ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Measurement Computing / ComputerBoards CIO-DAS16/JR* at our website: *Click HERE*

CIO-AD16JR-AT
(CIO-DAS16/330)
USER'S MANUAL

REVISION 1, OCTOBER 1991

&

CIO-AD16JR-AT UTILITY DISK, REV 1.0

Computer Boards, Inc.
44 Wood Ave
Mansfield, MA 02048

(508) 261-1123
FAX 261-1094

(C) 1991 Computer Boards, Inc.

COPYRIGHT NOTICE: This document and all the diagrams are the property of Computer Boards, Inc. and may not be reproduced or copied by any means or for any purpose.

Table of Contents

1 QUICK START 1

2 INTRODUCTION 2

3 CIO-AD16JR-AT - SPECIAL FEATURES OVERVIEW 3

4 INSTALLATION 4
- 4.1 BASE ADDRESS 4
- 4.2 MODE SWITCH 4
- 4.3 DMA LEVEL SELECT 5
- 4.4 1/10 MHz XTAL JUMPER 5
- 4.5 8/16 CHANNEL SELECT 5
- 4.6 INSTALLING THE CIO-AD16JR-AT IN THE COMPUTER 6
- 4.7 PROGRAMMABLE RANGE AND GAIN SETTING 6
- 4.8 SIGNAL CONNECTION TO TEST THE INSTALLATION 6

5 SOFTWARE INSTALLATION 8
- 5.1 UTILITY SOFTWARE 8
  - 5.1.1 IOTEST.EXE 8
  - 5.1.2 CAL16JR 8
  - 5.1.3 CIO16 CALL ROUTINE & EXAMPLES 8
- 5.2 LABLOG II 8
- 5.3 ACQUIRE 9

6 CALIBRATION AND TEST 10
- 6.1 FREQUENCY OF CALIBRATION 10
- 6.2 REQUIRED EQUIPMENT 10
- 6.3 CALIBRATING THE ANALOG INPUTS 10
- 6.4 TESTING THE COUNTER/TIMER & DIGITAL I/O 12

7 SIGNAL CONNECTION 13
- 7.1 CONNECTOR DIAGRAM 13
- 7.2 ANALOG INPUTS 13
  - 7.2.1 SINGLE ENDED 14
  - 7.2.2 FLOATING DIFFERENTIAL 14
  - 7.2.3 DIFFERENTIAL 14
- 7.3 DIGITAL OUTPUTS & INPUTS 15

8 BASIC PROGRAMMING 16
- 8.1 SYNTAX OF THE EXAMPLES 16
- 8.2 BASIC CALL STATEMENT GUIDELINES - INTERPRETERS 16
  - 8.2.1 HOW TO CALCULATE SEG 17
  - 8.2.2 TWO CIO-AD16JR-AT BOARDS IN ONE PC 17
- 8.3 BASIC CALL STATEMENT GUIDELINES - COMPILERS 18
  - 8.3.1 HOW TO LOAD QUICK BASIC AND CIO16 18
- 8.4 CIO16 CALL FORMAT RULES 18
- 8.5 MODE SUMMARY 19
- 8.6 MODE 0 INITIALIZE 21
- 8.7 MODE 1 SET MUX LIMITS 22
- 8.8 MODE 2 READ MUX 23
- 8.9 MODE 3 DO ONE A/D 24
- 8.10 MODE 4 DO N A/D TO ARRAY 25
- 8.11 MODE 5 DO N A/D TO MEMORY 26
- 8.12 MODE 6 DO N A/D VIA DMA 27
- 8.13 MODE 7 DISABLE INTERRUPT/ DMA 29
- 8.14 MODE 8 READ INTERRUPT/ DMA 30
- 8.15 MODE 9 TRANSFER DATA TO ARRAY 31
- 8.16 MODE 10 CONFIGURE COUNTER 0 32
- 8.17 MODE 11 LOAD COUNTER 0 33
- 8.18 MODE 12 READ COUNTER 0 34
- 8.19 MODE 13 WRITE DIGITAL OP0-3 35
- 8.20 MODE 14 READ DIGITAL IP0-3 36

8.21 MODE 17 LOAD A/D PACER CLOCK CTR 1&2 37
8.22 MODE 19 ANALOG TRIGGER 38
8.23 MODE 20 A/D SCAN ON INTERRUPT 39
8.24 MODE 40 330KHz FOREGROUND A/D 40
8.25 MODE 41 330KHz BACKGROUND A/D 42
8.26 MODE 42 PRE/POST TRIGGER A/D FOREGROUND 44
8.27 MODE 43 PRE/POST TRIGGER A/D BACKGROUND 46
8.28 MODE 44 TRANSFER MODE 43 DATA TO BASIC 48
8.29 MODE 45 FORCE COMPATIBILITY MODE 49
8.30 MODE 50 EASY TO USE A/D 50
8.31 MODE 51 LOAD COUNTER 1&2 WITH Hz 52
8.32 CALL ERROR CODES 53
8.33 SPECIAL START/STOP CHANNEL & SPEED LIMITATIONS 55

9 CIO-AD16JR-AT ARCHITECTURE 56
9.1 PC/XT/AT BUS INTERFACE 57
9.1.1 BASE ADDRESS 57
9.1.2 MODE SWITCH 57
9.1.3 BUS POWER CONSUMPTION 58
9.1.4 DATA TRANSFERS 58
9.2 CONTROL & DATA REGISTERS 58
9.2.1 A/D DATA & CHANNEL REGISTERS 60
9.2.2 CHANNEL MUX SCAN LIMITS REGISTER 61
9.2.3 4 BIT DIGITAL I/O REGISTERS 61
9.2.4 STATUS REGISTER 62
9.2.5 DMA, INTERRUPT & TRIGGER CONTROL 62
9.2.6 PACER CLOCK CONTROL REGISTER 63
9.2.7 ANALOG INPUT RANGE REGISTER 64
9.2.8 PACER CLOCK DATA & CONTROL REGISTERS 65
9.2.9 ENHANCED FEATURES PACER CLOCK DATA & CONTROL REGISTERS 66
9.3 ANALOG INPUT 67
9.4 DIGITAL INPUT & OUTPUT 67
9.4.1 OUTPUT 67
9.4.2 INPUT 67
9.5 INTERRUPT & TRIGGER CONTROL 67
9.6 DMA CONTROL LOGIC 68
9.7 SPECIFICATIONS 69

10 APPLICATION NOTES 70
10.1 UP-TIME / DOWN TIME & LOAD MONITOR 71
10.2 PC COLLECTS DATA FROM TEXTURE ANALYZER 72
10.3 FREQUENCY MEASUREMENT WITH A CIO-AD16JR-AT 76

11 ANALOG ELECTRONICS 79
11.1 VOLTAGE DIVIDERS 79
11.2 DIFFERENTIAL & SINGLE ENDED INPUTS 80
11.3 LOW PASS FILTERS 83
11.4 A/D RESOLUTION & ENGINEERING UNITS 83
11.5 CURRENT LOOP 4-20 mA 84
11.6 NOISE 85
11.7 DIGITAL INTERFACING 86
11.7.1 TTL TO TTL SIGNAL CONNECTION 86
11.7.2 LS244 OUTPUT BUFFERS 86
11.7.3 TRANSISTORS 86
11.7.4 POWER MOSFET's 88
11.7.5 TTL TO SOLID STATE RELAYS 89

12 PC ARCHITECTURE 90
12.1 **CPU SPEED** 90
12.2 **EGA & VGA SCREENS** 90
12.3 **PC/AT 16 BIT DATA BUS** 90
12.4 PC TECHNICAL REFERENCE 91
12.4.1 **TABLE OF I/O ADDRESS** 91
12.4.2 **MEMORY MAP** 91

12.4.3 **INTERRUPTS** 91
12.5 DIRECT MEMORY ACCESS 92

13 DIAGNOSIS & DEBUG 94

14 HELPFUL DIAGRAMS 98

Table of Figures

I/O TEST Program Digital to A/D loopback 1
Base address switch at 300H 4
DMA Level Switch 5
XTAL jumper block in 1 MHz position. 5
Channel Select Switch 6
I/O TEST Progarm Digital to A/D loop-back 7
CIO-AD16JR-AT Board layout showing calibration trimpots 11
Analog 37 pin connector pin outs. 13
Single Ended input signal connection. 14
Floating Differential signal connection. 14
Differential signal connections. 15
CIO-AD16JR-AT board potentiometer and jumper locations. 56
Base address switch. 57
8254 Counter/Timer and control register diagram. 64
8254 Counter/Timer and control register diagram. 67
DMA level select switch. 68
Industrial electric motor load monitoring. 71
Voland Texture Analyzer chart recorder interface plug. 72
A/D board connectors with Voland TA signals highlighted 73
CIO-AD16 to Voland TA cable diagram. 73
Texture analyzer to CIO-AD08 cable. 74
CIO-AD16 connector wireing schematic. 76
Frequency measurement timing chart. 77
Voltage divider schematic. 79
Differential inputs. 81
Single ended inputs. 81
Common mode range graph. 82
Low pass filter schematic. 83
Schematic of 4-20 mA to analog voltage input. 84
Transistor diagram. 87
TTL and transistor controlling a flashlight bulb. 87
Transistor schematic for active low signal. 88
TTL to power MOSFET schematic. 88
CIO-AD16Jr Signal Connector 98
Base Address switch. 98
DMA level select switch 98
8/16 channel select switch. 98
XTAL clock speed jumper. 98

1 QUICK START

The CIO-AD16JR-AT is easy to use. Here is the quick start procedure for those who know how to open the PC an install expansion boards, and want to dive right in.

BOARD SETUP

The CIO-AD16JR-AT is setup at the factory with:

BASE ADDRESS	300H (768 Decimal) Same as data sheet.
DMA CHANNEL SELECT	1 Choices are 1 & 3. 3 is used by the hard disk in an XT.
1/10 MHz jumper	1 MHz[1]
8/16 CHANNEL SELECT	8 Differential input channels.
GAIN/RANGE	Programmable. Power-on default is +/- 5 volts
MODE SWITCH	UP Position 7 of the Base Address DIP switch Only AD16 Compatible when UP. No extended fetures.

Open your PC (after turning off the power) and install the board. Leave the switches as they were set at the factory or refer to the data sheet to change the settings. They are described in detail on the data sheet and in this manual.

After the board is installed and the computer is closed up, turn the power back on.

SOFTWARE SETUP

Insert the CIO-AD16JR-AT UTILITY disk in drive A:, or...

Answer your software program setup questions with the information above. The LABLOG II program shipped with your disk has a CIO-AD16JR-AT installed. So does the LabTech Acquire program. The Compatible I/O Series utility software and drivers are also preconfigured.

Keep in mind, the CIO-AD16JR-AT is 100% compatible with the CIO-AD16, CIO-AD16/F and MetraByte DAS-16 series. Software written for these other boards is already configured for the CIO-AD16JR-AT.

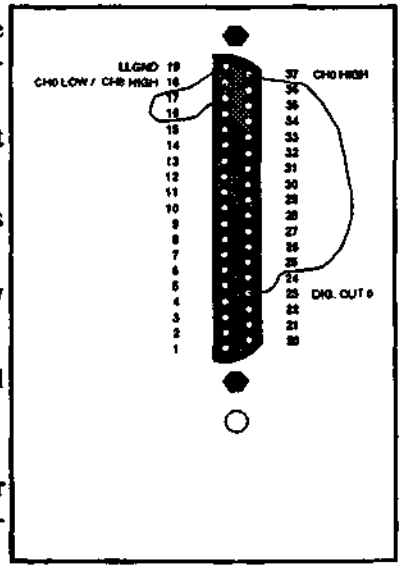
SIGNAL CONNECTION

The board is factory configured in differential input mode. Differential Input mode requires 3 connections be made from the signal source to the CIO-AD16JR-AT. These are SIGNAL HIGH, SIGNAL LOW & GROUND.

To test the analog inputs you need a known signal to connect to them. The simplest signal to connect is the Digital Output 0 from the CIO-AD16JR-AT.

The DAS-16 compatible analog inputs are on the 37 pin connector which extends through the faceplate out the back of the PC when the board is installed.

Using the shorting wires supplied with the board, short pins 18 to 19 and pins 37 to 9 as shown to the right.



Run the program 'IOTEST' on the CIO-AD16JR-AT utility disk. Select Digital output channel 0 and Analog input channel 0.

You should see a slow moving square wave on the screen.
The CIO-AD16JR-AT is installed and running correctly. You may now run other programs from the utility disk or install and run LABLOG II or LabTech Acquire.

---

1 Some programs, such as Unkelscope, expect the 1/10 MHz jumper to be in the 1 MHz position. Please refer to the software user's manual for guidence. Computer Boards' BASIC CALL routines assume the jumper is in the 1MHz position.

2 INTRODUCTION

This manual is intended to assist your application of the CIO-AD16JR-AT to your measurement or control problem. We are interested in your corrections and suggestions toward that end. There is no form in this manual for that purpose because we would prefer to talk to you. Please pick up the phone and speak with one of our Technical Support/Sales Engineers. They will note all of your suggestion for implementation in the next revision of this manual.

As a CIO-AD16JR-AT board owner, the latest revision of the manual is your's free, any time you request it.

A recent Personal Engineering Magazine reader's poll showed that documentation is the number one complaint among I/O board users. Help us change this. Tell us how to make the manuals better and what additional information would be helpful.

ORGANIZATION

This manual is organized in the order that you are likely to need the information in it. To fully understand the CIO-AD16JR-AT and the implications of installing and using it, we suggest that you read the manual and follow along. It will take only an hour or so and when you have finished you will be prepared to use your CIO-AD16JR-AT.

INSTALLATION	How to install the CIO-AD16JR-AT.
SOFTWARE	How to install and run the supplied software.
CALIBRATION	How to calibrate and test the board.
SIGNAL CONNECTION	How to connect analog and digital signals.
BASIC PROGRAMMING	How to program with the supplied CALL routine.
ARCHITECTURE	How the board works and how to program the registers.
APPLICATION NOTES	Some common applications fully explained.
ANALOG ELECTRONICS	A *very* basic course in the subject.
PC ARCHITECTURE	How the architecture of the PC affects board performance.
DIAGNOSIS & DEBUG	What to do if you think the board is broken.

3 CIO-AD16JR-AT - SPECIAL FEATURES OVERVIEW

Important!

Please read this section through.

JR-AT IS TWO ARCHITECTURES ON ONE BOARD

The CIO-AD16JR-AT is two architectectures on one A/D board; standard DAS-16 registers and extended registers.

To maintain compatibility with existing DAS-16 software, such as Labtech, SnapShot and Language Drivers, a complete set of DAS-16 (CIO-AD16) registers exist at BASE + 0 through BASE + 15. When in *Compatible Mode*, the JR-AT behaves as a CIO-AD16 (DAS-16G) would with the exception of much faster sample rates.

A second set of registers exists at BASE + 16 through BASE + 19. In addition, a special register at BASE + 11 opens up when the JR-AT is in *enhanced mode.*

INITIATING ENHANCED MODE

The JR-AT is placed in enhanced mode by setting one switch, then by writing to a specific address. When in enhanced mode, the JR-AT occupies 32 I/O addresses and so may only be placed on even-tens I/O addresses. Examples of achievable base addresses are; 300H, 320H, 340H. Addresses such as 310H and 330H are not possible in enhanced mode.

WHAT DOES ENHANCED MODE GIVE ME?

Enhanced mode opens up additional counters and control registers which allow:

- 16 bit bus transfers.
- Transfer rates of 330KHz using the REP IN SW command.
- Pre-trigger and post-trigger sample buffers.

Enhanced mode is fully supported by the BASIC language driver which was shipped with your CIO-AD16JR-AT. Using it, you can acquire data at rates to 330KHz and store pre/post trigger buffers of unlimited size (system RAM limits). Software packages such as Labtech Notebook will support the enhanced features of the JR-AT in the near future.

AS YOU FOLLOW THE INSTALLATION PROCEDURES, CONSIDER WHICH MODE YOU WANT TO RUN IN.

4 INSTALLATION

The CIO-AD16JR-AT has one bank of switches, two single function switches and one jumper block which must be set before installation of the board inside your computer. The calibration and test program included with the CIO-AD16JR-AT will show how these switches are to be set and should be run before you open your computer.

To run the calibration and test program, first install the software included on the CIO-AD16JR-AT Utility disk. Create a directory on your hard disk using the command:

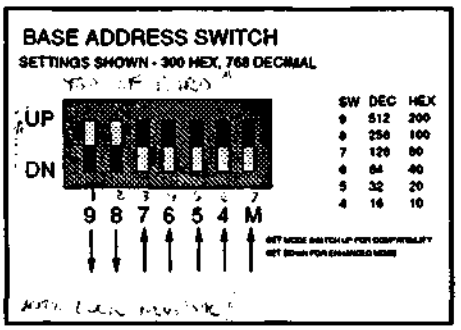
[enter]	This means hit enter, don't type the brackets.
MD \CIO16 [enter]	Create a hard disk directory named \CIO16.
CD \CIO16 [enter]	Change to that directory.
	Now put the utility disk in the A: drive.
COPY A:*.* [enter]	The files will be copied.
AD16JR-ATSW [enter]	De-compress the programs from the distribution file.
CAL16JR [enter]	The CIO-AD16JR-AT calibration and test program.

If you have encountered difficulty and cannot diagnose it, please find someone at your facility to help you. These are simple DOS commands and there are many more you will need to be familiar with to use a data acquisition board and analyze data. Possibly someone can show you how to use DOS.

4.1 BASE ADDRESS

Menu choice number 1 from the CAL16JR main menu is 'SELECT BASE ADDRESS' and runs a program that allows you to choose a desired base address and adjusts the base address if necessary to place it on a 32 or 16 bit boundary.

After a base address is chosen, a diagram of the switch setting is drawn on the PC screen.[2] Set the switches on your base address switch as shown on the diagram. Unless there is already a board in your system which uses address 300 HEX (768 Decimal) then you can leave the switches as they are set at the factory.



In the example shown here, the CIO-AD16JR-AT is set for base address 300H.

Switch #7 of the base address switch is a MODE SWITCH and will not affect the base address of the board.

4.2 MODE SWITCH

Switch #7 of the base address switch doubles as a mode switch. The mode switch enables and disables extended addresses and features of the CIO-AD16JR-AT.

The state of the mode switch may be read back at Base Address + 11. When UP (OFF), Base Address + 11, BIT 4 reads back as 0 (zero). When the mode switch is down, Base Address + 11, BIT 4 reads back as 1 (one).

When the switch is ON (UP), the CIO-AD16JR-AT is compatible with the MetraByte DAS-16, ComputerBoards CIO-AD16 and CIO-AD16Jr. Only 16 I/O address are used by the CIO-AD16JR-AT when the mode switch is up. Because only 16 I/O addresses are used when the mode switch is off, the CIO-AD16JR-AT may be placed on 16 bit boundaries, such as 300H, 310H 320H, 330H and 340H.

When the mode switch is OFF (DOWN), the CIO-AD16JR-AT is in extended features mode, or full PC/AT mode. Additional control bits are present in the upper nibble of Base Address + 11 and a second 8254 counter is addressable for pre/post trigger control. When the mode switch is down, addresses such as 300H, 320H and 340H are available while addresses such as 310H and 330H are not. In enhanced mode, the CIO-AD16JR-AT occupies 32 I/O addresses.

The mode switch must be down to access extended features through software. The CIO-AD16JR-AT programming language CALL routines and examples which use high-speed transfers or extended features require the modes switch be down.

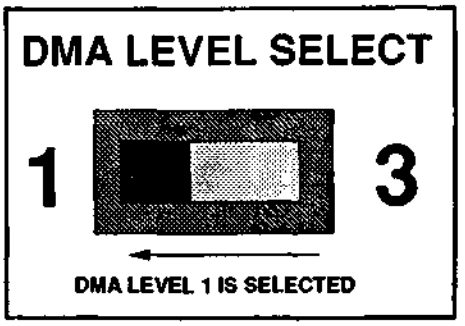
---

2 This program attempts to locate conflicts but can only do so if the device occupying the base address you have chosen has read-back registers.

4.3 DMA LEVEL SELECT

What kind of computer are you installing the board in? If it is an XT then there are only two DMA levels available and level 3 is probably used by the hard disk controller in your XT computer. Set the DMA level switch to the level 1 position.

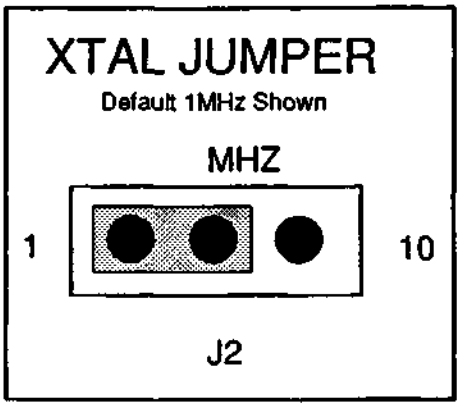
If you have an AT or 386 type computer the hard disk controller is not at level 1 or 3 so either level may be used.



There are other boards that use DMA levels. Some network boards do and so do some IEEE-488 interface boards. Do you have other boards in your computer with DMA level switches on them?

4.4 1/10 MHz XTAL JUMPER

The 1/10 MHz XTAL jumper selects the frequency of the square wave which is connected to OUT 2, the output of 8254 counter 2. Counter 2 is used to pace the A/D start conversion trigger.



Why is this jumper on the board? Because the original DAS-16, designed in 1984, had a 1 MHz crystal. When MetraByte redesigned the DAS-16 and added the faster 10MHz crystal, a jumper was provided to maintain compatibility with older software. The CIO-AD16JR-AT has the jumper because the DAS-16 has the jumper and some software expects the jumper to be in the 1 MHz position and some software expects the 10 MHz position.

The CIO-AD16JR-AT is shipped with the jumper in the 1 MHz position. The BASIC CALL routine, CIO16, allows you to specify a 1 or 10 MHz position in the setup[3]. Certain MODES, 50 and 51 for example, allow the sample rate to be specified in Hz rather than having to independently set the counter speed. To calculate the counter load values the counter 2 input frequency must be known.

Some software programs, such as UNKELSCOPE, ASYST and ASYSTANT+ require that the jumper be in the 1MHz position. Please refer to the software program user's manual for guidence. Othe programs, such as Labtech Notebook, have a 1/10MHz choice in the set-up menu.

4.5 8/16 CHANNEL SELECT

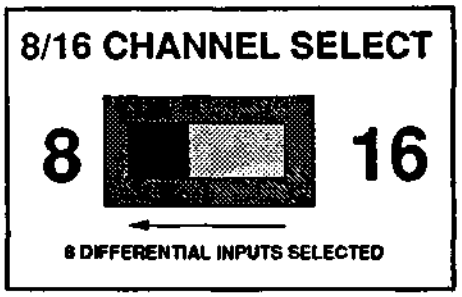
The analog inputs of the CIO-AD16JR-AT may be configured as 8 differential or 16 single ended[4]. Using single ended inputs means you have more separate analog input channels available to connect signals. Using differential inputs allows up to 10 volts of common mode (ground loop) rejection and can be more noise immune.

---

3 See the explaination of MODE 0 in the section on BASIC CALL software.

4 For a full discussion of the differences, benefits and applications of single ended and differential inputs see the sections on signal connection, the introduction to analog electronics and the application notes.

The CIO-AD16JR-AT comes from the factory configured for 8 differential inputs so the 8/16 switch is in the position shown here. Set it for the type and number of inputs you desire.



4.6 INSTALLING THE CIO-AD16JR-AT IN THE COMPUTER

Turn the power off.

Remove the cover of you computer. Please be careful not to dislodge any of the cables installed on the boards in your computer as you slide the cover off.

Locate an empty expansion slot in your computer.

Push the board firmly down into the expansion bus connector. If it is not seated fully it may fail to work and could short circuit the PC bus power onto a PC bus signal. This could damage the motherboard in your PC as well as the CIO-AD16JR-AT.

4.7 PROGRAMMABLE RANGE AND GAIN SETTING

The CIO-AD16JR-AT analog input range is fully programmable. The range is set by writing to a register at base address + 11. The 4 lower bits of that register correspond to settings of:

Range	Uni/Bip	G1	G0	Input Range	Decimal
1	0	0	0	±10V	8
0	0	0	0	±5V	0
0	0	0	1	±2.5V	1
0	0	1	0	±1.25V	2
0	0	1	1	±0.625V	3
0	1	0	0	0-10V	4
0	1	0	1	0-5V	5
0	1	1	0	0-2.5V	6
0	1	1	1	0-1.25V	7

Of course, the bits and control words are only of interest to programmers. Applications software takes care of programming the analog input range for you, as does the calibration program.

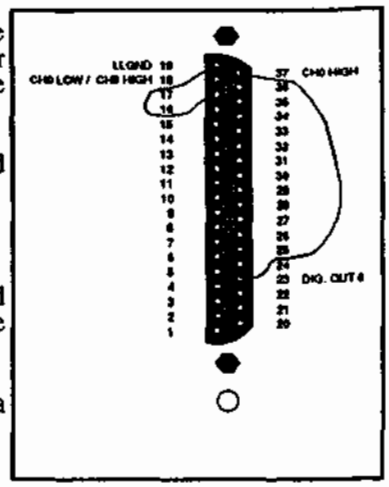
For those wishing more information on programming the analog inputs, examine the section of register base+11 and the section on BASIC programming.

4.8 SIGNAL CONNECTION TO TEST THE INSTALLATION

To test the installation of the CIO-AD16JR-AT there must be electrical signals for it to read and display. You can supply the signal with a function generator or other voltage source or you can loop-back the CIO-AD16JR-AT output signals onto the CIO-AD16JR-AT inputs.

Please, before you connect a voltage from a signal generator or other source, be sure that the signal does not exceed +/- 10V, the maximum analog input range of the board, or 0-5V, the maximum digital input range. The analog inputs are protected to 30V but why prove it.

Supplied with the CIO-AD16JR-AT are some wires with plugs on each end. These are intended to be plugged directly on the CIO-AD16JR-AT 37 pin connector for loop-back testing. If you purchased a screw terminal then connect it up and use it instead[5].



To test the installation of the CIO-AD16JR-AT the digital output 0 will be looped back into the A/D 0 input.

If you are using a signal generator feed the signal directly into A/D channel 0.

The CIO-AD16JR-AT is ready to test. You can try running the software supplied with your board now, or you can continue reading the next two sections on Software Installation and Calibration.

The program IOTEST will display the loopback Digital output into the A/D on a strip chart on the screen.

5 The CIO-TERMINAL and CIO-MINITERM are fully explained in the appendix. Briefly, the connector numbers on the screw terminals correspond to the numbers on the CIO-AD16JR-AT 37 pin connector.

5 SOFTWARE INSTALLATION

There are 3 types of software supplied with the CIO-AD16JR-AT. A BASIC CALL routine and examples, including calibration and test programs are on the disks marked CIO-AD16JR-AT UTILITY PROGRAMS. A slick acquisition and logging program is on the disk marked LABLOGII. Labtech ACQUIRE is on the disk marked FREESOFT.

All of these programs may be installed in the same directory.

You have already created that directory using the name CIO16 or one of your choosing, and installed the CIO-AD16JR-AT UTILITY software in it.

Install the remaining software by copying the contents of both disks into that directory.

5.1 UTILITY SOFTWARE

IOTEST	.EXE	Loop-back graphic display test program.
CAL16JR	.EXE	Graphic calibration and test program.
EX##	.BAS	BASIC Example programs where ## corresponds to a mode #.
CIO16	.BIN	BASIC CALL Routine.
CIO16	.OBJ	BASIC Library for use with Quick BASIC.
CIO16	.LIB	QUICK BASIC compiler library files.
CIO16	.QLB	QUICK BASIC run-time library.
QBEX##	.BAS	QUICK BASIC examples.

5.1.1 IOTEST.EXE

IOTEST exercises the CIO-AD16JR-AT by stimulating the digital output ports and reading the analog and digital inputs. The inputs are displayed on the screen on a line graph.

Program setup is easy. The program was written for the CIO-AD16JR-AT and is low speed only, as a result only the base address and analog input range need be supplied.

IOTEST can provide one output and read one input at a time. The activity on an output channel is fixed by the program. Digital outputs produce a square wave. Counter outputs produce a square wave.

5.1.2 CAL16JR

CAL16JR.EXE is a modified version of the calibration and test procedure used at the factory. The differences lie in the steps required to complete a calibration procedure. The factory uses a test fixture which, under program control, sequences calibration signals onto the inputs. CAL16JR does not autosequence but is still fast and easy to use.

This program exercises all the features of the CIO-AD16JR-AT and is explained fully in the section on Calibration and Test.

5.1.3 CIO16 CALL ROUTINE & EXAMPLES

The CIO16 CALL routine is of interest only to those who intend to write programs in BASIC. It and the example programs, all written in BASIC are covered in depth in the section on BASIC programming.

