random.

Chapter 6

USING THE PC-32 DRIVER SOFTWARE

6.1. Introduction.

The software provided together with the PC-32 has the ability to perform the following functions :

- i. Input a single value from the A/D.
- ii. Output a single value to the D/A.
- iii. Input a set of values from the A/D into a block of memory either under program control or via interrupts. These values may be obtained from any sequential set of input channels. The interval between samples may be set either by the card's internal clock source or via the external clock input to the card. Transfer rates up to the maximum possible for the A/D are supported on a standard 4.77 MHz PC. This routine also provides a sophisticated triggering facility, similar to that in used on digital oscilloscopes. The following trigger parameters may be set :
 - iii.i Trigger value. This sets the voltage at which triggering is to occur.
 - iii.ii Trigger slope. Triggering can occur either on the positive or negative slope of the input waveform.
 - iii.iii Trigger channel.
 - iii.iv A trigger filter value. This allows noise or glitches to filtered out.
 - iii.v Number of samples to be stored before and after the trigger position. Hence either pre- or post-triggering is possible.

- iv. Output a set of values from memory to the D/A either under program control or via interrupts. The interval between samples may be set either by the card's internal clock source or via the external clock input to the card. This routine also provides the ability to output the set an arbitrary number of times, so providing a waveform synthesis capability. Transfer rates up to the maximum possible for the D/A are supported on a standard 4.77 Mhz PC.
- v. Set the internal clock source for both the A/D and the D/A.
- vi. Perform diagnostic functions on the PC-32.
- vii. Object modules are provided for use in conjunction with the following languages :
 - Microsoft C, Pascal and Fortran, utilizing the large, medium or small memory models. Virtually all other compiled languages in use on the IBM PC can also be interfaced to by using one or the other of these modules.
 - IBM BASICA.
 - Turbo Pascal.

All of the above modules provide identical capabilities, within the inherent limitations of the host language.

6.2. Language interface.

All versions of the PC-32 driver software have the following in common :

- i. The software consists of a single module, to which three parameters are passed. These are the following :
 - i.i Address of a common data area. This area is used for communication between the driver and the user's software. It must be allocated by the user. The contents of this area are described in the next section.
 - i.ii Address of a data block. This address is used by the various block in and block out functions.
 - i.iii Function number. This function number selects the operation to be performed.

6.2.1. Format of the common data area.

The common data area consists of an array of 256 integer (16 bit) data locations. This array may either be declared as an array, or for greater clarity, as a structure in those languages which support structures. Examples of declaration as both structures and arrays are given in the demonstration programs. The common data area contains the following :

Address in array	Variable.
0	Portnum. This contains the base address of the card, and must be set by the user.
1	Status. This returns a value indicating whether or not the request could be processed. This value is returned by all functions.
2	Int_lvl. Interrupt level. This is used by the interrupt driven routines to determine which interrupt level is to be used. It must be set by the user prior to use of either of the interrupt routines. Valid values are 3, 4, 5 and 7.
3	Out_val. This value is used by several functions as the value to be written to hardware.
4	Ret_val. This value is used by several input routines to return values
5	Cnt_md. Used to set the mode of the on-board clock generators. See the description of the clock mode function.
6	Prescale. Used to set the division ratio of the prescaler which drives the on-board clock generators. See the description of the clock mode function.
7	Da_cnt. Used to set the division ratio of the on-board D/A clock generator.
8	Ad_cnt. Used to set the division ratio of the on-board A/D clock generator.
9	Da_low_ch. Used for compatibility with the driver software for other cards. Must always be 0.
10	Da_high_ch. Used for compatibility with the driver software for other cards. Must always be 0.
11	Da_n_samp. Used by D/A block output routines, and contains the number of data values in the block.
12	Da_n_rep. Used by D/A block output routines, and contains the number times which the block should be sent to the the D/A. A value of 0 results in a continuous output signal.
13	Ad_low_ch. The lowest A/D channel to be sampled. Must be in the range 0 to 15.
14	Ad_high_ch. The highest A/D channel to be sampled. Must be in the range 0 to 15, and greater than or equal to Ad_low_ch.

15	Ad_gain. This value sets the A/D gain, for those cards which have the programmable gain option. This value ranges from 0 to 3.
16	Ad_n_samp. This value reflects the size of the array into which the A/D block in routines are to place their data. If more than this number of data samples are obtained, then the latest samples wrap round into the beginning of the data area.
17	Ad_pretrig. Number of samples to be obtained on the trigger channel prior to a trigger condition.
18	Ad_posttrig. Number of samples to be obtained on the trigger channel after detecting a trigger condition.
19	Ad_trig_md. This value sets the trigger mode as follows : 0 - No triggering. 1 - Positive edge triggering. 2 - negative edge triggering.
20	Ad_trig_val. Value at which triggering is to occur. (Note that this value is encoded in the same way as the A/D - either offset binary or complimentary binary, depending on the board's switch setting.)
21	Ad_trig_ch. Channel on which to look for a trigger condition.
22	Ad_trig_filt. Number of samples for which the selected channel must be either above or below the specified trigger value for a valid trigger to occur. See the description of the AD block in function for a complete explanation.
23 - 255	Reserved. This part of the data area is reserved for the use of the driver, and must NOT be changed in any way.

6.2.2. Format of the data area

The data area is an array of data, either to be sent to the D/A (for waveform synthesis etc), or obtained from the A/D. In each case the format is as follows :

- i. D/A output block. The block consists of a series of offset binary values.

ii. A/D input block. The block consists of a series of either offset binary or complimentary binary values (depending on the card's switch setting), arranged in order of channel sampled. For example, if channels 2 to 4 were sampled, the contents of the array would be as follows :

- ii.i Channel 2 sample 1
- ii.ii Channel 3 sample 1
- ii.iii Channel 4 sample 1
- ii.iv Channel 2 sample 2 etc.

iii. Function numbers. The function number is normally passed to the driver as a number, rather than the address of a variable, except in the case of the FORTRAN and BASIC versions of the software. The following functions numbers are used : (A complete description of each function is given later in the chapter.)