5.2 LABLOG II

Lablog II is a pull-down menu controlled program for data acquisition, logging and alarming. It acquires both analog and digital signals and has six different types of real time display.

Lablog II is an input only program. It is supplied free with every board through the generosity of Quinn-Curtis Software. This program is available with output for control applications; see the data sheet on EG-Control ($495). Quinn-Curtis also produces a number of other software tools, all of which are detailed on product promotion screens which are menu choices from Lablog II.

You installed Lablog II when the contents of the Lablog II disk were copied into the CIO16 directory of your hard disk, if not, do so now.

Lablog II is so easy to use that you will probably never need a manual. To run the program just type LABLOG2 at the DOS prompt. There are many context-sensitive help screens which will guide you as you experiment. If you do want to read a manual, the help screens are stored in a file on the Lablog II disk in a .DOC file. Print it out using the DOS command COPY LABLOG2.HLP PRN: or edit it with your word processor then print it.

Lablog2 Hint: If you get lost, press the F10 key to get back to the main menu.

NOTE: If you feel you must have a printed manual for Lablog2, you may purchase CONTROL-EG, which is what Lablog2 was derived from and includes a complete manual.

5.3 ACQUIRE

There is a complete manual on disk. Please print it out and use it to learn about acquire.

One quick note. To select the board type for ACQUIRE, select Options, place the cursor on the Hardware Interface and press F1. Place the cursor on DAS-16 and hit enter, then you can change the base address.

6 CALIBRATION AND TEST

The CIO-AD16JR-AT is supplied with software for calibration and test.

6.1 FREQUENCY OF CALIBRATION

Every board was fully tested and calibrated before being placed in finished goods inventory at the factory. *Your brand new CIO-AD16JR-AT should not require calibration before initial use*. For normal environments a calibration interval of 6 months to one year is recommended. If frequent variations in temperature or humidity are common then recalibrate at least once every three months. It takes less than 30 minutes to calibrate the CIO-AD16JR-AT.

6.2 REQUIRED EQUIPMENT

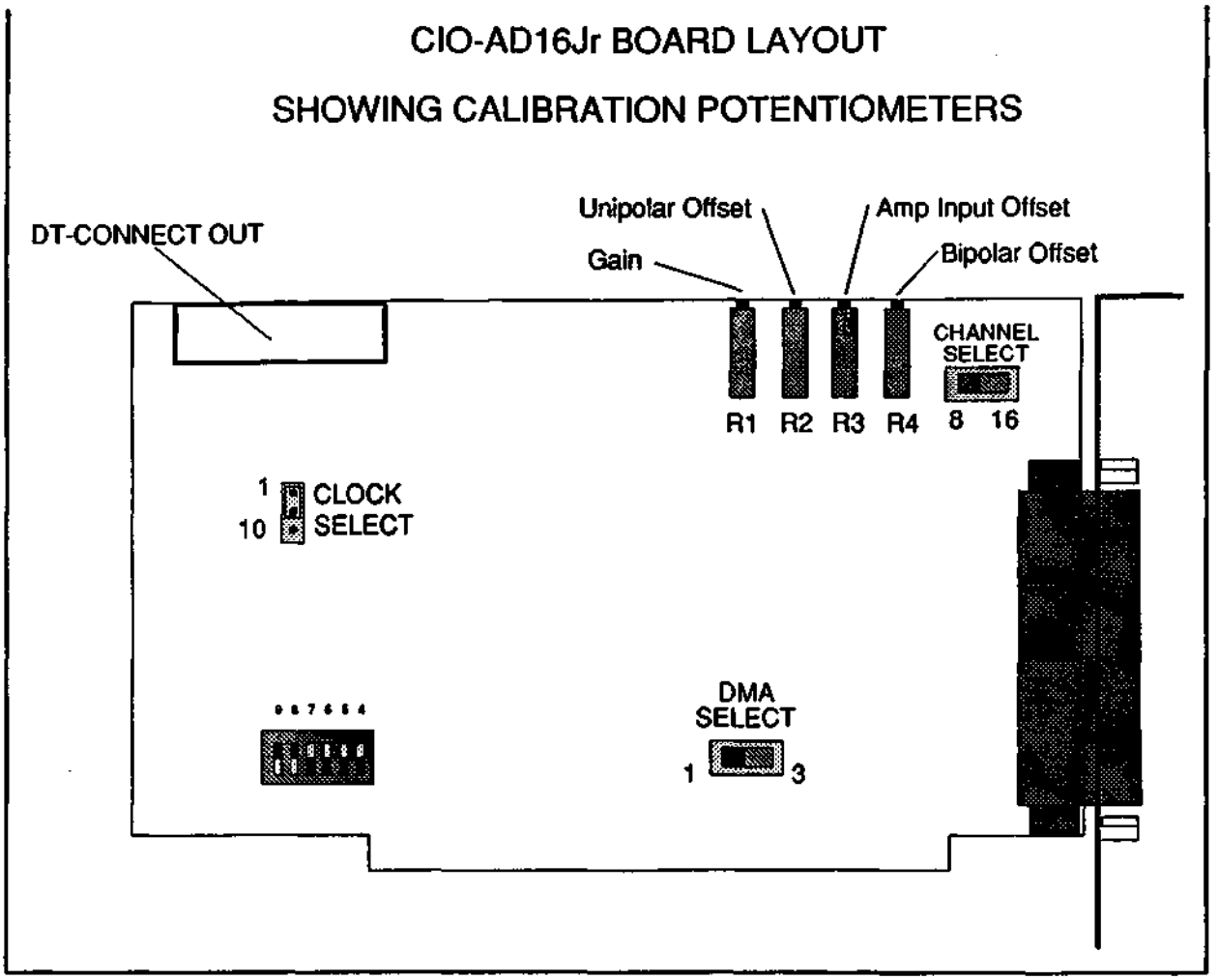
Ideally, you will need a precision voltage source, a calculator and some pieces of wire. If you do not have a precision voltage source, you will need a non-precision source and have to make a few calculations.

You will not need an extender card to calibrate the board but you will need to have the cover off you computer with the power on, so trimpots can be adjusted during calibration. For that reason a plastic screwdriver has been supplied with your CIO-AD16JR-AT. In the event that the screwdriver is dropped into the PC, no damage will result from short circuits.

6.3 CALIBRATING THE ANALOG INPUTS

The A/D is calibrated by applying a known voltage to any analog input channel and adjusting trimpots for gain, amplifier input offset, bipolar offset and unipolar offset. There are four trimpots requiring adjustment to calibrate the analog input section of the CIO-AD16JR-AT.

The CIO-AD16JR-AT should be calibrated for the range you intend to use it in. When the range is changed, slight variation in Zero and Full Scale may result. These variations can be measured and removed in software if desired.



CALIBRATION PROCEDURE

The entire calibration procedure is provide in the program CAL16JR.EXE which is on the CIO-AD16JR-AT utility disk. Please refer to the automated procedure if you want to calibrate your CIO-AD16JR-AT. The calibration procedure provided here is informational only.

BIPOLAR OFFSET ADJUSTMENT

1) Set the gain register to the bipolar range which is likely to be used most often.

2) Apply 0V + 1/2 LSB of the specified range to the channel 0 input.

3) Adjust R4, bipolar offset pot, until A/D reads 0/1 counts.

UNIPOLAR OFFSET ADJUSTMENT

1) Set the gain register to the unipolar range which is likely to be used most often.

2) Apply 0V + 1/2 LSB of the specified range to channel 0 input.

3) Adjsut R2, unipolar offset, until A/D reads 0/1 counts.

AMPLIFIER INPUT ADJUSTMENT

1) Apply 0V to the channel 0 input. Shorting cahnnel 0 to LLGND (pin 37 to pin 19) is ideal.

2) Set the gain register to the ±0.625 volt range (code = 3).

3) Adjust R3 until A/D reads 0 counts.

GAIN ADJUSTMENT

1) Set the gain register to the range likely to be used most often.

2) Apply +Full Scale - 1.5 * LSB to the channel 0 input

3) Adjust R1, gain adjust pot, until:
for unipolar, adjust to 4094/4095 counts.
for bipolar, adjust to 2046/2047 counts.

6.4 TESTING THE COUNTER/TIMER & DIGITAL I/O

There is no calibration to be performed on the counter timer, which is crystal controlled, or the digital I/O, which are TTL.

CAL16JR can test the cascaded counters 1 and 2 against the system clock to ensure that they are functioning. It can also stimulate the digital outputs and read the digital inputs. A better dynamic test of the counters and digital I/O can be performed by the IOTEST.EXE program.

7 SIGNAL CONNECTION

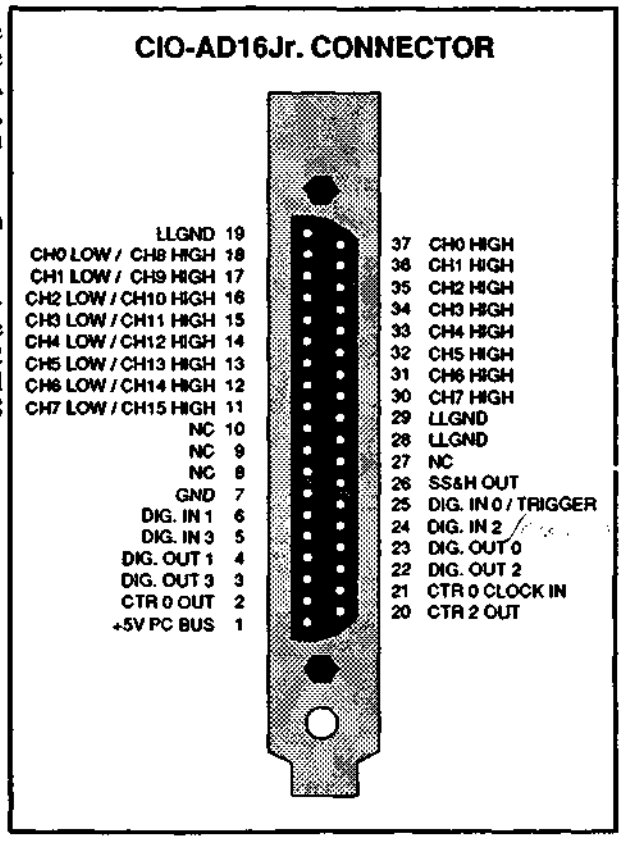
Signal connection can be one of the most challenging aspects of applying a data acquisition board. If you are an *Analog* Electrical Engineer then this section is not for you, but if you are like most PC data acquisition people these simple examples should get you started. The objective here is to present 'how to connect' common signals while avoiding discussion of electrical theory and special symbols. If you need a connection we do not show, or invent one, please call and tell us so we can add it to this manual.

7.1 CONNECTOR DIAGRAM

The CIO-AD16JR-AT analog connector is a 37 pin D type connector accessible from the rear of the PC through the expansion backplate. With the exception of the missing D/A signals, the signals available are identical to the DAS-16, an additional signal, SS&H OUT, may be accessed at pin 26.

The connector accepts female 37 D type connectors, such as those on the C73FF-2, 2 foot cable with connectors.

If frequent changes to signal connections or signal conditioning is required, please refer to the information on the CIO-TERMINAL screw terminal board, CIO-MUXTC32 32 channels analog MUX/AMP, CIO-SSH-16 16 channel simultaneous sample & hold board or the ISO-RACK08 (and CMUXAD16-10 cable) 5B module interface rack.



7.2 ANALOG INPUTS

Analog inputs to the CIO-AD16JR-AT may be connected in three different configurations. In order of complexity, these are Single Ended, Floating Differential and Differential.

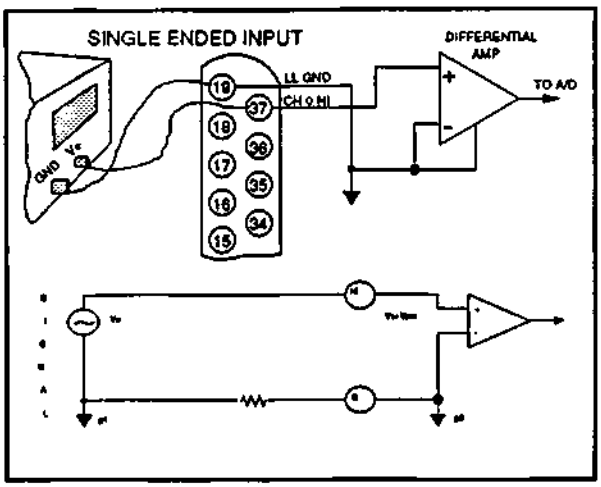
WARNING - PLEASE READ

Here is a good tip. Measure the voltage potential (difference) between the ground signal at the signal source and the PC. Use a volt meter and place the red probe on the PC ground and the black probe on the signal ground. If there is a difference of more the 10 volts, do not connect the CIO-AD16JR-AT to this signal source because you will not be able to make any reading. If the difference is more than 30 volts, DO NOT connect this signal to the CIO-AD16JR-AT because it will damage the board and possibly the computer.

7.2.1 SINGLE ENDED

A single ended input is two wires connected to the CIO-AD16JR-AT; a channel high (CH# HI) and a Low Level Ground (LLGND). The LLGND signal *must be the same ground the PC is on.* The CH# HI is the voltage signal source.

Examples are:

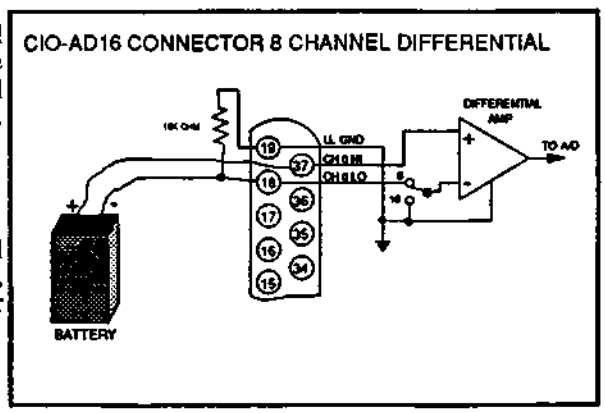


7.2.2 FLOATING DIFFERENTIAL

A floating differential input is two wires from the signal source and a 10K ground reference resistor installed at the CIO-AD16JR-AT input. The two signals from the signal source are Signal High (CH# HI) and Signal Low (CH# LO).

The reference resistor is connected between the CIO-AD16JR-AT CH# LO and LLGND pins.

A floating differential hookup is handy when the signal source is floating with respect to ground, such as a battery, 4-20mA transmitter or and the lead lengths are long or subject to EMI interference.



The floating differential input will reject up to 10V of EMI energy on the signal wires.

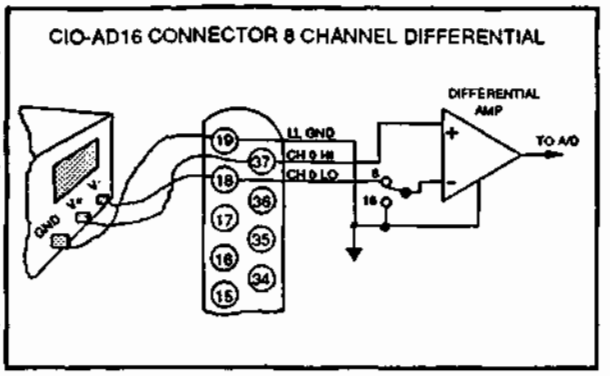
WARNING! Is the signal source really floating? Check it with a voltmeter before risking the CIO-AD16JR-AT and PC!

7.2.3 DIFFERENTIAL

A differential signal is three wires from the signal source. The signals are Signal High (CH# HI), Signal Low (CH# LO) and Signal Ground (LLGND).

A differential connection allows you to connect the CIO-AD16JR-AT to a signal source with a ground that is different than the PC ground, but less than 10V difference, and still make a true measurement of the signal between CH# HI and CH# LO.

EXAMPLE:
A laboratory instrument with its own wall plug. There are sometimes differences in wall GND between outlets.



7.3 DIGITAL OUTPUTS & INPUTS

All the digital outputs inputs on the CIO-AD16JR-AT are TTL level. TTL is an electronics industry term, short for Transistor Transistor Logic, with describes a standard for digital signals which are either at 0V or 5V. The binary logic inside the PC is all TTL or LSTTL (Low power Schotky TTL). Both specifications are presented here.

Digital signal connections are common and easy to understand when the simple rules presented in the section on analog electronics are applied.

8 BASIC PROGRAMMING

Every Compatible I/O Series board is supplied with a CALL routine which provides programmers access to the board's functions. A CALL routine is an extension of a programming language. Following the syntax of the CALL routine, instructions with specific functions may be added to the programming language, simplifying the use of products such as data acquisition boards.

The CALL routine explained here is written in assembly language and is intended for use with BASICA and all its compatible variants as well as Microsoft's compiled Quick Basic.

The following explanations are limited to the specific syntax of the CALL routine and its modes. For detailed examples of the CALL routine in BASIC programs, examine the various EX##.BAS files on the CIO-AD16JR-AT Utility Disk. The ## in EX##.BAS corresponds to the mode number the example program is designed to explain.

8.1 SYNTAX OF THE EXAMPLES

The syntax of the examples is as follows:

MODE 50: EASY TO USE A/D TO AN ARRAY

Take N samples from channels # through # at rate of R samples per second. Return the data to a BASIC array. All parameters are specified in engineering units so no calculations are necessary and the data is passed directly into a BASIC array.

```
70 DIM ADDATA%(24000)          ; Array for analog data.
80 DIM CHDATA%(24000)          ; Array for channel tags.

100 MD% = 50                   ; Set mode to 50.
110 D%(0) = 0                  ; Lower scan limit is Ch0.
120 D%(1) = 3                  ; Upper scan limit is Ch3.
130 D%(2) = 50                 ; Speed in KHz.
140 D%(3) = 0                  ; Speed in hundreds of Hz.
150 D%(4) = 24000              ; Number of samples to take.
160 D%(5) = VARPTR(ADDATA%(0)) ; Array pointer for data.
170 D%(6) = VARPTR(CHDATA%(0)) ; Array pointer for channel tags.
180 D%(7) = 1                  ; Gate from DIGITAL INPUT 0.
190 CALL CIO16(MD%, D%(0), F%) ; Execute the CALL.
```

Each mode is explained by example showing all the setup lines required to complete a successful CALL. Any special conditions are noted and all variable are explained.

8.2 BASIC CALL STATEMENT GUIDELINES - INTERPRETERS

This section applies only to interpreters. Do not refer to these instructions when using compilers, such as QUICK BASIC!

The BASIC statement 'CALL' transfers control of the processor from BASIC to a machine language program. The CALL statement passes data to the machine language program and the machine language program can pass data back to BASIC.

To successfully transfer control from BASIC to the machine language program, the CALL statement must pass the processor the starting memory address of the machine language program. 80X86 processors address memory using two 16 bit address registers which are summed to create a memory address equal to the processors address bus width. These two registers are the SEGMENT and OFFSET registers[6]. For program addresses, the registers are known as code segment (CS) and offset.

---

6 For a complete explanation of the addressing mechanism of the iAPX86 family, an Intel data book or any of the popular books on the subject should be read. You need not know the 'how to' as long as the guidelines here are followed.

The memory address that the machine language CALL program is loaded at must be declared. BASIC provides a way to specify both the CS and offset. The code segment is declared with the command 'DEF SEG = CIO-AD16JR-AT'. The offset is specified each time the call is made in the CALL statement name.

MUST DECLARE CODE SEGMENT

```
10 DEF SEG = &H6000
20 BLOAD "CIO16.BIN", 0
```

Here the CS is set for 60000 HEX (384,000 decimal). The offset is set for 0. CS + OFF = 384,000.

The memory location the CALL is loaded into ***must be above and away from all other memory used by DOS, BASIC and all other programs.***

8.2.1 HOW TO CALCULATE SEG

The CS declared in the statement DEF SEG = &HCIO-AD16JR-AT must be chosen by you, and it is easy to calculate following these rules.

FIRST: Load BASICA or GWBASIC.

SECOND: type the command

SHELL "CHKDSK" [hit return]

The DOS CHKDSK program will tell you the total memory available and the total memory free.

THIRD: Complete this calculation:

655,360	Total available
- 458,784	Total free

196,576	Top of BASIC
+ 64,000	BASIC workspace

256,878	Last occupied memory address.

The address 256,000 decimal is 40000 HEX. The CS is automatically shifted one power of sixteen so if you declare DEF SEG = &H4000 then the Code Segment is set at &H40000 which = 256,000 decimal. An easy rule is " every 1000 in the DEF SEG equals 64K of memory.

In the example above we would want to locate the CALL above 256,878 (&H40000) so a good choice for the DEF SEG would be &H5000 (320K decimal).

8.2.2 TWO CIO-AD16JR-AT BOARDS IN ONE PC

To use two CIO-AD16JR-AT boards in on PC, both boards must be installed in the PC with different base addresses, different DMA and interrupt levels.

In addition, two complete CALL routines must be BLOADed in the BASIC program which accesses the two CIO-AD16JR-ATs. Each CALL is particular to one of the boards, and must be used to read to and write from the board it is assigned to. An example illustrates this best.

```
10 SEG1 = &H6000                    ; First segment is 60000 HEX.
20 DEF SEG = SEG1
30 BLOAD "CIO16.BIN", 0             ; Load the call for board #1.

40 SEG2 = SEG1 + 6144/16            ; Second segment at 60000H + 6K
50 DEF SEG = SEG2
60 BLOAD "CIO16.BIN", 0             ; Load the call for board #2.
```

Now call MODE 0 for each board, using different settings for each.

```
100 DEF SEG = SEG1                      ; The CALL must be preceded by a DEF
                                        ; SEG command to the correct segment.
110 MD% = 0
    etc.
150 CALL CIO16(MD%, D%(0), F%)          ; Execute the CALL to board #1.
200 DEF SEG = SEG2                      ; Set segment for CALL to board #2.
210 MD% = 0
    etc.
250 CALL CIO16(MD%, D%(0), F%)          ; Execute the CALL to board #2.
```

8.3 BASIC CALL STATEMENT GUIDELINES - COMPILERS

QUICK BASIC is a great environment for developing programs. The editor and debugger are easy to use and powerful, and best of all, Quick BASIC programs *run fast!*

The CIO16 CALL routine is easy to use with Quick BASIC and there are example programs on the CIO-AD16JR-AT utility disk. These are labeled QBEX##.BAS and are designed to be run from Quick BASIC.

8.3.1 HOW TO LOAD QUICK BASIC AND CIO16

When Quick BASIC is loaded from DOS, you must instruct it to include the CIO16 CALL program. This is done by typing:

C> QB/L CIO16 [hit return]

The /L tells Quick BASIC to link the CIO16.QLB file into Quick BASIC.

NOTE ON QUICK BASIC

There are example programs on the CIO-AD16JR-AT utility disk which show how to load and CALL the CIO16 Quick BASIC CALL. The procedure for Quick BASIC is unlike that for GWBASIC. The examples in this manual are all of the GWBASIC syntax and will not run if used in Quick BASIC.

PLEASE PRINT OUT AT LEAST ONE QUICK BASIC EXAMPLE PROGRAM AND READ IT IF YOU ARE A QUICK BASIC PROGRAMMER.

8.4 CIO16 CALL FORMAT RULES

The CALL routine has several format conventions which must be followed with compiled and interpreted BASIC.

THE CALL PARAMETERS ARE POSITIONAL

The CALL program expects variables in a given order and interprets the information in the variable based on that order. Names of variable mean nothing to the CALL.

The syntax of the CALL is:

CALL CIO16(MD%, D%(0), F%)

If you were to type the CALL as:

CALL CIO16(D%(0), MD%, F%)

The CALL routine would take the data in D%(0) as the mode number, etc.

THE CALL ROUTINE ACCEPTS AND RETURN INTEGER VALUES ONLY

The % on the variable MD% signifies that the data in MD% is an integer. If the CALL is passed a non-integer variable, it will not function.

NO CALCULATIONS ARE ALLOWED INSIDE THE BRACKETS

CALL CIO16(MD% + 2, D%(0), F%) will not work.

CONSTANTS CANNOT REPLACE VARIABLES INSIDE THE BRACKETS

CALL CIO16(50, D%(), F%) will not work.

8.5 MODE SUMMARY

MODE	FUNCTION	SPEED
0	Initialize CIO16 CALL routine. Provide base address, interrupt level & DMA.	
1	Set multiplexer scan limits as start channel and stop channel.	
2	Read multiplexer current channel and scan limits.	
3	Do one A/D and increment the MUX address.	Slow Foreground
4	Perform N, A/D conversions after trigger. The A/D rate is based on the on-board pacer clock (8253 channels 1 & 2) or by external pacer. The data is transferred to a BASIC array.	Medium Foreground
5	Same as MODE 4 except that the data from A/D conversion is transferred into a memory buffer by an interrupt service routine.	Medium Background
6	Perform N, A/D conversions after trigger. A/D rate is based on the on-board pacer clock or by external pacer. Data is transferred via DMA to memory.	
7	Disable the DMA and interrupt operation of modes 5,6,18,20 or 50.	
8	Check status of DMA or interrupt operation of modes 5,6,18,20 or 50.	
9	Transfer data from the memory buffer where data was stored in modes 5 and 6, into a BASIC array.	
10	Load the control register of Counter 0. Load 8254 control codes from data sheet.	
11	Load counter 0 data.	
12	Read counter 0.	
13	Set digital outputs OP0 OP1, OP2 & OP3 as one 4 bit nibble.	
14	Read 4 bit nibble of IP0, IP1, IP2, & IP3.	
15	Output a value between 0 and 4095 to D/A 0 or D/A 1.	

16	Output two values between 0 and 4095, one to each D/A.	
17	Set the on-board pacer clock rate by providing two numbers between 2 and 65,536, one to each counter. Requires calculation.	
18	On interrupt a D/A output on either channel is optionally followed by an A/D scan.	
19	Define an analog trigger level and slope and wait until it occurs.	
20	On trigger, sample analog inputs between lower and upper scan limits. Do this N times.	
40-45	**ENHANCED MODE CALLS** Mode switch must be set down.	
40	**REP INSW Foreground.** Take n samples from channel low through channel high at R samples per second (330KHZ max) using REP INSW and transfer samples directly to a BASIC array. Opperates in FOREGROUND	
41	**REP INSW Background.** Take n samples at R samples per second (330KHZ max) using REP INSW and transfer samples directly to memory. Opperates in BACKGROUND	
42	**Pre-trigger & post-trigger buffer, foreground.** Hold N samples in a pre-trigger buffer and on trigger collect M more sample in a post-trigger buffer. After collecting last sample, transfer all data directly to a BASIC array.	
43	**Pre-trigger & post-trigger buffer, background.** Hold N samples in a pre-trigger buffer and on trigger collect M more sample in a post-trigger buffer. Keep data in memory.	
44	**Transfer data from memory to array.** Transfer N samples of data acquired in modes 41 or 43 from memory to a BASIC array.	
45	**Force compatibility mode.** Forces the CIO-AD16JR-AT from enhanced mode to compatible (CIO-AD16 & DAS16) mode.	
50	**Easy to use A/D.** Take N samples from CH low to CH high at R samples per second using DMA and transfer samples directly to a BASIC array.	
51	**Pacer clock in Hz.** Set the on board pacer clock rate by specifying the rate in KHz and Hz. Calculations to load counter are done by the mode software.	
NOTE	TO SET THE PROGRAMMABLE ANALOG INPUT RANGE, WRITE TO BASE+11 USING BASIC's OUT COMMAND. SEE THE REGISTER BASE+11 FOR GAIN VALUES.	

8.6 MODE 0 INITIALIZE

Set up base address, interrupt level and DMA level. *Must be run before other modes.*

```
10 DIM D%(10)                          ; Inititalize data passing array.
20 F% = 0                              ; Inititalize error flag variable

30 MD% = 0                             ; SET MODE = 0
40 D%(0) = &H300                       ; SET BASE ADDRESS (HEX 300 HERE )
50 D%(1) = 2[7]                        ; SET INTERRUPT LEVEL (LEVEL 2 HERE)
60 D%(2) = 1[8]                        ; SET DMA LEVEL (LEVEL 1 HERE)
70 CALL CIO16(MD%, D%(0), F%[9])       ; EXECUTE CALL
```

RETURN VARIABLES
D%(0) - D%(2) — UNCHANGED

DEFAULTS

Channel Low Scan Limit	0
Channel High Scan Limit	7 in 8 Channel Differential Mode 15 in 16 Channel Single Ended Mode
Interrupts	Not Enabled
DMA	Not enabled
Pacer Clock (8254 1&2)	10 KHz

PREREQUISITE MODES — Declare D%(), F%
None. Mode 0 must be first.

IMPORTANT NOTE

The interrupt level 2 and DMA level 1 are good choices to begin with. Before using different values, please read about interrupts and DMA in the section on the architecture of the PC.