Function number	Function.
0	Version. This returns the version number of the PC-32 driver.
1	Diagnostics. This function checks the PC-32/33/34.
2	Auto_calibrate. Not used by the PC-32.
3	Open. This function initializes the common data area.
4	Rtc_off. Disables the PC's realtime clock.
5	Rtc_on. Enables the PC's realtime clock.
6	Key_abort. If this function is called, then pressing any key will abort any block operation in progress.
7	Close. Restores all interrupt vectors. This function MUST be called prior to exiting a program which has made use of the PC-32 driver.
8	Da_out. Outputs a single value to the D/A.
9	Ad_in. Inputs a single value from the A/D.
10	Clock_mode. Sets the mode of the PC-32's internal clock sources, as well as the prescaler division ratio.
11	Da_clock. Sets the internal D/A clock rate.
12	Ad_clock. Sets the internal A/D clock rate.
13	Da_block_out. Outputs a block of data to the D/A.

Languages which use the the DOS linker

14	Ad_block_in. Inputs a block of data from the A/D.
15	Da_block_out_i. Outputs a block of data to the D/A using interrupts.
16	Ad_block_in_i. Inputs a block of data from the A/D using interrupts.
17	Int_check. Checks to see if an interrupt operation has completed.

6.3. Calling the PC-32 driver.

In all cases, the PC-32 driver is called as a procedure. This procedure has three three parameters which are the common data area, the data space and the function number. There are two distinct forms of the driver, the first of which is intended for inclusion into compiled languages which use the DOS linker, and the second for languages which do no use the DOS linker.

6.3.1. Languages which use the the DOS linker

For languages which use the the DOS linker, object modules to suit various memory models and naming conventions are provided. These object modules must be linked to the object code of your program. Three languages are specifically supported, but the various modules supplied are general enough to operate in conjunction with almost any any language. The three languages are Microsoft C, Microsoft/IBM PASCAL and Microsoft/IBM FORTRAN. If your language uses the DOS linker, but is not one of these, consult Appendix B.

6.3.1.1. Microsoft C

Microsoft C is supported for the small, medium and large memory models. In all three cases no declaration of the PC-32 driver module is required, and it is called as follows :

```
pc32(&c_d, val, fnum);
```

where `c_d` is the common data area, `val` an array of sample data, and `fnum` is the function number.

The program should be linked to object modules as follows:

Programs using the small memory model : link to PC32CS.OBJ.

Programs using the medium memory model : link to PC32CM.OBJ.

Programs using the large memory model : link to PC32CL.OBJ.

Demonstration programs DEMO8.C, DEMO9.C, DEMO10.C, DEMO11.C and DEMO12.C demonstrate the use of the PC-32 driver in C programs. These programs were compiled using version 4.0 of the Microsoft C compiler.

6.3.1.2. Turbo C

Turbo C programs make use of the same object modules as does Microsoft C. The small, medium and large models only are supported. Calling sequence and declarations are the same as for Microsoft C.

Note. The linker supplied with Turbo C defaults to a case sensitive link. This must either be disabled, or the driver procedure called as PC32, not pc32.

6.3.1.3. Microsoft/IBM Pascal

To be called by a PASCAL program, the PC-32 procedure must first be declared as an external procedure. Two techniques are shown below :

The first makes use of a normal PASCAL call, and uses the medium memory model. Programs using this declaration must be linked with PC32PM.OBJ.

```
procedure pc32(var parameters: integer;

var data: integer;

func_type: integer); extern;
```

The second technique generates a large memory model call sequence. Programs using this declaration must be linked with PC32PL.OBJ.

```
procedure pc32(vars parameters: integer;

vars data: integer;

func_type: integer); extern;
```

In both cases the call is identical, and is shown below :

```
pc32(c_d.portnum, d_a[0], fnum);
```

where c_d.portnum is the first integer in the common data area, d_a an array of sample data, and fnum is the function number.

Demonstration programs DEMO4.PAS and DEMO5.PAS demonstrate the use of the PC-32 driver in Microsoft PASCAL programs. These programs were compiled using version 2.00 of the IBM PASCAL compiler.

Languages which do not use the DOS linker

6.3.1.4. Microsoft/IBM FORTRAN compiler

No declaration is required for the PC-32 driver, and it is called as follows :

```
call pc32(cd(1), da(1), fnum)
```

where cd(1) is the first integer in the common data area, da an array of sample data, and fnum is the function number.

Note that as arrays in FORTRAN begin at 1 rather than 0, all addresses given in section 2.1.2. must be increased by one.

Microsoft FORTRAN programs should be linked to PC32FL.OBJ.

Demonstration program DEMO6.FOR demonstrates the use of the PC-32 driver in Microsoft FORTRAN programs. This program was compiled using version 3.2 of the Microsoft FORTRAN77 compiler.

6.3.2. Languages which do not use the DOS linker

Of the languages which do not use the linker, DOS BASIC and TURBO PASCAL are provided for. Other languages may very easily be used in conjunction with the PC-32 driver as discussed in Appendix B.

The driver for BASIC and TURBO PASCAL is in a somewhat different form to that of the version supplied for use with the linkable languages. In the case of the linkable languages the driver is linked to the program using it, forming a single executable file. In the case of the languages which do not use the linker, the driver is in the form of a memory resident program. This program is run once, prior to the user's program. It then remains permanently in memory, the user's program communicating to it via software interrupts. This has two advantages. Firstly, only a small routine is required to generate the software interrupt, so saving memory space. (Both BASIC and TURBO PASCAL have severe memory space limitations.) Secondly, the driver is not limited to the forms of addressing allowed by the language in question.

6.3.2.1. Installing the PC-32 driver into memory

If you wish to use either BASIC or TURBO PASCAL, the driver must first be made resident in memory. This is done as follows :

Run the program PC32DR.COM. This installs the driver. The program will respond with the following message : "PC-32 Driver installed using vector 66H".

This should be done as soon as you power up your computer, prior to running any other program. It is convenient to make this part of your AUTOEXEC.BAT file. Once the driver is installed in memory, it is there permanently. Only resetting your system will remove it.