---

7 For further information on the available interrupt levels, see the section the architecture of the PC.

8 For an explanation of DMA levels and how to select one, see the section on the architecture of the PC.

9 For a list of error codes and their meaning, see the section on error codes in this chapter.

8.7 MODE 1 SET MUX LIMITS

Set the A/D channel lower and upper scan limits.

```
100 MD% = 1
110 D%(0) = 0          ; Lower scan limit.
120 D%(1) = 3          ; Upper scan limit.
130 CALL CIO16(MD%, D%(0), F%)   ; Execute the CALL.
```

RETURN VARIABLES D%(0) - D%(1)	Unchanged
DEFAULTS	After execution the MUX is set at the lower scan limit.
PREREQUISITE MODES	Declare D%(), F% Mode 0

8.8 MODE 2 READ MUX

Read A/D channel multiplexor current setting and scan limits.

```
100 MD% = 2                          ; Set mode to 2
110 CALL CIO16(MD%, D%(0), F%)
```

RETURN VARIABLES

D%(0)	Channel address for next conversion.
D%(1)	Lower scan limit.
D%(2)	Upper scan limit.

DEFAULTS — None

PERQUISITE MODES — Declare D%(), F%
Mode 0

8.9 MODE 3 DO ONE A/D

Do one A/D conversion and increment A/D channel MUX.

```
100 MD% = 3                          ; Set mode = 3
110 CALL CIO16(MD%, D%(0), F%)       ; Execute the CALL
```

RETURN VARIABLES	
D%(0)	A/D Data 0 to 4095 Unipolar -2048 to 2048 Bipolar
D%(1)	Channel number
DEFAULTS	None
PREREQUISITE MODES	Declare D%(), F% MODE 0
OPTIONAL SETUP MODES	MODE 1 To set scan limits.

This mode executes one A/D conversion per CALL. Because the timing of the conversion is uncertain, placing this mode of A/D conversion in a loop is not recommended if successive samples are to be used for spectrum analysis. Use modes 4, 5 or 6.

8.10 MODE 4 DO N A/D TO ARRAY

Perform N A/D conversions and transfer the data directly to a BASIC integer array.

```
90 DIM ADATA%(1000)                          ; Dimension the BASIC array.

100 MD% = 4                                  ; Set mode = 4
110 D%(0) = 1000                             ; Total samples to take.
120 D%(1) = VARPTR(ADATA%(0))                ; The array to store the samples in.
130 D%(2) = 1                                ; Trigger source (8254 Pacer here)
140 CALL CIO16(MD%, D%(0), F%)               ; Execute the CALL
```

RETURN VARIABLES	
D%(0) - D%(2)	Unchanged
ADATA%(0)	First sample
ADATA%(1)	Second sample
....	
ADATA%(999)	1000 the sample.
DEFAULTS	None
PREREQUISITE MODES	Dimension array for data Declare D%(), F% MODE 0
OPTIONAL SETUP MODES	MODE 1 To set scan limits. MODE 17 To set pacer clock. MODE 51 Same as Mode 17 but in Hz.

TRIGGER METHOD variable may be 0 or 1.

Trigger method 0 expects an external TTL trigger pulse on the TRIGGER 0 (pin #25 of the 37 pin connector) input. Conversions are triggered on the rising edge of the TRIGGER 0 input. The CALL routine will execute until the number of triggers specified in D%(0) have occurred. Failure to cause the required number of trigger pulses will prevent BASIC from recovering control of the processor.

Trigger method 1 provides A/D pacing via the on board 8254 pacer clock counter 1 & 2. The first conversion will occur after the level of TRIGGER 0 input is brought high (the TRIGGER 0 input is really a GATE in this mode). This input floats high so if the input is not pulled low, conversions start immediately upon executing the mode. To delay the start of A/D conversions, hold pin 25 low by grounding it with a 10K Ohm resistor. Break the connection to ground to begin conversion. Conversions, once started, cannot be stopped by grounding TRIGGER 0 input.

VARPTR COMMAND Cautions.

The VARPTR command places a pointer into the variable specified (D%(1) above). This pointer shows the CALL where to place the A/D data. *Never define new variables on the lines between the VARPTR command and the CALL.*

```
90 DIM ADATA%(1000)                          ; Dimension the BASIC array.

100 MD% = 4                                  ; Set mode = 4
110 D%(0) = 1000                             ; Total samples to take.
120 D%(1) = VARPTR(ADATA%(0))                ; The array to store the samples in.
130 D%(2) = 1                                ; Trigger source (8254 Pacer here)
140 X = 20                                   ; This lines should not be here.
150 CALL CIO16(MD%, D%(0), F%)               ; Execute the CALL
```

8.11 MODE 5 DO N A/D TO MEMORY

Do N A/D conversions and transfer the data to memory via interrupt.

```
100 MD% = 5                          ; Set mode to 5.
110 D%(0) = &H7000[10]               ; Data Segment for conversion data.
120 D%(1) = 1                        ; Trigger source (internal here).
130 D%(2) = 0                        ; One Cycle/ Re-cycle (one cycle here).
140 CALL CIO16(MD%, D%(0), F%)       ; Execute CALL.
```

200 Now set up and execute MODE 9 to move the data into an array.

RETURN VARIABLES	
D%(0) - D%(3)	Unchanged
DEFAULTS	
Interrupts	Enabled
DMA	Not enabled
PREREQUISITE MODES	Declare D%(), F% MODE 0
OPTIONAL SETUP MODES	MODE 1 To change scan limits. MODE 17 To set pacer clock. MODE 51 Same as Mode 17 but in Hz.

TRIGGER METHOD variable may be 0 or 1.

Trigger method 0 expects an external TTL trigger pulse on the TRIGGER 0 (pin #25 of the 37 pin connector) input. Conversions are triggered on the rising edge of the TRIGGER 0 input. The CALL routine will execute until the number of triggers specified in D%(0) have occurred. Failure to cause the required number of trigger pulses will prevent the CALL routine from un-installing the interrupt handler, although BASIC will execute.

Trigger method 1 provides A/D pacing via the on board 8254 pacer clock counter 1 & 2. The first conversion will occur after the level of TRIGGER 0 input is brought high (the TRIGGER 0 input is really a GATE in this mode). This input floats high so if the input is not pulled low, conversions start immediately upon executing the mode. To delay the start of A/D conversions, hold pin 25 low by grounding it with a 10K Ohm resistor. Break the connection to ground to begin conversion. Conversions, once started, cannot be stopped by grounding TRIGGER 0 input. NOTE: If pin 25 is held low perpetually, the CALL routine will never return control to BASIC

SINGLE CYCLE / RE-CYCLE

Single Cycle / Re-cycle: This mode can run once and exit after collecting the number of samples specified in D%(0) (Single Cycle Mode), or it can continuously Re-cycle.

In re-cycle mode the number samples specified in D%(0) are taken over and over again. The first sample of the cycle is always in the memory location specified in D%(1).

MODE 5 installs an interrupt handler. The interrupt handler is automatically un-installed at the termination of the Nth sample in Single Cycle Mode only. In Re-Cycle mode, the handler must be un-installed by executing MODE 7.

The maximum number of samples which may be taken in this mode are 32,000 because of the BASIC 64K data segment limit.

The maximum conversion speed achievable in this mode is dependent on the processor clock. On a PC/XT at 4.77 MHz, 4,000 samples per second is the limit. On a 386 at 20MHz that limit rises to 20,000 samples per second.

---

10 For information on segment registers and how to locate free memory to load the CALL routine and store data, see the introduction to programming at the beginning of this chapter and the section on the architecture of the PC.
When choosing a memory location to place samples in, a good, safe rule is to choose the page above the address the CALL routine was loaded at. For example, if the CALL was BLOADed at &H6000 (384,000 decimal), then set the data segment to &H7000 (448,000 decimal).

8.12 MODE 6 DO N A/D VIA DMA

Do N A/D conversions and transfer the data to memory via Direct Memory Access (DMA[11]) transfer. Conversions may be triggered by the on board pacer clock or by an external rising edge trigger. Muliple (two) CIO-AD16JR-AT boards on different DMA level may operate simultaneously using this mode.

```
100 MODE = 6                          ; Set mode = to 6
110 D%(0) = 1000                      ; The number of sample to take.
120 D%(1) = &H7000[12]                ; Data segment address for samples.
130 D%(2) = 1                         ; Trigger source (internal here)
140 D%(3) = 0                         ; Cycle/ Re-cycle (One cycle here.)
150 CALL CIO16(MD%, D%(), F%)         ; Execute the call.
```

200 Now run Mode 9 which transfers the data from memory to an array.

RETURN VARIABLES	
D%(0) - D%(3)	Unchanged
DEFAULTS	
Interrupts	Not enabled
DMA	Enabled
PREREQUISITE MODES	Declare D%(0), F% MODE 0
OPTIONAL SETUP MODES	MODE 1 Set scan limits. MODE 17 Set pacer clock. MODE 51 Set pacer clock n Hz.
WARNINGS	Do not write to EGA or VGA screens until the complete DMA cycle is done when sampling at speeds over 60 KHz.[13]

TRIGGER METHOD variable may be 0 or 1.

Trigger method 0 expects an external TTL trigger pulse on the TRIGGER 0 (pin #25 of the 37 pin connector) input. Conversions are triggered on the rising edge of the TRIGGER 0 input. The CALL routine will execute until the number of triggers specified in D%(0) have occurred. Failure to cause the required number of trigger pulses will prevent the CALL routine from un-installing the interrupt handler, although BASIC will execute.

Trigger method 1 provides A/D pacing via the on board 8254 pacer clock counter 1 & 2. The first conversion will occur after the level of TRIGGER 0 input is brought high (the TRIGGER 0 input is really a GATE in this mode). This input floats high so if the input is not pulled low, conversions start immediately upon executing the mode. To delay the start of A/D conversions, hold pin 25 low by grounding it with a 10K Ohm resistor. Break the connection to ground to begin conversion. Conversions, once started, cannot be stopped by grounding TRIGGER 0 input. NOTE: If pin 25 is held low perpetually, the CALL routine will never return control to BASIC

SINGLE CYCLE / RE-CYCLE

Single Cycle / Re-cycle: This mode can run once and exit after collecting the number of samples specified in D%(0) (Single Cycle Mode), or it can continuously Re-cycle.

In re-cycle mode the number samples specified in D%(0) are taken over and over again. The first sample of the cycle is always in the memory location specified in D%(1).

---

11 The PC's DMA controller manages the high speed transfer of data between memory and I/O devices without tying up the CPU.

12 When choosing a memory location to place samples in, a good, safe rule is to choose the page above the address the CALL routine was loaded at. For example, if the CALL was BLOADed at &H6000 (384,000 decimal), then set the data segment to &H7000 (448,000 decimal).

13 This resetriction applies only to EGA and VGA screens because those screens employ the Non Maskable Interrupt (NMI) signal on the bus. This has priority over DMA transfers and will cause the DMA controller to miss a transfer request. The result will be bad channel tags, bad data or both. MetraByte's example program has this error in programming in it. Do not follow that example.

MODE 6 initializes and initiates DMA operation of the 8237 DMA controller. The DMA channel is automatically disabled at the termination of the Nth sample in Single Cycle Mode only. In Re-Cycle mode, the DMA channel must be disabled by executing MODE 7.

The maximum number of samples which may be taken in this mode are 32,000 because of the BASIC 64K data segment limit.

The maximum conversion speed achievable in this mode is dependent on the throughput of the A/D section of the board. The main limiting factor being the throughput of the A/D chip. Two versions of the CIO-AD16JR-AT, the CIO-AD16JR-AT/50K (50 KHz max S/S) and CIO-AD16JR-AT/100K (100 KHz max S/S) are available. On a PC/XT at 4.77 MHz, 100,000 samples per second with a CIO-AD16JR-AT/100K is the limit.

On AT class machines with processor clocks slower than 10MHz, the speed of the DMA controller chip will limit the speed of the system to less than 100KHz. The reason is that the AT DMA clock is 1/2 the speed of the CPU clock, therefore an 8 MHz AT has a DMA clock of 4MHz, which is 770Khz less than an XT!

8.13 MODE 7 DISABLE INTERRUPT/ DMA

Immediately disables any running interrupt or DMA operation. Should be run after MODES 5, 6, 18, or 20 when re-cycle was chosen.

```
100 MD% = 7                          ; Set mode to 7
110 CALL CIO16(MD%, D%(), F%)        ; Execute the call
```

RETURN VARIABLES	None
DEFAULTS	Turns off all IR & DMA.
PREREQUISITE MODES	Declare D%(), F% MODE 1

8.14 MODE 8 READ INTERRUPT/ DMA

Monitor the status of background interrupt and DMA operations.

```
100 MD% = 8                          ; Set mode = 8.
110 CALL CIO16(MD%, D%(), F%)        ; Execute the call.
```

RETURN VARIABLES	
D%(0)	0 = No background operation enabled. 1 = DMA mode 6 enabled. 2 = Interrupt, mode 5. 3 = Interrupt, mode 18. 4 = Interrupt, mode 20.
D%(1)	0 = Enabled operation is completed. 1 = Enabled operation is active.
D%(2)	Number of conversions completed.
DEFAULTS	None
PREREQUISITE MODES	Declare D%(), F% MODE 0

8.15 MODE 9 TRANSFER DATA TO ARRAY

Transfer A/D data gathered in mode 5 or 6 from memory into two BASIC arrays. One array holds conversion data, the other holds the channel tags.

```
80 DIM ADATA%(1000)[14]          ; Dimension the array for A/D data.
90 DIM CHTAG%(1000)              ; Dimension the channel tag array.

100 MD% = 9                      ; Set mode = 9.
110 D%(0) = 1000                 ; Number of samples to transfer.
120 D%(1) = &H7000               ; Data segment used in mode 5 or 6.
130 D%(2) = 0                    ; Conversion to put in ADATA%(0).
140 D%(3) = VARPTR(ADATA%(0))    ; Array pointer for A/D data.
150 D%(4) = VARPTR(CHTAG%(0))[15] ; Array pointer for channel tag.
160 CALL CIO16(MD%, D%(0), F%)   ; Execute the CALL.
```

RETURN VARIABLES	
D%(0) - D%(4)	Unchanged
ADATA%(0) - ADATA%(999)	Conversions 1 - 1000
CHTAG%(0) - CHTAG%(999)	Channel tags 1 - 1000
DEFAULTS	None
PREREQUISITE MODES	Dimension ADATA%() and CHTAG%() Declare D%(), F% MODE 0

NUMBER OF SAMPLES TO TRANSFER

The number of samples to transfer may be any number less than the dimension of the array ADATA%(). A number greater than the dimension of the array will cause overwrites of other variables, and may crash BASIC.

SOURCE SEGMENT IN MEMORY

The source segment to transfer data from may be any segment you choose. It only makes sense that it would be the segment data was transfered to using modes 5 or 6.

CONVERSION TO PUT IN ADATA%(0)

Mode 9 can offset the data within the data segment. If you wanted the first element of the array ADATA%(0) to be the 100th sample acquired by MODE 6, then set D%(2) = 100.

FORMAT OF THE A/D DATA

The data in array ADATA%() is in the form:

0 to 4095	Unipolar data.
-2048 to +2048	Bipolar data.

The position of the UNIPOLAR/BIPOLAR switch can be determined by reading the status of the U/B bit of the status register, and MODE 9 does this before transferring the data. For this reason, *do not change the position of the UNI/BIP switch between running MODEs 5 or 6 and MODE 9.*

FORMAT OF THE CHANNEL TAG DATA

The data in the channel tag array is in the form of a channel number, 0 to 15. The channel tag is derived from the channel MUX address as each sample is acquired and is stored in memory with the A/D sample. It is not calculated by MODE 9.

---

14 The variable names ADATA% and CHTAG% are chosen for clarity of this example only. Choose any variable names you wish, that BASIC will allow.

15 Assign no new variables after the VARPTR command and before the CALL. See MODE 4 for a complete explanation.

8.16 MODE 10 CONFIGURE COUNTER 0

Program the control register of 8254 COUNTER 0 in one of 6 configurations.

```
100 MD% = 10                          ; Set mode to 10.
110 D%(0) = 2                         ; Set CTR 0 configuration (rate here).
120 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
```

RETURN VARIABLES — None

DEFAULTS — CTR 0 source is internal.

PREREQUISITE MODES — Declare D%(), F%
MODE 0

COUNTER 0 CONFIGURATIONS

0	OUT 0 transitions from low to high on terminal count and remains high until reset.
1	OUT 0 transitions from high to low to high (pulse) for one clock cycle on terminal count.
2	OUT 0 pulses low for one count on terminal count and reloads the counter and re-cycles.
3	OUT 0 is high for count < 1/2 terminal count then low until terminal count, whereupon it recycles. This mode generates a square wave.
4	OUT 0 pulses low for one clock cycle on terminal count. Count starts after counter is loaded.
5	OUT 0 pulses low for one clock cycle on terminal count. Count starts on rising edge at GATE 0 input.

COUNTER 0 SOURCE

Using MODEs 10 (set counter configuration) and 11 (load counter 0 count register) do not provide a means of determining the *source* of the clock pulses that decrement the count register.

There are two sources of pulses into the CLK 0 input of counter 0, they are pin 21 of the 37 pin connector and the on-board frequency generator. The frequency of the on-board CLK 0 source is either 1MHz or 100KHz, and is determined by the XTAL jumper[16].

16 The XTAL jumper is an artifact of pre-1986 DAS-16 boards. When MBC revised the DAS-16 in 1986 they changed the on-board oscillator frequency from 1MHz to 10MHz, and provided a 1/10 chip and jumper. All CIO-AD16JR-AT boards are shipped with this jumper in the 1 MHz position, and it should not be changed except for the purpose stated here.

8.17 MODE 11 LOAD COUNTER 0

Load the CTR 0 count down register with a total count value between 2 and 65,535.

```
100 MD% = 11                          ; Set mode to 11
110 D%(0) = 20000                     ; Count value for count register 0
120 CALL CIO16(MD%, D%(), F%)         ; Execute the CALL.
```

RETURN VARIABLES — None

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

OPTIONAL SETUP MODES — MODE 10
(If mode 10 has not been run, counter 0 control is in configuration 0)

CALCULATING THE CORRECT FOR D%(0)

Integer values can only be -32,767 <= D%(0) <= 32,767 yet the counter will accept a count down value from 0 to 65,535. This creates a small problem in formatting count down values greater than 32,767. The following code sample, leading the CALL to MODE 11 will format the data correctly for you.

```
90 COUNT% = 50000 - 65536             ; Offsets count value for integer
                                      ; COUNT% = -15,536

100 MD% = 11                          ; Set mode to 11
110 D%(0) = COUNT%                    ; Count value for count register 0
120 CALL CIO16(MD%, D%(), F%)         ; Execute the CALL.
```

To load a desired count greater than 32,767 subtract 65,536 from the desired count and load the result into D%(0).

8.18 MODE 12 READ COUNTER 0

```
100 MD% = 12                          ; Set mode = 12
110 D%(0) = 1                         ; Read CTR 0 'on the fly'.
120 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL
```

RETURN VARIABLES

D%(0)	Unchanged
D%(1)	CTR 0 data.

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

CTR 0 READ OPTIONS

There are two ways to read the CTR register on an 8254 counter/ timer chip, they are static and 'on the fly'.

Set D%(0) = 0 for a static read. This read should be done only when no clock input pulses are being received by the CTR 0 count register.

Set D%(0) = 1 for read 'on the fly'. This read may be executed at any time.

FORMAT OF THE CTR 0 DATA

Data read back from the CTR 0 count register by MODE 12 is in integer form and so has the same format which count data needs to be in when loading the count register in MODE 11.

For CTR 0 reads greater than 32,767, the read data must be subtracted from 65,536. For example, if the data in D%(1) = -12,482, then the counts in CTR 0 are equal to 65,536 - 12,482 = 53,054.

Remember when writing the line of code to calculate this value that the data in D%(1) is negatively signed, *and should be added to 65,536.*

```
100 MD% = 12                          ; Set mode = 12
110 D%(0) = 1                         ; Read CTR 0 'on the fly'.
120 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL

130 IF D%(1) > 32767 THEN COUNT = 65,536 + D%(1) ELSE COUNT = D%(1)
```

8.19 MODE 13 WRITE DIGITAL OP0-3

Write data to the digital output lines OP0, OP1, OP2, & OP3, as one decimal coded (0 - 15) nibble.

```
100 MD% = 13                          ; Set mode = 13.
110 D%(0) = 2                         ; Set OP1 high, OP0, OP2, & OP3 low.
120 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
```

RETURN VARIABLES — None

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

FORMAT OF THE DATA

The data written to OP0 - OP3 is a decimal number between 0 and 15. The number is constructed by determining which bits should be high and low, and assigning a 0 or 1 to the bit position, for example:

To set OP0 and OP3 high, and set OP1, and OP3 low.

OP0 = 1 position	1
OP1 = 2 position	+ 0
OP2 = 4 position	+ 4
OP3 = 8 position	+ 0

Data for D%(0) in CALL.	5

Frankly, this is quite awkward. In addition to calculating the correct number for D%(0), it must be kept in mind that all the OP# lines are set with each CALL to MODE 13. If one of the OP# lines was previously set and you want it to retain its previous level, MODE 14 (read digital bits) must first be executed to see what that state was! *THERE IS A BETTER WAY.* Use the 24 digital lines on the on-board 8255 (PIO-12) and program it with MODE 55 - 60.

We recommend that digital lines OP0 to OP3 not be used for digital I/O. These lines perform double duty as the External MUX Control Lines when using a MUX-32 or EXP-16.

8.20 MODE 14 READ DIGITAL IP0-3

Read the high / low status of digital inputs IP0, IP1, IP2, & IP3 as a decimal number between 0 & 15.

```
100 MD% = 14                          ; Set mode to 14.
110 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
```

RETURN VARIABLES
D%(0) — IP0 - IP3 data

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

The data is returned in the same format as the data must be written in MODE 13.

We recommend that IP0 - IP3 not be used as general purpose digital I/O lines as the also serve as TRIGGER (IP0) and GATE 0 (IP2) inputs.

Use the on-board 8255 instead and read it with MODE 55 - 60.

8.21 MODE 17 LOAD A/D PACER CLOCK CTR 1&2

Load CTR1 & CTR2 count down register with data.

```
100 MD% = 17                              ; Set mode = 17.
110 D%(0) = 10                            ; Value for CTR 2.
120 D%(1) = 1000                          ; Value for CTR1.
130 CALL CIO16(MD%, D%(0), F%)            ; Execute the CALL.
```

RETURN VARIABLES — None

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

FORMAT OF THE DATA

The data loaded into CTR1 & CTR2 form a two stage divider of the input to CTR2. The rate is determined by one of two equations.

$$RATE = \frac{1{,}000{,}000}{(CTR2 * CTR1)}$$ When the XTAL jumper is set for 1 MHz, which is the factory default.

$$RATE = \frac{10{,}000{,}000}{(CTR2 * CTR1)}$$ When the XTAL jumper is set for 10 MHZ

In the example above, CTR2 = 10 and CTR 1 = 1000.

$$RATE = \frac{10{,}000{,}000}{(10 * 1000)} = 1000, \text{ or } 1KHz.$$

The A/D pacer clock is set at 1KHz. This is the pacer clock used in MODEs 4,5 & 6.

This mode is cumbersome to use since the only useful purpose of setting CTR2 & CTR1 is to pace the A/D conversions, which most of us think of in Samples/Second or Hertz (Hz). An easier to use MODE is MODE 51 since it accepts data in Hz and KHz and does the calculation for you.

8.22 MODE 19 ANALOG TRIGGER

Hold off further execution until analog value V with slope S is present on channel C.

```
100 MD% = 19                          ; Set mode to 19.
110 D%(0) = 5                         ; Channel # to trigger from.
120 D%(1) = 410                       ; Bit value to trigger on.
130 D%(2) = 0                         ; Positive slope.
140 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
```

RETURN VARIABLES — None

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

FORMAT OF THE DATA

D%(0) = the channel number. 0-7 Differential, 0-15 Single ended.

D%(1) = the bit value of the trigger level. You must calculate this value using one of these two formulas:

For Bipolar Inputs: ((Vt * 4095) / Vfs) - 2048

For Unipolar Inputs: (Vt * 4095) / Vfs

Where — Vt = Trigger Voltage
Vfs = Full Scale Volts.

D%(1) = Slope of the signal. 0 = positive, 1 = negative. MODE 19 is a slope dependent analog trigger. That means that if the Vin is above the Vtrig and the slope is positive, the trigger will not occur until the Vin falls below Vtrig and then rises to Vtrig.

TIMING

This MODE puts the CPU into a tight loop, poling the analog input and comparing its value and slope to the trigger value and slope. The analog input is not latched, so it is possible that a valid trigger value, as part of a fast pulse, could come and go between readings. For this reason we recommend a 1 millisecond signal

8.23 MODE 20 A/D SCAN ON INTERRUPT

On trigger, sample analog channels between the lower scan limit and upper scan limit as specified in MODE 1 then wait for the next trigger. When the trigger comes, sample the channels. Repeat this until N total samples have been taken and stop or optionally re-cycle.

```
100 MD% = 20                          ; Set mode to 20.
110 D%(0) = 1000                      ; Take 1000 samples
120 D%(1) = &H7000                    ; Segment for data is 70000 Hex.
130 D%(2) = 0                         ; External trigger source.
140 D%(3) = 0                         ; One cycle only.
150 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
```

RETURN VARIABLES	None
DEFAULTS	None
PREREQUISITE MODES	Declare D%(), F% MODE 0 MODE 1
OPTIONAL MODES	MODE 17 set pacer clock

***NOTE:** Before using this MODE, please read about MODEs 1, 4, 5 and 6.*

FORMAT OF THE DATA

D%(0) = Number of total conversions. 1 <= N <= 32,767. This number is the total conversions, not the number of scans. The number of scans is equal to total conversions divided by the number channels specified in MODE 1.

D%(1) = Memory segment to receive data. See MODE 4.

D%(2) = External / Internal sample pacer.

If D%(2) = 0 then each scan is initiated by a high pulse on DIGITAL INPUT 0 (pin 25). The time between samples is machine dependent. On a 4.77 MHz PC it is about 100uS.

If D%(2) = 1 then each scan is initiated by a pulse from the on board pacer clock. Pulses from the pacer clock (see MODE 17) should not be faster than the channel to channel time (100 uS on a PC), *times*, the number of samples per scan. The first scan may be held off by holding DIGITAL INPUT 0 low. Once it goes high further transitions have no effect.

D%(3) = One Cycle or Re-cycle.

If D%(3) = 0 then one cycle of the total number of samples from D%(0) is taken then the interrupts are cleared.

If D%(3) = 1 the mode continually re-cycles. MODE 7 should be used to shut off interrupts when the program is through with MODE 20.

8.24 MODE 40 330KHz FOREGROUND A/D

Take N samples from channels Lower Scan Limit through Upper Scan Limit at R samples per second directly to memory using the REP INS/W command, then transfer to a BASIC ARRAY after all samples are taken. Optionally trigger externally.

```
70 DIM ADDATA%(24000)          ; Array for analog data.
80 DIM CHDATA%(24000)          ; Array for channel tags.

100 MD% = 40                   ; Set mode to 40.
110 D%(0) = 0                  ; Lower scan limit is Ch0.
120 D%(1) = 3                  ; Upper scan limit is Ch3.
130 D%(2) = 50                 ; Speed in KHz.
140 D%(3) = 0                  ; Speed in hundreds of Hz.
150 D%(4) = 24000              ; Number of samples to take.
160 D%(5) = VARPTR(ADDATA%(0)) ; Array pointer for data.
170 D%(6) = VARPTR(CHDATA%(0)) ; Array pointer for channel tags.
180 D%(7) = 1                  ; Gate from DIGITAL INPUT 0.
190 D%(8) = 0                  ; Range set to +/-5V
200 D%(9) = 0                  ; DT CONNECT disabled.
210 CALL CIO16(MD%, D%(0), F%) ; Execute the CALL.
```

RETURN VARIABLES	
D%(2)	KHz achieved. See MODE 51.
D%(3)	Hz achieved.
ADDATA%	Sample data
CHDATA%	Channel tags.
DEFAULTS	None
PREREQUISITE MODES	Declare D%(), F% MODE 0

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 10 MHz. All the calculations will be wrong if the jumper is in the 1 MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

D%(1) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

Specifying D%(0) = D%(1) means only one channel is sampled. D%(0) may not be greater than D%(1).

D%(2) is the desired sampling rate component desired in KHz (50 above = 50,000 S/S). On return from the CALL, this variable will contain the actual KHz the samples were taken at. See MODE 51.

D%(3) is the desired sampling rate component in hundreds of Hz, and may not be greater than 999. This number is summed with D%(2). On return from the CALL, this variable will contain the actual Hz the samples were taken at.