6.3.2.2. Calling the driver from BASIC

In order to call the PC-32 driver from BASIC, the machine language subroutine contained in the file PC32B.M must be loaded into memory. This can be done in two ways:

- i. Loading outside of the BASIC work area. This is the simplest technique, and also the technique used in the BASIC demonstration program. Basic uses approximately 64K of memory, in addition to the space used by the operating system and the PC-32 driver, as well as any other memory resident programs. Hence, if your system has sufficient memory, the subroutine can be loaded above the BASIC interpreter.
- ii. Loading inside the BASIC work area. In order to load the subroutine inside the BASIC work area, a section of the work area must be set aside by use of the /M option. This is further described in the BASIC Reference Manual.

In either case, the instructions used are identical, only the actual address used changing. In this case the subroutine will be loaded at address 3FF00H, which is well above the BASIC work area, assuming no other memory resident programs are loaded. This choice of address will work on a PC with 256K or more of memory. Note that PC32B.m can be loaded at position in memory (it is fully relocatable), and is less than 128 bytes in length.

```
10 def seg=&H3FF0

20 blood "pc32b.m", 0

30 pc32 = 0

40 call pc32 (cd%(0), valb%(0), f%)
```

where cd%(0) is the first integer in the common data area, valb%(0) the first integer of an array of sample data, and f% is the function number.

Demonstration program DEMO13.BAS demonstrates the use of the PC-32 driver in BASIC programs. This program was run using version 3.0 of Microsoft BASIC.

6.3.2.3. Calling the driver from TURBO PASCAL

In order to call the driver from TURBO PASCAL, the external subroutine PC32, contained in PC32P.BIN must be used. This is declared as follows :

```
procedure pc_32(var parameters: integer;

var data: integer;
```

```
func_type: integer); external 'pc32p.bin';
```

The procedure is then called as follows :

```
pc_32(c_d.portnum, d_a[0], fnum);
```

where `c_d.portnum` is the first integer in the common data area, `d_a` an array of sample data, and `fnum` is the function number.

Demonstration programs DEMO1.PAS, DEMO2.PAS, DEMO3.PAS and DEMO7.PAS demonstrate the use of the PC-32 driver in TURBO PASCAL programs. These programs were compiled using version 3.01A of the TURBO PASCAL compiler.

6.4. Functions : Detailed Descriptions.

The following should be taken into account when calling the various functions.

- a) The Close function should always be called at the end of a program if the `Key_abort`, `Rtc_off`, `Da_block_out_i` or `Ad_block_in_i` functions were used.
- b) If the `Da_block_out_i` function is active then the `Clock_mode`, `Da_out` and `Ad_block_in` functions should not be called.
- c) If the `Ad_block_in_i` function is active then the `Clock_mode`, `Ad_in` and `Da_block_out_i` functions should not be called.
- d) The Diagnostics function destroys all settings on the card, as well as the contents of the common data area.

6.4.1. Function 0 - Version.

Description :

This returns the version number of the PC-32 driver.

Common Data Area Parameters used

None.

Returns :

Status Always returns 0.

Ret_val Version number in ret_val, encoded as major version number in the high byte, minor version number in the low_byte.

Demonstration programs using this function

DEMO1.PAS, DEMO2.PAS, DEMO3.PAS, DEMO4.PAS, DEMO5.PAS, DEMO6.FOR, DEMO7.PAS, DEMO8.C, DEMO9.C, DEMO10.C, DEMO11.C, DEMO12.C, DEMO13.BAS.

6.4.2. Function 1 - Diagnostics.

Description :

Checks the PC32/33/34, and returns a code signifying what was found. Note that the diagnostic software cannot detect the presence or absence of a D/A converter. The routine can, and does, check the D/A support circuitry, but this is present on all versions of the card.

Common Data Area Parameters used :

PortnumBase address of the PC-32 card as set on the board address jumpers.

Returns :

Status	Always returns 0.
Ret_val	0 - PC-32 or PC-33 found functioning correctly. 1 - PC-34 found functioning correctly or A/D failure (PC-32/33). 2 - A/D support circuit fault. 3 - D/A support circuit fault. 4 - Internal clock circuit failure. 5 - Card not found or bus interface fault. (This code is normally as the result of incorrect base address setting.)

Demonstration programs using this function

DEMO1.PAS, DEMO2.PAS, DEMO3.PAS, DEMO4.PAS, DEMO5.PAS, DEMO6.FOR, DEMO7.PAS, DEMO8.C, DEMO9.C, DEMO10.C, DEMO11.C, DEMO12.C, DEMO13.BAS.

6.4.3. Function 3 - Open.**Description :**

This function initializes a common data area. It must be called prior to any function other than Version and Diagnostics. Only one Data area may be open at a time. A data area is closed (deactivated) by the Close function.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
---------	---

Returns :

Status	Always returns 0.
--------	-------------------

Demonstration programs using this function

DEMO2.PAS, DEMO3.PAS, DEMO7.PAS, DEMO9.C, DEMO10.C, DEMO11.C, DEMO12.C.

6.4.4. Function 4 - Rtc_off.**Description :**

This function disables the PC's real time clock. This can significantly speed the operation of the block I/O functions. The effect on each function is described individually for each function.

Common Data Area Parameters used

None.

Returns :

Status	Always returns 0.
--------	-------------------

Demonstration programs using this function :

DEMO9.C, DEMO10.C, DEMO11.C, DEMO12.C.

6.4.5. Function 5 - Rtc_on.**Description :**

This function enables the PC's real time clock.

Common Data Area Parameters used

None.

Returns :

Status Always returns 0.

Demonstration programs using this function :

DEMO9.C, DEMO10.C, DEMO11.C, DEMO12.C.

6.4.6. Function 6 - Key_abort.

Description :

This function causes any key which is pressed on the PC's keyboard to abort any block operation which is in progress.

Common Data Area Parameters used :

None.

Status

Always returns 0.

Demonstration programs using this function :

DEMO10.C, DEMO12.C.

6.4.7. Function 7 - Close.

Description :

This function restores the PC to state in which it was when the common data area in use was opened. Specifically the following steps are performed :

Interrupts on the PC-32 are disabled, the interrupt vector in use is returned to its original value, and the PC's interrupt hardware is also returned to its original state.