D%(4) is the total number of samples to take. A scan is a group of samples between the lower scan limit and upper scan limit. The number of scans is the total number of samples divided by the number of channels per scan.

D%(5) is the pointer to the array where the samples are stored.

D%(6) is the pointer to the array where the channel tags are stored.

D%(7) = 0, use external pacer on pin 25, D%(7) = 1 use internal pacer clock. Hold off on first sample until pin 25 is high.

D%(8) is the gain code to set the input range of the board. Gain codes are:

8	+/- 10V	4	0 - 10V
0	+/- 5V	5	0 - 5V
1	+/- 2.5V	6	0 - 2.5V
2	+/- 1.25V	7	0 - 1.25V
3	+/- 0.625V		

D%(9) = 0, DT-CONNECT is disabled. D%(9) = 1, DT-CONNECT is enabled. DT-CONNECT is automatically disabled upon exit from this routine weather or not it was enabled upon entry.

8.25 MODE 41 330KHz BACKGROUND A/D

Take N samples from channels Lower Scan Limit through Upper Scan Limit at R samples per second directly to memory using the REP INS/W command. Trigger internally or externally.

Data gathered by Mode 41 must be transfered to a BASIC array by mode 9. Mode 8 may be used to check background status.

Mode 41 operates in the background. When a program executes the call to Mode 41, program control returns to BASIC immediately. Data is acquired in the background by the interrupt service routine and BASIC may continue with other functions, check the mode's status with mode 8 or transfer the samples taken using mode 9

```
70 DIM ADDATA%(24000)          ; Array for analog data.
80 DIM CHDATA%(24000)          ; Array for channel tags.

100 MD% = 41                   ; Set mode to 41.
110 D%(0) = 0                  ; Lower scan limit is Ch0.
120 D%(1) = 3                  ; Upper scan limit is Ch3.
130 D%(2) = 50                 ; Speed in KHz.
140 D%(3) = 0                  ; Speed in hundreds of Hz.
150 D%(4) = 24000              ; Number of samples to take.
160 D%(5) = VARSEG(SEGNUM%(0)) ; Segment address for data.
170 D%(6) = 1                  ; Trigger method. Internal here.
180 D%(7) = 1                  ; Recycle sample set.
190 D%(8) = 0                  ; Range set to +/-5V
200 D%(9) = 0                  ; DT CONNECT disabled.
210 CALL CIO16(MD%, D%(0), F%) ; Execute the CALL.
```

RETURN VARIABLES

D%(2)	KHz achieved. See MODE 51.
D%(3)	Hz achieved.

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 10 MHz. All the calculations will be wrong if the jumper is in the 1 MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

D%(1) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

Specifying D%(0) = D%(1) means only one channel is sampled. D%(0) may not be greater than D%(1).

D%(2) is the desired sampling rate component desired in KHz (50 above = 50,000 S/S). On return from the CALL, this variable will contain the actual KHz the samples were taken at. See MODE 51.

D%(3) is the desired sampling rate component in hundreds of Hz, and may not be greater than 999. This number is summed with D%(2). On return from the CALL, this variable will contain the actual Hz the samples were taken at.

D%(4) is the total number of samples to take. A scan is a group of samples between the lower scan limit and upper scan limit. The number of scans is the number of samples divided by the number of channels per scan.

D%(5) is the pointer to the memory segment where the sample data is to be stored for later retrieval by Mode 9. See mode 6, 8 & 9.

D%(6) Trigger source variable may be 0 or 1.

Trigger method 0 expects an external TTL trigger pulse on the TRIGGER 0 (pin #25 of the 37 pin connector) input. Conversions are triggered on the rising edge of the TRIGGER 0 input. The CALL routine will execute until the number of triggers specified in D%(0) have occurred. Failure to cause the required number of trigger pulses will prevent the CALL routine from un-installing the interrupt handler, although BASIC will execute.

Trigger method 1 provides A/D pacing via the on board 8254 pacer clock counter 1 & 2. The first conversion will occur after the level of TRIGGER 0 input is brought high (the TRIGGER 0 input is really a GATE in this mode). This input floats high so if the input is not pulled low, conversions start immediately upon executing the mode. To delay the start of A/D conversions, hold pin 25 low by grounding it with a 10K Ohm resistor. Break the connection to ground to begin conversion. Conversions, once started, cannot be stopped by grounding TRIGGER 0 input. NOTE: If pin 25 is held low perpetually, the CALL routine will never return control to BASIC

D%(7) Single Cycle / Re-cycle: This mode can run once and exit after collecting the number of samples specified in D%(0) (Single Cycle Mode), or it can continuously Re-cycle.

In re-cycle mode the number samples specified in D%(0) are taken over and over again. The first sample of the cycle is always in the memory location specified in D%(1).

D%(8) is the gain code to set the input range of the board. Gain codes are:

8	+/- 10V	4	0 - 10V
0	+/- 5V	5	0 - 5V
1	+/- 2.5V	6	0 - 2.5V
2	+/- 1.25V	7	0 - 1.25V
3	+/- 0.625V		

D%(9) = 0, DT-CONNECT is disabled. D%(9) = 1, DT-CONNECT is enabled. DT-CONNECT is automatically disabled upon exit from this routine weather or not it was enabled upon entry.

8.26 MODE 42 PRE/POST TRIGGER A/D FOREGROUND

Take N pre-trigger samples from channels Lower Scan Limit through Upper Scan Limit at R samples per second directly to memory using the REP INS/W command as a continuous, rolling, pre-trigger buffer of samples. When the trigger is received on pin 25 (active high), take M post-trigger samples. When all post-trigger samples are taken, transfer both pre and post trigger samples to a BASIC ARRAY after all samples are taken. This mode runs in the foreground and control will not return to the BASIC program until all samples are taken.

```
70 DIM ADDATA%(24000)                    ; Array for analog data.
80 DIM CHDATA%(24000)                    ; Array for channel tags.

100 MD% = 42                             ; Set mode to 42.
110 D%(0) = 0                            ; Lower scan limit is Ch0.
120 D%(1) = 3                            ; Upper scan limit is Ch3.
130 D%(2) = 50                           ; Speed in KHz.
140 D%(3) = 0                            ; Speed in hundreds of Hz.
150 D%(4) = 24000                        ; Number of pre-trigger samples.
160 D%(5) = 50000                        ; Total number of samples to take.
170 D%(6) = VARPTR(ADDATA%(0))           ; Array pointer for data.
180 D%(7) = VARPTR(CHDATA%(0))           ; Array pointer for channel tags.
190 D%(8) = 0                            ; Range set to +/-5V
200 CALL CIO16(MD%, D%(0), F%)           ; Execute the CALL.
```

RETURN VARIABLES	
D%(2)	KHz achieved. See MODE 51.
D%(3)	Hz achieved.
ADDATA%	Sample data
CHDATA%	Channel tags.
DEFAULTS	None
PREREQUISITE MODES	Declare D%(), F% MODE 0
HARDWARE	Hold Pin 25 Low until trigger.

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 10 MHz. All the calculations will be wrong if the jumper is in the 1 MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

D%(1) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

Specifying D%(0) = D%(1) means only one channel is sampled. D%(0) may not be greater than D%(1).

D%(2) is the desired sampling rate component desired in KHz (50 above = 50,000 S/S). On return from the CALL, this variable will contain the actual KHz the samples were taken at. See MODE 51.

D%(3) is the desired sampling rate component in hundreds of Hz, and may not be greater than 999. This number is summed with D%(2). On return from the CALL, this variable will contain the actual Hz the samples were taken at.

D%(4) is the total number of PRE-Trigger samples to take. This number of samples is taken into a rolling buffer. After a high signal is received on pin 25, the pre-trigger buffer is frozen and further samples are taken into the post-trigger buffer.

D%(5) is the total number of samples to take; the sum of pre-trigger and post-trigger samples. A scan is a group of samples between the lower scan limit and upper scan limit. The number of scans is the total number of samples divided by the number of channels per scan. The number of post-trigger samples must be greater than 512.

D%(6) is the pointer to the array where the samples are stored.

D%(7) is the pointer to the array where the channel tags are stored.

D%(8) is the gain code to set the input range of the board. Gain codes are:

8	+/- 10V	4	0 - 10V
0	+/- 5V	5	0 - 5V
1	+/- 2.5V	6	0 - 2.5V
2	+/- 1.25V	7	0 - 1.25V
3	+/- 0.625V		

8.27 MODE 43 PRE/POST TRIGGER A/D BACKGROUND

Take N pre-trigger samples from channels Lower Scan Limit through Upper Scan Limit at R samples per second directly to memory using the REP INS/W command as a continuous, rolling, pre-trigger buffer of samples. When the trigger is received on pin 25 (active high), take M post-trigger samples. When all post-trigger samples are taken, transfer both pre and post trigger samples to a BASIC ARRAY after all samples are taken.

Data gathered by Mode 43 must be transfered to a BASIC array by mode 44 Mode 8 may be used to check background status.

Mode 43 operates in the background. When a program executes the call to Mode 43, program control returns to BASIC immediately. Data is acquired in the background by the interrupt service routine and BASIC may continue with other functions, check the mode's status with mode 8 or transfer the samples taken using mode 44.

```
70 DIM ADDATA%(24000)          ; Array for analog data.
80 DIM CHDATA%(24000)          ; Array for channel tags.

100 MD% = 43                   ; Set mode to 43.
110 D%(0) = 0                  ; Lower scan limit is Ch0.
120 D%(1) = 3                  ; Upper scan limit is Ch3.
130 D%(2) = 50                 ; Speed in KHz.
140 D%(3) = 0                  ; Speed in hundreds of Hz.
150 D%(4) = 24000              ; Number of pre-trigger samples.
160 D%(5) = 50000              ; Total number of samples to take.
170 D%(6) = VARSEG(SEGNUM%(0)) ; Segment address for data.
180 D%(7) = 0                  ; Range set to +/-5V
200 CALL CIO16(MD%, D%(0), F%) ; Execute the CALL.
```

RETURN VARIABLES

D%(2)	KHz achieved. See MODE 51.
D%(3)	Hz achieved.

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

HARDWARE — Hold pin 25 low until trigger.

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 10 MHz. All the calculations will be wrong if the jumper is in the 1 MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

D%(1) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

Specifying D%(0) = D%(1) means only one channel is sampled. D%(0) may not be greater than D%(1).

D%(2) is the desired sampling rate component desired in KHz (50 above = 50,000 S/S). On return from the CALL, this variable will contain the actual KHz samples were taken at. See MODE 51.

D%(3) is the desired sampling rate component in hundreds of Hz, and may not be greater than 999. This number is summed with D%(2). On return from the CALL, this variable will contain the actual Hz the samples were taken at.

D%(4) is the total number of PRE-Trigger samples to take. This number of samples is taken into a rolling buffer. After a high signal is received on pin 25, the pre-trigger buffer is frozen and further samples are taken into the post-trigger buffer.

D%(5) is the total number of samples to take; the sum of pre-trigger and post-trigger samples. A scan is a group of samples between the lower scan limit and upper scan limit. The number of scans is the total number of samples divided by the number of channels per scan. The number of post-trigger samples must be greater than 512.

D%(6) is the pointer to the memory segment where the sample data is to be stored for later retrieval by Mode 44. See modes 44 & 8.

D%(7) is the gain code to set the input range of the board. Gain codes are:

8	+/- 10V	4	0 - 10V
0	+/- 5V	5	0 - 5V
1	+/- 2.5V	6	0 - 2.5V
2	+/- 1.25V	7	0 - 1.25V
3	+/- 0.625V		

8.28 MODE 44 TRANSFER MODE 43 DATA TO BASIC

Retrieve samples and channel tags from memory and store them in a BASIC array.

```
100 MD% = 44                          ; Set mode to 44.
110 D%(0) = 50000                     ; Total number of samples to retrieve.
120 D%(1) = VARSEG(SEGNEM%(0))        ; Segment to retrieve from.
130 D%(2) = 0                         ; Offset (Usually 0).
140 D%(3) = VARPTR(ADDATA%(0))        ; Array for sample data.
150 D%(4) = VARPTR(CHDATA%(0))        ; Array for channel tags.
190 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
```

D%(0) is the total number of samples to take; the sum of pre-trigger and post-trigger samples. A scan is a group of samples between the lower scan limit and upper scan limit. The number of scans is the total number of samples divided by the number of channels per scan.

D%(1) is the pointer to the memory segment where the sample data is to be stored for later retrieval by Mode 44. See modes 44 & 8.

D%(2) Mode 44 can offset the data within the data segment. If you wanted the first element of the array ADATA%(0) to be the 100th sample acquired by MODE 6, then set D%(2) = 100. Usually, this variable would be zero (0).

D%(3) is the pointer to the array where the samples are stored.

D%(4) is the pointer to the array where the chan , are stored.

8.29 MODE 45 FORCE COMPATIBILITY MODE

Call this mode to force the CIO-DAS16/330 back into 16 register CIO-AD16 (DAS-16) compatible mode. This mode is needed if you want to return to compatibility mode after executing modes 41 or 43, or if there was an unplanned interruption in the operation of modes 40 or 42.

```
100 MD% = 45                          ; Set mode to 45.
190 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
```

8.30 STREAMING A/D DATA TO A FILE - MODES 46-49

Streaming is a term commonly used to describe analog or digital input software which acquires data directly from a data acquisition board, into a file, at a very high rate. Streaming software differs from data acquisition routines which acquire data into an array in three fundamental ways:

1) Data sets are limited only by file size, not by the maximum data set supported by the language.

2) Data is acquired directly into a file not into a language data type. To use the data programatically, it must be read from the file. Streaming is not for real time applications.

3) Streaming software takes control of system resources to achieve the maximum transfer rate from board to file. For best performance, streaming should be run with as few concurrent programs as possible.

ComputerBoards is proud to provide these modes for streaming free with every CIO-DAS16/330. Other vendors charge quite a bit for stand-alone streamer software and od not support streaming from within programs.

There are 4 modes which support streaming, they are:

46 Create a file to stream data to.

47 Stream data to a file.

48 Stream data to a file employing pre/post trigger buffers.

49 Transfer data from a stream file into a BASIC array.

OPTIMIZING STREAM FILES

The file created by mode 46 may not be contiguous. We have no way to control that since DOS allocates the disk space we request for the file. When DOS allocates disk space, that space may not exist as a single block of disk area with each byte immideately following the previos one.

To achieve maximum streaming rates, a contiguous file is required.

To make a fragmented file into a contiguous file you might run Norton Speed Disk, a disk optimixation program. Any other disk optimization file should do the trick. Most of these programs have a means of informing you if a file is fragmented or not.

The bottom line is, the streaming modes will work with fragmented files but will not run at the maximum rate possible for your hardware.

8.30.1 MODE 46 CREATE STREAM FILE

This mode creates the file into which data acquired by modes 47 and 48 will be streamed. The same can be accomplished with the program MAKESTRM.EXE. Creating your file using this mode will not optimize the file, which may affect the maximum streaming speed.

The file may be on a hard disk or RAM disk. RAM disks provide the maximum speed. Please refer to your DOS manual or other software to create the RAM disk prior to running mode 46.

The disk:\pathname must be specified to the RAWFILENAME$ statement if the current drive and directory are not desired.

```
70 RAWFILENAME$ = EXAMPLE.DAT          ; File name to create.
80 COMMON /QBLOCK/RAWFILENAME$         ; Common statement.

100 MD% = 46                           ; Set mode to 46.
110 D%(0) = 200                        ; 200K sample file. File size = 400K.
120 CALL CIO16(MD%, D%(0), F%)         ; Execute the CALL.
```

RETURN VARIABLES
None

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0

The file created by this mode will be a few bytes larger than twice the size in Kilobytes as the variable defined for D%(0). A sample takes two bytes of file space.

8.31 MODE 47 ACQUIRE & STREAM DATA TO FILE

Take N samples from channels Lower Scan Limit through Upper Scan Limit at R samples per second directly to stream file via REP INSW. Optionally trigger externally.

```
70 RAWFILENAME$ = EXAMPLE.DAT          ; File name to stream to.
80 COMMON /QBLOCK/RAWFILENAME$         ; Common statement.

100 MD% = 47                           ; Set mode to 47.
110 D%(0) = 0                          ; Lower scan limit is Ch0.
120 D%(1) = 8                          ; Upper scan limit is Ch8.
130 D%(2) = 250                        ; Speed in KHz.
140 D%(3) = 0                          ; Speed in hundreds of Hz.
150 D%(4) = 800                        ; Number of samples to take in 000s.
160 D%(5) = 1                          ; Internal A/D pacing.
170 D%(6) = 0                          ; ±5V Range.
180 D%(7) = 0                          ; DT Connect disabled.
190 CALL CIO16(MD%, D%(0), F%)         ; Execute the CALL.
```

RETURN VARIABLES

D%(2)	KHz achieved. See MODE 51.
D%(3)	Hz achieved.
D%(4)	The number of samples collected.

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0
MODE 46 or MAKEFILE.EXE

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 10MHz. All the calculations will be wrong if the jumper is in the 1 MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

D%(1) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

Specifying D%(0) = D%(1) means only one channel is sampled. D%(0) may not be greater than D%(1).

D%(2) is the desired sampling rate component desired in KHz (50 above = 50,000 S/S). On return from the CALL, this variable will contain the actual KHz the samples were taken at. See MODE 51.

D%(3) is the desired sampling rate component in hundreds of Hz, and may not be greater than 999. This number is summed with D%(2). On return from the CALL, this variable will contain the actual Hz the samples were taken at.

D%(4) is the total number of samples to take in thousands of hertz (200 = 200KHz). A scan is a group of samples between the lower scan limit and the upper scan limit. The number of scans is the total number of samples divided by the number of channels per scan.

D%(5) = 0, use external pacer on pin 9. D%(7) = 1 use internal pacer clock. When the internal pacer clock is used, sampling will be held off until pin 9 is high.

D%(6) is the gain range to be used for all channels. Gain codes are:

0 = ±5V	4 = 0V to 10V
1 = ±2.5V	5 = 0V to 5V
2 = ±1.25V	6 = 0V to 2.5V
3 = ±0.625V	7 = 0V to 1.25V
8 = ±10V	

D%(7) = 0, DT Connect is disabled. D%(7) = 1 DT Connect is enabled.

8.32 MODE 48 STREAM WITH PRE/POST TRIGGER

Take N pre-trigger samples from channels Lower Scan Limit through Upper Scan Limit at R samples per second directly to stream file using the REP INS/W command as a continuous, rolling, pre-trigger buffer of samples. When the trigger is received on pin 25 (active high), take M post-trigger samples.

```
70 RAWFILENAME$ = EXAMPLE.DAT          ; File name to stream to.
80 COMMON /QBLOCK/RAWFILENAME$         ; Common statement.

100 MD% = 48                           ; Set mode to 48.
110 D%(0) = 0                          ; Lower scan limit is Ch0.
120 D%(1) = 3                          ; Upper scan limit is Ch3.
130 D%(2) = 300                        ; Speed in KHz.
140 D%(3) = 0                          ; Speed in hundreds of Hz.
150 D%(4) = 10                         ; Number of pre-trigger (K) samples.
160 D%(5) = 500                        ; Number of post-trigger (K) samples.
180 D%(6) = 0                          ; Range set to +/-5V
200 CALL CIO16(MD%, D%(0), F%)         ; Execute the CALL.
```

RETURN VARIABLES	
D%(2)	KHz achieved. See MODE 51.
D%(3)	Hz achieved.
D%(4)	Number of pre-trigger samples taken.
D%(5)	# of post-trigger samples taken.
DEFAULTS	None
PREREQUISITE MODES	Declare D%(), F% MODE 0
HARDWARE	Hold pin 25 low until trigger then bring high and hold high.

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 10 MHz. All the calculations will be wrong if the jumper is in the 1 MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

D%(1) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

Specifying D%(0) = D%(1) means only one channel is sampled. D%(0) may not be greater than D%(1).

D%(2) is the desired sampling rate component desired in KHz (50 above = 50,000 S/S). On return from the CALL, this variable will contain the actual KHz the samples were taken at. See MODE 51.

D%(3) is the desired sampling rate component in hundreds of Hz, and may not be greater than 999. This number is summed with D%(2). On return from the CALL, this variable will contain the actual Hz the samples were taken at.

D%(4) is the total number of PRE-Trigger samples to take in thousands. This number of samples is taken into a rolling buffer. After a high signal is received on pin 25, the pre-trigger buffer is frozen and further samples are taken into the post-trigger buffer. The maximum size of this variable is 15,000 samples, or 15, for 15K.

D%(5) is the total number of post-trigger samples to take in thousands. A scan is a group of samples between the lower scan limit and upper scan limit. The number of scans is the total number of samples divided by the number of channels per scan.

D%(6) is the gain code to set the input range of the board. Gain codes are:

8	+/- 10V	4	0 - 10V
0	+/- 5V	5	0 - 5V
1	+/- 2.5V	6	0 - 2.5V
2	+/- 1.25V	7	0 - 1.25V
3	+/- 0.625V		

8.33 MODE 49 MOVE DATA FROM FILE TO BASIC

Move data from stream file into a BASIC array in chnunks. A BASIC array may be no larger than 32,767 elements. When moving the data from the file, optionally create a channel tag array,

```
70 RAWFILENAME$ = EXAMPLE.DAT          ; File name to move from.
80 COMMON /QBLOCK/RAWFILENAME$         ; Common statement.

100 MD% = 49                           ; Set mode to 49.
110 D%(0) = 20000                      ; Number of samples to load into array.
120 D%(1) = 100                        ; Retrieve file offset in thousands.
130 D%(2) = 0                          ; Retrieve file offset in hundreds.
140 D%(3) = VARPTR(ADDATA%(0)          ; BASIC array for samples.
150 D%(4) = VARPTR(CHDATA%(0)          ; BASIC array for channel tags.
160 CALL CIO16(MD%, D%(0), F%)         ; Execute the CALL.
```

RETURN VARIABLES

D%(0)	The number of points placed in the BASIC array durring ...ion of the mode.
D%(1)	...ow channel in the scan when the data was collected. This ...ored in the file.
D%(2)	... high channel in the scan when the data was collected. This ... ...ored in the file.
D%(3)	... KHz component of the data collection rate from the sample run. not valid for externally paced sample runs.
D%(4)	The hundreds of Hz component of the data collection rate from the sample run. not valid for externally paced sample runs.
D%(5)	The thousands component of the total number of samples in the file.
D%(6)	The hundreds component of the total number of samples in the file.
D%(7)	The number of pre-trigger samples in the file. May also be interpreted as the last pre-trigger sample in the file.
D%(8)	The gain used to gather the samples in the file. Same codes at modes 47 & 48.

DEFAULTS — None

PREREQUISITE MODES — Declare D%(), F%
MODE 0
Name file to draw from.

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 1 MHz. All the calculations will be wrong if the jumper is in the 1 MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) is the number of samples to load into the BASIC array during this mode 49 call. This may not exceed the maximum BASIC array size of 32,767.

D%(1) is the offset within the sample file, in thousands, to retrieve samples from. For example, if you wanted to retrieve samples 200000 through 220000 you would set D%(0) to 20000, D%(1) to 200, and D%(2) to 0.

D%(2) is the hundreds component of th... ...et within the sample file to retrieve samples from.

D%(3) is the name of the BASIC array to place the samples into. The name ADDATA%(0) is chosen only as an example.

D%(4) is the name of the BASIC array to place the channel tags into. The name CHDATA%(0) is chosen only as an example. If you set D%(4) = 0 then the channel tags are not retrieved. For example, if you sampled only a single channel, why load up an array with channel tags that are all the same.

8.30 MODE 50 EASY TO USE A/D

Take N samples from channels Lower Scan Limit through Upper Scan Limit at R samples per second directly memory via DMA, then transfer to a BASIC ARRAY after all samples are taken. Optionally trigger externally and optionally re-cycle.

```
70 DIM ADDATA%(24000)          ; Array for analog data.
80 DIM CHDATA%(24000)          ; Array for channel tags.

100 MD% = 50                   ; Set mode to 50.
110 D%(0) = 0                  ; Lower scan limit is Ch0.
120 D%(1) = 3                  ; Upper scan limit is Ch3.
130 D%(2) = 50                 ; Speed in KHz.
140 D%(3) = 0                  ; Speed in hundreds of Hz.
150 D%(4) = 24000              ; Number of samples to take.
160 D%(5) = VARPTR(ADDATA%(0)) ; Array pointer for data.
170 D%(6) = VARPTR(CHDATA%(0)) ; Array pointer for channel tags.
180 D%(7) = 1                  ; Gate from DIGITAL INPUT 0.
190 CALL CIO16(MD%, D%(0), F%) ; Execute the CALL.
```

RETURN VARIABLES	
D%(2)	KHz achieved. See MODE 51.
D%(3)	Hz achieved.
ADDATA%	Sample data
CHDATA%	Channel tags.
DEFAULTS	None
PREREQUISITE MODES	Declare D%(), F% MODE 0
WARNINGS	Do not write to EGA or VGA screens until the complete DMA cycle is done when sampling at speeds over 60 KHz.[17]

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 1 MHz. All the calculations will be wrong if the jumper is in the 10MHz position. Factory default is 1MHz.

FORMAT OF THE DATA

D%(0) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

D%(1) may be 0 to 7 in differential mode, 0 - 15 in single ended mode.

Specifying D%(0) = D%(1) means only one channel is sampled. D%(0) may not be greater than D%(1).

D%(2) is the desired sampling rate component desired in KHz (50 above = 50,000 S/S). On return from the CALL, this variable will contain the actual KHz the samples were taken at. See MODE 51.

D%(3) is the desired sampling rate component in hundreds of Hz, and may not be greater than 999. This number is summed with D%(2). On return from the CALL, this variable will contain the actual Hz the samples were taken at.

D%(4) is the total number of samples to take. A scan is a group of samples between the lower scan limit and upper scan limit. The number of scans is the total number of samples divided by the number of channels per scan.

D%(5) is the pointer to the array where the samples are stored.

D%(6) is the pointer to the array where the channel tags are stored.

---

17 This restriction applies only to EGA and VGA screens because those screens employ the Non Maskable Interrupt (NMI) signal on the bus. This has priority over DMA transfers and will cause the DMA controller to miss a transfer request. The result will be bad channel tags, bad data or both. MetraByte's example program has this error in programming in it. Do not follow that example.

D%(7) = 0, use external pacer on pin 25, D%(7) = 1 use internal pacer clock. Hold off on first sample until pin 25 is high.

NOTE: Unlike MODE 6, MODE 50 does not have a recycle option. The recycle option requires a sophisticated programming strategy where two independent devices, the CPU and DMA controller, operate on the same memory location. See MODE 6.

8.31 MODE 51 LOAD COUNTER 1&2 WITH Hz

Load the on board pacer clock, specifying a frequency in Hz. Mode calculates closest achievable frequency and reports it back.

```
100 MD% = 51                          ; Set mode to 51.
110 D%(0) = 10                        ; Set KHz to 10.
120 D%(1) = 500                       ; Set Hz to 500. Rate is 10,500 Hz.
130 CALL CIO16(MD%, D%(0), F%)        ; Execute the CALL.
140 KHRATE% = D%(2)                   ; Store current KHz rate.
150 HRATE = D%(3)                     ; Store current Hz rate.
```

RETURN VARIABLES	
D%(0), D%(1)	Unchanged
D%(2)[18]	Divisor in counter 2
D%(3)	Divisor in counter 1
DEFAULTS	Assumes counter input rate is the on-board 1 MHz source.
PREREQUISITE MODES	Declare D%(), F% MODE 0

NOTE - This mode assumes the jumper which selects 1 MHz or 10 MHz as the source frequency to the 8254 pacer clock is set for 1 MHz. All the calculations will be wrong if the jumper is in the 10MHz position. Factory default is 1MHz.

DESIRED vs. ACHIEVABLE FREQUENCY

The 8254 pacer clock divides the 1 MHz input by some number to provide an output rate. The two cascaded counters, 1 & 2, each may be loaded with a number between 2 and 64,536, providing rates as high as 100,000 Hz (the maximum sampling rate of the board) and as low as 0.0024 Hz (1 sample every 7 minutes).

One very nice feature of this mode is the return variables D%(2) and D%(3). The values from these variables should be stored with the data.

SHAPE OF THE WAVE

The shape of the wave from counter 2 out as a result of programming counters 1 and 2 with Mode 51 is a pulsed output. The output of counter 2 is high until terminal count when it goes low for one count.

---

18 It is a simple matter to calculate the actual rate of counter 2 out by this formula:

OUT2 = 1,000,000 / D%(2) / D%(3)

If your 1/10 MHz jumper is in the 10MHz position, substitute 10MHz in the formula.

8.32 CALL ERROR CODES

The variable F% which is passed with each use of the CALL CIO16(MD%, D%(0), F%), is always returned with the error status of the CALL.