The PC's real time clock is enabled.

The Key_abort routine is deactivated.

NOTE . All programs which used the interrupt system, disabled the real time clock or used Key_abort MUST call this function before ending.

This function must also be called if D/A interrupts must be used after A/D interrupts have been used or vice-versa. It is good practice to call it immediately after either of the interrupt functions terminates.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
---------	---

Returns :

Status	Always returns 0.
--------	-------------------

Demonstration programs using this function :

DEMO7.PAS, DEMO10.C, DEMO12.C.

6.4.8. Function 8 - Da_out.

Description :

Outputs the word in Out_val to the DAC.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Out_val	Contains the output word. Coding is offset binary.
Da_low_ch	Must be 0 for compatibility.

Returns :

Status	0 - Function performed.
	1 - Invalid parameter.

Demonstration programs using this function :

DEMO2.PAS, DEMO3.PAS, DEMO7.PAS.

6.4.9. Function 9 - Ad_in.**Description :**

Inputs a single data value from one of the input channels. Note that this function will hang up the system if called using an incorrect base address, or on a PC-34.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Ad_low_ch	Channel to obtain data from. Must be in the range 0 to 15.
Ad_gain	The gain setting for cards with digitally programmable gain. Must be in the range 0 - 3.

Returns :

Status	0 - Function performed. 1 - Invalid parameter.
Ret_val	Data word. Coding is dependent on the jumpering of the card.

Demonstration programs using this function :

DEMO2.PAS.

6.4.10. Function 10 - Clock_mode.**Description :**

Sets the mode of the internal clock sources as well as the prescaler setting. Note that this function destroys the D/A and A/D clock settings.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Prescale	Count value to which the prescaler should be set. This ranges from 1 to FFFFH.
Cnt_md	This sets the count mode. Values are as follows. 0 - No internal clocks generated. 1 - D/A clock only generated. 2 - A/D clock only generated.

3 - D/A and A/D clocks generated.

Returns :

Status	0 - Function performed.
	1 - Invalid parameter.

Demonstration programs using this function :

DEMO3.PAS, DEMO7.PAS, DEMO9.C, DEMO10.C, DEMO11.C, DEMO12.C.

6.4.11. Function 11 - Da_clock.

Description :

Sets the division ratio of the D/A clock generator.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Da-cnt	Count value to which the D/A clock divider is to be set. This ranges from 1 to FFFFH.

Returns :

Status	0 - Function performed.
	1 - Invalid parameter.

Demonstration programs using this function :

DEMO9.C, DEMO10.C.

6.4.12. Function 12 - Ad_clock.

Description :

Sets the division ratio of the A/D clock generator.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Da-cnt	Count value to which the A/D clock divider is to be set. This ranges from 1 to FFFFH.

Returns :

Status	0 - Function performed.
	1 - Invalid parameter.

Demonstration programs using this function :

DEMO3.PAS, DEMO7.PAS, DEMO11.C, DEMO12.C.

6.4.13. Function 13 - Da_block_out.**Description :**

This function outputs a block of data to the D/A, so effectively functioning as a waveform synthesizer. The D/A must be clocked by either the D/A internal clock or an external clock.

The block waveform represented by the block of data may be generated a certain number of times, or continuously. If continuous generation is selected, the function must be terminated via the Key_abort function.

Maximum frequency of operation for a standard 4.77 MHz PC is approximately 15 Khz with the host PC's realtime clock enabled, and 23 Khz with it disabled.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Da_low_ch	Must be 0. (Used for compatibility).
Da_high_ch	Must be 0. (Used for compatibility).
Da_n_rep	Number of times the entire waveform should be generated. A value of 0 selects continuous generation. The maximum value for this parameter is 32767.
Da_n_samp	Number of data points (words) in the waveform. This value ranges from 1 to 32767.

Returns :

Status	0 - Function performed.
	1 - Invalid parameter.

Demonstration programs using this function :

DEMO9.C.

6.4.14. Function 14 - Ad_block_in.

Description :

This function obtains a block of data from any contiguous set of input channels. Several trigger modes are available :

i. Mode 0. In this case no triggering is performed, and the function simply collects as many samples as are specified in the Ad_posttrig parameter.

ii. Mode 1. This is positive edge triggered mode. The sequence of events is as follows :

ii.i The number of samples specified in the Ad_pretrig parameter are collected from the set of channels.

ii.ii The driver then acquires data until the trigger channel (specified in the Ad_trig_ch parameter) has been less than the trigger value (specified in the Ad_trig_val parameter) for the number of samples specified in the Ad_trig_filt parameter.

ii.iii The driver then acquires data until the trigger channel has been greater than the trigger value for the number of samples specified in the Ad_trig_filt parameter.

ii.iv The driver then acquires as many samples as are specified in the Ad_posttrig parameter.

iii. Mode 2. This is negative edge triggered mode. The sequence of events is as follows :

iii.i The number of samples specified in the Ad_pretrig parameter are collected from the set of channels.

iii.ii The driver then acquires data until the trigger channel (specified in the Ad_trig_ch parameter) has been greater than the trigger value (specified in the Ad_trig_val parameter) for the number of samples specified in the Ad_trig_filt parameter.

iii.iii The driver then acquires data until the trigger channel has been less than the trigger value for the number of samples specified in the Ad_trig_filt parameter.

iii.iv The driver then acquires as many samples as are specified in the Ad_posttrig parameter.

Note that the once the data area is full, the driver wraps round, and begins to overwrite the firsts samples taken.

Maximum frequency of operation for a standard 4.77 MHz PC is approximately 2.5 Khz with the host PC's realtime clock enabled, and 2.8 Khz with it disabled.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Ad_low_ch	First channel to acquire. Ranges from 0 to 15.
Ad_high_ch	Last channel to acquire. Ranges from 0 to 15, and must be greater than Ad_low_ch.
Ad_gain	The gain setting for cards with digitally programmable gain. Must be in the range 0 - 3.
Ad_n_samp	Size of the data buffer in words. This value ranges from 1 to 32767.
Ad_pretrig	Number of samples to obtain of each channel before attempting to trigger. Note that if this parameter is set to 2 then 2 samples of each of the channels from Ad_low_ch to Ad_high_ch will be acquired. This value ranges from 0 to 32767.
Ad_posttrig	Number of samples to obtain of each channel after a valid trigger. Note that if this parameter is set to 2 then 2 samples of each of the channels from Ad_low_ch to Ad_high_ch will be acquired. This value ranges from 0 to 32767.
Ad_trig_md	Sets the trigger mode. 0 - No triggering. 1 - Positive edge triggering. 2 - Negative edge triggering.
Ad_trig_val	Trigger level. The coding for this value is the same as for the A/D (either offset binary or complimentary binary), and is set by the switches on the PC-32 card.
Ad_trig_ch	This sets the channel to which the trigger conditions apply. This must be greater than or equal to Ad_low_ch, and less than or equal to Ad_high_ch.
Ad_trig_filt	This sets the number of samples for which the trigger channel must be above or below the trigger value to be a valid trigger. It used for filtering noise. This value ranges from 0 to 32767.

Returns :

Status	0 - Function performed. 1 - Invalid parameter.
--------	---

Ret_val Returns the offset (in words) of the last sample taken. This allows the determination of the trigger position. If the function was aborted, Ret_val is set to -1.

Demonstration programs using this function :

DEMO3.PAS, DEMO11.C.

6.4.15. Function 15 - Da_block_out_i.

Description :

This function outputs a block of data to the D/A using the interrupt system, so effectively functioning as a waveform synthesizer. It is identical in operation to function 13 except that it functions as a background task. The D/A must be clocked by either the D/A internal clock or an external clock.

The block waveform represented by the block of data may be generated a certain number of times, or continuously.

Maximum frequency of operation for a standard 4.77 MHz PC is approximately 11 KHz with the host PC's realtime clock enabled, and 18 KHz with it disabled.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Da_low_ch	Must be 0. (Used for compatibility).
Da_high_ch	Must be 0. (Used for compatibility).
Da_n_rep	Number of times the entire waveform should be generated. A value of 0 selects continuous generation. The maximum value for this parameter is 32767.
Da_n_samp	Number of data points (words) in the waveform. This value ranges from 1 to 32767.
Int_lvl	Interrupt level to be used. Valid settings are 3, 4, 5 or 7. The level must correspond to that set on the card jumpers.

Returns :

Status	0 - Function performed. 1 - Invalid parameter.
---------------	---

Demonstration programs using this function :

DEMO10.C.

6.4.16. Function 16 - Ad_block_in_i.**Description :**

This function obtains a block of data from any contiguous set of input channels via the interrupt system. This function is very similar to Ad_block_in, with the exception that it runs in the background. Several trigger modes are available :

i. Mode 0. In this case no triggering is performed, and the function simply collects as many samples as are specified in the Ad_posttrig parameter.

ii. Mode 1. This is positive edge triggered mode. The sequence of events is as follows :

ii.i The number of samples specified in the Ad_pretrig parameter are collected from the set of channels.

ii.ii The driver then acquires data until the trigger channel (specified in the Ad_trig_ch parameter) has been less than the trigger value (specified in the Ad_trig_val parameter) for the number of samples specified in the Ad_trig_filt parameter.

ii.iii The driver then acquires data until the trigger channel has been greater than the trigger value for the number of samples specified in the Ad_trig_filt parameter.

ii.iv The driver then acquires as many samples as are specified in the Ad_posttrig parameter.

iii. Mode 2. This is negative edge triggered mode. The sequence of events is as follows :

iii.i The number of samples specified in the Ad_pretrig parameter are collected from the set of channels.

iii.ii The driver then acquires data until the trigger channel (specified in the Ad_trig_ch parameter) has been greater than the trigger value (specified in the Ad_trig_val parameter) for the number of samples specified in the Ad_trig_filt parameter.

iii.iii The driver then acquires data until the trigger channel has been less than the trigger value for the number of samples specified in the Ad_trig_filt parameter.

iii.iv The driver then acquires as many samples as are specified in the Ad_posttrig parameter.

Note that the once the data area is full, the driver wraps round, and begins to overwrite the firsts samples taken.

Maximum frequency of operation for a standard 4.77 MHz PC is approximately 2.5 Khz with the host PC's realtime clock enabled, and 2.6 Khz with it disabled.

Common Data Area Parameters used :

Portnum	Base address of the PC-32 card as set on the board address jumpers.
Ad_low_ch	First channel to acquire. Ranges from 0 to 15.
Ad_high_ch	Last channel to acquire. Ranges from 0 to 15, and must be greater than Ad_low_ch.
Ad_gain	The gain setting for cards with digitally programmable gain. Must be in the range 0 - 3.
Ad_n_samp	Size of the data buffer in words. This value ranges from 1 to 32767.
Ad_pretrig	Number of samples to obtain of each channel before attempting to trigger. Note that if this parameter is set to 2 then 2 samples of each of the channels from Ad_low_ch to Ad_high_ch will be acquired. This value ranges from 0 to 32767.
Ad_posttrig	Number of samples to obtain of each channel after a valid trigger. Note that if this parameter is set to 2 then 2 samples of each of the channels from Ad_low_ch to Ad_high_ch will be acquired. This value ranges from 0 to 32767.
Ad_trig_md	Sets the trigger mode. 0 - No triggering. 1 - Positive edge triggering. 2 - Negative edge triggering.
Ad_trig_val	Trigger level. The coding for this value is the same as for the A/D (either offset binary or complimentary binary), and is set by the switches on the PC-32 card.
Ad_trig_ch	This sets the channel to which the trigger conditions apply. This must be greater than or equal to Ad_low_ch, and less than or equal to Ad_high_ch.
Ad_trig_filt	This sets the number of samples for which the trigger channel must be above or below the trigger value to be a valid trigger. It used for filtering noise. This value ranges from 0 to 32767.

Chapter 7

Calibration

7.1. Introduction

This chapter contains information on the calibration procedures for the A/D and D/A subsystems on the PC-32 series of boards.

These procedures should be performed at six month intervals, or whenever the input or output range jumpers are changed.

NOTE : Allow the host PC and the board to warm up for at least one hour before calibration.