0	No error.
1	Driver not initialized using MODE 0
2	MODE number out of range. *NOTE*: Metrabyte used only modes 0 - 20, we have added modes 50 - 60.
3	Base Address out of range. Range is 512 (200H) - 1008 (3F0H).
4	Interrupt level out of range. Range is 2 - 7.
5	DMA level out of range. Range is 1 or 3.
6	MUX scan high limit > 7 in differential mode.
7	MUX scan high limit > 15 in single ended mode.
8	Not used.
9	A/D hardware timeout error. No EOC
10	Counter load attempted < 2 in MODE 17.
11	Number of conversions < 1 in MODEs 4, 5 or 6.
12	Counter configuration # > 5 in MODE 10.
13	OP# data > 15 in MODE 13.
14	D/A data > 4095 in MODEs 15 or 16.
15	D/A channel number > 1 in MODEs 15 or 18.
16	Counter read operation request > 1 in MODE 12.
17	Memory Transfer offset < 0 in MODE 9.
18	Word count for transfer < 1 in MODE 9.
19	Trigger choice > 1 in MODE 4, 5, or 6.
20	DMA or Interrupt currently active in MODE 5 or 6.
21	DMA page overrun in MODE 6.
22	Board read-back registers not readable. Board probably has the wrong base address.
23	Trigger channel not as configured in MODE 19.
24	Trigger out of range in MODE 19
25	A/D wave slope > 1 in MODE 19.
40	MODE 55 configuration error.
41	Data out of range, MODEs 50 - 60.
42	A mode requiring word (16 bit) operation has been called and the hardware override switch for byte compatibility has been set. Returned by modes 40, 41, 42, 43.
43	Inticates that pin 25 is high upon entr[illegible] modes 42 or 43. Pin 25 should be low upon entry.
44	Hardware failure. A/D EOC line is not responding.

45	Trigger received prior to acquiring first complete set of pre-trigger samples. Reduce the size of the pre-trigger buffer or hold off the trigger.
46	FIFO data over run error. Data has been lost due to overrun at the FIFO buffer or the DT-CONNECT port.

The error codes are most useful in dubugging programming errors. With the exception of ERROR 9 (A/D time-out) and ERROR 22 (No readback), the error codes only refer to mistakes which would prevent proper execution of your program. The new modes, 40 - 45 have additional run-time error correction which is usefull for monitoring programs which use the advanced features.

These kind of mistakes will be found during your development and verification cycle.

A BASIC routine that will trap error 9 and 22 follows:

```
150 CALL CIO16(MD%, D%(0), F%)          ; The CALL routine in your program.
160 IF F% <> 0 THEN GOSUB 1000          ; Test for any error code.

1000 IF F% <> 9 THEN GOTO 1100
1010 PRINT "THERE HAS BEEN A FAILURE OF THE A/D CONVERTER DETECTED."
1020 PRINT "PLEASE CALL THE SYSTEM OPERATOR AT 508-261-1123."
1030 INPUT A                            ; A dummy input to hold the screen.
1030 RETURN                             ; Return to the main program

1100 IF F% <> 22 THEN GOTO 1200
1110 PRINT "THERE IS A PROBLEM WITH THE I/O BOARD."
1120 PRINT "PLEASE CALL THE SYSTEM OPERATOR AT 508-261-1123".
1130 INPUT A
1140 RETURN

1200 PRINT "THERE HAS BEEN AN UNEXPECTED ERROR NUMBER "; F%
1210 PRINT "PLEASE RECORD THIS NUMBER,"
1220 PRINT "AND CALL THE SYSTEM OPERATOR AT 508-261-1123."
1230 INPUT A
1240 RETURN
```

In this routine there is much room for improvement but it shows how to use the error codes.

8.33 SPECIAL START/STOP CHANNEL & SPEED LIMITATIONS

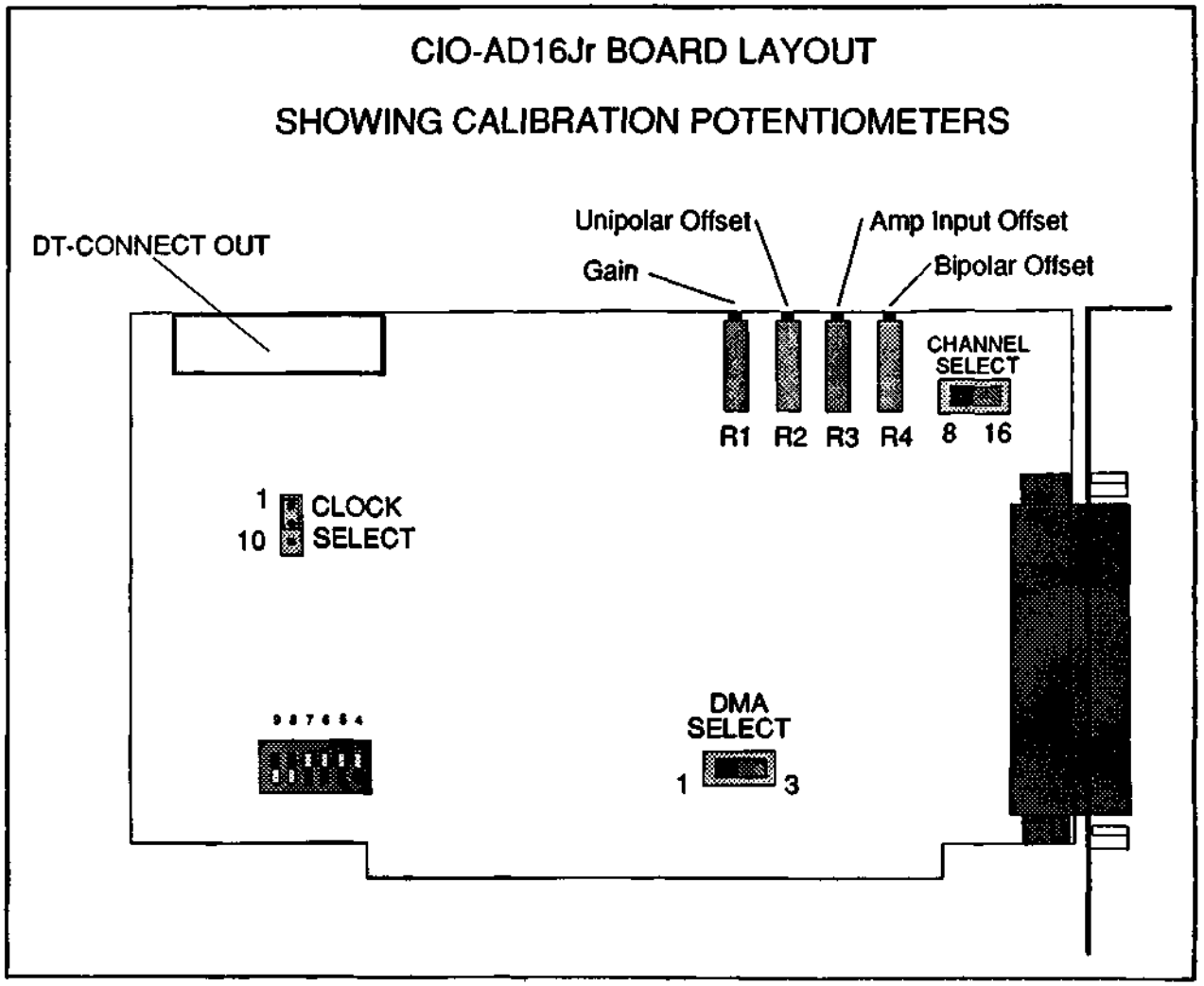
There is a limitation on acquisition speed in given start/stop channel selections. Here is a table which explains the limitations.

START	STOP	MAXIMUM A/D SPEED
0 same	0 same	330 KHz. No limitations when the start channel and stop channel are the same channel.
0 even	3 odd	330 KHz. No limitations when the start channel is odd and the stop channel is even.
1 odd	4 even	330 KHz. No limitations when the start channel is even and the stop channel is odd.
0 same even	4 same even	150 KHz. When the start channel AND stop channel are even then A/D rate is limited to 150 KHz due to mux settling time.
1 same odd	3 same odd	150 KHz. When the start channel AND stop channel are odd then A/D rate is limited to 150 KHz due to mux settling time.

9 CIO-AD16JR-AT ARCHITECTURE

In addition to the control and data registers, there are 9 major functional blocks on the CIO-AD16JR-AT which are explained in this manual, in order of explanation, they are:

- **PC/XT/AT Bus Interface.** Address, power consumption and data transfers.
- **Control & Data Registers.** Function and format of the register.
- **Analog Input.** Signal connection, multiplexing, amplification and conversion.
- **Analog Output.** Digital value, reference voltage, signal output.
- **Counter / Timer.** Basic function, board usage and external connection & control.
- **Digital Input (4 bit).** Acquisition and signal connection.
- **Digital Output (4 bit MUX control).** Control and signal connection.
- **PIO-12 Compatible I/O.** Programming and signal connection.
- **Trigger Logic.** External and on board IR & DMA triggers.
- **DMA Control Logic.** Timing charts and logic.
- **Interrupt Control.** Timing charts and logic.



9.1 PC/XT/AT BUS INTERFACE

The CIO-AD16JR-AT interface to the PC is via the gold finger connector on the lower right corner of the board. This edge connector plugs into the bus expansion connector of a PC/XT or PC/AT.

When IBM introduced the PC/AT, the original XT bus expansion connector and signals were retained and a second bus expansion connector was added to support the 16 bit data bus, 7 additional interrupts and 4 DMA levels. Because the AT bus is 100% backward compatible with the XT, user's investment in hardware is protected. All of the recent PC/386 machines use the PC/AT bus specification.

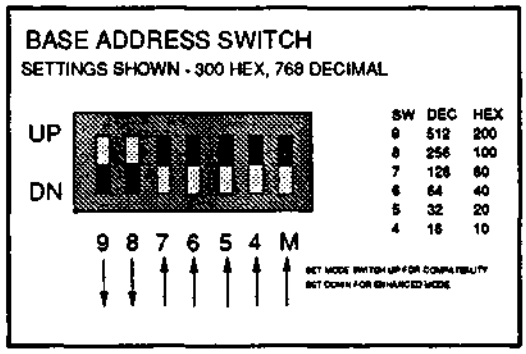
The new microchannel, or PS/2 bus, is not at all compatible with the PC/XT/AT bus architecture, signals or connector.

9.1.1 BASE ADDRESS

The base address switch fixes the CIO-AD16JR-AT registers at specific locations in the PC I/O address space. The CIO-AD16JR-AT uses 16 or 32 consecutive addresses at which 16 or 32 consecutive 8 bit registers reside.

The base address switch fixes the first of the 16 or 32 consecutive registers at an I/O address[19]. This address, for example 300 HEX (768 decimal), is then called the CIO-AD16JR-AT base address and all registers are referenced as being some count up from the base address. For example, the register at BASE + 2 is the Multiplexor control register. If the CIO-AD16JR-AT base address switch is set for 300H, the MUX control register is at 302H.

The base address switch is shown here. The base address switch is a bank of dip switches which control the 74LS682 comparator, U46. The comparator compares the address on the PC bus with the state of the switches and responds to any address from BASE to BASE + 15 with a signal which places the CIO-AD16JR-AT transceivers on the PC
bus so board and CPU can communicate.



The base address switch has 6 switches numbered 1 through 6 on the DIP switch. The numbers printed on the board are the address lines these switches correspond to; address lines 9, 8, 7, 6, 5, and 4.

When a switch is DOWN, the CIO-AD16JR-AT responds to that address line. A CIO-AD16JR-AT base address is constructed by selecting the address lines, 5 through 9, that you want the CIO-AD16JR-AT to respond to.

For example, BASE 300 HEX is constructed by switching address line 9 (200H) and line 8 (100H) down and leaving switches 5, 6 & 7 up.

9.1.2 MODE SWITCH

Switch #7 of the base address switch doubles as a mode switch. The mode switch enables and disables extended addresses and features of the CIO-AD16JR-AT.

The state of the mode switch may be read back at Base Address + 11. When UP (OFF), Base Address + 11, BIT 4 reads back as 0 (zero). When the mode switch is down, Base Address + 11, BIT 4 reads back as 1 (one).

When the switch is ON (UP), the CIO-AD16JR-AT is compatible with the MetraByte DAS-16, ComputerBoards CIO-AD16 and CIO-AD16Jr. Only 16 I/O address are used by the CIO-AD16JR-AT when the mode switch is up. Because only 16 I/O addresses are used when the mode switch is off, the CIO-AD16JR-AT may be placed on 16 bit boundaries, such as 300H, 310H 320H, 330H and 340H.

---

19 The iAPX86 family of processors have two types of address space, memory addresses and I/O addresses. Access to memory and I/O space is controlled by the type of ASSEMBLY instruction being executed via the hardware pin IO/M. For a full discussion of IO/M space, see the Intel iAPX86 family data books.

When the mode switch is OFF (DOWN), the CIO-AD16JR-AT is in extended features mode, or full PC/AT mode. Additional control bits are present in the upper nibble of Base Address + 11 and a second 8254 counter is addressable for pre/post trigger control. When the mode switch is down, addresses such as 300H, 320H and 340H are available while addresses such as 310H and 330H are not. In enhanced mode, the CIO-AD16JR-AT occupies 32 I/O addresses.

The mode switch must be down to access extended features through software. The CIO-AD16JR-AT programming language CALL routines and examples which use high-speed transfers or extended features require the modes switch be down.

TABLE OF I/O ADDRESS

HEX RANGE	FUNCTION	HEX RANGE	FUNCTION
000-00F	8237 DMA #1	2C0-2CF	EGA
020-021	8259 PIC #1	2D0-2DF	EGA
040-043	8253 TIMER	2E0-2E7	GPIB (AT)
060-063	8255 PPI (XT)	2E8-2EF	SERIAL PORT
060-064	8742 CONTROLLER (AT)	2F8-2FF	SERIAL PORT
070-071	CMOS RAM & NMI MASK (AT)	300-30F	PROTOTYPE CARD
080-08F	DMA PAGE REGISTERS	310-31F	PROTOTYPE CARD
0A0-0A1	8259 PIC #2 (AT)	320-32F	HARD DISK (XT)
0A0-0AF	NMI MASK (XT)	378-37F	PARALLEL PRINTER
0C0-0DF	8237 #2 (AT)	380-38F	SDLC
0F0-0FF	80287 NUMERIC CO-P (AT)	3A0-3AF	SDLC
1F0-1FF	HARD DISK (AT)	3B0-3BB	MDA
200-20F	GAME CONTROL	3BC-3BF	PARALLEL PRINTER
210-21F	EXPANSION UNIT (XT)	3C0-3CF	EGA
238-23B	BUS MOUSE	3D0-3DF	CGA
23C-23F	ALT BUS MOUSE	3E8-3EF	SERIAL PORT
270-27F	PARALLEL PRINTER	3F0-3F7	FLOPPY DISK
2B0-2BF	EGA	3F8-3FF	SERIAL PORT

NOTE: I/O address not used are free, for example, addresses 330 through 370 are not used and so may be used by the CIO-AD16JR-AT.

9.1.3 BUS POWER CONSUMPTION

Voltage	Max.	Typ.
+5	450mA	375mA
+12	0V	
-12	0V	

The CIO-AD16JR-AT analog input section is powered by the ±15V from the DC/DC converter.

9.1.4 DATA TRANSFERS

The CIO-AD16JR-AT bus interface is a PC/XT/AT bus interface. In compatibility mode (mode switch up) it the JR-AT interfaces to the XT bus only. In enhanced mode, the JR-AT employs the full 16 bit PC/AT bus.

Because 16 bit data transfers are faster, the CIO-AD16JR-AT can transfer A/D samples taken with old DAS-16 software at speeds over 150KHz. Old DAS-16 software use DMA to transfer A/D data.

Of course, using the REP INSW command which can only be employed by the CIO-AD16JR-AT, data transfers of 330KHz and sample sets of any size up to the size of available memory may be taken.

9.2 CONTROL & DATA REGISTERS

The CIO-16 is controlled and monitored by writing to and reading from 16 or 20 consecutive 8 bit I/O addresses. The first address, or BASE ADDRESS, is determined by setting a bank of switches on the board.

A register is easy to read and write to. Most often, register manipulation is best left to ASSEMBLY language programs as most of the CIO-AD16JR-AT possible functions are implemented in easy to use BASIC CALL routines or PASCAL, C FORTRAN libraries.

The register descriptions follow all follow the format:

7	6	5	4	3	2	1	0
A/D9	A/D10	A/D11	A/D12 LSB	CH8	CH4	CH2	CH1

Where the numbers along the top row are the bit positions within the 8 bit byte and the numbers and symbols in the bottom row are the functions associated with that bit.

To write to or read from a register in decimal or HEX, the following weights apply:

BIT POSITION	DECIMAL VALUE	HEX VALUE
0	1	1
1	2	2
2	4	4
3	8	8
4	16	10
5	32	20
6	64	40
7	128	80

To write control or data to a register, the individual bits must be set to 0 or 1 then combined to form a Byte. Data read from registers must be analyzed to determine which bits are on or off.

The method of programming required to set/read bits from bytes is beyond the scope of this manual. It will be covered in most Introduction To Programming books, available from a book store.

In summary form, the registers and their function are listed on the following table. Within each register are 8 bits which may constitute a byte of data or 8 individual bit set/read functions.

ADDRESS	READ FUNCTION - ALL MODES	WRITE FUNCTION
BASE	A/D Bits 9 - 12(LSB) & Channel #	Start A/D Conversion
BASE + 1	A/D Bits 1(MSB) - 8	None
BASE + 2	Channel MUX Set	Channel MUX Read
BASE + 3	Digital 4 Bit Input	Digital 4 Bit Output
BASE + 4	None	None
BASE + 5	None	None
BASE + 6	None	None
BASE + 7	None	None[20]
BASE + 8	Status EOC, UNI/BIP etc.	None
BASE + 9	DMA, Interrupt &Trigger Control	Set DMA, INT etc
BASE + 10	Pacer clock control register.	None
BASE + 11	Gain setting read-back.	Gain control
BASE + 12	Counter 0 Data	Counter 0 Data
BASE + 13	CTR 1 Data - A/D Pacer Clock	CTR 1 Data - A/D Pacer
BASE + 14	CTR 2 Data - A/D Pacer Clock	CTR 2 Data - A/D Pacer
BASE + 15	None. No read back on 8254.	Pacer Clock Control (8254)
	ENHANCED MODE ONLY	
BASE + 11	Upper nibble, enhanced features status.	Upper nibble, enhanced features control.
BASE + 16	Total sample count (1/2)	CTR 0. Total counter (1/2).
BASE + 17	Total sample count (1/2)	CTR 1. Total counter (1/2).
BASE + 18	Pre-trigger count.	CTR 2. Pre-trigger counter
BASE + 19	None. No read back on 8254	Adnavced features counter control (8254).

9.2.1 A/D DATA & CHANNEL REGISTERS

BASE ADDRESS Example, 300 HEX, 768 Decimal

7	6	5	4	3	2	1	0
A/D9	A/D10	A/D11	A/D12 LSB	CH8	CH4	CH2	CH1

A read/write register.

READ

On read, it contains two types of data. The least significant 4 digits of the Analog input data and the channel number which the current data was taken from.

These 4 bits of analog input data must be combined with the 8 bits of analog input data in BASE + 1, forming a complete 12 bit number. The data is in the format 0 = minus full scale. 4095 = +FS.

20 These four addresses, Base + 4 through Base + 7, are the addresses used by the analog output chips on the CIO-AD16 (MetraByte DAS-16). In order to stay 100% software compatible, these four registers are empty on the CIO-AD16JR-AT.

The channel number is binary. The weights are shown in the table above. If the current channel were 5 then bits CH4 and CH1 would be high, CH8 and CH2 would be low.

WRITE

Writing any data to the register causes an immediate A/D conversion.

BASE ADDRESS + 1 301 HEX, 769 Decimal

7	6	5	4	3	2	1	0
A/D1 MSB	A/D2	A/D3	A/D4	A/D5	A/D6	A/D7	A/D8

A Read-only register.

On read the most significant A/D byte is read.

9.2.2 CHANNEL MUX SCAN LIMITS REGISTER

BASE ADDRESS + 2 302 HEX, 770 Decimal

7	6	5	4	3	2	1	0
CH H8	CH H4	CH H2	CH H1	CH L8	CH L4	CH L2	CH L1

A read and write register.

READ

The current channel scan limits are read as one byte. The high channel number scan limit is in the most significant 4 bits. The low channel scan limit is in the least significant 4 bits.

WRITE

The channel scan limits desired are written as one byte. The high channel number scan limit is in the most significant 4 bits. The low channel scan limit is in the least significant 4 bits.

That is one way of looking at this register, and probably the most sensible way. In fact, the bits 3-0 contain the starting channel number and bits 7-4 contain the ending channel number. If you wanted to scan channels 1, 2, 3 in that order, you could do so by placing the 3 in bits 7-4 and the 1 in bits 3-0.

NOTE: Every write to this register sets the current A/D channel MUX setting to the number in bits 7-4. See BASE + 8.

9.2.3 4 BIT DIGITAL I/O REGISTERS

BASE ADDRESS + 3 303 HEX, 771 Decimal

When read...

7	6	5	4	3	2	1	0
0	0	0	0	DI3	DI2, CTR 0 GATE	DI1	DI0, TRIG

READ

The signals present at the inputs are read as one byte, the most significant 4 bits of which are always zero. The pins 25 (digital input 0) and 24 (digital input 2) digital inputs have two functions each.

The TRIG function of digital input 0 may be used to hold of the first sample of an A/D set by holding it low (0V) until you are ready to take samples, which are then paced by the 8254. It can also be used as the source of an external start conversion pulse, synchronizing A/D conversions to some external event.

When written to..

7	6	5	4	3	2	1	0
X	X	X	X	DO3	DO2	DO1	DO0

WRITE

The upper four bits are ignored. The lower four bits are latched TTL outputs. Once written, the state of the inputs cannot be read back because a read back would read the separate digital input lines (see above).

9.2.4 STATUS REGISTER

BASE ADDRESS + 8 308 HEX, 776 Decimal

7	6	5	4	3	2	1	0
EOC	U/B	MUX	INT	CH8	CH4	CH2	CH1

A read mostly, one-function-write register.

READ

EOC = 1, the A/D converter is busy. EOC = 0, it is free.

U/B = 1, the amplifier is in Unipolar mode. U/B = 0, is bipolar.

MUX = 1, Channels are configured 16 single ended. MUX = 0, 8 differential.

INT = 1, an external pulse has been received. INT = 0, the flip-flop is ready to receive a pulse..

There is a flip-flop on the TRIGGER input (pin 25) which will latch a pulse as short as 200 nanoseconds. Once triggered, this flip-flop must be reset by a write to this register. Your interrupts service routine must do this before another interrupt trigger can be received.

CH8, CH4, CH2 & CH1 are a binary number between 0 and 15 indicating the channel number that the MUX is currently set to and is valid only when EOC = 0. The channel MUX increments shortly after EOC = 1 so may be in a state of transition when EOC = 1. The binary weight of each bit is shown in the table above.

WRITE

A write of any data to this register resets the flip-flop on the pin 25 input and sets the INT bit to 0.

9.2.5 DMA, INTERRUPT & TRIGGER CONTROL

BASE ADDRESS + 9 309 HEX, 777 Decimal

7	6	5	4	3	2	1	0
INTE	IR4	IR2	IR1	Don't Care	DMA	TS1	TS0

A read and write register.

READ

INTE = 1, Interrupts are enabled. An interrupt generated will be placed on the PC bus interrupt level selected by IR4, IR2 & IR1. INTE = 0, interrupts are disabled.

IR4, IR2, IR1 are bits in a binary number between 0 and 7 which map interrupts onto the PC bus interrupt levels 2 - 7. Interrupts 0 & 1 may not be asserted by the CIO-AD16JR-AT.[21]

DMA = 1, DMA transfers are enabled. DMA = 0, DMA transfers are disabled. It is worth noting that this bit only allows the CIO-AD16JR-AT to assert a DMA request to the PC on the DMA request level selected by the DMA switch on the CIO-AD16JR-AT. Before this bit is set to 1, the PC's 8237 (or appropriate) DMA controller chip must be set up.[22]

TS1 & TS0 control the source of the A/D start conversion trigger according to the table below.

TS1	TS0	
0	X	Software triggered A/D only.
1	0	Start on rising TRIGGER (Digital input 0, Pin 25)
1	1	Start on Pacer Clock Pulse (CTR 2 OUT, no external access)

9.2.6 PACER CLOCK CONTROL REGISTER

BASE ADDRESS + 10 30A HEX, 778 Decimal

7	6	5	4	3	2	1	0
X	X	X	X	X	X	CTR0	TRIG0

Write only

CTR0 = 1. When CTR0 = 1, an on-board 100KHz clock signal is ANDed with the COUNTER 0 CLOCK INPUT (pin 21). A high on pin 21 will allow pulses from the on-board source into the 8254 Counter 0 input.

CTR0 = 0. When CTR0 = 0, the input to 8254 Counter 0 is entirely dependent on pulses at pin 21, COUNTER 0 CLOCK INPUT.

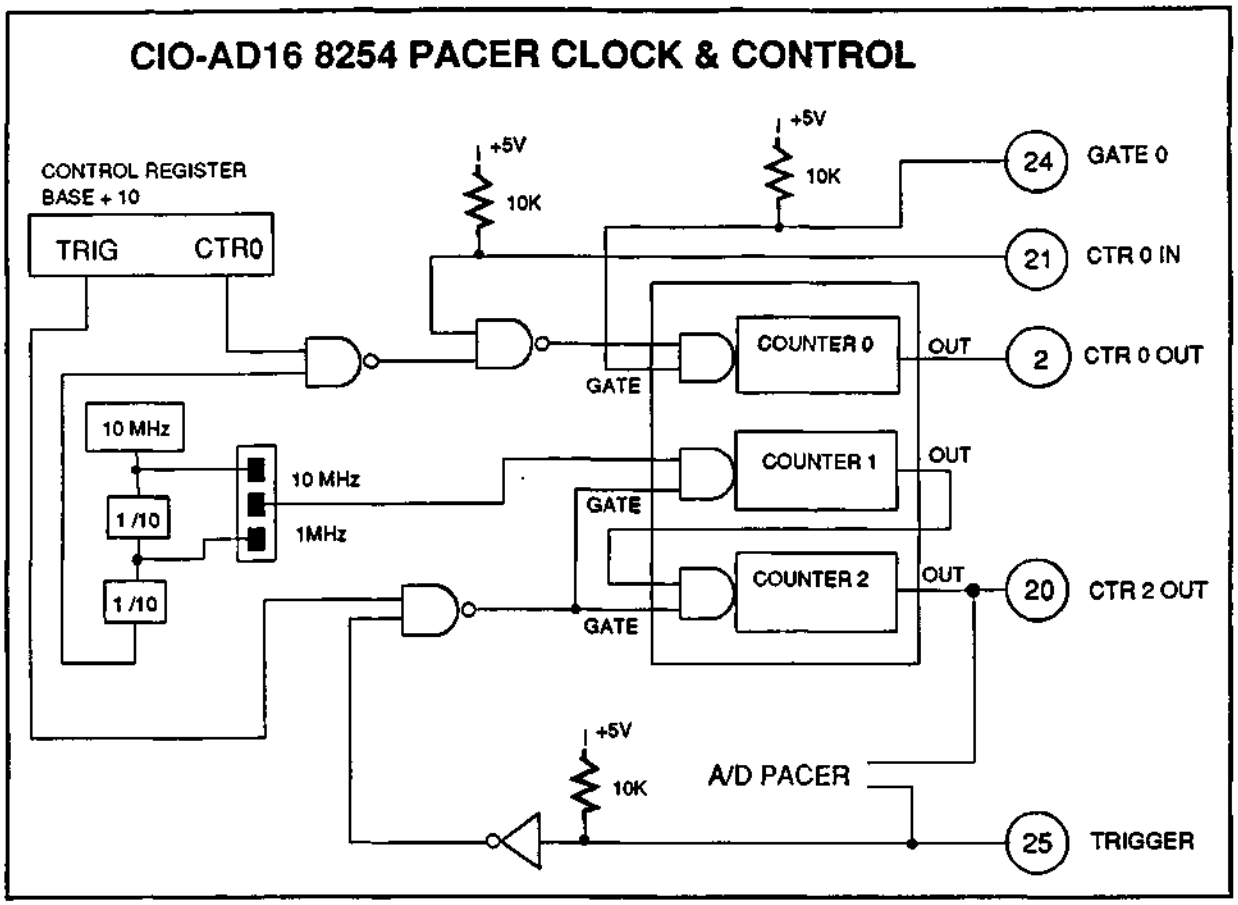
TRIG0 = 1. When TRIG0 = 1, the TRIGGER input at pin 25 is ANDed with TRIG0 which must therefore be high for the pulses from the on-board pacer clock (8254) to start A/D conversions. The input at pin 25 is pulled up and will always be high unless pulled low externally.