7.2. A/D calibration.

A/D calibration is performed by adjusting two trimpots, R21 and R22. These trimpots are easily located from the labels on the PC-32 board's overlay.

7.2.1. Requirements

- i. Calibration is done on channel 1. The recommended connector wiring is shown in figure 7.1. These connections are suitable for either single ended or differential mode calibration.
- ii. The calibration procedure does not take user installed gain components into consideration. The jumper from location 41 to 40 must be removed for this procedure.
- iii. Calibration is performed with the board jumpered into its intended operating mode.

iv. All cables should be as short as possible.

7.2.2. Equipment required

i. Precision voltage source. Range +10 to -10 V, accuracy 1 μ V.

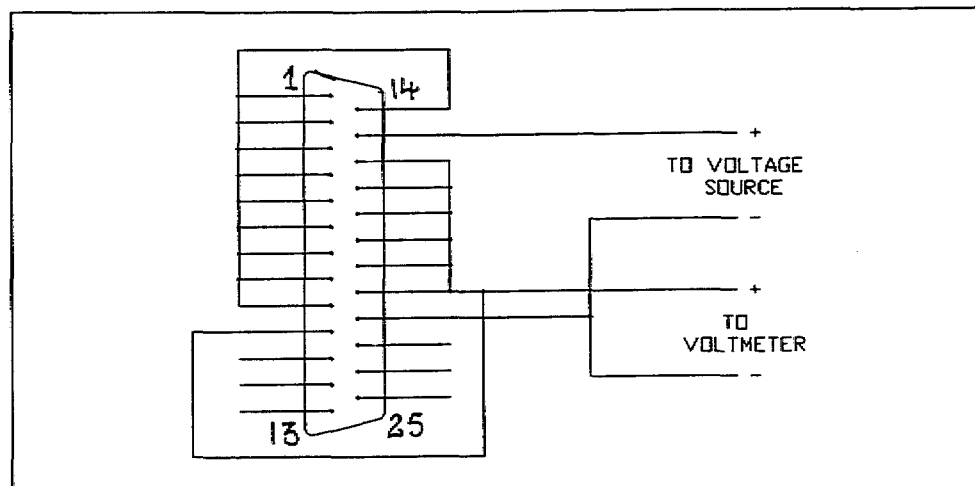
7.2.3. Procedure

7.2.3.1. Unipolar Inputs

i. Set the voltage source to 0.5 LSB (76 μ V for 10 V A/D range).

ii. Use R21 to adjust the A/D reading such that the LSB is switching between 0 and 1, and all other bits are 0. If the range of R21 is not sufficient to calibrate the board, the offset trimpot on the A/D module may also be adjusted.

Figure 7.1. PC-32 calibration interconnections.



iii. Set the voltage source to Full Scale - 1.5 LSB (9.99977V for 10 V A/D range).

iv. Use R22 to adjust the A/D reading such that the LSB is switching between 0 and 1, and all other bits are 1. If the range of R22 is not sufficient to calibrate the board, the gain trimpot on the A/D module may also be adjusted.

These two adjustments interact to some extent, and hence the sequence should be repeated until no further adjustment in either trimpot is required.

7.2.3.2. Bipolar Inputs

i. Set the voltage source to (- Full Scale + 0.5 LSB) (-9.99985V for 10 V A/D range).

- ii. Use R21 to adjust the A/D reading such that the LSB is switching between 0 and 1, and all other bits are 0. If the range of R21 is not sufficient to calibrate the board, the offset trimpot on the A/D module may also be adjusted.
- iii. Set the voltage source to Full Scale - 1.5 LSB (9.99954V for 10 V A/D range).
- iv. Use R22 to adjust the A/D reading such that the LSB is switching between 0 and 1, and all other bits are 1. If the range of R22 is not sufficient to calibrate the board, the gain trimpot on the A/D module may also be adjusted.
- v. These two adjustments interact to some extent, and hence the sequence should be repeated until no further adjustment in either trimpot is required.

7.2.4. A/D Calibration Software.

The program CAL32A.EXE, supplied on the distribution disk, automates the above procedure, repeating the entire sequence several times. For each adjustment, two numbers are displayed, the first being a coarse adjust, and the second a fine adjust. For correct calibration, the first number should be 0, and the second alternating about 0.

7.3. D/A calibration.

D/A calibration is performed by adjusting two trimpots, R1 and R2. These trimpots are easily located from the labels on the PC-32 board's overlay.

7.3.1. Requirements

- i. The recommended connector wiring is shown in figure 7.1.
- ii. Calibration is performed with the board jumpered into its intended operating mode.
- iii. All cables should be as short as possible.

7.3.2. Equipment required

- i. Precision Multimeter. Range +10 to -10 V, accuracy 10 μ V for voltage, range +1 mA to -1 mA, accuracy 1 nA for current output.

7.3.3. Procedure

7.3.3.1. Unipolar Output

- i. Set the D/A code to 0000.
- ii. Use R1 to adjust the D/A output to 0.
- iii. Set the D/A code to FFFF (hex).

- iv. Use R2 to adjust the D/A output to (- Full scale + 1 LSB).

These two adjustments interact to some extent, and hence the sequence should be repeated until no further adjustment in either trimpot is required.

7.3.3.2. Bipolar Output

- i. Set the D/A code to 0000.
- ii. Use R1 to adjust the D/A output to + Full Scale.
- iii. Set the D/A code to FFFF (hex).
- iv. Use R2 to adjust the D/A output to (- Full scale + 1 LSB).
- v. Set the D/A code to 8000 (hex).
- vi. Use R1 to adjust the D/A output to 0.
- vii. Set the D/A code to FFFF (hex).
- viii. Use R2 to adjust the D/A output to (- Full scale + 1 LSB).

These adjustments interact to some extent, and hence the last two adjustments (0 V and - Full Scale) should be repeated until no further adjustment in either trimpot is required.

NOTE. The + Full Scale and - Full Scale adjustments must be performed prior to attempting to set 0 volts out.

7.3.4. D/ACalibrationSoftware.

The program CAL32U.EXE, supplied on the distribution disk, automates the procedure for unipolar output, and the program CAL32B.EXE automates the procedure for bipolar output. The procedure is repeated several times in both cases.