TRIG0 = 0. When TRIG0 = 0, the GATEs of counter 1 & 2 are held high, preventing any pulses from acting as a source of A/D start pulses.

The following diagram may help understand these registers, and is further explained in the section covering the 8254.

21 Information on the assignment of interrupt levels to standard PC peripherals is found in the section on the architecture of the PC. In other words, not all interrupts 2-7 are available without conflict.

22 An example of how to set up the DMA controller may be found in the BASIC CALL source code. The code is the same for PC/XT/AT and 386. The source code is for sale only after executing a non-disclosure agreement. Please contact our technical support for a copy of the agreement.



9.2.7 ANALOG INPUT RANGE REGISTER

BASE ADDRESS + 11 - Compatible Mode 30B HEX, 779 Decimal

A write to this register sets the analog input range for all 8/16 analog inputs. The lower 4 bits set the analog input range. The upper 4 bits are not used in compatibility mode.

7	6	5	4	3	2	1	0
X	X	X	X	Range	Uni/Bip	G1	G0

Range	Uni/Bip	G1	G0	Input Range	Decimal
1	0	0	0	±10V	8
0	0	0	0	±5V	0
0	0	0	1	±2.5V	1
0	0	1	0	±1.25V	2
0	0	1	1	±0.625V	3
0	1	0	0	0-10V	4
0	1	0	1	0-5V	5
0	1	1	0	0-2.5V	6
0	1	1	1	0-1.25V	7

To set the analog inpput range of the CIO-AD16JR-AT programatically, write the correct input range code to the base address + 11. For example, from BASIC:

If the board's base address is 300 HEX (768 decimal), then the gain register is at 768 + 11 = 779

```
100  OUT 779, 5                    'Set analog output range to 0-5V
```

The decimal range codes are in the far right column above.

BASE ADDRESS + 11 - Enhanced Mode **30B HEX, 779 Decimal**

A write to this register sets the analog input range for all 8/16 analog inputs as well as enabling and disabling extended featuresand controlling extended features for pre-triggering and DT-CONNECT. The lower 4 bits set the analog input range. The upper four bits control and provide status of enhanced features.

7	6	5	4	3	2	1	0
DT-Connect	Pre-Trigger	A/D Overrun	Set/Read Mode	Range	Uni/Bip	G1	G0

GAINS: Gain control is identical in both compatibility and enhanced mode.

BIT 4 - Set Compatible/Enhanced Mode. Read Mode Status

WRITE: 0 = disable enhanced mode. 1 = enable enhanced mode (mode switch must be down).

READ: 0 = Switch 7 of base address switch, the mode switch, is up. It is not possible to enable enhanced features when the mode switch is up. When Bit 4 = 1, the mode switch is down and enhanced features may be enabled.

BIT 5 - FIFO or DT-CONNECT Over Run - Read Only.

0 = No over run. 1 = FIFO buffer full (> 1024 samples) or DT-CONNECT over run if DT-CONNECT is enabled. This signal is the logical OR of these two possible over run signals. To determine the source of the over run error: If DT-Connect is not enabled then the over run is FIFO. If DT-CONNECT is enabled, issue a clear to the FIFO then re-read the status bit. If it is still active then the over run was at the DT-CONNECT.

BIT 6 - Pre-Trigger Enable. Trigger Status.

WRITE: 0 = Disable pre-trigger mode. 1 = enable pre-trigger. The trigger index counter (CTR 2 of 2nd 8254) is gated to the packet interrupt. Because it is gated to the FIFO packet interrupt, the trigger flushes the FIFO at the instant of the trigger, ensuring the only samples prior to and all samples prior to the trigger are in the pre-trigger buffer.

READ: Valid only when pre-trigger is enabled. 0 = Trigger has not yet occurred. 1 = Trigger occurred.

BIT 7 - DT-CONNECT Eanble/Disable & Status

WRITE: 0 = Disable DT-Connect. 1 = Enable DT-CONNECT.

READ: 0 = DT-CONNECT is disabled. 1 = DT-CONNECT is enabled.

9.2.8 PACER CLOCK DATA & CONTROL REGISTERS

8254 COUNTER 0 DATA
BASE ADDRESS + 12 30C HEX, 780 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

8254 COUNTER 1 DATA
BASE ADDRESS + 13 30D HEX, 781 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

8254 COUNTER 2 DATA
BASE ADDRESS + 14 30E HEX, 782 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

The three 8254 counter/timer data registers may be written to and read from. Because each counter will count as high as 64,536, it is clear that loading or reading the counter data must be a multi-step process. The operation of the 8254 is explained in the section on the counter/time and the Intel 8254 data sheet.

8254 COUNTER CONTROL
BASE ADDRESS + 15 30F HEX, 783 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

This register controls the operation and loading/reading of the counters. The configuration of the 8254 codes which control the 8254 chip is explained in the section on the counter timer and the Intel 8254 data sheet.

9.2.9 ENHANCED FEATURES PACER CLOCK DATA & CONTROL REGISTERS

8254 COUNTER 0 DATA - Chained to counter 1
BASE ADDRESS + 16 310 HEX, 784 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

8254 COUNTER 1 DATA - Chained to counter 2
BASE ADDRESS + 17 311 HEX, 785 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

8254 COUNTER 2 DATA
BASE ADDRESS + 18 312 HEX, 786 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

The three 8254 counter/timer data registers may be written to and read from. Because each counter will count as high as 64,536, it is clear that loading or reading the counter data must be a multi-step process. The operation of the 8254 is explained in the section on the counter/time and the Intel 8254 data sheet.

8254 COUNTER CONTROL
BASE ADDRESS + 19 313 HEX, 787 Decimal

7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1

This register controls the operation and loading/reading of the counters. The configuration of the 8254 codes which control the 8254 chip is explained in the section on the counter timer and the Intel 8254 data sheet.

9.3 ANALOG INPUT

Analog signals connected to P3, the 37D connector which protrudes from the expansion slot of the PC, are first fed into the two HI-0508 analog multiplexors. A multiplexor's (MUX) function is to select on of several (8) inputs and connect that input to the MUX output. MUX U27 connects CH0-CH7 high inputs. MUX U28 connects CH0-CH7 Low input (differential input mode) or CH8-CH15 High inputs (single ended mode) depending on the state of the channel configuration switch located at the upper right of the board and marked 8/16.

From the output of the MUX, the analog signal is fed into a programmable differential amplifier.

The A/D converter chip has an integral sample & hold circuit, greatly simplifying design and improving signal integrity. The A/D converter is capable of sampling rates to 330KHz but the DMA transfer circuitry of the personal computer's 8 bit bus is limited to a maximum synchronous transfer rate of about 130KHz. Therefore the maximum sampling rate of the CIO-AD16JR-AT is up to 130KHz, depending on the computer.

9.4 DIGITAL INPUT & OUTPUT

There are 4 bits of output only and 4 bits of input only on the CIO-AD16JR-AT analog connector. From the original DAS-16 design these were the only 8 bits of digital I/O.

For complete programming information turn to the section on CIO-AD16JR-AT registers.

9.4.1 OUTPUT

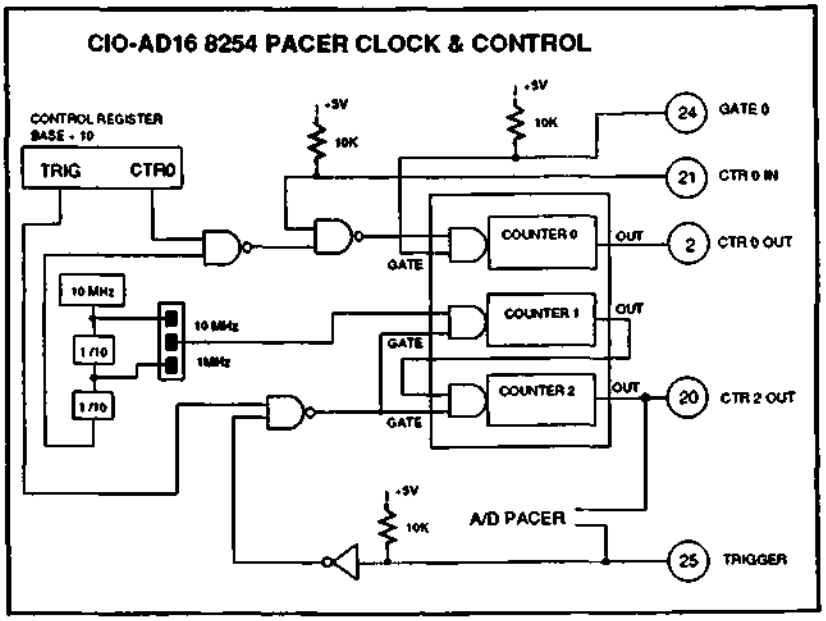
The output bits are part of chip U39, a 74LS197 output buffer. The other half of the chip is used for on board control. If the digital output lines are blown by overload or high voltage connection you may replace this chip.

9.4.2 INPUT

The input bits are part of chip U38, a 74LS244 buffer. The other half of this chip is used for on board functions. This chip is socketed.

9.5 INTERRUPT & TRIGGER CONTROL

The interrupt and trigger control of the CIO-AD16JR-AT is accessible via programming at register BASE + 9 and is really quite simple. Interrupts are enabled by setting a bit. The PC bus interrupt number, 2 to 7 is register programmed. The source, either from the 8254 or external is programmed also.



BASE ADDRESS + 9

309 HEX, 777 Decimal

7	6	5	4	3	2	1	0
INTE	IR4	IR2	IR1	Don't Care	DMA	TS1	TS0

A read and write register.

READ

INTE = 1, Interrupts are enabled. An interrupt generated will be placed on the PC bus interrupt level selected by IR4, IR2 & IR1. INTE = 0, interrupts are disabled.

IR4, IR2, IR1 are bits in a binary number between 0 and 7 which map interrupts onto the PC bus interrupt levels 2 - 7. Interrupts 0 & 1 may not be asserted by the CIO-AD16JR-AT.[23]

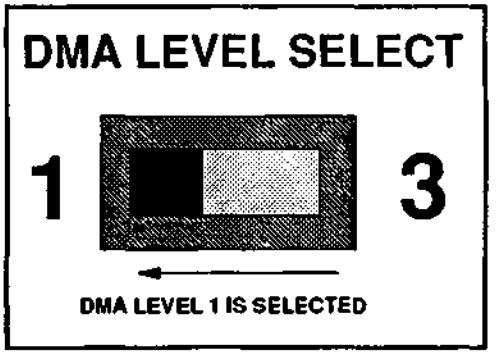
TS1 & TS0 control the source of the A/D start conversion trigger according to the table below.

TS1	TS0	
0	X	Software triggered A/D only.
1	0	Start on rising TRIGGER (Digital input 0, Pin 25)
1	1	Start on Pacer Clock Pulse (CTR 2 OUT, no external access)

9.6 DMA CONTROL LOGIC

The Direct Memory Access (DMA) controller is on the personal computer CPU board, not on the CIO-AD16JR-AT. The CIO-AD16JR-AT has the logic on board to request a DMA transfer. In addition to the on board logic, the DMA controller must be programmed.

Complete register specifications for the CIO-AD16JR-AT DMA control registers will be found in the previous section on the CIO-AD16JR-AT control registers.



A discussion of the CIO-AD16JR-AT DMA level selection and the PC's DMA controller will be found in the section on the architecture of the PC.

23 Information on the assignment of interrupt levels to standard PC peripherals is found in the section on the architecture of the PC. In other words, not all interrupts 2-7 are available without conflict.

9.7 SPECIFICATIONS

POWER CONSUMPTION

+5	450mA MAX.	375mA Typical
+12	0V	
-12	0V	

ANALOG INPUT

Channels	8 Differential	16 Single Ended
Resolution	12 Bits	1 part in 4095
Accuracy	0.01% of reading, ±1 bit	
Input Range	±10V MAX	± 0.625 MIN
	See table in section on Analog Input	
Coding	Bipolar	Offset binary
	Unipolar	True binary
Overvoltage	±35V continuous	
Input Current	250 nA MAX	12 nA Min @ 25 deg C
Input Impedance	10 Meg Ohm	
Temp. Coeff.	Full Scale	±25 ppm/deg C MAX
	Zero	±12 ppm/dec C MAX
Type	Successive Approximation	
Conversion Time		3 uSec MAX
Model		
Monotonicity	Guaranteed over operating temp. range.	
Linearity	±1 Bit	
Zero drift	±10 ppm/deg C MAX	
Gain drift	±30 ppm/deg C MAX	
Vref Output	-5V ±0.05V	

PROGRAMMABLE TIMER

Type	82C54	
Counters	3, 16 Bit down counters	
	2 dedicated to A/D pacer	
XTAL	1 or 10 MHz	
Output Drive	2.2mA @ 0.45V	
Input	TTL	
Frequency	DC to 10MHz	
Active count edge	Negative	
Clock pulse width	30nSec High	50nSec Low

DIGITAL I/O

Input only	4 bits	74LS244
Output only	4 bits	74LS374
Input low volts	74LS244	0.8V MAX
Input high volts	74LS244	2.0V MIN @ 20uA
Output low volts	74LS374	0.5V MAX @ 8.0mA
Output high volts	74LS374	2.4V MIN @-0.4mA

INTERRUPT

Level	2-7	Software programmable
Enable	INTE Bit	CIO-AD16JR-AT Control register
Trigger	On board	8254 pacer clock
	External	User supplied TTL pulse

DMA A/D TRANSFER

Level	1 or 3	Switch selectable
Enable	DMA Bit	CIO-AD16JR-AT Control register
Termination	Single Cycle	Interrupt on terminal count
	Recycle	
Pacing	Internal	8254 pacer clock
	External	User supplied TTL pulse

POWER OUTPUTS

Direct from PC bus		See you PC spec.
Voltage	±5V ±5%	Depends on PC power
Loading	22AWG	64.9 Ohms/ 1000 feet

ENVIRONMENTAL

CIO-AD16JR-AT	*Normal Temp Model*
Operating Temp	0 to 50 Deg C
Storage Temp	-20 to 70 Deg C
Humidity	0 to 90% non-condensing
Weight	10.25 Oz
CIO-AD16JR-AT	*Extended Temp Model*
Operating Temp	-30 to 60 Deg C
Storage Temp	-165 to 150 Deg C
Humidity	0 to 90% non condensing
Weight	10.25 Oz

10 APPLICATION NOTES

Application notes detail specific implementations of a PC, CIO-AD16JR-AT, software, and signal source. Some information about the goals of the application may be provided.

Please contribute if you can.

If you provide measurement solutions and there is one that uses a Computer Boards board, we will be glad to mention your company and your solution in our application notes.

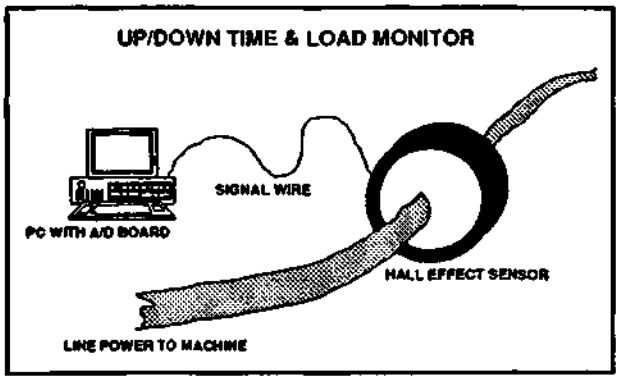
These application notes are presented to illustrate applications of I/O boards and are meant to be a guide or to suggest applications and get *your* creative juices flowing. In some cases these notes are donated by customers and are mearly outlines of a project, lacking complete technical details required to complete the application. If the application is of interest to you, you may need to acquire additional electrical or programming information on you own. *Good luck!*

10.1 UP-TIME / DOWN TIME & LOAD MONITOR

The current load on industrial machinery which is powered with electric motors can be monitored with a personal computer and data acquisition board connected to Hall Effect transducers (current to voltage). Information on power consumption can provide manufacturing managers with:

- Up time vs. down time.
- Percentage of work time (heavy current load) vs. non work time (machine on but under light load).
- Sharpness of cutting tools (dull tools use more power for a given output).

Each of these parameters may be monitored by one analog input channel. A board such as the CIO-AD08 can monitor 8 machines while a board such as the CIO-AD16JR-AT can monitor 16 machines. If more machines are to be monitored and external analog multiplexor, CIO-MUXTC32, can be employed to monitor as many as 256 machines.



This application was brought to my attention by an engineer from a large paper company. They were having trouble monitoring up/down time and load on the bark removal chipper in the wood lots adjacent to the pulp mill. The chippers were dispersed over a wide area and were difficult to supervise.

The chipper operators also needed better information about the blade condition. Blade wear was unpredictable. If the logs had been kept out of the mud then blade wear was low. If there was a high content of mud and gravel on the logs blade wear was high. Dull blades meant that motors and bearing may be overstressed. Sharpening blades cost money and should only be done when needed. Only experienced chipper operators were good at judging blade condition, but, current load per output was a very accurate measure.

These problems are analogous to problems in any production cutting tool environment and the concepts may apply to most industrial equipment powered by electrical motors.

The free software, LABLOG2, supplied with your CIO-AD board is more than adequate for the task of monitoring the load of electrical motors. If higher speeds or specialized displays and reports are needed, a program may be written in BASIC using the CALL routine software.

Hall effect transducers are available from a number of sources. We have had good luck with F.W.Bell.

10.2 PC COLLECTS DATA FROM TEXTURE ANALYZER

OR ANY LABORATORY INSTRUMENT WITH A CHART RECORDER OUTPUT

Many laboratory instruments may be interfaced directly to a PC even though no provision for such a connection was made in the original design of the instrument. A PC equipped with an A/D board can read the signals from the instrument's chart recorder plug. The A/D board converts the variable voltage from the chart recorder plug into numbers. These numbers represent the magnitude of the voltage at a specific point in time and are exactly proportional to the engineering units of the instrument. These numbers may be read directly into many software programs designed to aid analysis.

An example of such an instrument is the Voland texture analyzer 1000 penetrometer (Voland Corp., Hawthorne, NY. Penetrometers (also called Densitometers) are used to measure adhesion, bloom strength, breaking point, crispness, density, tackiness and other parameters. I first encountered the texture analyzer in a donut franchisor's food laboratory, where studies of perceived freshness were being correlated to measurements of the donut's crust, texture and tackiness.

To understand why anyone would want to interface a texture analyzer to a PC, consider the case of the donut franchisor's perceived freshness studies. Hundreds of samples had to be taken from donuts in varying states of freshness. Each donut had to be sampled and the taster's impressions recorded. The donut was then put into the texture analyzer, where a chart was created on the chart recorder connected to the texture analyzer. Later, these charts had to be transcribed into numeric values and fed into a computer to compute correlation. That meant a great deal of measuring, writing, and keyboarding. In sheer frustration at the volume of work being done just to prepare the data for analysis, the food scientist began searching for a way to feed the data from the texture analyzer directly into a PC.

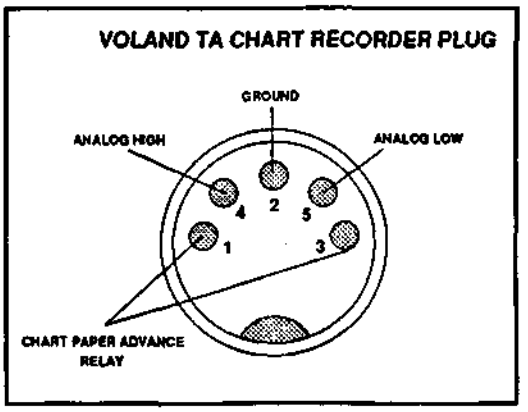
INTERFACE

This texture analyzer was not designed with a standard computer interface such as IEEE-488, RS422, or RS232C. Most laboratory data gathering instruments are not equipped with a communications interface but are equipped with a chart recorder interface. It is simple to monitor the signals from the chart recorder interface plug on the Voland TA.

The chart recorder interface plug on the Voland TA is a 5 wire, round DIN plug.

The mating connector may be purchased from Radio Shack.

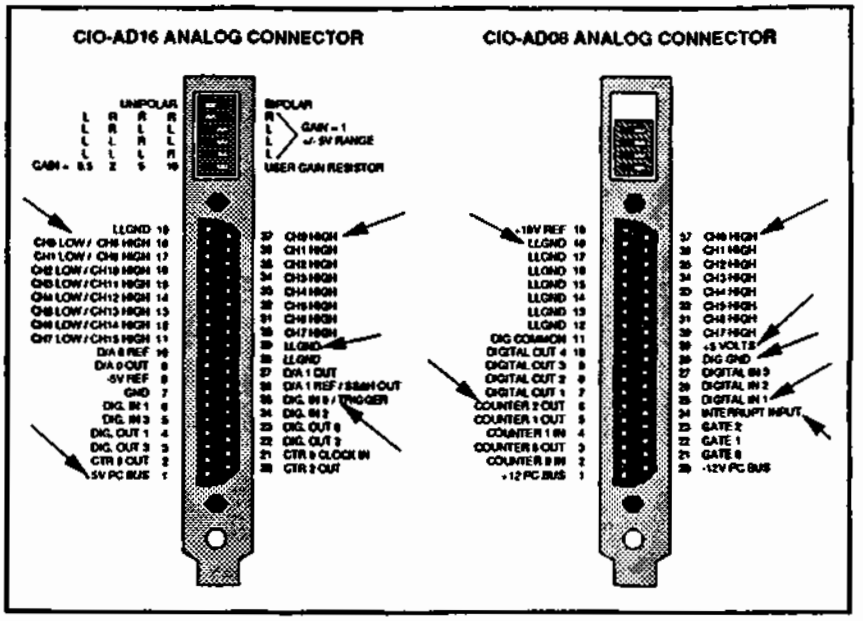
The voltage output of the TA varies between +/- 2.5 volts, and corresponds to the +/- 500 gram range of the TA.



The output of the Voland TA chart recorder plug is simple and easy to understand and interface. An analog voltage (pin 4) drives the chart recorder pen and a voltage return completes the circuit (pin 5). An electromechanical relay (pins 1 and 3) in the instrument closes when the probe makes contact with the subject. This relay is meant to start the paper advance motor in the chart recorder. A ground signal (pin 2) provides the needed ground reference for the interface.

The connector diagram for the CIO-AD16JR-AT and CIO-AD08 is shown here with the signals of interest highlighted.

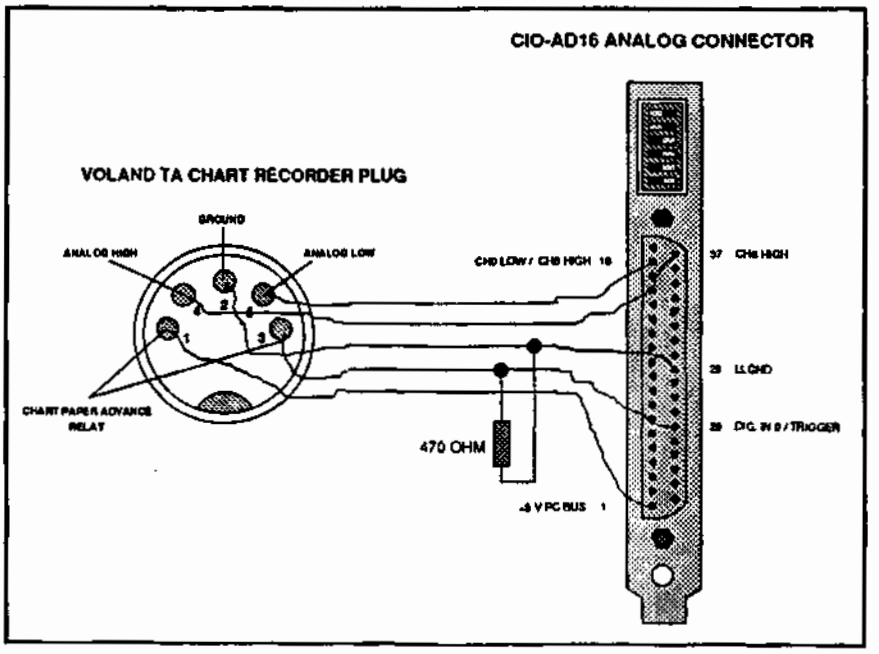
The CIO-AD16JR-AT input range may be exactly matched to the TA, at +/- 2.5 volts. The CIO-AD08 may be set for +/- 5 volts.

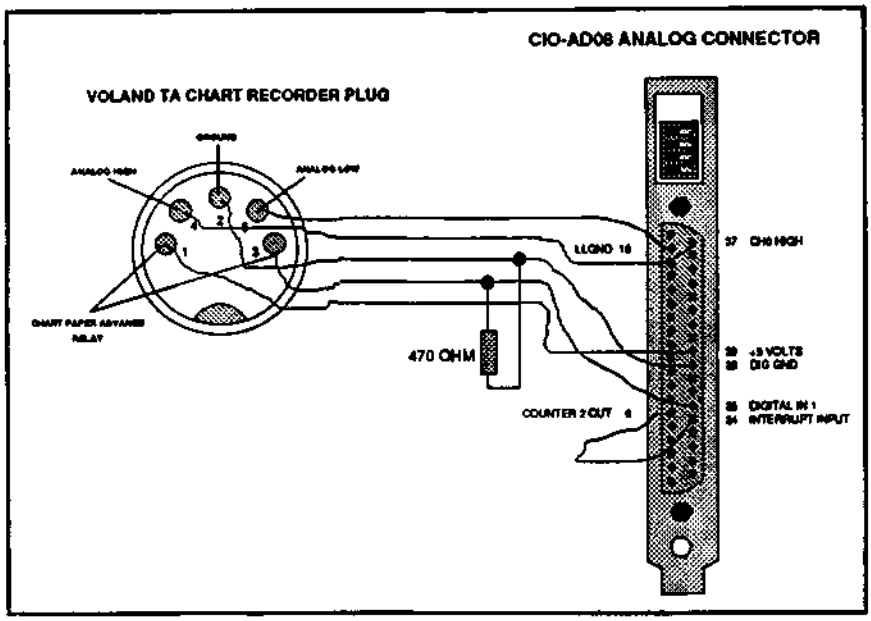


A high quality cable may be constructed of materials available from Radio Shack. You will need to purchase one 5 pin DIN connector assembly, one 37 pin D type connector assembly and a length of shielded cable. The instrument manufacture will specify a maximum cable length. Voland recommends it be as short as possible or 2 meters maximum. If you cannot find the material for a cable, call Computer Boards' Tech Support and we will make one for you.

The two cables shown above exactly match the signal and triggering requirements of the two A/D boards and the Voland TA.

The CIO-AD16JR-AT has an on board sample pacer clock which is hard wired to the A/D timing circuitry. A trigger input (pin 25) monitors the TA's chart motor relay so no samples are taken until the probe makes contact with the material under test.





CIO-AD08 has an on board pacer clock output (pin 6) which must be used to supply interrupts to the interrupt input (pin 24). A trigger input (pin 25) monitors the TA's chart motor relay so no samples are taken until the probe makes contact with the material under test.

The specifications for the Voland TA chart recorder interface were taken directly from the Voland manual. Most instruments with chart recorder plugs have instructions for connecting a chart recorder. Those instructions will have a chart recorder plug pin out like the one shown in the diagram above. Using that information and the information in this manual it should be easy to design the cable which connects the instrument to the CIO-AD board. If you have questions or trouble, FAX a copy of the instruments cable pin out to Tech Support and we will help you.

VERIFICATION

To test the interface between the instrument and A/D board, run the program I/O test while putting the instrument through a data gathering exercise. The program I/O test will monitor the analog input channel and display a strip chart of the voltage input on the PC screen. The data may or may not look 'right', but the point of this test is merely to verify that a variable voltage from the instrument is being measured by the A/D board.

SOFTWARE

To gather and analyze data, some software to control the A/D board, gather and store the data, and analyze or graph the results will be needed. Here is the sequence of events as they occur in a data gathering cycle which define the software requirements.

SETUP: The software must be configured to sample data from one analog channel at a known rate of samples per second. No sampling should begin until valid data is present on the instrument's chart output. That means that the digital line described in the cable interface must be monitored for the chart paper motor start signal.

DISPLAY: Depending on the rate of data collection and your real-time observation needs, real-time graphing of the data (graphing as it is being gathered) may or may not be possible. The maximum speed at which data can be graphed is dependent on processor and display board type. The software manufacture should have that specification in their manual. The IOTEST program mentioned above will run at 130 Hz on a 16MHz 386 PC. Much faster rates are available with Labtech Notebook and other packages.

TRIGGERING: The software must be able to hold off data acquisition and storage until there is valid data. To see why this is so, imaging that you have set up the PC to gather data at 100 samples per second and that you start it then walk over to the instrument. If one were very quick only 500 or so (5 seconds) of worthless samples would have been gathered. These would have to be removed from the data files. Even more important, the time between the trigger event and a peak in the samples may be important.

PACING: The software must be able to pace the acquisition of analog samples precisely. The CIO-AD boards provide the resources for precise pacing.

DATA ACQUISITION & STORAGE: Data must be gathered and stored, and it is best if the program offers flexibility in the storage format. The data file may then be read by other programs, such as SPS, which do not have data acquisition capability.

ANALYSIS: Most of the good data acquiring programs are weak on analysis. The food scientist who introduced me to the Voland TA was using Lotus 123 for analysis.

CHARTING & DISPLAY: Once I heard the job of research scientist defined as a three step process involving data gathering, analysis and communication of conclusions. If that is the case then the communication of conclusions and the recognition that go with it are the payoff for the other two steps. Any written communication is improved by attractive exhibits and the PC excels at attractive graphs and charts. Lotus 123 and other spreadsheet packages have extensive chart preparation and production facilities. So do advanced word processors and desktop publishing programs. The task of preparing presentation material is best left to these packages as none of the current data acquisition packages provide the level of presentation capability the spreadsheet and publishing packages do.

DATA ANALYSIS

The data in the file after an acquisition run are the A/D counts from the analog input data converter on the CIO-AD board. These counts are proportional to the voltage present on the analog input at the time the conversion was made. The voltage present on the analog input (the output of the chart plug) is proportional to the engineering units of the instrument.

Here is a simple example.

When the Texture Analyzer reads 250 grams of pressure,

the chart output is at 1.25 volts. (0 grams = 0V, 500 grams = 2.5V)

1.25 Volts of analog input = 512 counts (the number in the data file)

If the example is not obvious don't be discouraged. The math and the concepts are simple. They look hard because it is new to you. Please read the explanation of converting A/D data to engineering units in the section on analog electronics.

CONCLUSION

A new technology may be interesting but its value must come in the form of a measurable benefit. For the food scientist measuring the parameters of perceived freshness with a texture analyzer, the benefit was the elimination of hours of tedium and expense as well as immediate feedback of the test results. New recipes could be developed faster. For a laboratory the benefits could be the following:

- Upgrading existing equipment to computer-controlled capability at minimal expense.
- Elimination of tedious data entry.
- Immediate feedback of test results.
- Ability to share data with corporate or other laboratories on standard IBM PC format disks.
- More complete and easy to access archiving of government or insurance related measurements of quality and fitness.

Instruments are being developed with more and more microprocessor-controlled features. In many instances the features are hardwired into the design of the instrument and thus raise its cost. A PC interfaced to an instrument provides total flexibility and control and allows the addition of new features as new hardware and software are introduced. This flexibility, coupled with the popularity of personal computing, is why PCs and instruments together are such an attractive solution to laboratory measurement and analysis challenges.

10.3 FREQUENCY MEASUREMENT WITH A CIO-AD16JR-AT

It is possible to measure frequency with a CIO-AD16JR-AT using all three of the 16 bit counters, one digital input and a short BASIC program to control the counters and calculate the frequency. You will need some BASIC programming skills and an 8254 data sheet to complete this application.

It is suggested that and oscilloscope be available while you are learning to program the 8254 counters. The output of the counters should be checked visually to see that the wave form and timing are as you believe your program to be setting them.

THEORY OF OPERATION

The unknown frequency provides the clock input to a counter. The gate of the counter is controlled by a known frequency that is at least 1/10th but not less than 1/64,000th the rate of the unknown frequency.

SIGNAL CONNECTION

CLK1: The input of counter 1, CLK1 is hardwired to the on board XTAL clock. The XTAL frequency is 1MHz. There is a divide by 10 the output of which is 1MHz. By choice of jumper selection, either 10MHz or 1MHz may be the input of CLK1.

OUT1: The counter 1 is programmed so that the output of counter 1, OUT1, is a rate = to CLK1 divided by the divisor loaded into counter 1. The minimum is 2, the maximum is 65,535.

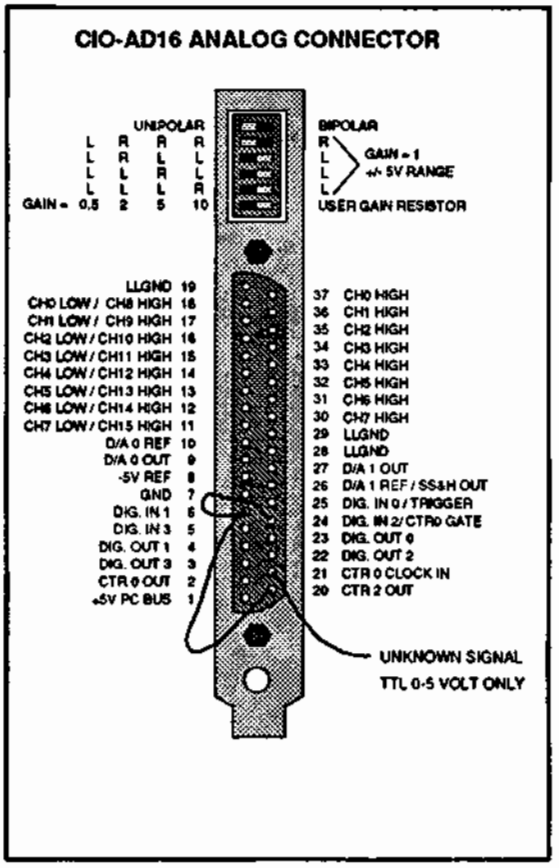
CLK2: The input to counter 2 is hardwired to the output of counter 1, OUT1

OUT2: The counter 2 is programmed so that the output of counter 2, OUT2, is a rate = to CLK2 divided by the divisor loaded into counter 2. The minimum is 2, the maximum is 65,535.

IP1: Digital input 1, IP1, must be wired to the output of counter 2, OUT2. IP1 is monitored to sense transitions on OUT2.

GATE0: The output of counter 2, OUT2, must be wired to the gate of counter 0. The gate to counter 0, GATE0, is controlled by the output of counter 2, OUT2. Pulses on counter 0 input, CLK0, are inhibited when the GATE0 is low.

CLK0: The input to counter 0 must be connected to the unknown frequency. Counter 0 is programmed as a count down register, counting down from 65,535 on each low-to-high transition.



SOFTWARE SEQUENCE

1) Program counters 1 and 2 as rate dividers. You will need an 8254 data sheet (from Intel or AMD). Program the counters 1 and 2 as square wave rate dividers and program counter 0 as a count down counter. Load counters 1 and 2 with the appropriate divisors to generate a counter output square wave of the correct frequency (no more than 1/10th the unknown frequency minimum). See the timing diagram.
2) Program counter 0 for using BASIC CALL mode 10 for 8254 mode 0, count down.
3) Start monitoring IP1 using BASIC CALL mode 14. When IP1 makes a high to low transition, load counter 0 with 65,535 using BASIC CALL mode 11.

4) Continue monitoring IP1. When IP1 makes a second high to low transition, read the contents of counter using BASIC CALL mode 12. You must read the contents of the counter before IP1, GATE0 goes high again.

5) The unknown frequency may be determined by dividing the known rate, OUT2, by the value in counter 0 * 2 (times two because the GATE0 is open for 1/2 the duty cycle of OUT2)

6) Check your CIO-AD16JR-AT utility disk. Doug Haynes of Computer Boards may have finshed writing and debugging a QBASIC program that impliments this application note.

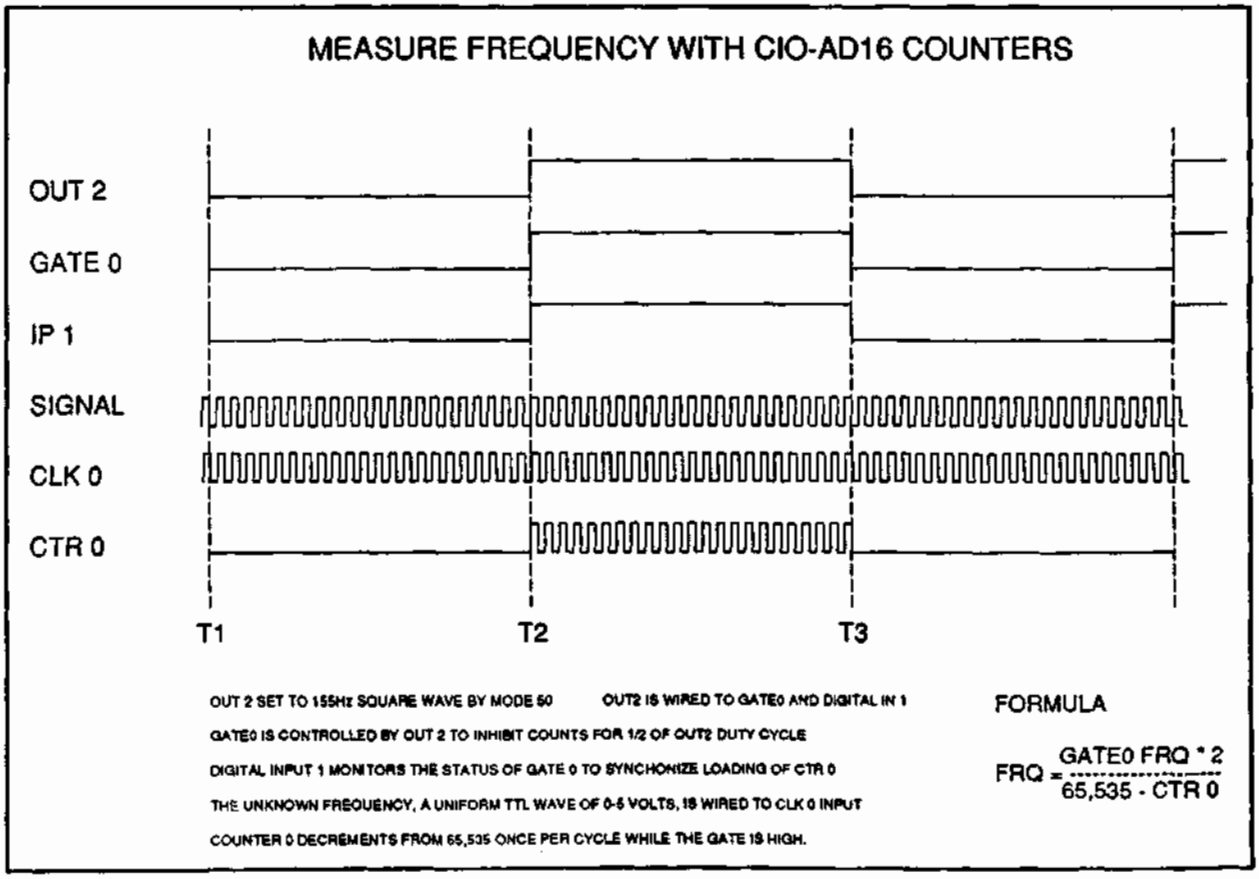
FREQUENCY MEASUREMENT LIMITS

The following calculations are for the XTAL jumper in the 1MHz position. If the XTAL jumper were moved to the 10MHz position, the limits below would increase by a factor of 10. The limit for the unknown frequency, CLK1, assumes that 10 duty cycles of the unknown frequency are required to make an accurate measurement. Adjust this assumption as you see fit. The 8254 is limited to 10MHz inputs.

SIGNAL	MIN	MAX	EQUATION
OUT1	15.26Hz	500,000Hz	CLK1 ÷ (65,535 ≥ D ≥ 2)
OUT2	0.00023Hz	250,000Hz	CLK2 ÷ (65,535 ≥ D ≥ 2)
CLK1	0.0023Hz	10,000,000Hz	10MHz ≥ CLK ≥ (OUT2 * 10)

Interestingly enough, loading counters 1&2 for an output of 155Hz allows you to measure unknown frequencies between 1500Hz and 10Mhz. An OUT2 of 155Hz is more than slow enough to allow for all the software overhead of monitoring IP1.

A timing chart and formula for counters 1&2 set to 155Hz is shown here.



NOTE: The frequency of OUT2 (GATE0) ***MUST*** be less than the unknown frequency; 1/10th is recommended.

11 ANALOG ELECTRONICS

This short, simple introduction to the analog electronics most often needed by data acquisition board users covers a few key concepts. They are:

- Voltage dividers.
- Differential vs. Single Ended Inputs.
- Isolation vs. Common Mode Range
- Low pass filters for analog and digital inputs.
- A/D Resolution
- Conversion to Engineering units.
- 4-20 mA inputs
- Noise; sources and solutions.
- Digital interfacing.

Each deals with the impact on measurements made with data acquisition boards. If you are truly interested in the subject of data converters and analog electronics, Radio Shack has written an excellent Op Amp handbook and has an Op Amp experimenters kit.

A more advanced treatment may be found in the Analog-Digital Conversion Handbook (32.95) and the Transducer Interfacing Handbook ($14.50) published by Prentice-Hall.

11.1 VOLTAGE DIVIDERS

If you wish to measure a signal which varies over a range greater than the input range of an analog or digital input, a voltage divider can drop the voltage of the input signal to the level the analog or digital input can measure.

A voltage divider takes advantage of Ohm's law, which states,

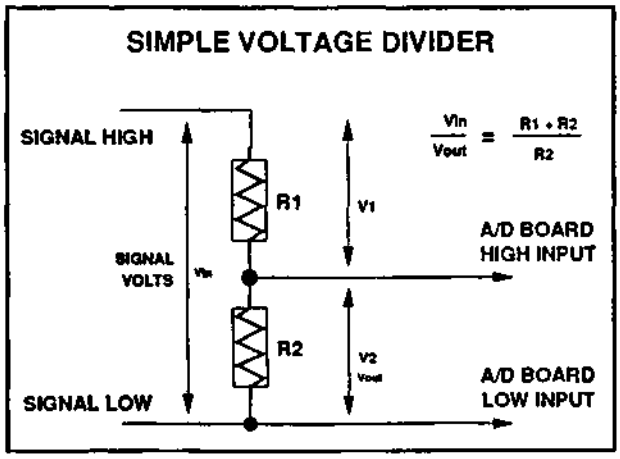
Voltage = Current * Resistance

and Kirkoff's voltage law which states,

The sum of the voltage drops around a circuit will be equal to the voltage drop for the entire circuit.

Implied in the above is that any variation in the voltage drop for the circuit as a whole will have a *proportional* variation in all the voltage drops in the circuit.

A voltage divider takes advantage of the fact that the voltage across one of the resistors in a circuit is proportional to the voltage across the total resistance in the circuit.
The trick to using a voltage divider is to choose two resistors with the proper proportions relative to the full scale of the analog or digital input and the maximum signal voltage.



The phenomena of dropping the voltage proportionally is often called attenuation. The formula for attenuation is:

Attenuation =	R1 + R2 -------- R2	The variable *Attenuation* is the proportional difference between the signal voltage max and the full scale of the analog input.
2 =	10K + 10K ---------- 10K	For example, if the signal varies between 0 and 20 volts and you wish to measure that with an analog input with a full scale range of 0 to 10 volts, the *Attenuation* is 2:1 or just 2.
R1 =	(A - 1) * R2	For a given attenuation, pick a handy resistor and call it R2, then use this formula to calculate R1.

Digital inputs also make use of voltage dividers, for example, if you wish to measure a digital signal that is at 0 volts when off and 24 volts when on, you cannot connect that directly to the CIO-AD digital inputs. The voltage must be dropped to 5 volts max when on. The *Attenuation* is 24:5 or 4.8. Use the equation above to find an appropriate R1 if R2 is 1K. Remember that a TTL input is 'on' when the input voltage is greater than 2.5 volts.

IMPORTANT NOTE: The resistors, R1 and R2, are going to dissipate all the power in the divider circuit according to the equation Current = Voltage / Resistance. The higher the value of the resistance (R1 + R2) the less power dissipated by the divider circuit. Here is a simple rule:

For Attenuation of 5:1 or less, no resistor should be less than 10K.

For Attenuation of greater than 5:1, no resistor should be less than 1K.

The CIO-TERMINAL has the circuitry on board to create custom voltage dividers. The CIO-TERMINAL is a 16" by 4" screw terminal board with two 37 pin D type connectors and 56 screw terminals (12 - 22 AWG). Designed for table top, wall or rack mounting, the board provides prototype, divider circuit, filter circuit and pull-up resistor positions which you may complete with the proper value components for your application.

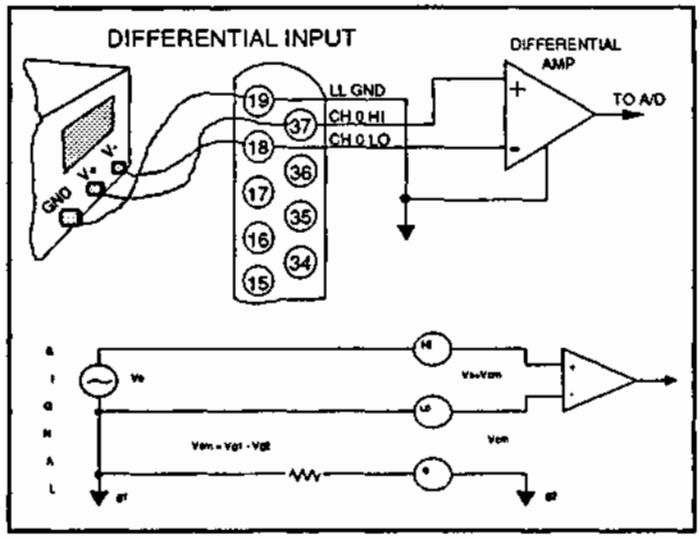
11.2 DIFFERENTIAL & SINGLE ENDED INPUTS

Two type of analog inputs are commonly found on A/D boards, they are differential and single ended. Single ended is the less expensive of the two.

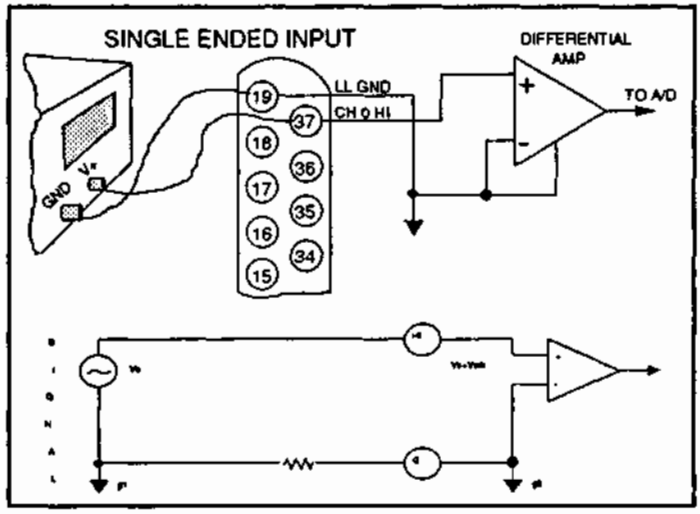
COMMON MODE RANGE

Differential inputs have a common mode range (CMR) (Vcm). Single ended inputs have no CMR . Common mode range is the voltage range over which differences in the low side of the signal and A/D input ground have no impact on the A/D's measurement of the signal voltage. A differential input can reject differences between signal ground and PC ground.

Shown here is a CIO-AD16JR-AT in differential mode.



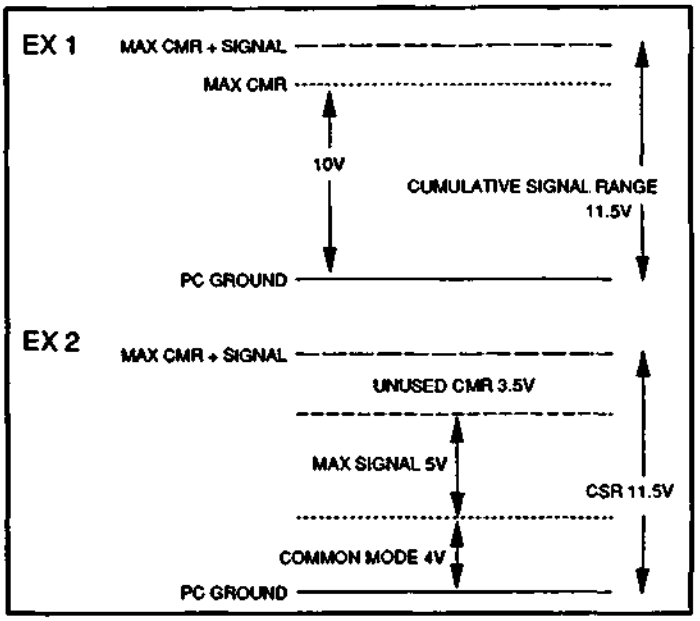
A single ended input has no common mode range because there is only one LOW wire, which is assumed to be at the same level at the signal and at the A/D board.



The maximum difference which may be rejected is the CMR.

For example, the CIO-AD16JR-AT has a common mode plus signal range of 11.5 volts, common mode not to exceed 10 volts.

This specification is illustrated graphically here and will be referred to as **Cumulative Signal Range (CSR).**



Most manufactures of A/D boards specify the CMR directly from the component data sheet, ignoring the effect of the board level system on that specification. A data sheet of that type might claim 10 volts of CMR. Although this is a factual specification and the designer of the board (or other EE) would be able to translate that into a systems specification, most A/D board owners are confused or mislead by such specs.

COMMON MISUNDERSTANDINGS

The CMR specification of a differential input is often confused with an isolation specification, which it is not. It makes sense. doesn't it, that 10 volts of CMR is the same as 10 volts of isolation? No. The graph above shows why.

Also, failure to specify the common mode plus signal *system* specification leads people to believe that a DC offset equal to the component CMR can be rejected regardless of the input signal voltage. It cannot as the graph above illustrates.

When is a differential input useful? The best answer is whenever electromagnetic interference (EMI) or radio frequency interference (RFI) may be present in the path of the signal wires. EMI and RFI can induce voltages on *both* signal wires and the effect on single ended inputs is generally a voltage fluctuation between signal high and signal ground.

A differential input is not affected in that way. When the signal high and signal low of a differential input have EMI or RFI voltage induced on them, that ***common mode voltage*** is rejected, subject to the system constraint that common mode plus signal not exceed the A/D board's CSR specification.

GROUND LOOPS

Ground loops are circuits (E=I*R) created when the signal ground and the PC ground are not the same. Ground loop inducing voltage differential may be a few volts of hundreds of volts. They may be constant or transient (spikes). A differential input will prevent a ground loop as long as the CSR specifications is not exceeded.

If ground differences greater than the CMR are encountered, isolation is required.

WHY USE SINGLE ENDED?

Unanswered still is the question of why use single ended inputs? First, single ended inputs require fewer parts so they cost less money. On an A/D board the parts cost to go from 16 single ended channels to 16 differential channels is about $6.00 so that cannot be the reason. The real reason is connector space. Single ended inputs require one analog high input per channel and one LLGND shared by all inputs. Differential inputs require signal high and signal low inputs for each channel and one common shared LLGND.

Single ended inputs save connector space, parts cost and in all cases where there is no common mode or EMI/RFI they work just as well as differential inputs and are less complex to wire up.

11.3 LOW PASS FILTERS

A low pass filter is placed on the signal wires between a signal and an A/D board. It stops frequencies greater than the cut off frequency from entering the A/D board's analog *or* digital inputs.

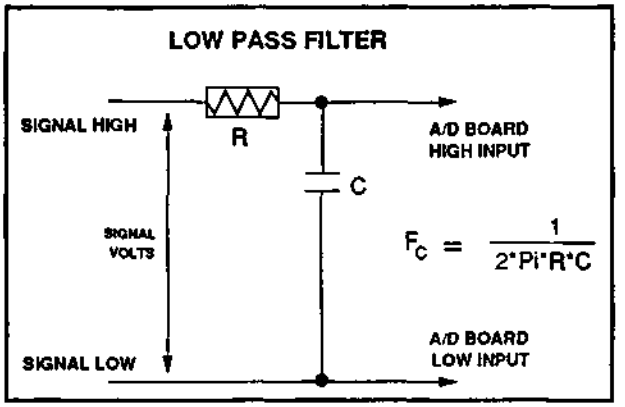
The key term in a low pass filter circuit is **cut off frequency**. The cut of frequency is that frequency above which no variation of voltage with respect to time may enter the circuit. For example, if a low pass filter had a cut off frequency of 30 Hz, the kind of interference associated with line voltage (60Hz) would be filtered out but a signal of 25Hz would be allowed to pass.

Also, in a digital circuit, a low pass filter might be used to de-bounce an input from a momentary contact button pushed by a person.

A low pass filter may be constructed from one resistor (R) and one capacitor (C). The cut off frequency is determined according to the formula:

$$F_c = \frac{1}{2 * Pi * R * C}$$

$$R = \frac{1}{2 * Pi * C * F_c}$$



11.4 A/D RESOLUTION & ENGINEERING UNITS

Resolution is specified in bits, such as 8, 10 or 12 bits. The 12 bits are really a power of 2 indicating the number of divisions of full scale the converter can resolve. For example, a 12 bit converter can resolve (2^12 - 1) = 4095 divisions of full scale. If the input of the board were +/- 5 volts full scale, each of the 4095 steps would be equal to 0.00244 volts.

Reading from the A/D converter would be thus:

Converter #	Volts
4095	5.0
4094	4.9976
4093	4.9951
2028	0.0
1	- 4.9976
0	- 5.0

The concept of resolution is the ability to differentiate between one voltage and another. Obviously, the more bits of resolution (13 bits = 8192 counts) the more divisions of full scale. The more divisions of full scale the finer the measurement.

ENGINEERING UNITS

When a program uses an A/D board to acquire data, the data file is filled with numbers like those above.

To translate the A/D numbers back into the engineering units of the original measurement, we need to know:

The sensor's voltage output per engineering unit.

The full scale range of the board at the time the measurement was made.

The resolution of the converter.

Here is an example from the application note on interfacing a Voland TA to a PC found elsewhere in this manual.

The TA measures resistance in grams between +500 and -500 grams.

The voltage output of the instruments is +2.5 volts to -2.5 volts.

The voltage output corresponds to the grams of pressure exactly, so:

+/- 500 grams = 1000 grams.

+/- 2.5 volts = 5 volts.

5 volts / 1000 grams = 0.005 volts per gram.

The A/D was set for +/- 2.5 volts = 5 volts full scale.

5 volts / 4095 counts = 0.00122 volts per bit.

If the number in the file for one reading was 3061, then

3061 * 0.00122 = 3.7632 volts.

3.7632 volts / 0.005 volts per gram = 735 grams.

Now shift from full scale to +/- scale.

735 grams full scale - 500 = 235 grams of positive pressure.

It may look like a lot of steps because it is presented that way here for clarity only. It could be expressed as a single equation in a spreadsheet.

11.5 CURRENT LOOP 4-20 mA

Although the inputs of a CIO-AD board are voltage inputs, it is easy to convert a current to a proportional voltage which may be measured by the CIO-AD board. The current is converted to a proportional voltage by the formula V=I*R.

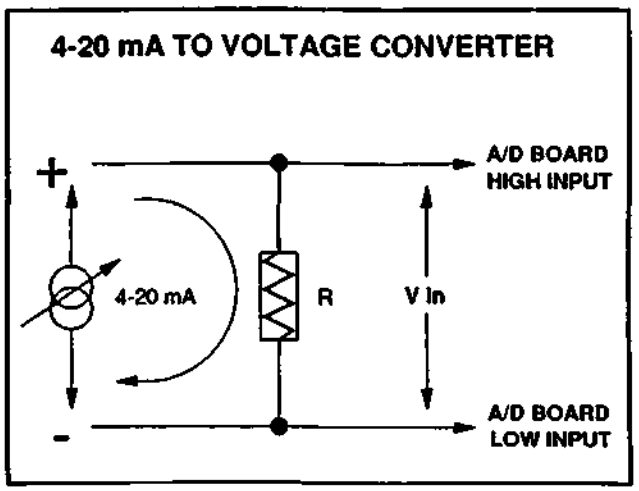
For example, if the CIO-AD board is set up to read 0 to 5 volts, then:

5 volts / 0.02 Amps = 250 Ohm shunt resistor.

So a full 20 mA will register 5 volts and 4 mA will register 1 volt.

To hook up the CIO-AD analog inputs to a 4-20 mA transducer or signal source, place the shunt resistor across the plus and minus terminals or signal wires of the 4-20 mA. Once the resistor is in place, connect the analog input CH# Hi to the plus terminal and the analog input CH# Lo to the minus input.

If they are backward, the A/D reading will be 0 or minus volts. Just reverse the connection.



11.6 NOISE

Noise is unavoidable in PC based data acquisition systems. There is board induced noise which can be measured by shorting an analog input to ground and taking a series of readings and plotting them in a histogram. There is EMI and RFI induced noise along the path of the signal wires. There is also noise at the signal source itself. All these sources of noise combine to create a region of uncertainty around the signal value.

Our objective here is to discover the sources of noise and discuss the means to reduce it.