Appendix A

Hardware Specifications

A.1. Inputs

Input Impedance	On channel 100M/120pF Off channel 100M/20pF
Resolution	16 Bits
Linearity	0.006%
Quantization Uncertainty	+ -1LSB
Input offset drift	10uV/degree C.
Gain drift	5 PPM/degree C (gain of 1). 10 PPM/degree C (gain 10 - 1000).
Gain	Resistor programmable from 1 to 1000 as standard. Card is also optionally available with digitally programmable gain.
Full scale input ranges	Card may be ordered for unipolar (full scale range from +5mV to +10V) or bipolar (+-5mV to +-10V range).
Number of inputs	Switch selectable 16 single ended or 8 differential.
Throughput rate	2.5KHz for a gain of 1.

A.2. Output

Resolution	16 Bits
Linearity	0.002%

A.2.1. Fullscale output ranges

Card is jumper selectable for current or voltage output, as well as for unipolar or bipolar operation.

Current output	0 to +2mA or -1 to +1 mA
Voltage output	0 to -10V or +5V or +10V
Settling time	100Us to +0.002% of final value.
Offset drift	8 PPM/degree C.
Gain drift	1 PPM/degree C.

A.2.2. Output compliance

Current mode	+ - 1 V.
Voltage mode	+ - 5 mA.

A.3. Card Timing

The A/D and D/A clocks are independently settable from either programmable internal sources or user supplied external sources. Both the A/D and D/A converter may also be triggered from an external source. Both the A/D and D/A can also either be operated as interrupt driven devices or as program transfer devices.

A.4. IBM PC Interface

Card base address	jumper selectable.
Card interrupt level	jumper selectable.
Power consumption	+5V, 1.5A

Appendix B

Software driver : technical specifications

B.1. Linkable object modules

A total of nine linkable object modules are supplied on the PC-32 disk. In order to understand these, it is necessary to understand certain calling conventions used on 8086 series processors. There are two aspects to this :

i. Memory model. Memory model refers to the length of the pointers used to access data and code. These may be either short (16 bit) or long (32 bit). Three memory models are in common use:

i.i Small model. Short data and code pointers. This means that data pointers consist of a single 16 bit word, and call are of NEAR type.

i.ii Medium model. Short data pointers and long code pointers. Hence call are of FAR type.

i.iii Large model. Long data and code pointers. Data pointers hence consist of 32 bit pointers and calls are of FAR type.

ii. Stack convention. Almost all software makes use of the same convention for placing data on the stack. This is to push the first parameter first, the second second etc. There is however a difference as to when these parameters are removed from the stack (e.i. the stack pointer readjusted).

ii.i PLM calling convention. In this case it is the responsibility of the called procedure to remove any parameters (normally by a RET N).

ii.ii C calling convention. In this case the calling program must remove any parameters.

The various object modules will now be discussed in detail :

B.1.1. Ccallablemodules

These modules make use of the C calling convention. Stack set up on entry to the driver is as follows :

- i. Address of the common data area
- ii. Address of the sample data area
- iii. Function number
- iv. Return address

B.1.1.1. PC32CS.OBJ :

Memory model: Small.

Procedure name: _PC32

Code segment name : _TEXT

Data segment name : _DATA

Data group name: DGROUP

B.1.1.2. PC32CM.OBJ :

Memory model: Medium.

Procedure name: _PC32

Code segment name : PC32_TEXT

Data segment name : _DATA

Data group name: DGROUP

B.1.1.3. PC32CLOBJ :

Memory model: Large.

Procedure name: _PC32

Code segment name : PC32_TEXT

Data segment name : _DATA

Data group name: DGROUP

B.1.2. Pascalcallablemodules

These modules make use of the PLM calling convention. Stack set up on entry to the driver is as follows :

- i. Address of the common data area
- ii. Address of the sample data area
- iii. Function number
- iv. Return address

B.1.2.1. PC32PS.OBJ :

Memory model: Small.

Procedure name: PC32

Code segment name : TEXT

Data segment name : DATA

Data group name: DGROUP

B.1.2.2. PC32PM.OBJ :

Memory model: Medium.

Procedure name: PC32

Code segment name : PC32_TEXT

Data segment name : DATA

Data group name: DGROUP

B.1.2.3. PC32PL.OBJ :

Memory model: Large.

Procedure name: PC32

Code segment name : PC32_TEXT

Data segment name : DATA

Data group name: DGROUP

B.1.3. Fortrancallablemodules

These modules make use of the PLM calling convention. They differ from the previous modules in that the address of a variable containing the function number is passed rather than the function number itself. Stack set up on entry to the driver is as follows :

- i. Address of the common data area
- ii. Address of the sample data area
- iii. Address of the function number
- iv. Return address

B.1.3.1. PC32FS.OBJ :

Memory model: Small.

Procedure name: PC32

Code segment name : TEXT

Data segment name : DATA

Data group name: DGROUP

B.1.3.2. PC32FM.OBJ :

Memory model: Medium.

Procedure name: PC32

Code segment name : PC32_TEXT

Data segment name : DATA

Data group name: DGROUP

B.1.3.3. PC32FL.OBJ :

Memory model: Large.

Procedure name: PC32

Code segment name : PC32_TEXT

Data segment name : DATA

Data group name: DGROUP

Code size for all the above modules is less than 3 Kbyte, and the data size is 0.

B.2. The resident PC32 driver

This driver is called by means of a software interrupt, and is the simplest way to incorporate the PC-32 driver into unsupported languages. All that is necessary is to set up the processor registers, and then to generate a software interrupt. This interrupt defaults to 66, but can easily be changed. This done by running the driver with the interrupt number in hex as a parameter. For example, to use interrupt number 68H the command line would be as follows :

PC32DR 68

Parameters are passed to the driver as follows :

At the time that the software interrupt is generated the registers should contain the following :

Function number	AX.
Common data area address	DS:BX.
Sample data area address	ES:DI.

Appendix C

Commonly asked questions.

Q : I can't get my card to work at all. The diagnostics report that no card is present.

This normally comes about because of an addressing problem. Check the base address setting jumpers. This is the most normally the problem. If the address is set correctly, check for address conflicts with other cards in the system.

Q : I can't get my card to work at all. The diagnostics report a timer problem.

This is normally due to an address conflict. The diagnostics find a card, but this fails the first test, which is the timer. Note that this fault may in fact be reported even if no PC-32 is present, if some other card is in the address space in which the diagnostics expect to find a PC-32.

Q : The diagnostics reports a problem. I am sure that the addressing is OK.

Firstly, try the card with as little as possible which may interfere with it. This includes any other cards in your PC (remove all cards not absolutely required for your PC to operate) and any memory resident programs. If the problem persists, try the card in another PC (preferably a genuine IBM). If the problem persists, return the card together with all information on the system (hardware configuration, serial number of the PC, operating system, any other software involved) to your dealer.

Q : The card passes diagnostics OK, but I have found a problem with the card/the software driver.

Firstly, perform the steps listed for the previous problem. If the problem persists, then preferable return the card with all the information listed above, as

well as the source code of whatever software shows up the problem (preferable on disk) as well as any listing and a COMPLETE description of the problem. If you wish to keep the card, send only the above information to your dealer. Please include as much information as possible, program versions etc.

Q : I would like to get source code for the device drivers/ manufacturing information, as this will save me a great deal of time.

Source code of the device drivers as well as engineering data is available to approved OEM users. This is not however stocked by dealers, and must be ordered for a specific client. Consult your dealer for further information.

Index

A

A/D channel	5-4
A/D clock	5-3
A/D configuration	3-1, 3-5, 3-6
A/D LSB	5-6
A/D MSB	5-6
A/D start	5-6
A/D strobcs	2-3
A/D Subsystem	1-2, 2-1
Accuracy	3-5
Ad_block_in	6-10, 6-18, 6-19
Ad_block_in_i	6-10, 6-21, 6-22
Ad_clock	6-16
Ad_cnt	6-3
Ad_gain	6-4
Ad_high_ch	6-3
Ad_in	6-10, 6-15
Ad_low_ch	6-3
Ad_n_samp	6-4
Ad_post_trig	6-4
Ad_pre_trig	6-4
Ad_trig_ch	6-4
Ad_trig_filt	6-4
Ad_trig_md	6-4
Ad_trig_val	6-4
ADCLK	2-4, 4-2
ADXFR	2-3
AGND	4-2
Amplifier	2-2
Analog inputs	4-2
Analog output	4-2, 4-3

B

Base address	2-2, 3-1, 3-2, 5-2
BASIC	6-9
Bus interface	1-3, 2-2, 3-1, 3-3

C

C language	B-2
Calibration	
A/D	7-1, 7-2
D/A	7-3, 7-4
Calling convention	B-1
Channel Address	2-1
Clock mode	5-2
Clock_mode	6-10, 6-15
Clocks	3-1, 3-4
Close	6-10, 6-13

Cnt_md	6-3
Common data area	6-2, 6-3
Configuration register	5-3

D

D/A clock	5-3
D/A configuration	3-1, 3-4
D/A LSB	5-4
D/A MSB	5-4
D/A Strobe	2-1
D/A strobes	2-3
D/A Subsystem	1-3, 2-1
D/A transfer	5-4
Da_block_out	6-17
Da_block_out_i	6-10, 6-20
Da_clock	6-16
Da_cnt	6-3
Da_high_ch	6-3
Da_low_ch	6-3
Da_n_rep	6-3
Da_n_samp	6-3
Da_out	6-10, 6-14
DACLK	2-4, 4-2
Data area	6-4, 6-5
DAXFR	2-3
DGND	4-2
Diagnostics	C-1, 6-10, 6-11
Differential inputs	4-3
Digital ground	4-2

E

External clock inputs	2-4
-----------------------	-----

F

FORTTRAN	B-4, 6-8
Functions	6-5

G

Gain	3-5, 5-5
Grounding	4-4

I

IBM backplane	4-1
Instrumentation amplifier	2-2
Int_check	6-23
Int_lvl	6-3
Interrupts	2-2, 3-1, 3-3

J

Jumpers	3-2
---------	-----

K

Key_abort	6-10, 6-13
-----------	------------

L

Languages	6-2
-----------	-----

M

Memory models	B-1
Memory resident	6-8
Microsoft C	6-6
Microsoft/IBM PASCAL	6-7
Multiplexer	2-1

O

Open	6-12
Out_val	6-3

P

Parameters	6-2
PASCAL	B-3
PC-32	1-1
PC-32 driver	6-8
PC-32/PGH	1-1
PC-32/PGL	1-1
PC-32/U	1-1
PC-32/U/PGH	1-1
PC-32/U/PGL	1-1

PC-33	1-1
PC-33/PGH	1-1
PC-33/PGL	1-1
PC-33/U	1-2
PC-33/U/PGH	1-2
PC-33/U/PGL	1-2
PC-34	1-2
PC32B.M	6-9
PC32CL.OBJ	B-2
PC32CM.OBJ	B-2
PC32CS.OBJ	B-2
PC32DR	B-5
PC32DR.COM	6-8
PC32FL.OBJ	B-4
PC32FM.OBJ	B-4
PC32FS.OBJ	B-4
PC32PL.OBJ	B-3
PC32PM.OBJ	B-3
PC32PS.OBJ	B-3
PLM	B-1
Portnum	6-3
Power supplys	4-2
Prescale	5-3, 6-3
Programmable gain	3-6

R

Registers	5-1, 5-6
Ret_val	6-3
Rtc_off	6-10, 6-12
Rtc_on	6-12

S

Sample and hold	2-2
Shielded inputs	4-4
Single ended inputs	4-3
Software	1-3, 1-4, B-1, B-4
Software interrupt	6-8
Software interrupts	B-5
Specifications	1-1, 1-3
Stack	B-1
Status	6-3
Status register	5-5

T

Timer	C-1
Trig	4-2
Trigger	6-1

Triggering	6-4
Turbo C	6-7
Turbo PASCAL	6-9
Twisted pair inputs	4-4

U

User inputs	4-2
-------------	-----

V

Version	6-10
---------	------