SOURCES OF NOISE

The first source of noise is the board itself. Manufacturers of A/D boards quote component specifications in their data sheets but rarely quote a system specification for general accuracy and noise. The reasons the system is not specified are that the system specification would be less accurate than component specification and that system specifications must also specify the conditions under which the specification was made. That means the PC, the PC's power supply and the connection to the front end.

No one likes to admit it but it is true. Take some very good components, put them on a circuit board and place that board in a PC and the system will be less accurate than the individual components. I have seen 12 bit A/D boards with the same components as a DAS-16 with as little as 9 bits of accuracy, due to board noise.

The system specification for the CIO-AD16JR-AT and CIO-AD08 is plus or minus 1 LSB. That means that if an analog input is tied to ground and the CIO-AD16JR-AT is on a bipolar scale, the reading will be 2048 90% of the time. The other 10% of the readings will be 2047 and 2049, which is +/- count (LSB). This is actually not very different from the component specifications.

You can verify this by grounding an analog input channel to LLGND and taking 1000 readings then plotting a histogram of those readings.

If your histogram is not +/- 1 LSB, check the +/- 12V PC power supply voltages or call Computer Boards' Tech Support.

SIGNAL WIRE NOISE

Signal wires, especially single ended inputs, are subject to EMI and RFI, both of which can induce noise on the wires carrying the transducer signal to the CIO-AD board. Fortunately, signal wire noise is often localized and can be reduced by repositioning the signal wire run.

To check for signal wire noise, first, short analog channel 0 to low level ground[24] at the connector and take 10,000 samples and plot the histogram. This is the best the signal can be and is what you will try to achieve with the signal wires in place.

After you have an ideal case histogram, remove the short between analog input 0 and low level ground. Attach the signal wires to the CIO-AD board inputs and run them to the sensor. Do not connect the sensor yet, just short the analog input(s) to LLGND.

Take data for the histogram and compare it to the best case data taken previously. If it shows noise, you can try to eliminate the noise by doing the following:

- Move the signal wires, trying to locate a 'quiet' run.
- Use a shielded twisted pair as the signal wire. Attach the shield at the PC **only.** If the shield is attached at both the PC and the sensor it may create a ground loop and add to signal interference.

SENSOR NOISE

When the signal wires have been tested and characterized for signal quality, connect the sensor and provide a known level to the sensor (ice bath for temp., etc.) then take data for the histogram plot. If additional noise has been introduced by the sensor *which exceeds the sensor specifications*, you can try moving the sensor or electrically isolating it from the device it is measuring.

---

24 Differential inputs must have CH Hi, CH Lo and LLGND shorted. Single ended inputs must have CH Hi shorted to LLGND.

SMOOTHING DATA

It is not always possible to eliminate all noise, especially with very low level sensors, but noise looks terrible when plotted and can raise doubts about otherwise excellent data.

There are two simple ways to eliminate noise from the data:

1) Apply a moving average to the data if you want to retain the same *apparent* accuracy.

2) Remove the information from the noisy range. For example, if the A/D is capable of , or, shift the data by the number of counts of noise. For example, if a 12 bit A/D converter is at +/- 5 volts (10 volts full scale) then one LSB = 10 / 4095 = 0.00244mV. If your system is inducing +/- 0.007mV of noise (+/- 3 counts), then just round all the readings by +/- 3 counts. In this way the reading's value reflects the true accuracy of the system.

11.7 DIGITAL INTERFACING

11.7.1 TTL TO TTL SIGNAL CONNECTION

TTL to TTL is the simplest digital connection to make. The voltage levels at the signal source and the CIO-DIO are the same and the load (current drain per connection) is easy to calculate.

INPUTS: Connect your signal source directly into CIO-DIO48 P1 compatible digital inputs or to any of the CIO-DIO96 P1 or P2 digital inputs.

OUTPUTS: The CIO-DIO48 and CIO-DIO96 P1 and P2 connectors may be used for TTL output. Connect your TTL device directly to the P1 or P2 connector.

The three varieties of TTL chip are TTL, LSTTL and CMOS TTL. In order of mention the power consumption of each type of chip decreases with CMOS being the lowest power consumption.

A TTL output pin will drive the input pins of at least 8 other TTL input pins or LSTTL input pins. A LSTTL output pin will drive at least 8 LSTTL input pins or TTL input pins. A CMOS TTL output pin will drive 1 TTL input pin or LSTTL input pins.

Two common examples of what a TTL output *will not* drive is a Light Emitting Diode and a solid state relay.

11.7.2 LS244 OUTPUT BUFFERS

There are two ways to drive 5 volt loads greater than 2 mili Amps. The easiest is to use an IC buffer chip, such as a 74244, and the second is to wire up a transistor suited to the load in Amps you wish to control.

74244 BUFFER CHIP: A 74244 is a common buffer chip which can be purchased from any Radio Shack store. The CIO-DIO48 P2 connector has 74LS244 output buffers installed between the 8255, U2, and the P2 connector.

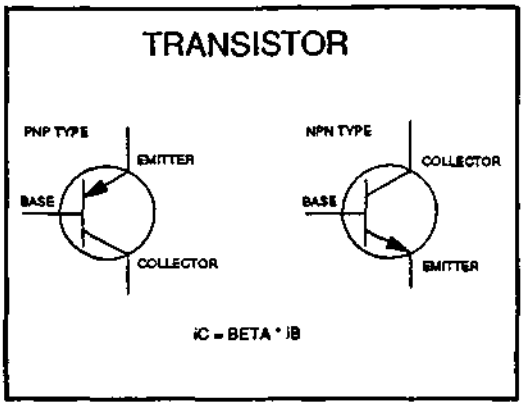
The inputs of a 74LS244 may be driven directly by the TTL outputs of an 8255.

A 74LS244 buffer has enough drive current on its outputs to turn on an LED or a solid state relay.

11.7.3 TRANSISTORS

All transistor data sheets specify operating voltages and currents according to the diagram to the right.

A transistor has three leads, a BASE, a COLLECTOR and an EMITTER.



A transistor, when used with a TTL device, is a switch that can be activated by one current while controlling another current. As such, it provides a convenient method of interfacing TTL devices to real world devices having far greater current or voltage levels. A transistor is a current amplifier. When a small current flows in the base of the transistor, it can control the flow of a large current in the collector of the transistor. The ratio of the base current to the collector current is called the beta (or gain) of the transistor. Typical betas range from 10 to 10,000. Values above 1,000 are generally found in special transistors called Darlington transistors.

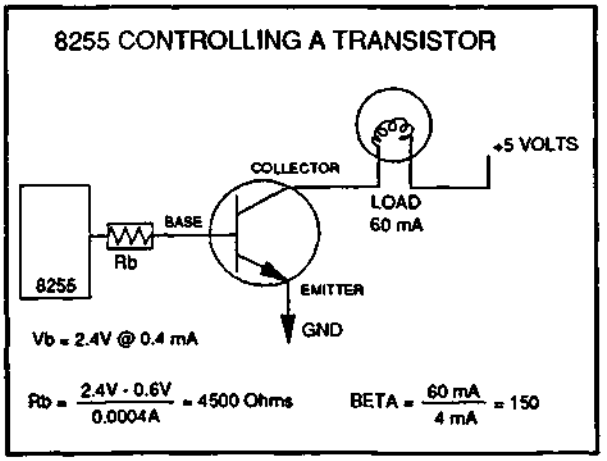
To use a transistor to interface between an I/O port and an external device we must first select a transistor which can operate the external device. The I/O port's operating limits must not be exceeded and the transistor's beta must be adequate to drive the real world device.

As an example, consider a flash light bulb which draws 60 mA at 5 volts, the circuit for which is shown below. The transistor must be able to switch 5 volts and tolerate a current flow of 60 mA. If the I/O port is connected to the base of the transistor, then we must include a resistor which limits the current flow through the I/O port to 0.4 mA. Since 60 mA divided by 0.4 mA is 150, we need a transistor with a minimum beta of 150.

This circuit turns the light bulb on when the 8255 output is high. This is probably the desired hook-up for any industrial or control application.

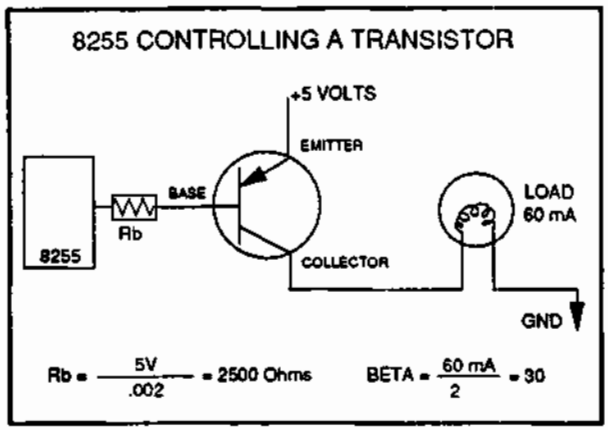
Often, books on transistors will show circuits that are active when the voltage on the base is low. They do this because the device controlling the base, such as an 8255, can sink more current than it can source.

The disadvantage of active low is that the device on the collector turns on the minute the computer with the I/O board is turned on.



An active low schematic is shown here. Don't hook this one up to the gas pipe valve!

For more details see Don Lancaster's TTL Cookbook, available in Radio Shack stores.



11.7.4 POWER MOSFET's

Power MOSFET's (Metal Oxide Semiconductor Field Effect Transistor) may be referred to by a variety of names including VMOSFET, DMOSFET and HEXFET. The difference is in the way the device is manufactured. The characteristics of power MOSFETs which make them so useful in interface circuits include high input impedance, low threshold voltages, fast switching times and no thermal runaway.

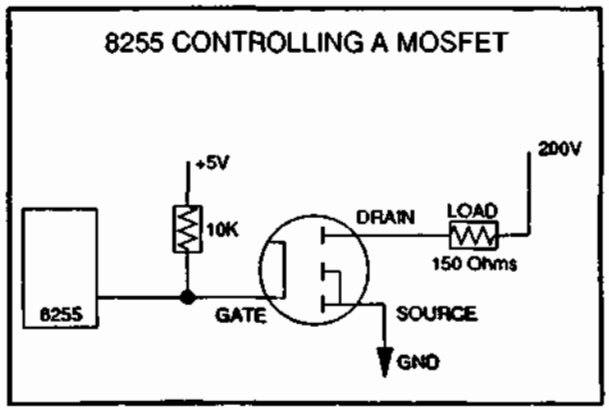
A MOSFET is a device in which the value of the voltage on the GATE relative to the SOURCE, controls the amount of current flowing from the DRAIN to the SOURCE. When the GATE voltage is below threshold, which ranges from 2 to 5 volts, no current flows from the DRAIN to the SOURCE. Above threshold the current flowing from the DRAIN to the SOURCE is a linear function of the GATE voltage.

The MOSFET has a maximum current flow for GATE voltages less than 10 volts. The DRAIN source currents may be 10 Amps or more while the voltage on the DRAIN may be hundreds of volts. A typical resistance is 1 Ohm. The current flow in the GATE circuit is in the neighborhood of 1 microAmp.

The major problem in interfacing a power MOSFET to an I/O pin is to ensure that the voltage on the GATE in the high state is as close to 5 volts as possible. The circuit below shows a 10K Ohm pull-up resistor installed which will provide a maximum voltage swing on the I/O pin (remember, TTL outputs may be as low as 2.4 volts in the high state). Many MOSFETs are designed to operate with TTL voltages.

The schematic shows a circuit which will switch 200 volts across 1 150 Ohm resistance load. The current switched is 1.3 Amps.

The IRF350 has a threshold at 25C of 3 volts. When the GATE voltage is 4 volts, it will allow 1.3 Amps to flow. Under these conditions the voltage drop across the IRF350 is about 0.5 volts.



To operate MOSFETs which require more than 5 volts, use an MOS device such as a 74C00 attached to +10 volts. Place the 74C00 between the 8255 I/O pin and the GATE of the MOSFET. For more details on MOSFETs, consult Stone and Berlin, Design of VMOS Circuits With Experiments.

11.7.5 TTL TO SOLID STATE RELAYS

Many applications require digital outputs to switch AC and DC voltage motors on and off and to monitor AC and DC voltages. These AC and high DC voltages cannot be controlled or read directly by the TTL digital lines of a CIO-DIO.

Solid State Relays, such as those available from Computer Boards, Inc. allow control and monitoring of AC and high DC voltages and provide 750V isolation. Solid State Relays (SSRs) are the recommended method of interfacing to AC and high DC signals.

The most convenient way to use solid state relays and a CIO-DIO board is to purchase a Solid State Relay Rack. A SSR Rack is a circuit board with output buffer chips which are powerful enough to switch the SSR and sockets to plug SSRs into.

SSR Racks are available from Computer Boards and most manufacturers of SSRs.

If you only want to drive one or two SSRs, all you need is a 74LS244 output buffer chip between the 8255 output and the SSR. Of course the SSR will need a 5 volt power source as well.

12 PC ARCHITECTURE

The architecture of the PC will not have an impact on the speed or accuracy of A/D boards operating at or under 100KHz DMA and 12 bits of resolution; with a few small exceptions.

12.1 CPU SPEED

The only performance parameter affected by computer type are CPU speed dependent operations, such as the time to service an interrupt. These variations have nothing to do with the A/D board alone but because board, software and CPU form a system, the differences are noticeable. For example, a 4.77MHz PC/XT acquires data from one of our interrupt driven routines at about 4,000 samples per second maximum while a 20 MHz 386 PC will process the same software at 20,000 samples per second.

EARLY PC/AT

CPU speed faster than 4.77 MHz has no impact on DMA operations at under 100 KHz with one exception. When IBM designed the AT the expansion bus clock was dropped to 1/2 the CPU clock speed, so, PC/ATs with clocks of 8MHz are running the DMA controller at 4MHz, and as such will not execute a DMA transfer as fast as a PC/XT. In a PC/AT with a 6MHz clock the maximum A/D sample transfer rate via DMA is bewteen 60KHz and 70KHz.

A PC/AT with a processor clock speed of 10MHz or better will allow A/D transfers via DMA at the full speed of 100KHz.

386 COMPUTERS

Computers based on the 386 and 386SX which have AT busses are excellent choices for data acquisition and analysis. All *CIO Series* boards operate in 386 machines and perform certain operations better as a result of the faster CPU speeds. In addition, 386 machines will run Deskview and Windows 386 and other multitasking shells. These programs really work and allow you to acquire data in one DOS task and analyze data or perform other operations in other DOS tasks.

12.2 EGA & VGA SCREENS

The EGA and VGA display boards employ the 80x86 signal NMI (non-maskable interrupt). There is no CPU pin with more priority than NMI and there is no way to mask it off without impairing the EGA screen. When an NMI is asserted, it gets serviced.

Because the EGA screen uses the NMI signal, any write to the EGA screen while a CIO-AD board DMA operation is in progress will interfere with the DMA transfer rate. *Any write to the EGA screen!* Even one character can introduce jitter at transfer rates over 70KHz. The jitter causes the DMA controller to miss DMA requests. The problem will manifest itself as bad data, missing samples or misplaced channel tags. Either slow down or stop writing to the screen while DMA transfers are in process.

12.3 PC/AT 16 BIT DATA BUS

It is commonly thought that adding the 8 additional bits of data transfer capability found on the PC/AT bus will speed the maximum data transfer rate of an A/D board. This is not the case with boards that have a maximum acquisition rate of 100KHz. In a PC/XT at 4.77MHz a CIO-AD16JR-AT will run at 100KHz, the maximum rate of the A/D converter chip. A faster DMA transfer or wider data bus cannot make the A/D chip work faster.

There are alternative A/D board architectures which employ multiple A/D converters and can take advantage of higher transfer speeds, at a higher cost.

12.4 PC TECHNICAL REFERENCE

12.4.1 TABLE OF I/O ADDRESS

HEX RANGE	FUNCTION	HEX RANGE	FUNCTION
000-00F	8237 DMA #1	2C0-2CF	EGA
020-021	8259 PIC #1	2D0-2DF	EGA
040-043	8253 TIMER	2E0-2E7	GPIB (AT)
060-063	8255 PPI (XT)	2E8-2EF	SERIAL PORT
060-064	8742 CONTROLLER (AT)	2F8-2FF	SERIAL PORT
070-071	CMOS RAM & NMI MASK (AT)	300-30F	PROTOTYPE CARD
080-08F	DMA PAGE REGISTERS	310-31F	PROTOTYPE CARD
0A0-0A1	8259 PIC #2 (AT)	320-32F	HARD DISK (XT)
0A0-0AF	NMI MASK (XT)	378-37F	PARALLEL PRINTER
0C0-0DF	8237 #2 (AT)	380-38F	SDLC
0F0-0FF	80287 NUMERIC CO-P (AT)	3A0-3AF	SDLC
1F0-1FF	HARD DISK (AT)	3B0-3BB	MDA
200-20F	GAME CONTROL	3BC-3BF	PARALLEL PRINTER
210-21F	EXPANSION UNIT (XT)	3C0-3CF	EGA
238-23B	BUS MOUSE	3D0-3DF	CGA
23C-23F	ALT BUS MOUSE	3E8-3EF	SERIAL PORT
270-27F	PARALLEL PRINTER	3F0-3F7	FLOPPY DISK
2B0-2BF	EGA	3F8-3FF	SERIAL PORT

NOTE: I/O address not used are free, for example, addresses 200 through 22F are not used and so may be used by the CIO-AD16JR-AT.

12.4.2 MEMORY MAP

00000-9FFFF	640K SYSTEM DRAM
00000-001DF	INTERRUPT VECTORS
001E0-002FF	USER VECTORS
00300-003FF	BIOS STACK
00500-005FF	DOS AND BASIC USE
00600-9FFFF	USER RAM
A0000-BFFFF	128K VIDEO BUFFER RAM
A0000-BFFFF	EGA/VGA VIDEO BUFFER
B0000-B7FFF	MDA VIDEO BUFFER
B8000-B8FFF	CGA VIDEO BUFFER
C0000-DFFF	128 ROM EXPANSION AREA
C0000-C3FFF	EGA BIOS
C6000-C63FF	PGA COMM AREA
C8000-CBFFF	XT HARD DISK BIOS
D0000-D7FFF	CLUSTER ADAPTER BIOS
E0000-EFFFF	ROM EXPANSION (XT) BIOS EXPANSION (AT)
F0000-FFFFF	BIOS
100000-FDFFFF	14.9MB AT EXTENDED RAM

12.4.3 INTERRUPTS

NAME	DESCRIPTION	NAME	DESCRIPTION
NMI	PARITY	IRQ8	REAL TIME CLOCK (AT)
IRQ0	TIMER	IRQ9	RE-DIRECTED TO IRQ2 (AT)
IRQ1	KEYBOARD	IRQ10	UNASSIGNED
IRQ2	RESERVED (XT) INT 8-15 (AT)	IRQ11	UNASSIGNED
IRQ3	COM OR SDLC	IRQ12	UNASSIGNED
IRQ4	COM OR SDLC	IRQ13	80287 NUMERIC CO-P
IRQ5	HARD DISK (XT)	IRQ14	HARD DISK

	LPT (AT)		
IRQ6	FLOPPY DISK	IRQ15	UNASSIGNED
IRQ7	LPT	Note: IRQ8-15 are AT only	

12.5 DIRECT MEMORY ACCESS

Direct Memory Access (DMA) is a special case of data transfer within a computer. DMA is managed by a DMA controller, which is a separate, programmable device managed by the CPU and other system resources. The DMA controller does one thing only and therefore does it very well. It transfers data from one location, I/O or memory, to another location, I/O or memory.

In the Intel iAPX86 architecture, the DMA controller (8237) is limited to page transfers (16 bit segments or 64K) due to its 16 bits of transfer address register.

The DMA controller bypasses the CPU's - point, read, point, write, increment pointers - data transfer method. The DMA controller transfers data directly from device to device.

The DMA controller transfers data faster than the CPU. It also can use the bus for transfers when the CPU is not actively on the bus, such as when the CPU is adding internal registers

OTHER FACTORS OF SPEED

In addition to shorter transfer cycles and stealing unused bus cycles, the DMA controller has higher priority than the 8259 interrupt controller and so is not subject to being slowed down by an interrupt request as the CPU would be[25].

A/D BOARD DMA REQUEST

Once the CIO-AD16JR-AT has completed a conversion it issues 2 DMA Requests (DRQ). The 8237 issues DMA Acknowledge (DACK) pulses for each DRQ. At each DACK two bytes of data (16 bits, 12 bits A/D + 4 bits channel tag) are transferred by the 8237. The entire operation takes about 8 microseconds.

At 100KHz an A/D conversion takes 10 microseconds.

iAPX86 DMA CYCLE

The iAPX86 family DMA controllers follow these steps:

- The DMA controller (8237) must be programmed with the data from address, the data to address and the number of bytes to transfer.
- Device issues a DMA request (DRQ) to the 8237.
- 8237 receives DRQ.
- 8237 issues a Hold Request (HRQ) to the CPU.
- The CPU releases the address and data bus and issues a Hold Acknowledge (HLDA) to the 8237.
- The 8237 places the 'data to' address on the address bus then issues a DMA Acknowledge (DACK) to the requesting device indicating that the requesting device should place its data on the data bus.
- The 8237 then issues simultaneously (in the case of the CIO-AD16JR-AT) a memory read pulse and an I/O write pulse.
- Control of the bus must be returned to the CPU for at least one bus cycle.

The PC/XT has one DMA control chip which supports 4 DMA channels. The PC/AT supports 8 DMA channels.

---

25 The one exception to this statement is the Non Maskable Interrupt (NMI) which is used by the EGA/VGA display controller and the parity error. Writing to an EGA/VGA screen while executing A/D board DMA operations will likely cause lost data. An A/D board, unlike a disk controller or other DMA device, cannot tell the external event to 'wait while I service NMI'. The external event and the A/D trigger just keep coming!

DMA Level	PC/XT	PC/AT
0	Memory Refresh	Memory Refresh
1	SDLC[26]	SDLC
2	Floppy Disk	Floppy Disk
3	Unassigned / Hard Disk	Unassigned/ Hard Disk
4		Unassigned
5		Unassigned
6		Unassigned
7		Unassigned

DMA level 1 is the best choice for use in PC/XT type machines as DMA level 3 may be used by a hard disk controller. Some network interfaces use DMA level 1.

The CIO-AD16JR-AT floats DREQ until DMA operation is activated so DMA levels may be shared with other devices as long as they are not activated simultaneously.

PROGRAMMING THE DMA CONTROLLER

The 8237 DMA controller may be programmed by any language that has I/O read/write commands. There are 5 registers which must be programmed, 4 of these are on the 8237 and the fifth is the page register which is on the CPU board.

I/O Address	Function
000-00F	8237 DMA #1
0C0-0DF	8237 DMA #2 (AT only)
083	Page register 1 (DMA level 1)
081	Page register 2 (DMA level 2)
082	Page register 3 (DMA level 3)

The 8237 is a 16 bit only device and so will address 64K of memory. To address any memory above 64K, page registers are required. No page register is required for DMA level 0.

Within each 8237, one set per DMA level, are registers for:

Mode Register	Operation type.
Address	Start address for memory.
Word count	Number of transfers.
Mask	Enables & disables DMA levels.
Command register	Set by BIOS on power-up.

The software supplied with the CIO-AD16JR-AT, and the library, CIO-PCF16, include acquisition routines which transfer A/D data to memory via DMA. If you wish to write your own DMA software you may purchase source code to either of these libraries after executing a simple non-disclosure agreement. The source code provides examples of programming the DMA controller and enabling the CIO-AD16JR-AT for DMA transfer.

26 Optional, usually not used.

13 DIAGNOSIS & DEBUG

Computer Boards maintains technical support lines staffed by experienced Electrical Engineers and Technicians. There is no charge to call and calls are returned promptly if you have called when the lines were busy.

Most of the problems encountered with data acquisition plug in boards can be solved over the phone.

Signal connection and programming are the common sources of difficulty and Computer Boards' Tech Support people can solve these type of problems with you, especially if you prepare for the call.

1) Have the phone near the PC where you can try the things we suggest.
2) Be prepared to open the PC, remove the board and read back or change switch and jumper settings.
3) Have a volt meter available to make measurements of the signals you are trying to measure and the signals on the board and PC power supply.
4) Isolate the program lines that are not working as you expect them to.
5) Have all the source code to the program you are having trouble with so preceding modes and prerequisite modes may be discussed.
6) Have the manual ready.
7) Have the CIO-ADXX Utility diskette available so the revision # and programs can be checked.

THIS LITTLE BIT OF PREPARATION WILL SPEED DIAGNOSIS AND MAY SAVE YOU THE TROUBLE OF A CALL BACK.

GENERAL DIAGNOSIS

If you would like to try a few things first, here is a list of common problems.

1) Check the PC bus power, especially the -12 volts. The +12 and -12 should be within 1/2 volt of 12.
2) Check the voltage level of the signal between the signal high and signal low. It cannot exceed the full scale range of the board.
3) Check the voltage between the signal ground and the PC ground. It should be 0 volts.
4) Check other boards in your PC for address and interrupt conflicts.
5) Check that the CALL routine is in the directory where your BASIC program is.
6) Refer to the example programs for techniques and as a baseline to check code against.

CIO-AD16JR-AT SPECIFIC DIAGNOSIS

Board does not collect valid data at full speed.

There are two probable things which prevent the CIO-AD16JR-AT from collecting valid data at full speed. *PC/AT CPU < 10MHz:* The PC/AT bus speed is 1/2 the processor clock speed. An 8 MHz PC/AT has a 4MHz bus speed which is slower than a PC/XT!

Writing to an EGA or VGA display at any time during DMA operation will cause the DMA controller to lag the CIO-AD16JR-AT and miss a sample. The EGA/VGA display employes the iAPX86 Non Maskable Interrupt (NMI). NMI has priority over everything, even DMA transfers. *Do not write to the screen during DMA transfers.*

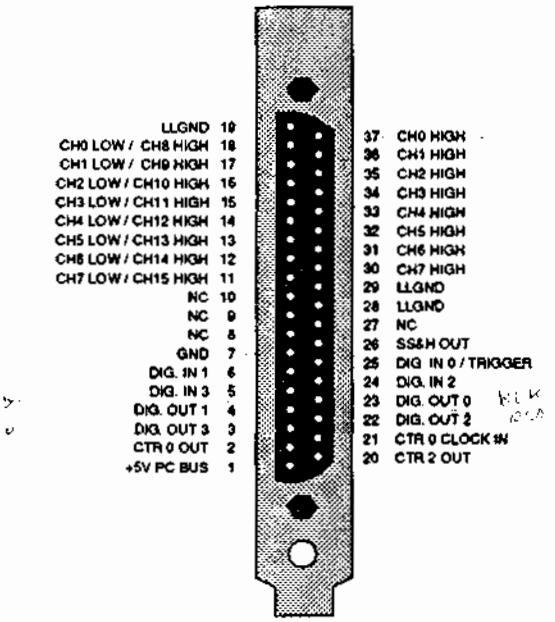
A/D is dead.

Boards returned for repair for A/D not working frequently have been subjected to overvoltage and one or more of the MUX, AMP, S&H or A/D is blown. If the voltage spike was so strong as to visibly burn the board you may return it for Computer Boards' exclusive '*Harsh Environment Warranty*'™.

If there is no visual damage to the board and you want to service the board on your own, place the CIO-AD16JR-AT in the computer where you can gain access to chips on the board; on top of a bus extension card is best. Run the program I/O test choosing A/D channel 0, user supplied signal. Attach a 1.5 volt battery and using a voltmeter and the table below or the CIO-AD16JR-AT analog input section diagram, follow the A/D input from the connector (input of the MUX) back to the A/D.

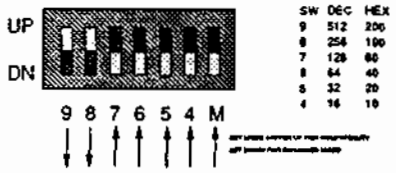
14 HELPFUL DIAGRAMS

CIO-AD16Jr. CONNECTOR

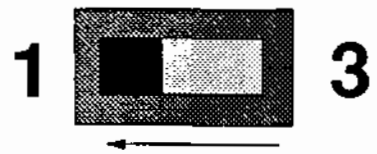


BASE ADDRESS SWITCH

SETTINGS SHOWN - 300 HEX, 768 DECIMAL

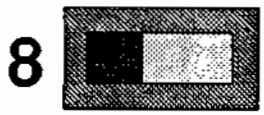


DMA LEVEL SELECT



DMA LEVEL 1 IS SELECTED

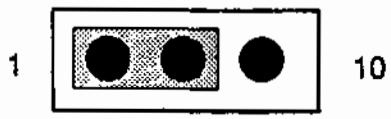
8/16 CHANNEL SELECT



8 DIFFERENTIAL INPUTS SELECTED

XTAL JUMPER

Default 1MHz Shown



J2

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP