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.

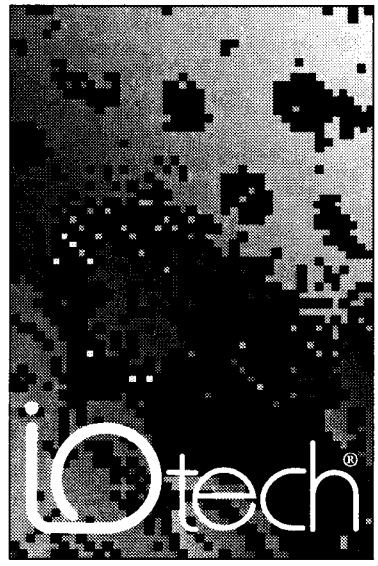
Custom engineering
so your equipment works exactly as you specify.

- 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 / IOtech ADC488/16* at our website: *Click HERE*



the smart approach to instrumentation ™

IOtech, Inc.
25971 Cannon Road Cleveland, OH 44146
Phone: (440) 439-4091 Fax: (440) 439-4093
Email: sales@iotech.com Internet: http://www.iotech.com

ADC488
User's Manual

ADC488-901, Rev. 4.0

1996-1997 by IOtech, Inc. Part No.ADC488-901, Rev. 4.0
Printed in the United States of America

Warranty Information - Your IOtech warranty is as stated on the *product warranty card.* You may contact IOtech by phone, fax machine, or e-mail in regard to warranty-related issues.
Phone: (440) 439-4091, fax: (440) 439-4093, e-mail: sales@iotech.com

Limitation of Liability - IOtech, Inc. cannot be held liable for any damages resulting from the use or misuse of this product.

Copyright, Trademark, and Licensing Notice - All IOtech documentation, software, and hardware are copyright with all rights reserved. No part of this product may be copied, reproduced or transmitted by any mechanical, photographic, electronic, or other method without IOtech's prior written consent. IOtech product names are trademarked; other product names, as applicable, are trademarks of their respective holders. All supplied IOtech software (including miscellaneous support files, drivers, and sample programs) may only be used on one installation. You may make archival backup copies.

FCC Statement - IOtech devices emit radio frequency energy in levels compliant with Federal Communications Commission rules (Part 15) for Class A devices. If necessary, refer to the FCC booklet *How To Identify and Resolve Radio-TV Interference Problems* (stock # 004-000-00345-4) which is available from the U.S. Government Printing Office, Washington, D.C. 20402.

CE Notice - Many IOtech products carry the CE marker indicating they comply with the safety and emissions standards of the European Community. As applicable, we ship these products with a Declaration of Conformity stating which specifications and operating conditions apply.



Warnings and Cautions - Refer all service to qualified personnel. This caution symbol warns of possible personal injury or equipment damage under noted conditions. Follow all safety standards of professional practice and the recommendations in this manual. Using this equipment in ways other than described in this manual can present serious safety hazards or cause equipment damage.



This warning symbol is used in this manual or on the equipment to warn of possible injury or death from electrical shock under noted conditions.



This ESD caution symbol urges proper handling of equipment or components sensitive to damage from electrostatic discharge. Proper handling guidelines include the use of grounded anti-static mats and wrist straps, ESD-protective bags and cartons, and related procedures.

Specifications and Calibration - Specifications are subject to change without notice. Significant changes will be addressed in an addendum or revision to the manual. As applicable, IOtech calibrates its hardware to published specifications. Periodic hardware calibration is not covered under the warranty and must be performed by qualified personnel as specified in this manual. Improper calibration procedures may void the warranty.



Quality Notice - IOtech has maintained ISO 9001 certification since 1996. Prior to shipment, we thoroughly test our products and review our documentation to assure the highest quality in all aspects. In a spirit of continuous improvement, IOtech welcomes your suggestions.

Table of Contents

1 Introduction 1.1

1.1 General Description 1.1

1.1.1 Analog Features 1.2

1.1.2 Data Collection and Storage Features 1.3

1.2 Specifications 1.3

1.3 Available Accessories 1.6

1.4 Abbreviations 1.6

1.5 Data Transfer Speeds 1.7

1.6 IEEE 488 Bus Implementation 1.8

2 Getting Started 2.1

2.1 Inspection 2.1

2.2 Front Panel Indicators 2.1

Figure 2.1: ADC488/8SA Front Panel Indicator Lights . . 2.2

Figure 2.2: ADC488/16A Front Panel Indicator Lights . . 2.3

2.3 Rear Panel Controls and Connectors 2.3

Figure 2.3: ADC488 Rear Panel Controls and Connectors 2.3

2.4 Power-On Self Test 2.4

2.5 Configuration 2.5

2.5.1 Line Voltage Selection 2.5

Figure 2.4: Fuse and Line Voltage Switch Locations . . . 2.6

2.5.2 IEEE 488 Bus Address Selection 2.7

Figure 2.5: S104 Default Bus Address Setting 2.7

2.6 Digital Input/Output Lines 2.8

2.6.1 Logic Levels 2.8

2.6.2 Digital I/O Port Pinouts 2.8

Figure 2.6: Rear Panel Digital I/O Connector Pinout . . . 2.8

2.7 Master/Slave Operation 2.9

2.7.1 Master/Slave Connector Pinouts 2.9

Figure 2.7: Digital Connector Pinout (Master/Slave) 2.9

2.7.2 Master/Slave Cable Connections 2.10

Figure 2.8: ADC488 Master/Slave Cable Connection . . . 2.10

2.8 External Trigger BNC Connector 2.10
2.9 Analog I/O Connectors 2.10
Figure 2.9: ADC488/16A Differential Input Connections . 2.11
Figure 2.10: ADC488/16A Single-Ended Input Connections 2.11
Figure 2.11: ADC488/8SA Differential Input Connections 2.11
2.10 RAM Expansion Board Installation 2.12
Figure 2.12: Top and Bottom Cover 2.13
Figure 2.13: Mounting Holes for MEMX Board 2.13
Figure 2.14: Installing the MEMX Option Board 2.14
2.11 Hardware Installation 2.14
2.11.1 Rack Mount 2.14
Figure 2.15: Enclosure (Side View) 2.14
Figure 2.16: Rack Mounting Hardware (Top View) 2.15
2.11.2 Bench Top 2.15
3 Data Acquisition Considerations 3.1
3.1 Sampling Theory 3.1
Figure 3.1: Sampling Theory Waveforms 3.1
3.1.1 Conversion Time 3.2
3.1.2 Frequency Folding 3.2
Figure 3.2: Frequency Folding 3.2
3.1.3 Aliasing 3.3
Figure 3.3: Aliasing 3.3
3.1.4 Sequential and Simultaneous Sampling 3.3
Figure 3.4: Sequential Sampling Phase Skew 3.4
3.2 Choosing Single-Ended or Differential Inputs 3.4
3.2.1 Instrumentation Amplifiers 3.5
Figure 3.5: ADC488 Instrumentation Amplifier Circuitry . 3.5
3.2.2 Single-Ended Inputs 3.6
Figure 3.6: ADC488 Single-Ended Inputs 3.6
3.2.3 Differential Inputs 3.6
Figure 3.7: ADC488 Differential Inputs 3.7
3.2.4 Input Bias Currents 3.7

Figure 3.8: Input Bias Currents 3.7
Figure 3.9: ADC488 Bias Current Resistors 3.8
4 Using the ADC488 . . . . . 4.1
4.1 Getting Started 4.1
4.2 Data Acquisition 4.2
4.2.1 Analog Input Mode 4.3
4.2.2 Scan Group 4.3
4.2.3 Range 4.3
4.2.4 Scan Interval 4.4
Figure 4.1: ADC488 Scan Intervals 4.4
4.3 ADC488 Triggering 4.5
Figure 4.2: ADC488 Trigger Logic 4.5
4.3.1 Trigger Source 4.5
4.3.2 Trigger Level 4.6
4.3.3 Trigger Mode 4.6
4.3.4 Pretrigger and Post-Trigger Scans 4.7
Figure 4.3: ADC488 Triggered Data Collection 4.7
Figure 4.4: ADC488 Triggered Data Collection Flowchart 4.8
4.3.5 Scan Count 4.9
4.3.6 Trigger Delay 4.9
4.3.7 Service Request 4.10
4.4 ADC488 Scan Buffer 4.10
4.4.1 Scan Buffer Size 4.10
4.4.2 Scan Buffer Organization 4.11
Figure 4.5 ADC488 Scan Buffer Organization 4.11
4.5 Scan Buffer Data Storage Examples 4.12
4.5.1 One-Shot Mode 4.12
Figure 4.6: One-Shot Mode 4.12
4.5.2 Continuous Mode With Pretrigger and Post-Trigger Scans 4.13
Figure 4.7: Continuous with Pre- & Post-Trigger Scans . 4.13
4.5.3 Continuous Mode With Post-Trigger and No Pretrigger . . 4.14
Figure 4.8: Continuous Mode with No Pretrigger Scans . 4.14

4.6 Data Output 4.15
Figure 4.9: ADC488 Scan Buffer Terminators 4.15
4.6.1 Scan Buffer Pointer 4.16
4.6.2 Buffer Select 4.16
4.6.3 Buffer Pointer Mode 4.16
4.6.4 End of Reading Delimiters 4.17
4.6.5 End of Scan Delimiters 4.17
4.6.6 User Terminator 4.17
4.6.7 Data Output Format 4.18
Figure 4.10: Data Output Logic 4.18
4.6.8 Data Correction 4.19
4.6.9 Digital I/O 4.19
4.7 Data Correction Theory 4.19
4.7.1 Analog Errors 4.19
Figure 4.11: Ideal A/D Converter Response 4.19
Figure 4.12: Deviation from Ideal Response 4.20
4.7.2 Calibration 4.20
4.7.2.1 Calibration Constants 4.21
4.7.2.2 Reading the Calibration Constants from ADC488 . . 4.21
4.8 ADC488 Commands 4.23
4.8.1 Command Options 4.24
4.8.2 Command Execution 4.24
4.8.3 Command Order 4.24
4.8.4 Query Options 4.25
4.8.5 Conflict Errors 4.25
4.9 Status Reports 4.25
4.9.1 Command Status 4.25
4.9.2 Digital Input Lines 4.26
4.9.3 Pretrigger and Post-trigger Address Limits 4.26
4.9.4 Internal Scan Counter 4.26
4.9.5 Calibration Constants 4.28
4.9.6 Scans Available 4.28

4.10 User Defined System Configurations 4.30
4.11 Calibration 4.30
4.12 Programming Examples 4.31
Figure 4.13: ADC488 Calibration Test Fixture 4.31
4.12.1 Interactive Commands 4.31
4.12.2 Using the ADC488 With BASIC 4.32
4.12.3 Setting Up the ADC488 to Acquire Data 4.33
4.12.4 Triggering 4.33
4.12.5 Serial Polling and Service Requests 4.33
4.12.6 Getting Readings From the Scan Buffer 4.34
4.12.7 High Speed Direct Memory Access 4.34
5 Command Descriptions 5.1
5.1 Introduction 5.1
5.2 ADC488 Command Set Index 5.1
Input Configuration (An) 5.5
Scan Buffer Pointer (Bn) 5.7
Figure 5.1 ADC488 Scan Buffer Organization 5.7
Scan Group (Cn,n,n,n) 5.9
Digital Output (Dn) 5.11
Error Query (E?) 5.12
Buffer Pointer Mode (Fn) 5.14
Output Format (Gn) 5.15
Calibration (Hn) 5.19
Scan Interval (In) 5.21
User Terminator (Jn) 5.23
Command Channel EOI (Kn) 5.24
Trigger Level (Lval) 5.25
SRQ Mask (Mn) 5.26
Figure 5.2: Serial Poll Response Byte & SRQ Mask . . . 5.27
Scan Count (Nn,m) 5.30
End of Reading Terminator (On) 5.32
Buffer Select (Pn) 5.33

Figure 5.3 ADC488 Scan Buffer Organization 5.33
End of Scan Terminator (Qn) 5.35
Range (Rn) 5.36
Setup (Save,Recall) (Sn,m) 5.37
Trigger (Tn) 5.39
Status (Un) 5.41
Version Query (V?) 5.43
Test (Wn) 5.44
Execute (X) 5.45
Command Channel Terminator (Yn) 5.46
Trigger Delay (Zn) 5.47
Query (?) 5.48

A Command Summary A.1

B Character Codes and IEEE Multiline Messages B.1

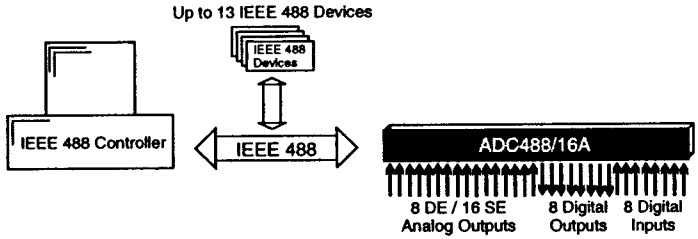
C Example Programs C.1

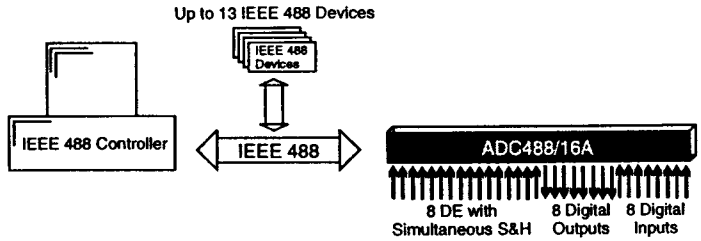
Introduction

1.1 General Description

The ADC488/16A and ADC488/8SA are analog-to-digital (A/D) converters for the IEEE 488 bus that can scan and sample analog signals at rates up to 100 kHz (10 µseconds per scan for one channel) with 16 bit resolution. The ADC488/16A will accept up to 16 single-ended analog input channels or 8 differential analog input channels and multiplex between them at a programmable scan interval. The ADC488/16A features 16-bit resolution, 96dB dynamic range, and 33 µV/bit sensitivity (at ±1 V range). This allows small variations in signals with large voltage ranges to be detected without the inconvenience of constantly reconfiguring range and gain settings.

The ADC488/8SA provides 8 differential input channels with simultaneous sample-and-hold. Individual sample and hold amplifiers "freeze" all analog input values with an inter-channel aperture uncertainty of less than 40 ns. These values are then sequentially converted at 10 µs intervals and stored to the resident memory buffer or transmitted directly to the IEEE bus.





The ADC488/16A and ADC488/8SA can record up to 2048 samples. Memory size can be increased to: 131,072 samples with the MEMX1; 262,144 samples with MEMX2; 1,048,576 samples with MEMX5; 2,097,152 samples with MEMX6 or 4,194,304 samples with the MEMX7 RAM Expansion Board. Channel capacity can be increased by linking and synchronizing ADC488 devices in a master/slave arrangement. When attached to a host computer that can maintain data transfers over the bus at greater than 200 kB/s, the ADC488/16 and ADC488/8S will collect and transfer data continuously.

Both ADC488 models are accurate enough to function as a digital voltmeter, and flexible enough in their data collection and recording modes to function as a low speed sampling oscilloscope. Triggering for data acquisition can be provided by an external TTL level change, a predefined level on one of the analog channels, or by an IEEE 488 GET or TALK command. The scan buffer can be programmed to store pretrigger samples, post-trigger samples, and to delay a specified number of samples after trigger.

Front panel LEDs show the state of Trigger, Talk, Listen, SRQ, Error, Test, and Power. Rear panel controls include a power switch and DIP switches to set the IEEE 488 bus address. Connectors are provided for power, external trigger (BNC), IEEE 488 bus, digital I/O, and analog inputs.

1.1.1 Analog Features

- 16 Single Ended/ 8 Differential Inputs (Model ADC488/16A)
- 8 Differential Inputs (Model ADC488/8SA)
- 16 bit A/D with 100 kHz sampling rate
- 10 volts Common Mode Voltage (CMV) Channel-to-Channel Isolation
- 500 volts CMV Channel-to-Earth (Digital Common) Isolation
- Four software programmable bipolar ranges (±1, ±2, ±5 and ±10 V).
- Basic Accuracy (23±5°C):

Range		Resolution	Accuracy (% of full scale)
±1 volt	16-bit	(33µV/bit)	±(0.02%)
±2 volt	16-bit	(66µV/bit)	±(0.02%)
±5 volt	16-bit	(166µV/bit)	±(0.02%)
±10 volt	16-bit	(333µV/bit)	+(0.02%)

- Common Mode Rejection (>70dB from dc to 100 Hz)

- Sequencer circuitry cycles channel and range at programmable intervals
- Digital Calibration

1.1.2 Data Collection and Storage Features

- Binary output on IEEE 488 bus at >200 kB/s (uncompensated raw binary)
- 2K scan buffer standard
- Optional RAM expansion (up to 4096 K readings)
- Programmable data output formats:
  - ASCII fixed point
  - ASCII hexadecimal
  - Decimal
  - Binary
- Multiple trigger sources
  - Level of analog input
  - External signal (TTL level)
  - GET
  - TALK
- Programmable Trigger Modes - one-shot or continuous
  - Flexible trigger management
  - Pre-trigger and post-trigger data collection
  - Programmable trigger delay
  - Trigger output for daisy-chaining multiple ADC488's

1.2 Specifications

CAUTION



Please read this manual carefully! If equipment is used in any manner not specified in this manual, the protection provided by the equipment may be impaired.

Analog

CAUTION

Analog Input: The maximum allowable analog input is 25 V peak-to-peak. The PGA range is not to exceed ±10 V. Isolation voltage between analog common and earth is not to exceed 125 volts.

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

Number of Channels:

- ADC488/16A- 16 Single-Ended or 8 Differential Inputs.
- ADC488/8SA- 8 Simultaneously Sampled Differential Inputs.

Conversion Rate:

- 100 kHz sampling 16-bit successive approximation A/D.

Conversion Time:

- 8.1 µsec (typical)

PGA Range and Resolution:

- ±1 volt range; 33µV/bit resolution
- ±2 volt range; 66µV/bit resolution
- ±5 volt range; 166µV/bit resolution
- ±10 volt range; 333µV/bit resolution

Operating Input Voltage Range:

- 13 volts peak (typical)

Maximum Allowable Input:

- **Standard:** 35 volts continuous.
- **CE:** 25 volts peak-to-peak.

Accuracy at 25± 5°C (Relative to calibration standards used):

- 1 volt range; ±0.02%
- 2 volt range; ±0.02%
- 5 volt range; ±0.02%
- 10 volt range; ±0.02%

Temperature Coefficient:

- (±0.002% ±100µV)/°C; 0-20°C and 30-50°C

Input Impedance:

- > 100 MΩ

A/D Acquisition Time and A/D Aperture Time:

- < 1.8 µsec (typical); and < 25 nsec (typical)

A/D Aperture Uncertainty:

- < 0.25 nsec (typical)

S/H Acquisition Time and S/H Aperture Time:

- < 2 µsec (typical) for ADC488/8SA; and < 200 nsec (typical) for ADC488/8SA

S/H Aperture Uncertainty:

- < 40 nsec (typical) for ADC488/8SA

Common Mode Rejection:

- > 70dB from dc to 100Hz

Channel-to-Channel Aperture Uncertainty:

- < 40 nsec (typical) for ADC488/8SA

Channel-to-Digital Low Isolation:

- **Standard:** 250 Volts maximum, 10^5 V-Hz
- **CE: (Analog Common-to-Earth Isolation)** 125 Volts (rms or DC) maximum, 10^5 V-Hz

Connector:

- Standard DB25 male connector. Mating connector is supplied.

Digital I/O

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

Number of Inputs:

- 8 bits, TTL level compatible.

Number of Outputs:

- 8 bits, TTL level compatible.

Master/Slave Arrangement:

- One master and four slaves can be connected, resulting in 80 single-ended, or 40 differential inputs. The master/slave connections are as follows: *Hold (High True), Trigger Transfer (Low True),* and *Start Scan Group (Low True).*

Connector:

- 20 pin card-edge. Mating connector supplied.

IEEE 488 Interface

CAUTION



The IEEE 488 terminal must only be used to control a non-isolated IEEE 488 system. The common mode voltage (cable to shell) must be zero.

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

Implementation:

- SH1, AH1, T6, TE0, L4, LE0, SR1, PP0, RL0, DC1, DT1, C0, E1

Programmable Parameters:

- Single-Ended/Differential Configuration
- Scan Buffer Pointer
- Scan Group
- Digital Output
- Buffer Mode
- Output Format
- Trigger Mode
- Scan Interval
- End-Or-Identify (EOI)
- Trigger Level
- SRQ Mask
- Scan Count
- Configuration
- Channel
- Pretrigger
- Post-Trigger
- Trigger Delay
- Terminators.

Data Transfer Speed:

- >200 kB/s (uncompensated raw binary)

Connector:

- Standard IEEE 488 connector with metric studs.

General

WARNING



Service: This product contains no operator serviceable parts. Service must be performed by qualified personnel. All terminals, including AC line and analog inputs, must be disconnected prior to opening the ADC488 case. Internal voltage potentials exist which could cause bodily injury or death.

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 2 for Line Voltage Input terminal. All other terminals are Category 1

Dimensions:

- 425 mm deep x 45 mm high x 203 mm wide (16.75" x 1.75"x 8").

Weight:

- 3.2 kg (7 lbs)

Operating Environment:

- **Standard:** 0 to 50°C; 0 to 70% RH to 35°C. Linearly derate 3% RH/°C from 35°C to 50°C.
- **CE:** Indoor use at altitudes below 2000 meters: 0 to 40°C; 80% maximum relative humidity up to 31°C decreasing linearly 4% RH/°C to 40°C.

Data Storage:

- 2K (readings) scan buffer standard. Optional scan buffers (readings) are: 128K, 256K, 1024K, 2048K, 4096K

Controls:

- Rear panel power switch and external switch for IEEE 488 bus address.

Indicators:

- Front Panel LEDs for Trigger, Talk, Listen, SRQ, Error, Test, & Power.

Power:

- 105-125 VAC, or 210-250 VAC, 50/60 Hz; 40 VA maximum.

CAUTION



Line Voltage: The protective conductor terminal on the AC line connector must be connected to an external protective earthing system. Failure to make such a connection will impair protection from shock. If common mode voltages connected to the analog input terminals exceed 60 VDC or 30 Vrms, the AC line connector must remain connected even when the unit is powered "off." If it becomes necessary to disconnect the AC line connector, you must first disconnect all power to the analog input terminals.

Fuse:

- 1/2A 250V, Slo Blo, 3AG (for 105-125V power line) or 1/4A 250V, Slo Blo, 3AG (for 210-250V power line)

CAUTION



Fuse failure: Fuse failure indicates a possible problem within the device circuitry. If a fuse blows, contact a qualified service representative. Replacement fuses are to be installed by qualified service personnel with the unit disconnected from the power source and with all other terminals disconnected. If the line voltage selector is changed, the fuse designated for use with that line voltage must be used.

Accessories Supplied

- Analog input Connectors (2).
- Digital I/O Connector (1).
- Power Cable.
- Instruction Manual.
- Rack Mount Kit.

(This page has been intentionally left blank.)

1.3 Available Accessories

Additional accessories for the ADC488/16A and ADC488/8SA include:

ABC488	IEEE 488 ABC data switch
CA-1	Power cable
CA-7-1	1.5 foot IEEE 488 cable
CA-7-3	6 foot shielded IEEE 488 cable
CA-7-4	6 foot reverse entry IEEE 488 cable
CA-7-5	3 foot shielded IEEE 488 cable
CA-7-6	12 foot shielded IEEE 488 cable
CN-6-2	20-pin card edge connector (digital I/O)
CN-20	Right angle IEEE 488 adapter, male to female
CN-21	IEEE 488 connector extender, male to female
CN-22	IEEE 488 multi-tap bus strip, four female connectors wired in parallel
CN-23	IEEE 488 panel mount feed-through connector, male to female
CN-40-12	12 position terminal strip plug (analog inputs)
MEMX1	128K samples RAM expansion board
MEMX2	256K samples RAM expansion board
MEMX5	1 Meg samples RAM expansion board
MEMX6	2 Meg samples RAM expansion board
MEMX7	4 Meg samples RAM expansion board

1.4 Abbreviations

The following IEEE 488 abbreviations are used throughout this manual.

addr n	IEEE bus address "n"
ATN	Attention line

CA	Controller Active
CR	Carriage Return
DCL	Device Clear
GET	Group Execute Trigger
GTL	Go To Local
LA	Listener Active
LAG	Listen Address Group
LF	Line Feed
LLO	Local Lock Out
MLA	My Listen Address
MTA	My Talk Address
PPC	Parallel Poll Configure
PPU	Parallel Poll Unconfigure
SC	System Controller
SDC	Selected Device Clear
SPD	Serial Poll Disable
SPE	Serial Poll Enable
SRQ	Service Request
TA	Talker Active
TAD	Talker Address
TCT	Take Control
term	Terminator
UNL	Unlisten
UNT	Untalk
*	Unasserted

1.5 Data Transfer Speeds

The ADC488 has been measured to perform at the approximate speeds listed below. All explanations discuss only one channel although the information given is valid for any channel. These figures are relative to the controller used. Speed may vary depending on the controller used with the ADC488.

Input of Data from Analog Channels

Since the A/D converter can convert once every 10 µs, the time required to do a complete scan is equal to the number of channels defined in a scan group, multiplied by 10µs. The maximum sampling frequency is limited by the number of channels to be scanned. One channel can be scanned at up to 100 kHz.

Output of Data to IEEE 488 Bus

Binary data can be read from the bus in high-byte/low-byte or low-byte/high-byte format at greater than 200 kB/s. When attached to a computer that can maintain data transfers over the bus at greater than 200 kB/s, the ADC488 will collect and transfer data continuously to the IEEE 488 bus.

Input of Data from the Digital Input Port

The digital input port may be read at an approximate rate of once every 75 milliseconds.

Output of Data to the Digital Output Port

Data may be sent to the digital output port at an approximate rate of once every 2.5 milliseconds.

1.6 IEEE 488 Bus Implementation

The ADC488 implements many of the capabilities defined by the IEEE 488 1978 specification. These are discussed in the following paragraphs.

Commands Not Supported by ADC488

Those bus uniline and multiline commands that the ADC488 does not support or respond to include:

Remote Enable (REN)	Parallel Poll (PP)
Go to Local (GTL)	Parallel Poll Configure (PPC)
Local Lockout (LLO)	Parallel Poll Unconfigure (PPU)
Take Control (TCT)	Parallel Poll Disable (PPD)

Commands Supported by ADC488

The following bus uniline and multiline commands are supported by the ADC488:

My Talk Address (MTA)

When the ADC488 is addressed to talk, it returns data from the scan buffer, if data is available. If no data is in the scan buffer when the ADC488 is addressed to talk, the device will hold the bus ("hang") and wait until data are available. The ADC488 can also return a status string for each port or for the entire A/D converter by using the Status (U0) command. Information about previously executed commands or data stored in the internal scan buffer may be returned by using the Query (?) command option.

My Listen Address (MLA)

When the ADC488 is addressed to listen, it accepts characters from the active talker and interprets these characters as commands and command parameters. These commands are explained in Chapter 5.

Device Clear (DCL and SDC)

Device Clear resets the ADC488 to its power-on state. The ADC488 will be in the Listener Idle and Talker Idle state. The power-on default settings for each command will be programmed.

Group Execute Trigger (GET)

When the ADC488 recognizes a GET, and the Trigger command (T1 or T7) has been used to arm the ADC488 to trigger on GET, the ADC488 will start to collect data and store it in the scan buffer. Refer to Chapter 4 on ADC488 Operation for further information.

Interface Clear (IFC)

IFC places the ADC488 in the Talker Idle and Listener Idle State.

Serial Poll Enable (SPE)

When sent a Serial Poll Enable (SPE), the ADC488 sets itself to respond to a serial poll with a serial poll status byte when addressed to talk. When the serial poll byte is accepted by the controller, any pending SRQs are cleared. The ADC488 will continue to try to output a serial poll response when addressed to talk until a Serial Poll Disable (SPD) is sent by the controller.

Serial Poll Disable (SPD)

Disables the ADC488 from responding to serial polls by the controller.

Unlisten (UNL)

UNL places the ADC488 in the Listener Idle State.

Untalk (UNT)

UNT places the ADC488 in the Talker Idle State.

Serial Poll Response

Whenever the ADC488 generates a service request (SRQ), a serial poll will return a serial poll status byte of at least 64 (decimal) showing that the SRQ was generated by the ADC488.

Getting Started

2.1 Inspection

The ADC488/16A and ADC488/8SA components were carefully inspected prior to shipment. When you receive your A/D converter, carefully unpack all items from the shipping carton and check for any obvious signs of physical damage which may have occurred during shipment. Immediately report any damage to the shipping agent. Retain all shipping materials in case you must return the unit to the factory.

Every ADC488/16A and ADC488/8SA is shipped with the following items:

- ADC488/16A or /8SA Analog-to-Digital IEEE 488 Converter
- ADC488-901 Instruction Manual
- ADC488-610 Programmed Disk in IBM Format
- ADC488-006 Accessory Kit, including:

  CA-1 Power Cable

  CN-6-20 20 Pin Card Edge Connector

  CN-40-12 12 Position Terminal Strip Plug (2)

  FE-1 Rubber Feet (4)

  FU-1-.25 1/4 Amp Replacement Fuse (for 220 volt operation)

  EN-6 Rack Ears (2)

  HA-41-6 Screws (4)

2.2 Front Panel Indicators

Seven indicator lights on the front panel of the ADC488 display the status of the A/D converter. The function of each indicator is described below.

TRIGGER — Flashing when ARMed (waiting for a trigger), on when triggered, off when data collection is finished. Also turned off by DCL or SDC.

TALK	On when the ADC488 is in the Talker state, off when the ADC488 is in the Idle or Listener state.
LISTEN	On when the ADC488 is in the Listener state, off when the ADC488 is in the Idle or Talker state.
SRQ	On when the ADC488 has generated a Service Request (SRQ), off when no SRQ is pending. Refer to the SRQ Mask (Mn) command for more information.
ERROR	On when an error has occurred, off when no error condition exists. See the Error Query (E?) command for more information.
TEST	Used to verify that communication has been established with the ADC488. The TEST indicator light is turned on and off with the Test (Wn) command.
POWER	On when power is applied to the ADC488 and the power switch on the back panel is in the on position (depressed). Off if power is not present.

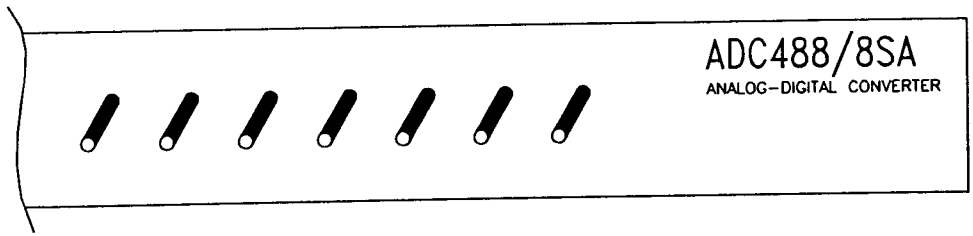


Figure 2.1: ADC488/8SA Front Panel Indicator Lights

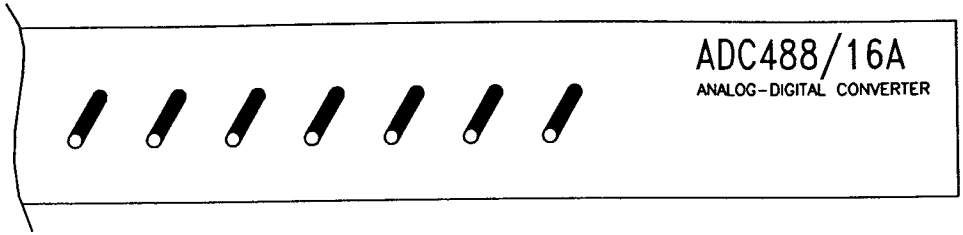


Figure 2.2: ADC488/16A Front Panel Indicator Lights

2.3 Rear Panel Controls and Connectors

Three switches and five connectors on the rear panel of the ADC488 provide power, IEEE 488 addressing, calibration, triggering, and I/O connections. The function of each is described below.

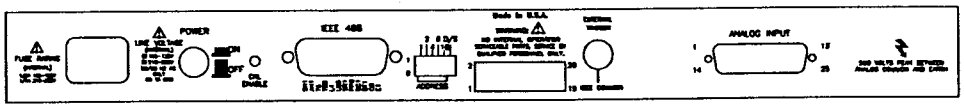


Figure 2.3: ADC488 Rear Panel Controls and Connectors

Power Connector	Provides power for the ADC488.
POWER Switch	Used to turn power to the ADC488 on and off. When pressed in, the power is on. When out, the power is off.
CAL ENABLE Switch	Used for calibration. When pressed in (enabled), calibration constants can be saved into calibration NVRAM.
IEEE 488 Connector	Connector for the IEEE 488 bus.
ADDRESS Switch	The IEEE 488 bus address is set with this switch. Refer to Section 2.3 for instructions on how to set this switch.
D/S Switch	D/S switch provides the option to bypass the MEMX memory test.
DIGITAL I/O Connector	Eight digital input lines, eight digital output lines, and signals for master/slave arrangement are on this card-edge connector.
EXTERNAL TRIGGER	BNC connector for the TTL level trigger input signal.
ANALOG I/O Connector	The analog input channels are connected through this DB-25 connector.

2.4 Power-On Self Test

Before applying power, check the factory configuration of the ADC488 for line voltage and IEEE 488 bus address. These are set at the factory to 110 V ac and bus address 14. If you need to change any of these settings, refer to Section 2.5 for details.

After plugging in the power cord, turn on the ADC488 by depressing the rear panel power switch. All of the front panel indicator lights should light for approximately one second while the ADC488 performs an internal ROM and RAM self-check. If you have a MEMX1 or MEMX2 5, 6, 7 Expansion Board installed, and the DS

switch is in the 0 position, the self-check will take longer and the indicator lights will step through a sequence that indicates the progress of the test. At the end of the self-check, all indicators should turn off except POWER.

If all of the indicator lights remain on, a ROM error has occurred. If all of the indicator lights (except POWER) flash continuously, a RAM error has occurred. Try turning the power off and on to see if the error is repeated. If indicator lights still flash and a MEMX1 or MEMX2 Expansion Board is installed in the ADC488, refer to Section 2.10, remove the Expansion board, and turn the power on. If the RAM error is not repeated after removing the Expansion Board, the ADC488 can be used with the smaller internal scan buffer (2048 locations).

If all of the indicator lights except POWER and ERROR go out, there was a checksum error in non-volatile RAM (NVRAM) and any previously saved configurations may be lost. The checksum error condition may be cleared saving a configuration using the System Defaults (Sn,m) command. Refer to Section 5.2 for details of these commands.

If the indicator lights do not flash and POWER does not light, there may not be any power supplied to the ADC488. Verify that power is supplied to the unit. If the problem persists, refer to the Service Information section of this manual.

2.5 Configuration

The ADC488 A/D Converter has been set at the factory to operate at 110 volts and IEEE 488 bus address 14. The ADC488 may be operated from 110 or 220 V ac and the IEEE 488 bus address can be changed. To change the operating voltage, it is necessary to open the enclosure. The IEEE 488 address can be changed from the rear panel. Before modifying any of these items, disconnect the power cord from the power line and disconnect any input or output connections from the rear panel of the ADC488.

2.5.1 Line Voltage Selection

The ADC488 may be operated from 110 or 220 V ac. The unit was shipped from the factory set for the operating voltage marked on the label placed over the power jack on the rear panel. To change the operating voltage, change the setting of internal switch S106 according to the following instructions. If the setting of S106 is changed, the power fuse must also be changed to avoid damage to the unit.

The locations of S106 and the fuse are shown in the Figure below.

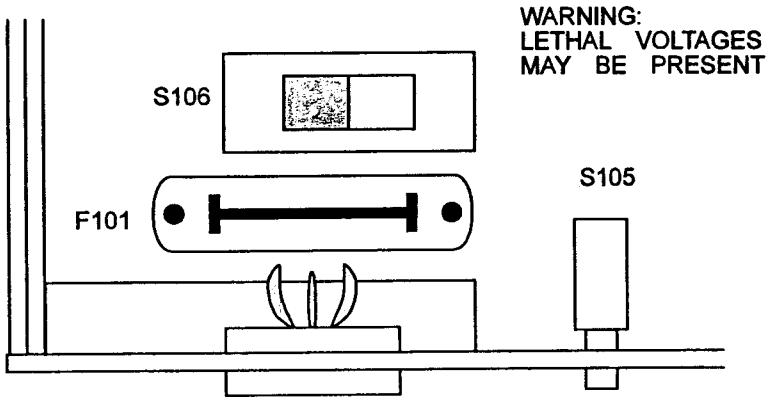


Figure 2-4: Fuse and Line Voltage Switch Locations

Warning: Never open the ADC488 case while it is connected to the power line. Internal voltage potentials exist which could cause personal injury or death.

1. Disconnect the power cord from the power line and from the ADC488. Disconnect any IEEE 488 or digital I/O cables prior to disassembly. Place the ADC488 on a flat surface. Remove the six screws on top of the case and remove the top cover.
2. The line voltage selection switch (S106) is located next to the main power supply transformer (the transformer near the power switch). Insert the tip of a small screwdriver into the slot of the switch and move the switch so the desired line voltage appears on the switch.
3. Install a power line fuse appropriate for the line voltage. The fuse is located next to the internal line voltage switch (S106). Gently pull upward on the plastic fuse housing. Remove the entire housing with the fuse inside. Select a fuse with the proper rating (see table below).

Line Voltage	Fuse Type
90 - 125V	½A 250V, Slo Blo, 3AG
210 - 250V	¼A 250V, Slo Blo, 3AG

Caution: A fuse with a rating higher than that specified may cause damage to the instrument. If the instrument repeatedly blows fuses, contact the factory.

4. Open the fuse housing by pushing up on the tab on the bottom of the housing.
5. Replace the fuse and close the housing. Insert the fuse housing into the fuse holder. Replacement fuses are included in the accessory kit supplied with the unit (part #ADC488-006).
6. Make note of the new voltage setting for later reference and carefully re-assemble the unit.
7. Carefully reassemble the unit.

Warning: The ADC488 is intended for INDOOR USE ONLY. Failure to observe this warning could result in equipment failure, personal injury or death.

2.5.2 IEEE 488 Bus Address Selection

The IEEE 488 bus address is set by S104-1 through S104-5 (located on rear of unit). The address can be set from 0 through 30. The address is selected by simple binary weighting, with switch 1 being the least significant bit and switch 5 the most significant bit. The factory default is address 14. If address 31 is selected, it defaults to address 30 because the IEEE 488 standard has reserved address 31.

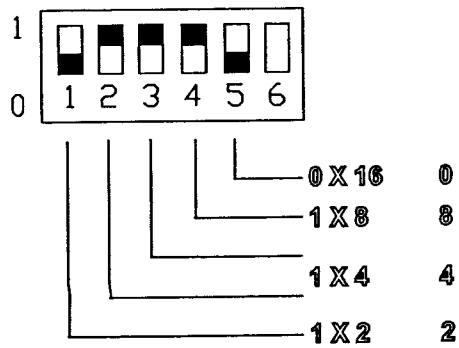


Figure 2.5: S104 Default Bus Address Setting

This switch is read only when power is applied to the unit and should be set before applying power. Set the switch positions using a small, jeweler's, flat-bladed screwdriver. The enclosure does not need to be opened to change S104 switch settings. The following figure shows the factory default IEEE 488 bus address setting.

2.6 Digital Input/Output Lines

The ADC488 has eight digital input lines and eight digital output lines available on a rear panel card edge connector. These lines can be set and/or read using ADC488 commands.

2.6.1 Logic Levels

Each digital output line will drive five (5) standard TTL loads. All digital input lines are one-eighth (0.125) TTL loads. All inputs are protected against damage from high static voltages. Normal precautions should be taken to limit the input voltages to –0.3 to +5.3 volts. All digital I/O lines are referenced to digital ground (Pin 20).

2.6.2 Digital I/O Port Pinouts

The Figure below illustrates the digital I/O edge connector as viewed from the rear of the ADC488.

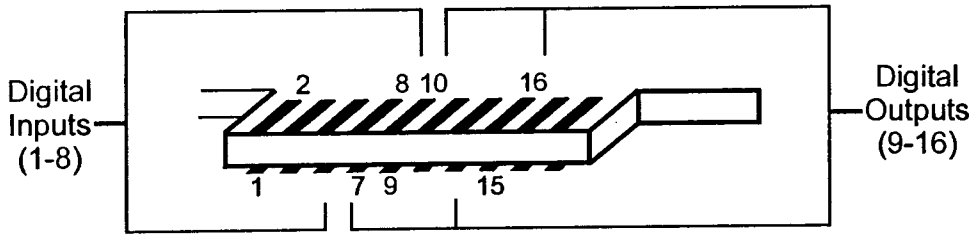


Figure 2.6: Rear Panel Digital I/O Connector Pinout

Pin	Description
1 - 8	Digital Inputs Pin 1 is bit 1 (LSB), Pin 8 is bit 8 (MSB).
9 - 16	Digital Outputs Pin 9 is bit 1 (LSB), Pin 16 is bit 8 (MSB).

2.7 Master/Slave Operation

Up to five ADC488 A/D Converters can be connected together in a master/slave arrangement to provide up to 80 single-ended or 40 differential analog input channels. One ADC488 is configured as a master and the others are configured as slaves and a cable is used to connect the devices tog er.

2.7.1 Master/Slave Connector Pinouts

In order to use the master/slave arrangement, the external hold, external trigger transfer, external start scan group, and digital ground signals on the rear panel digital connector must be connected.

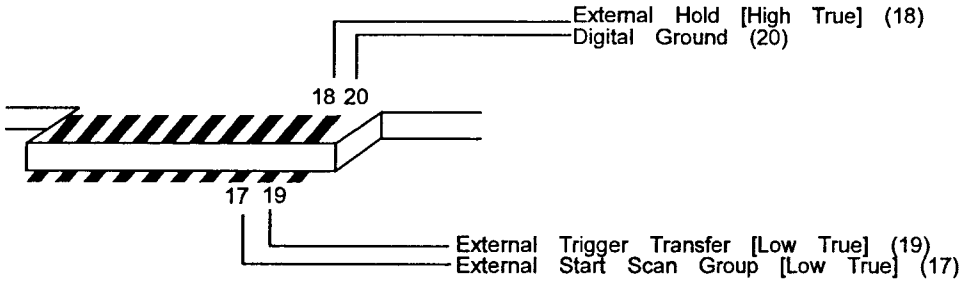


Figure 2.7: Rear Panel Digital I/O Connector Pinout

Pin	Description
17	External Start Scan Group (Low True)
18	External Hold (High True)
19	External Trigger Transfer (Low True)
20	Digital Ground

2.7.2 Master/Slave Cable Connections

The 20-pin Card Edge Connector (CN-6-20) that is supplied with the ADC488 can be used to tie the units together. The following diagram shows how to make a cable and connect the units for master/slave operation.

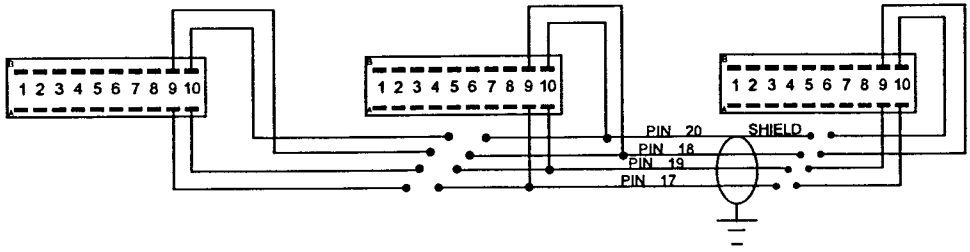


Figure 2.8: ADC488 Master/Slave Cable Connections

2.8 External Trigger BNC Connector

The rear panel external trigger BNC connector can be used to trigger the ADC488 once it has been properly armed. The ADC488 can be programmed to trigger on a positive-going or negative-going TTL level edge. Any TTL level signal may be used as a trigger pulse. A trigger pulse may also be used to generate a Service Request.

2.9 Analog I/O Connectors

Analog inputs are connected to the ADC488 through one DB-25 plug connector on the rear panel. The ADC488/16A accepts up to sixteen single-ended analog inputs or up to eight differential analog inputs. The ADC488/8SA accepts eight differential analog inputs. The following figures show how analog input connections are made to both the ADC488/16A and the ADC488/8SA.

2.7 Master/Slave Operation

Up to five ADC488 A/D Converters can be connected together in a master/slave arrangement to provide up to 80 single-ended or 40 differential analog input channels. One ADC488 is configured as a master and the others are configured as slaves and a cable is used to connect the devices tog er.

2.7.1 Master/Slave Connector Pinouts

In order to use the master/slave arrangement, the external hold, external trigger transfer, external start scan group, and digital ground signals on the rear panel digital connector must be connected.

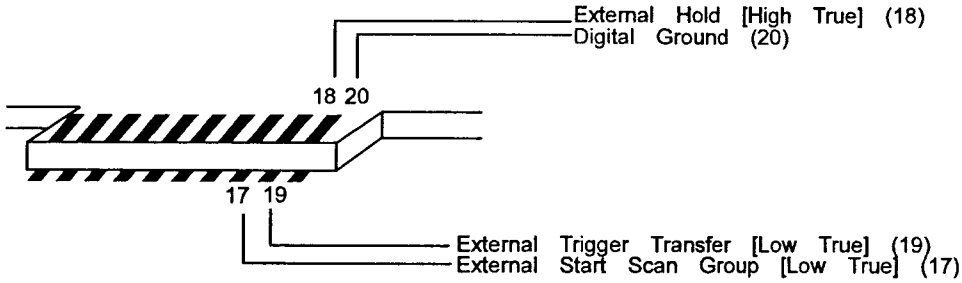


Figure 2.7: Rear Panel Digital I/O Connector Pinout

Pin	Description
17	External Start Scan Group (Low True)
18	External Hold (High True)
19	External Trigger Transfer (Low True)
20	Digital Ground

2.7.2 Master/Slave Cable Connections

The 20-pin Card Edge Connector (CN-6-20) that is supplied with the ADC488 can be used to tie the units together. The following diagram shows how to make a cable and connect the units for master/slave operation.

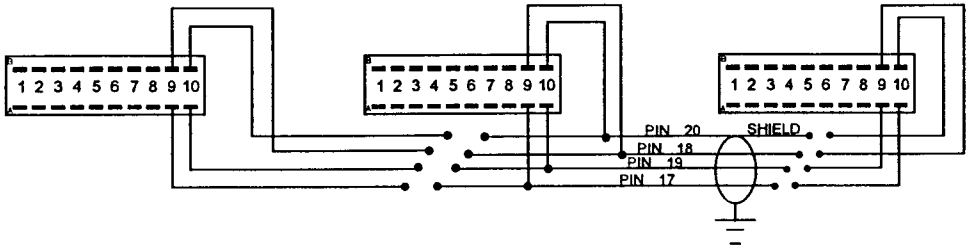


Figure 2.8: ADC488 Master/Slave Cable Connections

2.8 External Trigger BNC Connector

The rear panel external trigger BNC connector can be used to trigger the ADC488 once it has been properly armed. The ADC488 can be programmed to trigger on a positive-going or negative-going TTL level edge. Any TTL level signal may be used as a trigger pulse. A trigger pulse may also be used to generate a Service Request.

2.9 Analog I/O Connectors

Analog inputs are connected to the ADC488 through one DB-25 plug connector on the rear panel. The ADC488/16A accepts up to sixteen single-ended analog inputs or up to eight differential analog inputs. The ADC488/8SA accepts eight differential analog inputs. The following figures show how analog input connections are made to both the ADC488/16A and the ADC488/8SA.

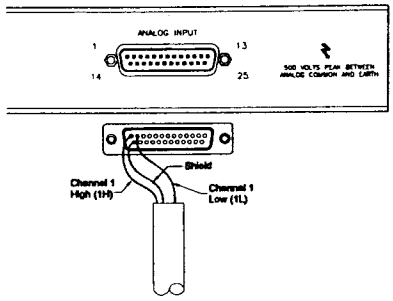


Figure 2.9: ACS488/8SA Differential Input Connections

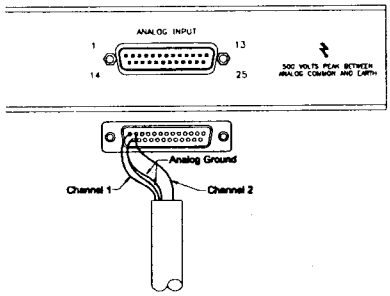


Figure 2.10: ADC488/16A Single-Ended Input Connections

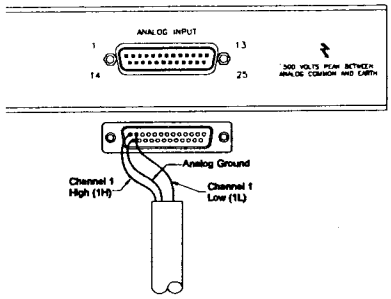


Figure 2.11: ADC488/16A Differential Input Connections

Input Configuration		DB-25M Connector Pin Number
Differential	Single-Ended	
1H	1	1
Analog Common	Analog Common	14
1L	2	2
2H	3	15
Analog Common	Analog Common	3
2L	4	16
3H	5	4
Analog Common	Analog Common	17
3L	6	5
4H	7	18
Analog Common	Analog Common	6
4L	8	19
5H	9	7
Analog Common	Analog Common	20
5L	10	8
6H	11	21
Analog Common	Analog Common	9
6L	12	22
7H	13	10
Analog Common	Analog Common	23
7L	14	11
8H	15	24
Analog Common	Analog Common	12
8L	16	25

Note: Shield is connected to pin 13 on the DB-25.

2.10 RAM Expansion Board Installation

The MEMX1, MEMX2, MEMX5, MEMX6 and MEMX7 RAM Expansion Board options can be installed in the ADC488 to increase the internal scan buffer size from 2048 scan buffer locations to: 131,072 for the MEMX1; 262,144 locations for the MEMX2; 1,048,576 for the MEMX5; 2,097,152 for the MEMX6 or 4,194,304 for the MEMX7. Included with the RAM Expansion Boards are four spacers, eight screws, and one PC board assembly with a 50 pin connector. To install the RAM

Expansion Board, it is necessary to open the enclosure, mount the PC board and spacers, and plug in the 50 pin header connector.

1. Disconnect the power cord from the power line and from the ADC488. Disconnect any IEEE488 or digital I/O cables prior to disassembly.

Warning: Never open the ADC488 case while it is connected to the power line. Internal voltage potentials exist which could cause personal injury or death.

2. Place the ADC488 on a flat surface. Remove the six screws on top of the case and remove the top cover. Turn the ADC488 over and remove the six screws on the bottom cover.
3. Located on the main PC Board Assembly are four mounting holes for the RAM Expansion Board. Insert one screw through each of the holes from the bottom of the ADC488 into the spacers. Tighten the screws from the bottom of the unit.

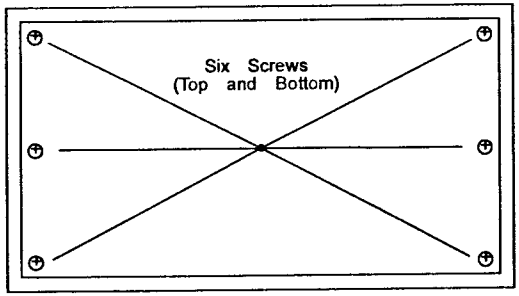


Figure 2.12: Top and Bottom Cover

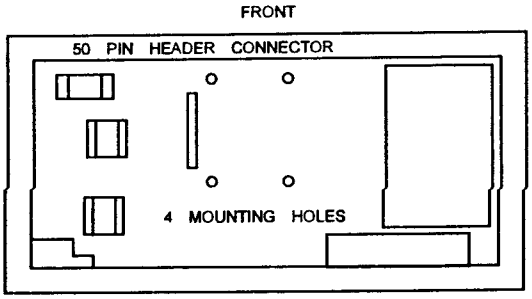


Figure 2.13: Mounting Holes for MEMX Board

4. Turn the ADC488 over. Plug the 50-pin connector onto the pins of the 50-pin header connector on the main PC Board Assembly as shown in Figure below.

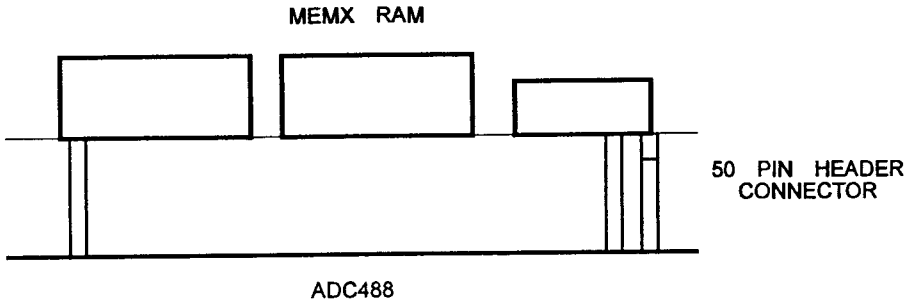


Figure 2.14: Installing the MEMX Option Board

5. Insert one screw through each of the holes in the RAM Expansion PC Board into the spacers. Tighten the screws.
6. Replace the top and bottom covers and screws.

2.11 Hardware Installation

The ADC488 includes accessories for rack or bench use.

2.11.1 Rack Mount

If rack mount installation is required, remove the two plastic screws from the predrilled holes on each side of the unit. Only remove the screws from the set of holes that will be toward the front of the rack (the unit can be mounted with the front or rear panel facing the front of the rack fixture).

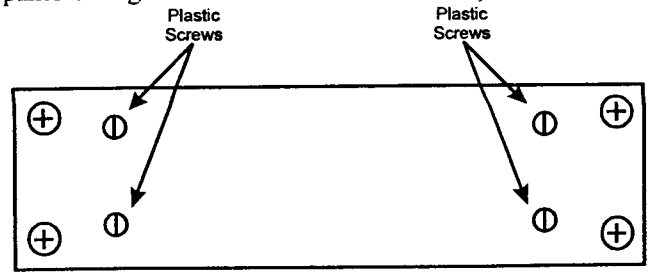


Figure 2.15: Enclosure (Side View)

Install the two rack ears using the enclosed screws as shown in the Figure below.

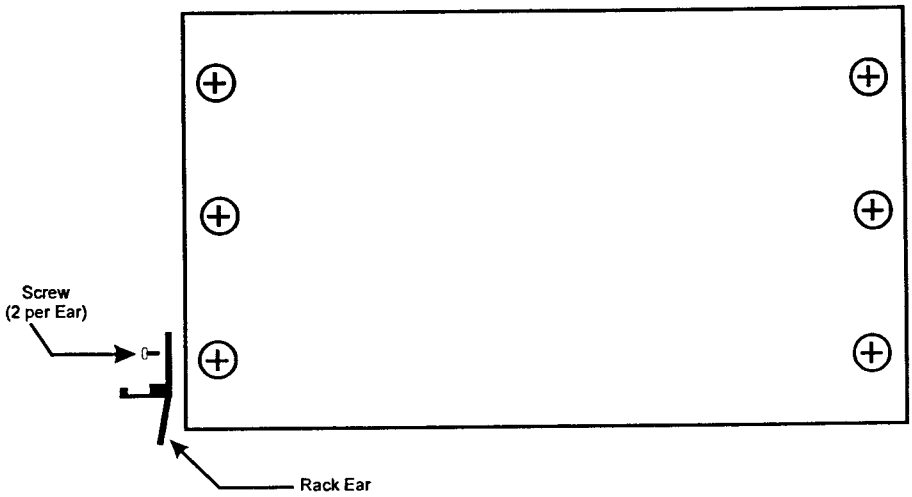


Figure 2.16: Rack Mounting Hardware (Top View)

2.11.2 Bench Top

If bench installation is required, install the self-adhesive rubber feet on the bottom of the unit approximately one inch from each corner.

Data Acquisition Considerations

3.1 Sampling Theory

In a data acquisition system using an ADC488, analog signals (readings) are sampled on a periodic basis. Once sampled, the ADC488 analog-to-digital (A/D) converter transforms the readings into digital form.

A train of sample pulses, as illustrated below, represents fast-acting switches which sample the analog signal for a very short time. The output of the fast-acting sampler is the modulated sampled signal shown, the product of the analog signal and the train of sampling pulses. The amplitude of the original reading is preserved in the modulation envelope of the sample pulses.

If the fast-acting sampler is replaced by a sample/hold circuit, the amplitude of each reading is stored between samples and the output is a reconstructed sample and hold signal. A sample/hold circuit samples a signal voltage and stores it in a capacitor for the period of the conversion time. While the sample/hold circuits are in the hold state, the A/D converter then sequentially converts each sample.

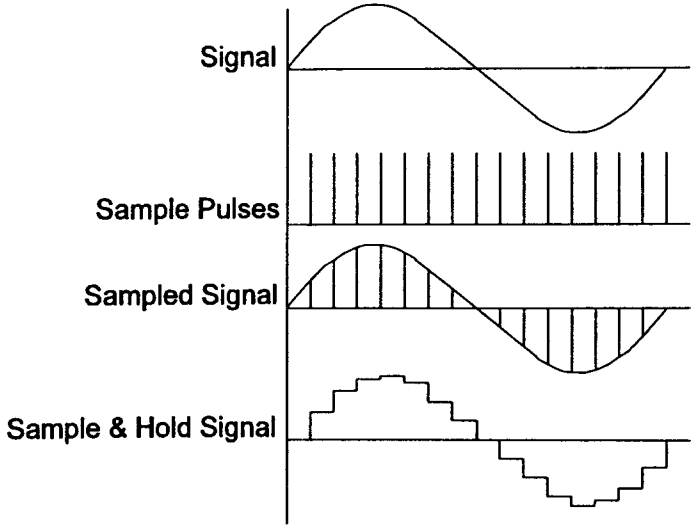


Figure 3.1: Sampling Theory Waveforms

3.1.1 Conversion Time

An A/D converter requires time to convert an analog value to a digital word. This time, called the conversion time or aperture time, represents a time window in the analog signal spectrum. The conversion time results in an uncertainty (read: error) if the signal changes during this time.

The conversion speed required for a particular application depends on the variations of the input signals and the accuracy desired. Information is lost if there is significant change in amplitude between samples. If a continuous, bandwidth-limited signal contains no frequency components greater than f_c, then the original signal can be recovered without distortion if it is sampled at a rate of at least $2f_c$ samples per second.

3.1.2 Frequency Folding

If the sampling frequency f_s is not high enough (at least twice the highest frequency component, f_c), part of the signal spectrum centered around f_s will fold over into the original spectrum. This effect is called frequency folding. The folded part of of the spectrum causes distortion in the recovered signal.

If the sampling rate is increased so that $f_s - f_c > f_c$ (that is, $f_s > 2f_c$), the two spectra are separated and the signal can be recovered without distortion.

In practice there is always some frequency folding due to high frequency signal components and noise. To reduce this effect a sampling rate much higher than $2f_c$ may be required.

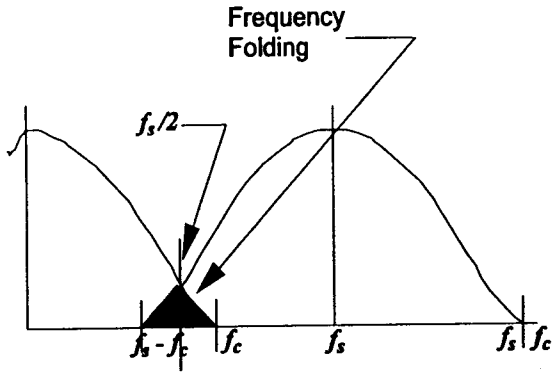


Figure 3.2: Frequency Folding

3.1.3 Aliasing

An alias frequency, the effect of too slow a sampling rate on a sine wave signal, is shown in the Figure below. This alias frequency can be significantly different from the original signal. A sampling rate slightly less than $2f_c$ returns the low frequency sine wave shown below. A sampling rate greater than $2f_c$ will prevent aliasing and preserve the original signal.

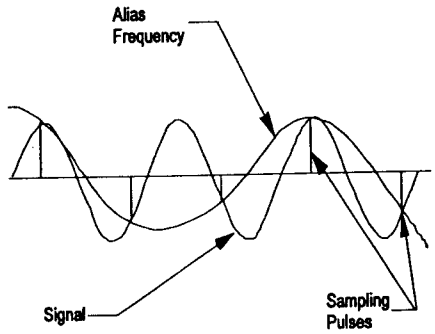


Figure 3.3: Aliasing

3.1.4 Sequential and Simultaneous Sampling

When sequential sampling is used (ADC488/16A), input signals are sampled and converted sequentially, one at a time. When simultaneous sampling is used (ADC488/8SA), all inputs are sampled at the same time and converted sequentially.

For example, assume that all eight channels are connected to a 1.25 kHz, 10 volt peak-to-peak sine wave. The scan rate is set to 100µs. The ADC488/16A will require 80µs to convert all eight channels. Channel #1 converts at t_1=0 and channel #8 converts at t_2=70µs. The difference in measurements between these two channels due to phase skew is:

$$A_2 \text{ Sin}(Wt_2) - A_1 \text{ Sin}(Wt_1) =$$

$$10 \text{ Sin } (7854 * 70\mu s) - 10 \text{ Sin } (7854 * 0\mu s) =$$

$$10 \text{ Sin } (0.55) - 10 \text{ Sin } (0) = 5.22 - 0 = 5.22 \text{ volts}$$

This value, 5.22 volts, corresponds to a phase skew of 31.5°. The effect of this skew is graphically shown in the Figure below.

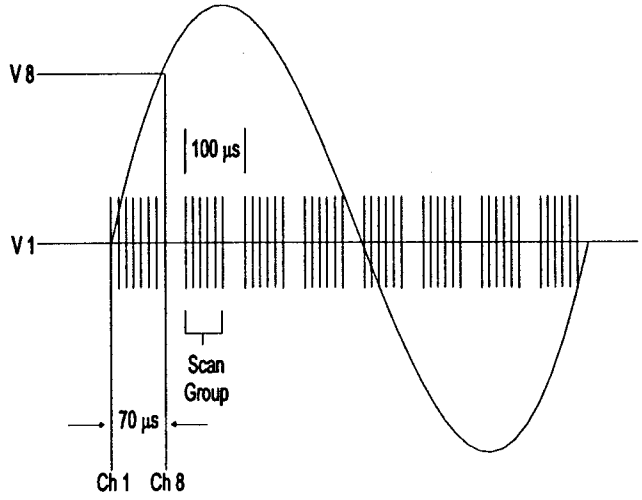


Figure 3.4: Sequential Sampling Phase Skew

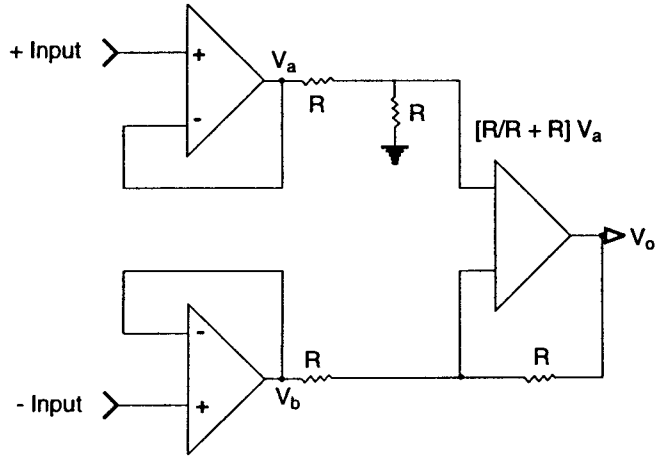
If simultaneous sampling is used, as with the ADC488/8S, the maximum time difference between each channels sample is 40 ns. This small time skew is accomplished by sampling all eight channels at the same time with individual sample/hold circuits. The ADC488/8SA measurements result in a voltage error of only 0.003 volts and a phase skew of 0.018° because of the 40 ns sample/hold uncertainty.

3.2 Choosing Single-Ended or Differential Inputs

The question of whether to use differential or single-ended inputs depends on signal level and signal ground references. In single-ended configuration, all inputs are referenced to a common ground. Problems arise if the ground potentials are different. In differential configuration, both the ground reference and the signal voltage are measured. If signal levels are very low or if ground reference problems are encountered, differential input configuration provides attenuation of the effects of common-mode voltage. The following paragraphs explain the difference between the single-ended and differential input configurations of the ADC488.

3.2.1 Instrumentation Amplifiers

The following figure illustrates the classical instrumentation amplifier topology used in the ADC488. The purpose of this differential input amplifier is to pass signals which are the difference between the +INPUT and –INPUT (normal mode) and to reject signals which are similar (common mode). The unity gain buffer operational amplifiers are provided to increase the input impedance. By reviewing the transfer function of the circuit it can be shown that the output voltage is equal to the difference between the inputs. In the following circuit discussions, this circuit will be illustrated as a simple operational amplifier designated by 'A'.



$$V_o = [R/R + R]\,[R + R/R]V_a - [R/R]V_b = V_a - V_b$$

Figure 3.5: ADC488 Instrumentation Amplifier Circuitry

3.2.2 Single-Ended Inputs

Using single-ended inputs is a good choice when one side of all the signal sources are tied to the same point, as in a star connection. When single-ended inputs are used, each signal source uses the same low signal path (ground). Differences in ground potentials between sources can lead to noise and errors in measurement.

The following figure shows two sources in a single-ended configuration. When switch S2 is closed, the signal applied to the amplifier input is equal to V2. But when S1 is closed, the applied signal is the sum of V1 and V_{cm} (the voltage potential between grounds G1 and G2). Measuring V1 may not only yield errors, but quite possibly noise.

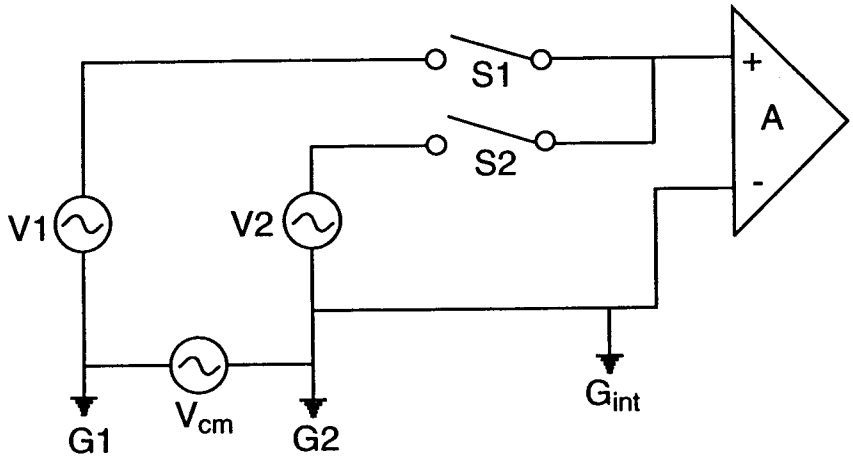


Figure 3.6: ADC488 Single-Ended Inputs

3.2.3 Differential Inputs

When it is impractical to connect all the sources to the same signal return point, differential inputs can be used to reduce errors and ground current induced noise.

The next figure shows such a configuration. When switch S2 is closed (S2a and S2b), the signal applied to the amplifier is V2. When switch S1 is closed (S1a and S1b), the signal applied to the amplifier is V1. The common-mode rejection ratio, the ratio of the differential voltage to the common-mode voltage (V_{cm}), is greater than 70dB from dc to 100 Hz.

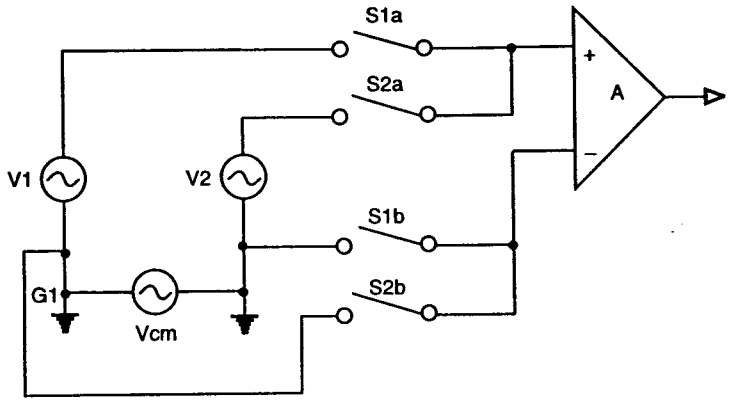


Figure 3.7: ADC488 Differential Inputs

3.2.4 Input Bias Currents

For differential input configuration, each input must have biasing currents. A certain amount of current must be flowing into the differential inputs or else capacitance-induced noise will make the inputs appear as noise.

Two banks of 100 kΩ resistors are provided to allow a current path when no current is flowing due to no common ground connections or a high impedance input source.

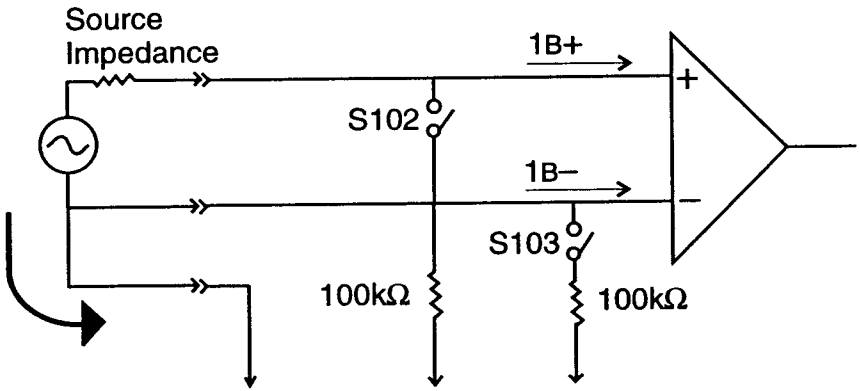


Figure 3.8: Input Bias Currents

With no common ground connection (LO), close switch S103 to put a 100kΩ resistor between the (L) input and ground and provide a common mode impedance of 100 kΩ. With a high impedance source, both S102 and S103 should be closed to provide 100 kΩ of normal mode impedance.



Figure 3.9: ADC488 Bias Current Resistors

Using the ADC488

4.1 Getting Started

Twenty-six commands are available in the ADC488 command set to control its operating and communications parameters. These commands are composed of a single alphabetic character (A to Z), followed by one or more command arguments. At power-up, all of the command states are set to 0 (that is, A0, B0, I0, etc.) except for the Scan Group command (Cn), which is set to C1. Until altered by sending commands to the ADC488, these default power-up states remain intact. You may find that the power-up state of the ADC488 needs only minor adjustments to suit your application.

To get a reading after power-up, simply send a T6X to the ADC488 to digitize the analog signal on channel one and prepare it for transfer to the IEEE 488 controller.

```
OPEN  "\DEV\IEEE...
.
.
.
PRINT  #1,  "OUTPUT  14;T6X"
PRINT  #1,  "ENTER  14"
INPUT  #2,  R$
PRINT  "The  value  at  channel  one  is  —>"  ;  R$
```

This program prints to the screen:

```
The  value  at  channel  one  is  —> +00.1349
```

The previous program commanded the ADC488 to convert one channel one time when it is placed into the TALK mode (the ENTER command places the ADC488 into the TALK mode). All other commands remain in their power-up states.

The IEEE 488 Selected Device Clear (SDC) command returns the ADC488 to its power-up state if the application program has changed any of the command states. This IEEE 488 command is issued by the IOtech Driver488 controller within the CLEAR command.

```
PRINT  #1,  "CLEAR  14"
```

The following program gets 24 readings from the ADC488's channel 2 and places them on the screen.

```
OPEN  "\DEV\IEEE...
.
.
.
PRINT  #1,  "OUTPUT  14;C2X"
PRINT  #1,  "OUTPUT  14;T6X"
FOR  I  =  1  TO  24
PRINT  #1,  "ENTER  14"
INPUT  #2,  R$
PRINT  R$
NEXT  I
```

The T6 command instructs the ADC488 to convert the analog signals in the scan group every time the ADC488 is placed into the TALK state. Since the ENTER command is within the FOR loop, the ADC488's talk address is issued by the control every time the loop is executed. The default scan group is channel 1 only, but this program changed the scan group to channel 2 only. A stream of numbers similar to the following will be printed on the screen when the program is executed.

```
+00.0023
+00.1273
+00.4932
+00.4936
.
.
.
```

To simplify the programming of the ADC488, issue only those commands whose default settings do not suite your application. The only command that must be issued to begin the acquisition process is the trigger command (Tn). Even if you wish to use the T0 setting (Continuous on TALK), it is necessary to issue that command to the ADC488 to begin the acquisition process.

4.2 Data Acquisition

The analog input channels are configured by using the ADC488 commands for Input Configuration (An), Scan Group (Cn,n,n,n), Range (Rn,n,n,n), and Scan Interval (In).

Changing the analog input mode (An command) is a major change in the operation of the ADC488. Changing the analog input mode forces default parameters to be set for many of the ADC488 commands. Any previously defined Scan Group will be nullified and will default to one channel, C1 (channel 1) and the scan interval will default to the fastest possible rate, I0 (10µs).

4.2.1 Analog Input Mode

The analog input mode is set by using the Input Configuration (An) command. Sixteen analog input connections are provided on the ADC488. On the ADC488/16A, the sixteen input connections can be configured to work in one of two modes: single-ended mode or differential mode. For the ADC488/8SA, only eight channels are available as differential inputs. The choice of input mode is universal to all input connections, that is, all input connections are single-ended or all input connections are differential.

In single-ended mode (ADC488/16A only), any of the analog input connections may be measured with respect to analog ground. In this mode, there can be sixteen analog channels to use for measurements.

In differential mode, the input connections are paired, and the measured value is the difference between one analog input connection and another. In this mode there are eight analog channels available for measurements.

4.2.2 Scan Group

The Scan Group (Cn,n,n,n) command is used to define the number and scanning order of the analog input channels. Scan Groups must be configured in groups of 1, 2 ,4, 8, or 16 channels. Channels can be scanned in any order and one channel can be specified and scanned more than once in a scan group.

The analog input connections are made to the rear panel terminal strip plugs. Any or all of the input connections can be used. Make sure to tie all the unused inputs to ground to minimize interchannel effects.

The scanning order specified in the Scan Group command also specifies the order of individual scan buffers referenced with the Buffer Select (Pn) command. The first channel specified in a Scan Group is the channel used for the Analog Level trigger source (Trigger commands T4 and T5).

4.2.3 Range

A bipolar range of ±1, ±2, ±5, or ±10 volts full scale may be specified for each input channel using the Range (Rn,n,n,n) command. The range for each channel can be set separately using one of two methods: by specifying all of the ranges for all of the channels (Rn,n,n,n,n,n,n,n), or specifying one range for one channel (R#c,n).

Ranges are set for each channel, starting with channel 1. Do not confuse the order of Range settings with the order of channels specified using the Scan Group (Cn,n,n,n) command. For instance, if a Scan Group command of C5,6,7,8 is specified and those four channels are to be set to the 2 volt range, then either the Range command Rn,n,n,n,1,1,1,1 (where n can be from 0 to 3) or R#5,1 R#6,1 R#7,1 and R#8,1 must be used.

Make sure the range is set correctly for the anticipated analog input signal voltage so one channel does not affect another. Input signals must not exceed ±15 volts.

4.2.4 Scan Interval

The Scan Interval (In) command is used to control the amount of time between scans. Scan intervals follow a 1, 2, 5, 10 sequence from a minimum of 10 microseconds up to a maximum of 50 seconds. The scan interval has a minimum resolution of one conversion time (10µs for one channel).

Since the A/D converter can convert one channel every 10 microseconds, the time required to do a complete scan is equal to the number of channels defined in a scan group multiplied by 10 microseconds.

Setting the Scan Interval to a value less than the time it takes to complete a scan will generate a conflict error (E4). For example, specifying 16 channels in the scan group (160µs/scan) and setting the scan interval to 50µs causes the front panel ERROR indicator light to light and a conflict error to be generated.

The maximum sampling frequency is limited by the number of channels to be scanned. One channel may be scanned at up to 100 kHz, two channels may be scanned at up to 50 kHz, and so on. Sixteen channels are scanned in 200 microseconds (5 kHz), not 160 microseconds. This is due to the 1, 2, 5, 10 sequence of the Scan Interval (In) command.

If a scan interval is not specified, the ADC488 automatically sets the fastest scan interval for the number of channels specified in a scan group.

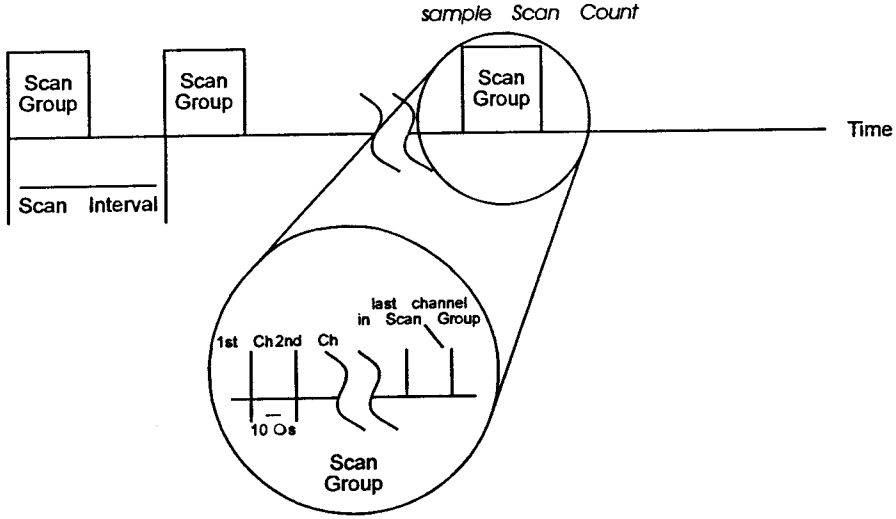


Figure 4.1: ADC488 Scan Intervals

4.3 ADC488 Triggering

The ADC488 can be configured to start data acquisition on one of many trigger sources. Commands are available to specify the trigger source and mode (Tn), trigger level for analog signals (Lval), trigger delay (Zn), and the number of scans that will be taken and saved in the scan buffer (Nn,m). A Service Request (SRQ) can also be sent to the controller when a trigger has been detected (Mn).

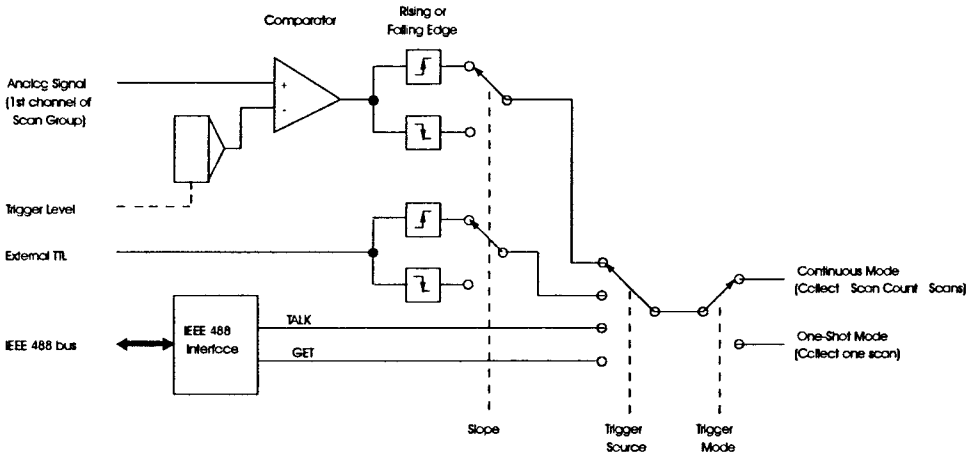


Figure 4.2: ADC488 Trigger Logic

4.3.1 Trigger Source

The Trigger (Tn) command is used to set up the trigger source and trigger mode of the ADC488. The Tn command is used to set the source of the trigger, the direction of the trigger source level if used (rising or falling edge), and the trigger mode, or how the ADC488 responds to successive triggers (one-shot mode or continuous mode).

Any of the following may be used as a source of trigger events:

TALK (T0 and T6)
GET (T1 and T7)
Analog level (T4 and T5)
External TTL Level (T2, T3, T8, and T9)

When TALK is selected as the trigger source (T0 or T6), data collection will be started every time the ADC488 is addressed to TALK and is not being serial polled. The ADC488 will not be triggered, however, if a Status (Un) command or Query (?) option is used.

When GET is selected as the trigger source (T1 or T7), data collection is started whenever the IEEE 488 Group Execute Trigger (GET) command is sensed on the IEEE 488 bus and ADC488 is addressed to LISTEN.

When Analog Level is selected as the trigger source (T4 or T5), the ADC488 monitors the signal on the first channel specified in a Scan Group (Cn,n,n,n) for a specified level (Lval) as it is transferred into memory.

When External TTL Level is specified as a trigger source (T2, T3, T8, or T9), a TTL level signal on the external trigger connector (rear panel BNC) will generate a trigger. The external trigger is edge sensitive, and triggering on the rising edge or falling edge can be specified.

The response to GET or External TTL Level trigger sources has a maximum latency of one Scan Group conversion time (number of channels in Scan Group multiplied by 10 microseconds) if trigger mode is one-shot.

The response to GET, Analog Level, or External TTL Level trigger sources has a maximum latency of the Scan Interval (In) if trigger mode is continuous.

4.3.2 *Trigger Level*

The Trigger Level (Lval) command is associated with the Analog Level trigger source command (T4 or T5). Trigger Level specifies the percentage value of the analog input voltage that is used as the trigger level. Once that level is detected with the proper slope (rising or falling edge), a trigger is generated and data collection starts. The analog level trigger always acts on the first channel specified in a Scan Group command (Cn,n,n,n). If no level is specified, L0 (0%) is used. When using Analog Level triggering, the scan buffer stores any readings collected before the trigger as pretrigger scans (continuous mode).

4.3.3 Trigger Mode

The trigger mode defines how the ADC488 responds to successive triggers on a selected trigger source. Trigger mode may be set to one-shot mode or continuous mode.

In one-shot mode, a trigger causes one scan to be taken and stored into the scan buffer. The trigger is then re-armed and the sequence is repeated. Once a specified number (Nm) of scans have been taken and stored into the scan buffer, triggers are disarmed.

In continuous mode, a trigger causes a specified number of scans to be taken (Nn,m) at the Scan Interval (In) and stored into the scan buffer. Triggers are then disarmed.

4.3.4 Pretrigger and Post-Trigger Scans

Although there may be a convenient trigger source available to use to capture data, a region of interest may actually occur some time before or after a trigger event. Pretrigger scans are those that occur before a trigger point and post-trigger scans are those scans which occur after a trigger. Delay scans are those that occur after the trigger, extending the "effective" trigger point. The Scan Count (Nn,m) command is used to specify the number of pretrigger and post-trigger scans captured into the scan buffer. The number of delay scans is specified with the Trigger Delay (Zn) command.

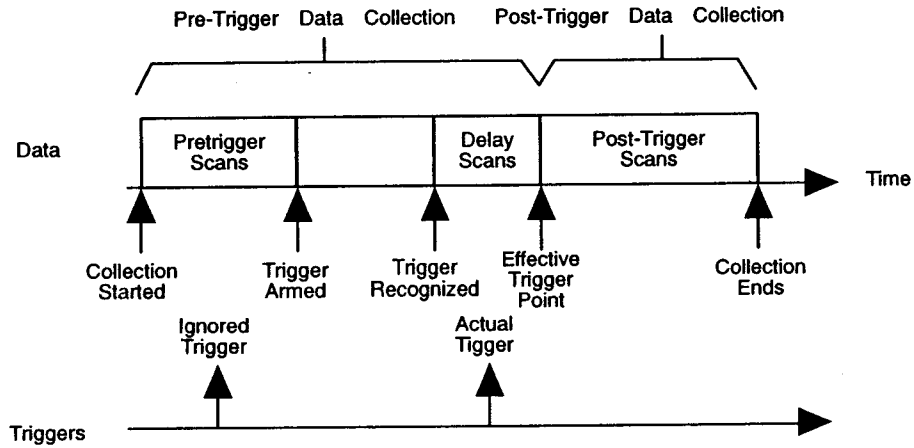


Figure 4.3: ADC488 Triggered Data Collection

When the ADC488 is set to continuous trigger mode and pretrigger scans are requested, or Analog Level trigger is selected, readings are collected from the moment that the trigger source is specified, even before the occurrence of a trigger (see the triggered data collection flowchart). Readings are collected and stored in the scan buffer to preserve pretrigger information. The maximum amount of pretrigger information available is equal to the scan buffer memory size minus the number of post-trigger scan counts that were specified.

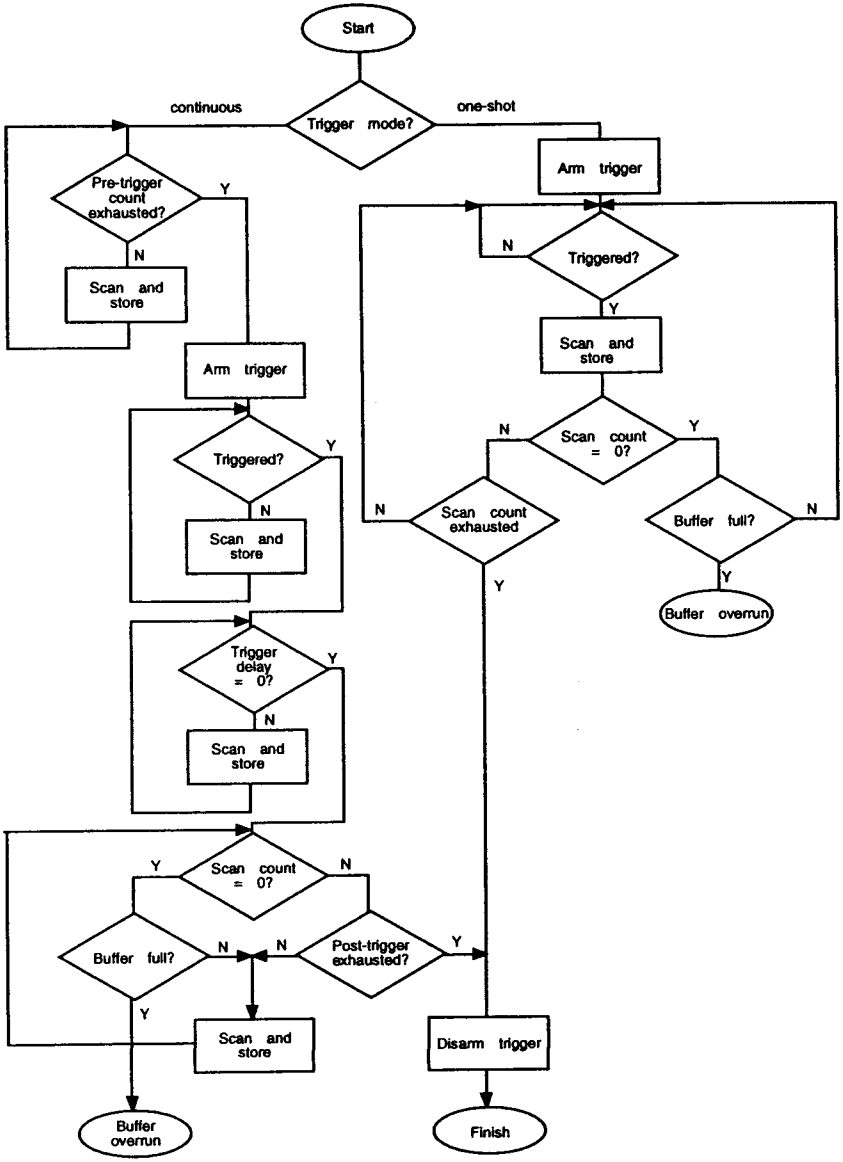


Figure 4.4: ADC488 Triggered Data Collection Flowchart

There will be times when a trigger event will occur immediately after starting data collection, thus providing little or no pretrigger information. To prevent this from happening, a pretrigger scan count can be specified to guarantee a minimum amount of pretrigger scans collected before a trigger is recognized. When pretrigger is specified, the minimum amount of pretrigger information available is equal to this amount. The maximum amount of pretrigger information available is equal to the scan buffer memory size minus the number of post-trigger scans specified in the Nn,m command. The number of pretrigger and post-trigger scans that were stored in the scan buffer can be determined by using the U2 Status command.

Any trigger that occurs during the specified minimum number of pretrigger scans is ignored. Triggers are not even enabled (Armed) until the specified number of pretrigger scans have been collected.

4.3.5 Scan Count

The number of scans that are taken in response to triggers is specified by the Scan Count (Nn,m) command. Both pretrigger scans (n) and post-trigger scans (m) can be specified. The number of scans may be set to any number from 0 to the size of any channel buffer (scan buffer size divided by the number of channels specified in a Cn,n,n,n command). Setting the scan count to 0 (N0 or N0,0) results in an infinite number of scans being taken.

Pretrigger is specified as a number of scans and cannot exceed 16,000,000 scans. Pretrigger scans will not be used for one-shot trigger mode.

If the IEEE 488 controller is capable of taking the readings from the ADC488 as fast as they are being acquired, data collection may proceed indefinitely. If the controller is not capable of maintaining this rate, or readings are not removed from the buffer at all, data collection proceeds until the buffer memory is full. A buffer overrun error is then generated and data collection continues.

4.3.6 Trigger Delay

Although there may be a convenient trigger source available to capture data, a region of interest may actually occur some time after a trigger point. The trigger point can be delayed, shifting the effective trigger point, by using the Trigger Delay (Zn) command. This trigger delay is used to shift the "real" trigger point to some later time (the Effective Trigger Point). The Effective Trigger Point becomes the trigger point (location 0) in the scan buffer. Any readings collected during the trigger delay period are stored as pretrigger data.

Trigger Delay is specified as number of scans. Trigger Delay will not be used for any of the one-shot trigger modes.

4.3.7 Service Request

A Service Request (SRQ) can be generated to inform the IEEE 488 bus controller that the ADC488 has been triggered or that it is finished with data collection. Issuing an SRQ Mask (M1) command generates a service request when a valid trigger is detected from the programmed trigger source. Issuing a SRQ Mask (M128) command generates an SRQ when data collection is completed.

4.4 ADC488 Scan Buffer

The ADC488 maintains an internal scan buffer consisting of 2048 locations, numbered 0 to 2047. The following scan buffer locations are available with optional MEMX expansion boards: 131,072 locations (MEMX1); 262,144 locations (MEMX2); 1,048,576 (MEMX5); 2,097,152 (MEMX6) or 4,194,304 (MEMX7). The scan buffer stores readings for each channel in a scan group at every scan. The Scan Buffer Pointer (Bn), Buffer Pointer Mode (Fn), and Buffer Select (Pn) commands are used to get the reading from the scan buffer either during data acquisition, or after data acquisition has been completed. These commands can be used to output data from one reading, one scan, one channel, or all the readings in the scan buffer. The scan buffer is cleared each time a Trigger (Tn) command is issued.

Trying to output readings from the scan buffer when no readings are available will "hang" the IEEE 488 bus until readings are available in the buffer. If pretrigger scans are specified, or if an Analog Level is specified as a trigger source, data from the scan buffer is available before the ADC488 has been triggered. Otherwise, trying to access the scan buffer after changing the Trigger command will hang the IEEE 488 bus until ADC488 has been triggered and readings are available.

4.4.1 Scan Buffer Size

The size (number of locations) of each channel buffer is dependent on the number of input channels that are defined using the An and C,n,n,n,n commands and the number of pretrigger and post-trigger scans specified with the Nn,m command. Channel buffers share the scan buffer equally. If one channel is defined, it can have all scan buffer locations available. If two channels are used and no Scan Buffer Expansion is installed, each buffer can contain 1024 locations. If four channels are used, each can have 512 locations. With eight channels, 256 locations are available. Sixteen input channels can be configured, each having a maximum of 128 buffer locations.

4.4.2 Scan Buffer Organization

The scan buffer is organized as shown in the Figure below. The scan buffer is shared by all input channels in a scan group and each channel has a separate channel buffer. Each reading (sample) is stored at a buffer location for every scan and the buffer pointer points to all the readings for one scan. The maximum size of any channel buffer is equal to the size of the scan buffer (number of locations) divided by the number of channels specified in the Scan Group (Cn,n,n,n) command.

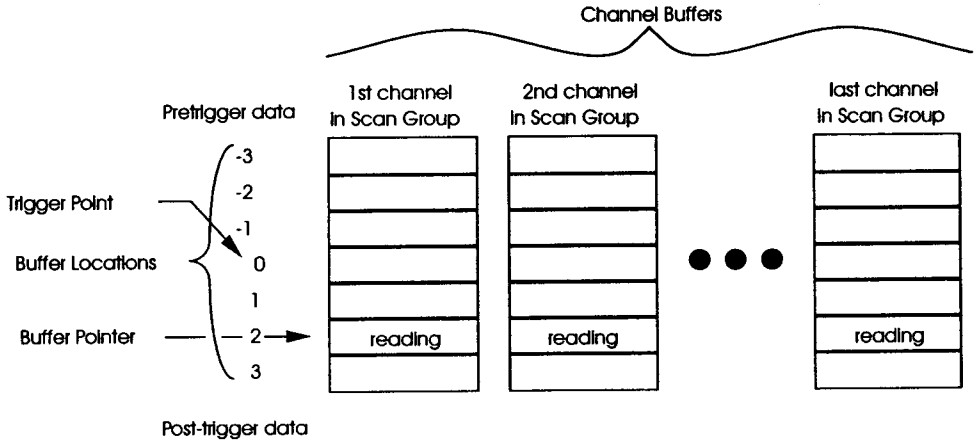


Figure 4.5 ADC488 Scan Buffer Organization

Scan buffer location 0 is always the trigger point (in the case of one-shot trigger mode, the first trigger point). Pretrigger readings, if available, are accessed with negative scan buffer locations. The oldest (least recent) pretrigger readings have the most negative value. Post-trigger readings, if available, are accessed with positive locations. The newest (most recent) post-trigger reading has the most positive value. The buffer locations containing the oldest (first) pretrigger scan and newest (last) post-trigger scan can be found using the U2 Status command.

4.5 Scan Buffer Data Storage Examples

The organization of data in the scan buffer after the ADC488 has been triggered will look like one of the following cases.

4.5.1 One-Shot Mode

For this example, trigger mode is one-shot. Every time the ADC488 is triggered, a scan is converted and stored in the scan buffer. Readings may be output to the bus as soon as they are available in the scan buffer.

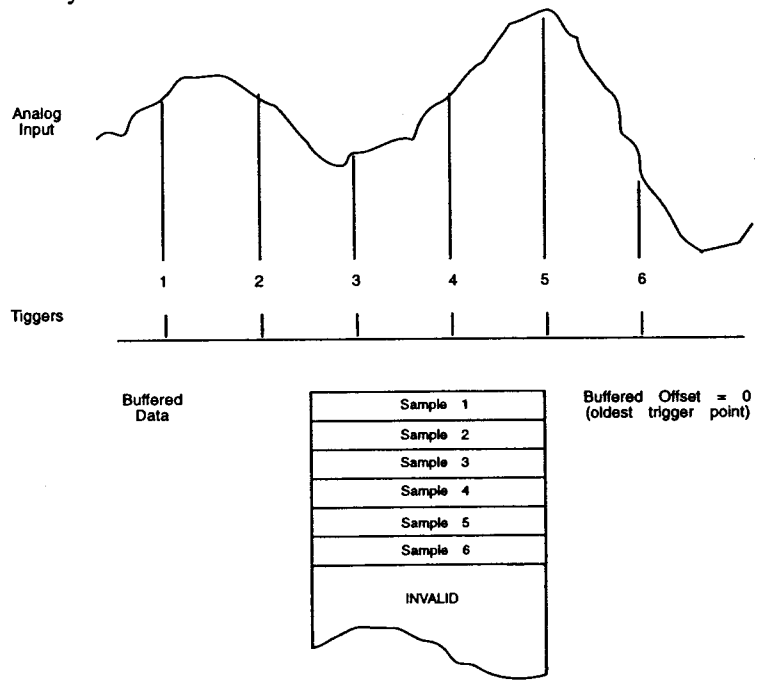


Figure 4.6: One-Shot Mode

If the scan buffer pointer mode is set to auto-increment, readings can be output as they are collected. If the trigger events are occurring slower than the rate at which the IEEE 488 controller can read them from the bus, the bus hangs until a trigger is detected and a new reading is available.

4.5.2 Continuous Mode With Pretrigger and Post-Trigger Scans

For this example, trigger mode is set to continuous and pretrigger readings are collected. When a trigger is detected, data collection is completed, and the scan buffer looks like that shown in the following figure.

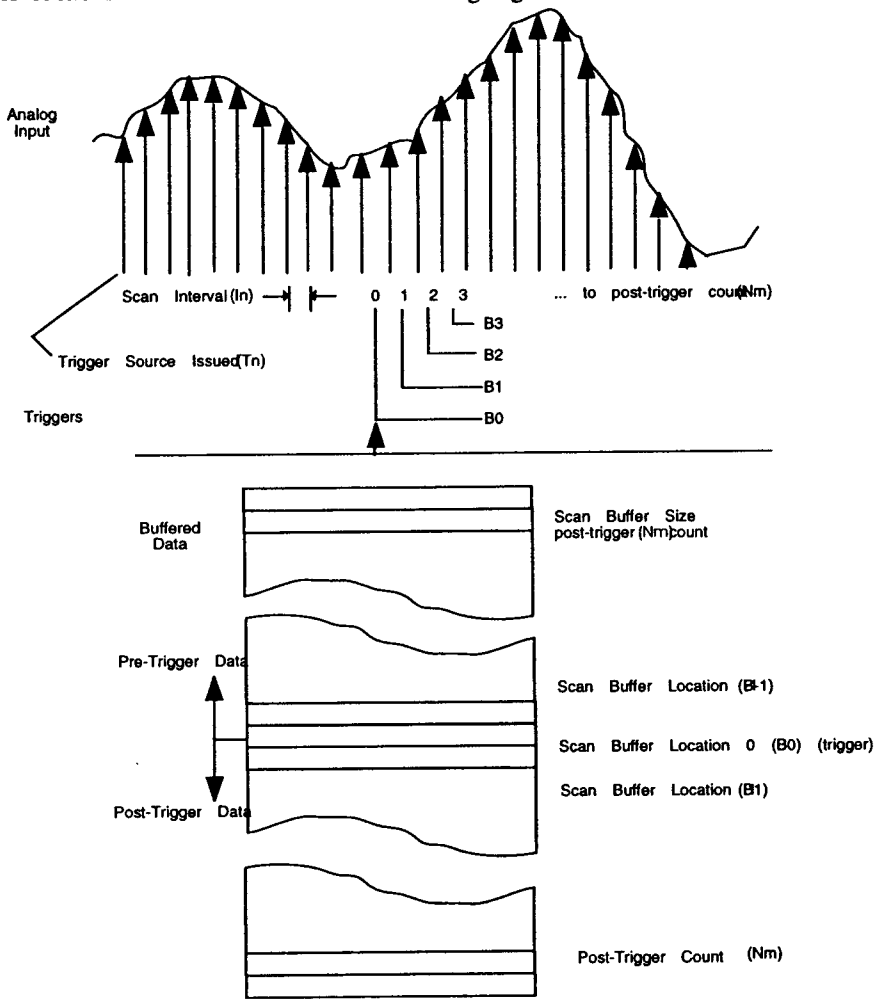


Figure 4.7: Continuous with Pre- & Post-Trigger Scans

4.5.3 Continuous Mode With Post-Trigger and No Pretrigger

For this example, trigger mode is set to continuous, and the scan count set to 0 (N0). The ADC488 will continue to transfer readings as long as the IEEE 488 controller can maintain data transfers at the same or greater rate than the rate data is being converted and buffered in the ADC488. If the IEEE 488 controller attempts to read data before the trigger event, the IEEE 488 bus will "hang" until the trigger event is detected. The scan buffer in this case is being used as a FIFO buffer.

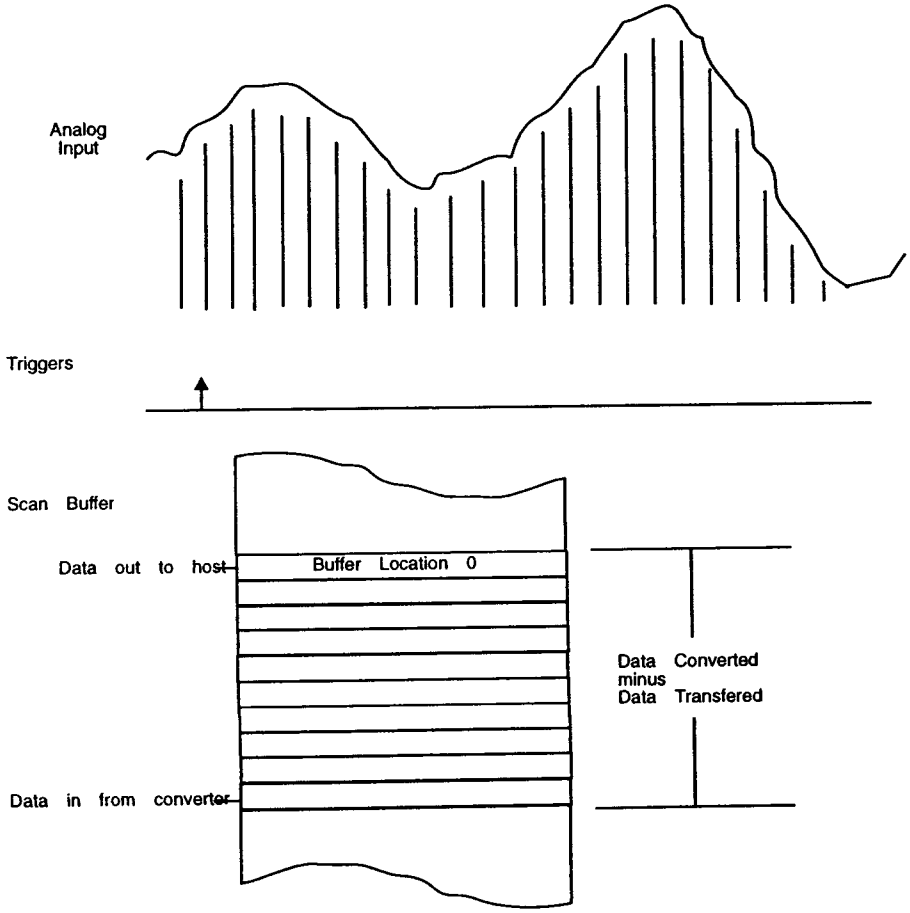


Figure 4.8: Continuous Mode with No Pretrigger Scans

This example causes destructive reads of the data from the IEEE 488 bus, that is, as a reading is output from the scan buffer, it is no longer available. The scan buffer location just accessed is freed up for a new reading from the converter. For this reason, it is important to output readings in this mode from channel 0 (P0), sending entire scan groups of readings each time. Sending the reading from a single channel will cause the loss of any information from other channels specified in the Scan Group.

If the IEEE 488 controller is unable to maintain data transfer across the bus at the rate at which it is acquired, eventually the scan buffer will fill, leaving no room for more readings. A buffer overrun error will be generated. Data collection will continue, but only the newest available readings will be output from the ADC488. Reading the scan buffer after a buffer overrun may cause discontinuities in the data. The ADC488 can be programmed to generate an SRQ if a buffer overrun error occurs by using the SRQ Mask (M4) command.

4.6 Data Output

Once scans have been made and readings stored in the scan buffer, the readings can be output to the IEEE 488 bus. The Scan Buffer Pointer (Bn), Buffer Pointer Mode (Fn), and Buffer Select (Pn) commands are used to access the scan buffer. End of Reading Terminator (On), End of Scan Terminator (Qn), and User Terminator (Jn) commands are used to control message terminators. The Output Format (Gn) command selects a particular data format.

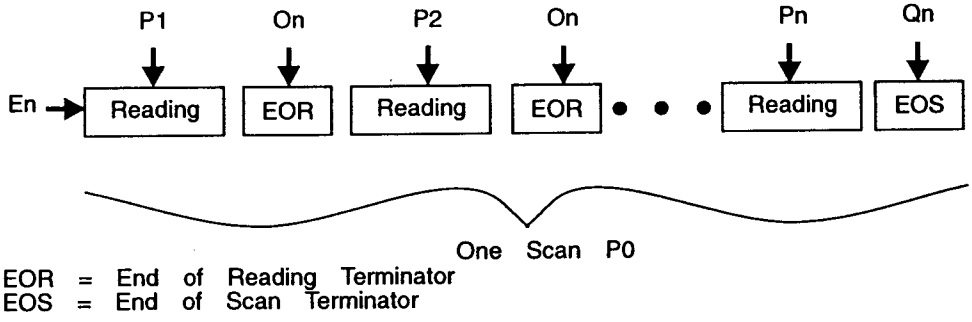


Figure 4.9: ADC488 Scan Buffer Terminators

Digital I/O values and command channel terminators can also be set. The Digital Output (Dn) and U1 Status control and monitor the lines on the rear panel digital connector. The Command Channel EOI (Kn) and Command Channel Terminator (Yn) commands set the terminators on query and status messages.

4.6.1 Scan Buffer Pointer

The Scan Buffer Pointer command (Bn) is used to set the scan buffer pointer to a specified location. The scan buffer pointer points to all the readings in one scan. Immediately after a data acquisition sequence has been completed, the buffer pointer is set to 0. The buffer locations containing the first (oldest) pretrigger scan and last (newest) post-trigger scan can be found using the U2 Status command.

4.6.2 Buffer Select

The Buffer Select command (Pn) is used to select a particular channel buffer, or all channel buffers. Once a channel buffer has been selected, the reading from that buffer can be output to the bus.

The readings for a complete scan may be accessed by specifying channel 0 (P0) of the ADC488, or the reading for a specific channel may be accessed by specifying a particular channel buffer (P1 through P16). If P0 is used, End of Reading (On) terminators are used between readings and an End of Scan (Qn) terminator is sent after the last reading in the scan. If a specific channel is accessed (Pn), only an End of Scan Terminator is sent after the reading.

The order of channel buffers corresponds to the order of input channels that are specified in the Scan Group (Cn,n,n,n) command. If, for example, a C1,3,2,1 command is used, four channel buffers will be available: P1 for channel 1, P2 for channel 3, P3 for channel 2, and P4 for another buffer of channel 1 readings.

4.6.3 Buffer Pointer Mode

The Buffer Pointer Mode (F0) command is an obsolete command that used to specify how the buffer pointer will move each time readings are returned from the scan buffer. The buffer pointer mode is always auto-increment, meaning that the buffer pointer is incremented after each scan is read.

The scan buffer pointer points to a group of channel buffers in the scan buffer. As an example, assume that a scan group is set for two channels (Channel 1 and 2), the scan buffer pointer is set to zero, and the buffer pointer mode is set to auto-increment. Two cases should be considered:

The first is when readings are requested from channel P0. When readings are requested from P0, the first reading sent is the reading in channel buffer number 1 at scan buffer pointer location 0. It is followed by the reading from channel buffer number 2, scan buffer pointer location 0. When the end of the scan group is reached (two channels in this example), the scan buffer pointer is auto-incremented to point to scan buffer pointer location 1. The next request for data will return the reading from channel buffer number 1, scan buffer pointer location 1. This continues until all the readings are output from the scan buffer.

The second case is when a reading is requested from one of the channel buffers. When a reading is requested from channel 1 (P1), the reading in channel buffer number 1 at scan buffer pointer location 0 is sent, and the scan buffer pointer is incremented immediately after sending the reading. Subsequently, requesting the reading from channel 2, (P2), will result in receiving, not the first reading in channel buffer 2, but the second (channel buffer number 2, buffer pointer location 1) due to the auto-incrementing of the scan buffer pointer.

Note that the scan buffer pointer is a global pointer for all channels. Requesting data from any one channel will affect the pointer.

4.6.4 End of Reading Delimiters

The End of Reading (On) command is used to define how readings from the scan buffer may be separated by unique characters when outputting scans from the scan buffer. The following delimiters may be used:

CR LF
LF CR
CR
LF
user-defined (any numeric value between 0 and 255)

In addition, EOI may be asserted on the last byte of each reading when returning readings in any format. End of Reading delimiters are not used in any of the binary output data formats.

4.6.5 End of Scan Delimiters

The End of Scan (Qn) command is used to define how entire scans that are output from the scan buffer may be separated by unique characters. The following delimiters may be used:

CR LF
LF CR
CR
LF
user-defined (any numeric value between 0 and 255)

In addition, EOI may be asserted on the last byte of each scan when returning readings in any format. End of Scan delimiters are not used in any of the binary output data formats.

4.6.6 User Terminator

The User Terminator (Jn) command can be used to define any character for an end-of-reading or end-of-scan terminator. Once a user terminator has been programmed,

it can be selected with the End Of Reading (O8 or O9) or End of Scan (Q8 or Q9) user terminator commands.

4.6.7 Data Output Format

Data output format is determined by the Output Format (Gn) command. Readings from the ADC488 may be output in six basic data formats: ASCII fixed point with or without a leading plus sign, signed integer decimal, ASCII hexadecimal, or binary with high byte first or low byte first. All six data formats are available with or without compensation for gain and offset for a total of twelve formats.

ASCII fixed point format returns the fixed point representation of the voltage for each converted value in the following format:

±nn.nnnnn

Decimal format returns a signed integer decimal representation of each reading.

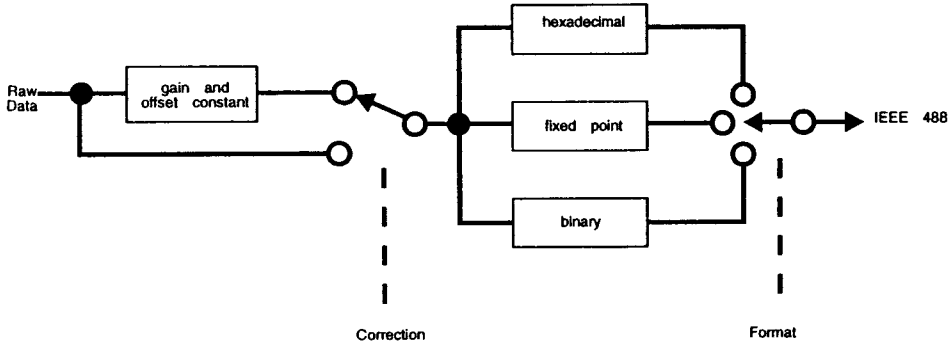


Figure 4.10: Data Output Logic

-nnnnn

ASCII hexadecimal format returns the four-digit hexadecimal representation of the twos-complement binary for each converted value in the following fixed format:

hhhh

Binary returns two bytes of two's-complement binary data for each converted value. End of Reading and End of Scan terminators are not used with binary data transfers. EOI is asserted on the last byte.

low byte first: <low byte> <high byte>
high byte first: <high byte> <low byte>

4.6.8 Data Correction

Data Correction can be applied to any of the basic output formats. When a format command that uses data correction is selected, the ADC488 will correct readings for offset and gain errors in the conversion process. The data correction process is performed by the microprocessor after readings have been stored in the scan buffer.

4.6.9 Digital I/O

The Digital Output (Dn) command is used to set the digital output state of the lines on the rear panel digital I/O connector. The digital input state can be determined by using the U1 Status command.

4.7 Data Correction Theory

Readings taken by the ADC488 are corrected by a software algorithm to compensate for errors introduced by the analog circuitry and the A/D converter used in ADC488 to convert analog voltages to digital values.

In order to provide overhead to apply correction to readings the full-scale values were reduced from ±32,767 to ±30,000. This does imply a reduction in resolution over the full range of A/D converter readings, but the difference is minuscule. Each count of difference represents 0.003333% of full-scale instead of 0.003051% of full scale.

4.7.1 Analog Errors

An ideal (that is, perfect) A/D converter would have a response to input voltages as shown in the Figure below.

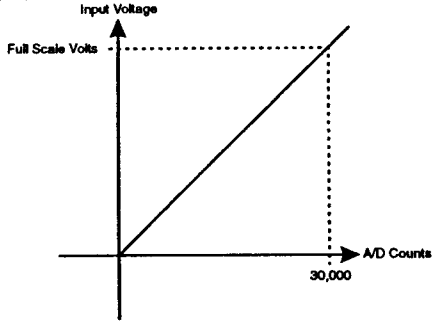


Figure 4.11: Ideal A/D Converter Response

Note that the response to input voltages is a perfectly straight line that crosses the origin (0 counts for 0 volts input) and a slope of 30,000 / full scale volts (30,000 counts for + full scale volts input).

A real A/D converter has two sources of error, however, that cause it to deviate from this ideal response to input voltages as shown in the Figure below.

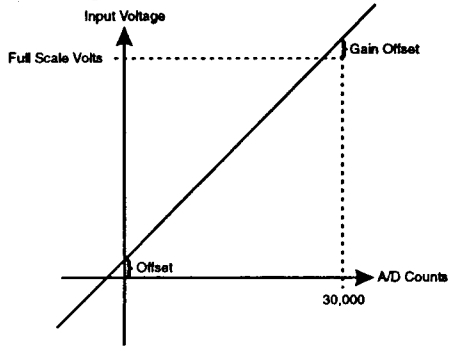


Figure 4.12: Deviation from Ideal Response

Offsets introduced by the analog circuitry preceding the A/D converter cause a small voltage to be added to or subtracted from the actual analog voltage input to the ADC488. This is a constant that is most noticeable when 0 volts *is* applied to the ADC488 input. Instead of 0 volts, the reading will be some small plus or minus voltage. This is the offset.

Gain errors are introduced by small differences in the gain stages preceding the A/D converter that either amplify or attenuate the signal before presenting it to the A/D converter input for conversion to a binary value. These differences actually cause a slight change in the slope of the line representing the response of the ADC488 to input voltages. This error has the most effect and is most noticeable at full scale. Gain errors cause the actual reading at full-scale voltage to be somewhat more or less than the ideal value of 30,000 counts.

4.7.2 Calibration

Data are compensated for two different sources of error: zero offset and gain. The calibration process for ADC488 allows the microprocessor in the ADC488 to measure the differences caused by offset and gain errors and to calculate calibration constants that are used during data collection to correct readings for offset and gain errors.

Note: To ensure accuracy, the ADC488 must be re-calibrated on a twelve month basis.

At 0 volts, the gain error is small enough to be negligible and offset error dominates the reading. At full-scale volts the reading error is the combination of both offset

and gain errors. For these reasons the calibration process first measures the offset error and calculates the offset compensation with 0 volts applied to the analog input. Gain errors are then measured with full-scale voltage applied to the analog input, and, with the offset error compensated for, the gain compensation is calculated. As the last step in the calibration process, the calculated compensation values are stored in battery-backed-up non-volatile RAM in the ADC488.

The exact calibration procedure and the commands to be used are described under the Hn command in Chapter 5. The diskette included with the ADC488 manual also includes a sample program that calibrates the ADC488.

4.7.2.1 Calibration Constants

When the ADC488 has been calibrated, there are two constants associated with each channel and each range on that channel to allow readings to be corrected. These constants relate to the offset errors and the gain errors for that channel and that range.

offset — The offset calibration constant is a signed 16-bit number that represents the number of A/D converter counts read from the channel at that gain with 0 volts applied. This number should be subtracted (signed subtract) from the raw uncorrected reading.

gain — The gain calibration constant is an unsigned 16-bit number that is returned from the ADC488 to indicate a channel's variation in gain. A "perfect" channel would have a value of 32,768 (8000H). Slight variations above or below the "perfect" gain slope are compensated for by correction values that are slightly more or less than 32,768. This number is divided by 32,768 and multiplied by the raw uncorrected reading to get corrected counts of A/D reading.

When the ADC488 corrects raw readings from the A/D Converter, it uses the following algorithm:

$$\text{Corrected Reading} = \frac{(\text{Uncorrected Reading} - \text{Offset Correction}) \times \text{Gain Correction}}{32{,}768}$$

4.7.2.2 Reading the Calibration Constants from ADC488

The calibration constants may be read from ADC488 by issuing a U4X command. ADC488 will then return 16 numbers in ASCII decimal, separated by commas:

```
+00030,32755,-00010,32787,+00012,32768,...  [etc.]
```

Note that these numbers are grouped in pairs, with the first of each pair being an explicitly signed number, and the second unsigned. Each of these pairs of numbers

represents the offset, then the gain calibration constant for the currently programmed range.

Sixteen pairs of numbers are returned, representing the 16 channels available on an ADC488/16 configured for single-ended inputs. For ADC488/8S or ADC488/16 configured for differential inputs, only the first eight pairs have any significance.

Calibration constants are returned for each channel specifically for the range each channel is programmed for when the U4X command is issued. If it is desired to have the calibration constants for all channels and all ranges, each channel must be re-programmed for different ranges and the calibration constants read again using the U4X command. As an example, the QuickBasic 4.5 code fragment below reads in the calibration constants for all 16 channels for all four ranges into two arrays whose first index is the channel number and whose second index is the range:

```
DIM offset%(16,4)
DIM gain&(16,4)
FOR range = 0 TO 3
    ' 1st, program all channels to the current range
    FOR channel = 1 TO 16
        PRINT #1, "output14; r#"; channel ; ","; range; "x"
    NEXT channel
    ' issue the command to read cal. constants from ADC
PRINT #1, "output14;u4x"
    ' read in all the cal. constants for current range
    PRINT #1, "enter14"
    FOR channel = 1 TO 16
        INPUT #2, offset%(channel, range), gain&(channel,range)
    NEXT channel
NEXT range
```

Note that the offset array is typed as 16-bit integer (%), while the gain array is typed as 32-bit integer (&). QuickBasic, as well as most other languages, cannot store numbers larger than ±32,767 into a 16-bit integer variable. Numbers larger than 32,767 are interpreted as negative. If the gain values were to be read into a 16-bit integer variable, either the program will report an "overflow error," or the value will be read in, but interpreted as a negative value. In either event, any math that attempts to use these values for later calculations will not be correct.

4.8 ADC488 Commands

Twenty six commands, most with query options, are available to control the ADC488. How these commands are used is described in this section. Each command is detailed in Chapter 5. GW-BASIC and QuickBASIC programming examples are included that use an IOtech Driver488 IEEE 488 board and associated software.

Input Configuration	An	Set input mode.
Scan Buffer Pointer	Bn	Set scan buffer pointer.
Scan Group	Cn,n,n,n	Define input channels.
Digital Output	Dn	Set the digital output lines.
Error Query	E?	Return error status.
Buffer Pointer Mode	Fn	Set scan buffer pointer action.
Output Format	Gn	Select data output format.
Calibration	Hn	Calibrate analog inputs.
Scan Interval	In	Select the time between scans.
User Terminator	Jn	Set user terminator.
Command Channel EOI	Kn	Enable or disable EOI.
Trigger Level	Lval	Set analog trigger level.
SRQ Mask	Mn	Set service request conditions.
Scan Count	Nn,m	Set pretrigger and post-trigger scans.
End of Reading Terminator	On	Set end-of-reading terminator.
Buffer Select	Pn	Select a scan buffer number.
End of Scan Terminator	Qn	Set end-of-scan terminator.
Range	Rn,n,n,n	Set analog input range.
Setup (Save,Recall)	Sn,m	Save or recall a configuration.
Trigger	Tn	Select trigger source.
Status	Un	Return system status.
Version Query	V?	Return firmware revision level.
Test	Wn	Test ADC488-to-bus communication.
Execute	X	Execute command string.
Command Channel Terminator	Yn	Set command channel terminator.
Trigger Delay	Zn	Set trigger delay.
Query	cmd?	Return command status.

4.8.1 Command Options

ADC488 commands are one alpha character followed by one or more parameters or options. If more than one parameter is used, they are separated by commas (,) as: C4,3,2,1. Spaces are ignored: A 0 is the same as A0. If commands are used without the required number of options, an Invalid Device Dependent Command Option (IDDCO) error will be generated. If parameters are selected that are inconsistent with other commands or modes, a Conflict Error will be generated. See Section 4.8.5 for a more detailed description of conflict errors.

4.8.2 Command Execution

The X command is used to execute ADC488 commands. Commands sent to the ADC488 are not executed until an X is sent. An X is usually the last command of a command string. The following example:

```
PRINT#1,"output14;A0C1R3I3T0X"
```

is the same as:

```
PRINT#1,"output14;A0"
PRINT#1,"output14;C1"
PRINT#1,"output14;R3"
PRINT#1,"output14;I3"
PRINT#1,"output14;T0"
PRINT#1,"output14;X"
```

4.8.3 Command Order

Once an X is sent to the ADC488, commands in the ADC488 command buffer are executed as a group. Some commands affect other commands so the order they are sent can be important. The An command, for instance, sets the input configuration to single-ended or differential inputs. Using the An command affects the Scan Group (Cn,n,n,n) command as well as the Scan Interval (In), Buffer Select (Pn), and Scan Count (Nn,m) commands by setting those command options to their default values.

For example, sending the following command string results in eight differential input channels, using channels one through four, scanned from channel four to channel one:

```
PRINT#1,"output14;A0X"
PRINT#1,"output14;C4,3,2,1X"
```

Sending the same command string in a different order results in a different configuration:

```
PRINT#1,"output14;C4,3,2,1X"
PRINT#1,"output14;A0X"
```

These commands result in one channel (channel 1). The A0X command resets the ADC488 scan group to the default value of the Scan Group command (C1,0,0,0,0,0,0,0).

4.8.4 Query Options

A question mark (?) following a command is a query option. It returns a string that shows the current value of the command option. Query commands usually return the last value programmed for that command. The Error Query (E?) command is used to return any ADC488 error conditions. The Version Query (V?) command returns the revision level of the ADC488 firmware.

4.8.5 Conflict Errors

Some combinations of commands and parameters can be sent to the ADC488 that are out of range for a particular configuration or inconsistent with other commands. For instance, specifying a scan interval (In command) to less time than the ADC488 can make a scan (the product of the number of defined channels times ten microseconds) results in a conflict error:

```
PRINT#1,"output14;c1,2,3,4i1x"
```

A conflict error lights the ERROR indicator light on the front panel and returns an E4 when queried with the Error Query (E?) command. Some conflict errors result in a default value for a conflicted command. For instance, in the example shown above, the scan interval defaults to the fastest possible scan interval for the number of defined channels.

4.9 Status Reports

Status commands are used to return information about various conditions of the ADC488. The status of the ADC488 may be read at any time without interfering with normal operation. Status strings are returned when the ADC488 is next addressed to Talk. If the ADC488 is configured to trigger on TALK, requesting a status report will not cause the ADC488 to trigger.

4.9.1 Command Status

The U0 Status command returns a fixed-length status message string that shows the current options in use for each command:

```
*.*A#B#######C##,##,##,##,##,##,##,##,##,##,##,##,##,##,
##D###E##F#G##H#I##J###K#L###M###N#######,#######O#P##Q#R#,
#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#,#S#,#T#U#W#Y#Z######
```

where each # shows the current option in use for that command. The leading information *.* is the revision level (V?) of the ADC488 firmware.

Any error conditions, except Calibration Errors and NVRAM Setup Errors, are cleared after the command status string is read by the controller. Calibration errors are cleared when the ADC488 is calibrated and NVRAM Setup errors are cleared when a setup is saved.

4.9.2 Digital Input Lines

The U1 Status command returns a string in the form: ###, where ### is the decimal equivalent of the binary value of the eight digital input lines of the rear panel digital I/O connector. For example, 054 would mean 00110110, (from pin 8 [MSB] to pin 1 [LSB]) where a 0 means the line is low (ground), and a 1 means that the line is high (+5V).

4.9.3 Pretrigger and Post-trigger Address Limits

The U2 Status command returns a string in the form: -#######,+#######

where ####### indicates the limits of the pretrigger (minus value) and post-trigger (positive value) scans in the scan buffer. The first number is the scan buffer location containing the first recorded sample (oldest reading) and the second number is the scan buffer location containing the newest sample (most recent sample recorded). These values are the limits of the scan buffer pointer that can be specified with the Scan Buffer Pointer (Bn) command.

4.9.4 Internal Scan Counter

The U3 Status command returns a string in the form: ±######, where ###### is the location within the ADC488's memory of the next already aquired scan to be read. The U3 Status command is valid at all times.

Once the Scan Buffer Pointer (Bn) command has been issued, every time the Buffer Select (Pn) command is issued, the internal scan counter will be reset to the value specified in the Scan Buffer Pointer (Bn) command. For example:

```
PRINT#1,"output14;b0x"
FOR P=1 to 4
     PRINT#1,"output14;p";P;"x"      The counter (U3) is reset to Bn
     FOR N=0 to MAXLIMIT
         PRINT#1,"enter16"
```

```
        INPUT#2,A$(P,N)
    NEXT N
NEXT P
```

Note that the value for the Scan Buffer Pointer (Bn) does not change as data is read from the scan buffer. Instead, the internal scan counter is automatically incremented. Use U3 to find the current pointer:

```
PRINT#1,"output14;b0x"
PRINT#1,"enter14"
INPUT#2,R$
PRINT R$
PRINT#1,"output14;u3x"
PRINT#1,"enter14"
INPUT#2,U$
PRINT U$
```

Returns: +00001; the pointer was incremented after scan 0 was read.

When reading data while data is still being collected, U3 may be used to determine how much of the data stored in the scan buffer has already been read from the ADC488. The internal scan counter, U3, is updated after data acquisition is complete and contains the scan buffer location that will be returned on the next data read from the ADC488. The following example illustrates this point by using U3 to determine whether the trigger point (b0) has already been buffered into the controller's memory, or if it must be read from the ADC488 after data acquisition has finished.

```
' Set up the ADC488 for Analog Level Trigger

' Wait until ADC488 is ready (pretrigger data is
' being collected).
GOSUB WaitOnReady
' Now start reading pretrigger data immediately
' without waiting until data acquisition is complete
FOR I=1 TO 150,000
        PRINT#1,"enter14"
        INPUT#2,A$(I)
        PRINT#1,"spoll14"
        INPUT#2,SPOLL$
        IF (VAL(SPOLL$) AND &H80) GOTO exitloop
NEXT I
exitloop:
' Data collection is finished. Use the U3 Status to
' determine if the trigger point is already buffered
```

```
' into the controller memory

PRINT#1,"output14;u3x"
PRINT#1,"enter14"
INPUT#2,LASTPTR
IF LASTPTR < 0 ' Did not read past the trigger
        PRINT#1,"output14;b0x"
' get ready from the ADC488
        PRINT#1,"enter14"
        INPUT#2,A$
' Print reading
        PRINT A$
END
END IF
' Trigger point is already in the controller memory
' Get it and print it
PRINT A$(I-LASTPTR)
' Print the trigger point
END
WaitOnTriggerArmed:
' Serial poll the Trigger Armed bit until ADC488 is ready
        PRINT#1,"spoll14"
        INPUT#2,SPOLL$
IF (VAL(SPOLL$) AND &H10) EXIT SUB ELSE GOTO WaitOnReady
```

4.9.5 Calibration Constants

The U4 Status command returns the calibration constants used by the ADC488 to correct data read from the Analog-to-Digital converter. These values are returned as sixteen pairs of numbers (only eight pairs have significance for an ADC488/8SA or an ADC488/16A configured for differential inputs). The values reflect the present offset and gain correction factors for each channel based on the range that is programmed for each channel.

4.9.6 Scans Available

The U5 Status command returns a string in the form: #######, where # to ####### is the number of scans aquired but not yet read. The U5 Status command is valid while an acquisition is in process, as well as when it is complete. This command is useful in applications that perform other tasks in addition to reading

data. Trying to read absent data will cause delays and ultimately truncate the task when the IEEE 488 controller times out. The following code fragment example demonstrates this functionality.

```
CONST BytesPerChannel = 2
CONST ChannelsPerScan = 4
CONST ScansPerBuf = 1000
CONST BufLen = BytesPerChannel * ChannelsPerScan * ScansPerBuf
DIM Buf AS STRING * BufLen ' Fixed length I/O buffer
'    Open files, set up the ADC488 for a fixed length acquisition, and trigger it.
        •
        •
        •
'    Read data and write it to disk 1000 scans at a time. Simultaneously, FAX a
'    flyer to folks in our database. (Presumes the ADC488 bus address is 14.)
Done% = 0
WHILE Done% = 0
        ' Serial poll the ADC488. All done if the acquisition is complete,
        ' buffer overflow occurred, or trigger overflow occurred.
        PRINT #1,"spoll14"
        INPUT #2,SerialPoll$
        ' 80 hex: acq complete. 4: buffer overflow. 2: trigger overflow.
        Done% = VAL(SerialPoll$) AND &H86
        ' Get the number of scans available to read from the ADC488
        PRINT #1,"output14;u5x"
        PRINT #1,"enter14"
        INPUT #2,Available&
        ' In case the FaxAnother routine takes a long time, catch up.
        WHILE Available& 999
                ' Read 1000 scans and write them to file #3 until 1000 left
                PRINT #1,"enter14#";STR$(BufLen);";BUFFER &H"; _
                        HEX$(VARSEG(Buf$));":&H";HEX$(VARPTR(Buf$))
                PUT #3,,Buf$
                Available& = Available& -ScansPerBuf
        WEND
        ' Send another FAX (or whatever task you choose)
        GOSUB FaxAnother
WEND
'
IF (Available& 0) AND (VAL(SerialPoll$) AND &H80 = &H80) THEN
        ' The last scans need to be read after the acquisition completed.
        ' Only ask for the # of scans available, so we don't wait forever.
        PRINT #1,"enter14#";STR$(Available&);";BUFFER H"; _
                HEX$(VARSEG(Buf$));":&H";HEX$(VARPTR(Buf$))
        TmpBuf$ = MID$(Buf$,1,Available&)
        ' Quick Basic doesn't like the MID$ above used directly in the PUT.
```

```
        PUT #3,,TmpBuf$
END IF
```

4.10 User Defined System Configurations

The ADC488 provides non-volatile RAM (NVRAM) storage for up to eight preset configurations. Two commands are provided to use the configuration storage. Save Configuration (Sn,1) saves present ADC488 settings to NVRAM. Recall Configuration (Sn,0) sets the ADC488 to a previously stored state.

The stored configurations include the following ADC488 settings:

An	Input Configuration
Bn	Scan Buffer Pointer
Cn,n,n,n	Scan Group
Dn	Digital Output
Fn	Buffer Pointer Mode
Gn	Output Format
In	Scan Interval
Jn	User Terminator
Kn	Command Channel EOI
Lval	Trigger Level
Mn	SRQ Mask
Nn,m	Scan Count
On	End of Reading Terminator
Pn	Buffer Select
Qn	End of Scan Terminator
Rn,n,n,n	Range
Tn	Trigger
Yn	Command Channel Terminator
Zn	Trigger Delay

4.11 Calibration

It is important that the ADC488 be calibrated on a twelve month basis so it can provide the necessary compensation to the measured voltage. Three commands are provided to calibrate the input channels of the ADC488. The H0 command specifies a zero volt reference and the H1 command establishes the full scale voltage. Once calibration is complete, send the H2 command to transfer the new calibration constants

into protected calibration memory. Refer to Figure 4.13 to view an ADC488 calibration test fixture. A sample calibration program is included on the disk provided with the ADC488. Refer to the program and the Calibration (Hn) command description in Chapter 5 for more information.

4.12 Programming Examples

Included on the disk supplied with the ADC488 are several GW-BASIC and QuickBASIC sample programs that show how to use the ADC488 for common bus operations such as configuration, calibration, polling, triggering, reading data, and using the device dependent commands.

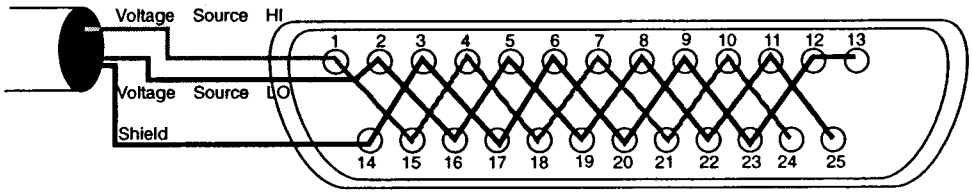


Figure 4.13: ADC488 Calibration Test Fixture

This section contains excerpts of the sample programs and some examples using the ADC488 with an IOtech Driver488 controller and the BASIC Keyboard Controller program provided with the Driver488.

4.12.1 Interactive Commands

The Keyboard Controller program supplied with the IOtech Driver488 allows direct communication with the ADC488 (or any IEEE 488 device on the bus) and is the most convenient method of exercising the ADC488 and becoming familiar with the commands and their actions.

First, make sure that S104 is set for IEEE 488 bus address 14. Connect an IEEE 488 cable from the ADC488 to the IEEE 488 connector on the Driver488 card. Turn on the ADC488 by depressing the rear panel power switch. Run the Keyboard Controller program and verify that the CMD> or IEEE> prompt appears on the screen.

The following is a listing of commands given to the ADC488 and the response received. Some responses are separated by extra vertical spaces for clarity, although these spaces will not actually appear when using the program.

OUTPUT 14;W1X	Turn on the TEST indicator light (TEST should light)
OUTPUT 14;W0X	Turn off the TEST indicator light (TEST should turn off)
OUTPUT 14;W?	Query the state of the indicator light.
ENTER 14	Address the ADC488 to Talk. Returns W0.
CLEAR 14	Clear the ADC488.
OUTPUT 14;A2X	Configure single-ended inputs (ADC488/16A only).
OUTPUT 14;C1,2,3,4,5,6,7,8X	Use eight channels.
OUTPUT 14;R3,3,3,3,3,3,3,3X	All channels set at ±10 volt range.
OUTPUT 14;I3X	Set Scan Interval to 100µs.
OUTPUT 14;N100X	Acquire one hundred scans when triggered.
OUTPUT 14;T1X	Trigger on GET Continuous Mode. (TRIGGER indicator light flashes [armed])
TRIGGER 14	Send a GET (TRIGGER indicator light turns off in 10 ms). (100 scans at 100µs scan interval).
OUTPUT 14;J44X	Set User Terminator to comma (,).
OUTPUT 14;O9X	Set End of Reading Terminator to use comma.
OUTPUT 14;Q1X	Set End of Scan Terminator to CRLF.
OUTPUT 14;P0B0X	Point to channel 0, scan buffer 0.
ENTER 14	Get the readings from the ADC488. Eight readings are returned.
OUTPUT 14;Q0X	Set the End of Scan Terminator to CRLF EOI.
ENTER 14 EOI	Get data from the ADC488 until EOI is asserted. Eight more readings are returned.

4.12.2 Using the ADC488 With BASIC

In order to use the Driver488 controller, the driver must be opened and reset. Two I/O buffers are used, one for input and one for output.

```
10  OPEN  "\dev\ieeeout"  FOR  OUTPUT  AS  #1
20  IOCTL#1,"BREAK"
30  PRINT#1,"RESET"
40  OPEN  "\dev\ieeein"  FOR  INPUT  AS  #2
```

4.12.3 Setting Up the ADC488 to Acquire Data

The ADC488 must be configured prior to use. First, the bus should be cleared and the ADC488 reset. The input configuration (single-ended or differential inputs), channels, ranges, and scan interval must then be specified.

100 PRINT#1,"abort"	Clear the bus.
110 PRINT#1,"clear 14"	Clear the ADC488.
120 PRINT #1,"output 14;a0"	Configure eight differential inputs.
130 PRINT #1,"output 14;c1,2"	Use two channels.
140 PRINT #1,"output 14;r0,0"	Set both to ±10 volt range.
150 PRINT #1,"output 14;i15"	Use a one second scan interval.
150 PRINT #1,"output 14;n1000"	Get 1000 readings.
160 PRINT #1,"output 14;X"	Execute the command string.

The whole command string could have also been sent in one line, as:

```
120  PRINT  #1,"output  14;a0c1,2r0,0i15n1000x"
```

4.12.4 Triggering

The ADC488 can use various triggers to start data acquisition. Two different modes (one-shot and continuous) can be used. Pretrigger and/or post-trigger scans can be saved, depending on the trigger mode.

200 PRINT #1,"output 14;t1x"	Setup to trigger on GET.
210 PRINT #1,"trigger 14"	Issue a GET.
200 PRINT #1,"output 14;t6x"	Trigger on TALK.
210 PRINT #1,"enter 14"	Address the ADC488 to TALK.
200 PRINT #1,"output 14;t4x"	Trigger on Analog Level.
210 PRINT #1,"output 14;l+80x"	At 80% of the level.
200 PRINT #1,"output 14;t8X"	Trigger on external TTL level.

4.12.5 Serial Polling and Service Requests

Once data has been requested, serial polling can be used to determine when data acquisition is complete:

300 PRINT #1,"spoll 14"	Serial poll the ADC488.
310 INPUT #2,STAT$	Get the status byte.
	Repeat until ready:

320 IF (VAL(STAT$) AND &H80) <> &H80 GOTO 300

Service Requests (SRQ) can also be used:

```
300 PRINT #1,"output 14;m1x"        Generate an SRQ on trigger.
310 PRINT #1,"spoll"                Poll the bus.
320 INPUT #2,SRQ$                   Get the SRQ status.
330 IF VAL(SRQ$)=0 THEN 310         Repeat until an SRQ is issued
```

4.12.6 Getting Readings From the Scan Buffer

Once readings are in the scan buffer, they can be extracted in several ways. Each channel buffer can be accessed individually or an entire scan can be output to the bus.

This example accesses the individual channel buffers.

```
400 PRINT#1,"output 14,b0x"
410 FOR I=1 to 16
420     PRINT #1,"output 14;p";I;"x"
430     FOR J=1 to 1000
440         PRINT #1,"enter 14"
450         INPUT #2,A$(I,J)
460     NEXT J
470 NEXT I
```

This example uses channel 0 to get all the readings:

```
400 PRINT #1,"output 14;p0x"
410 FOR J=1 to 1000
420     PRINT #1,"enter 14"
430     FOR I=1 to 16
440         LINE INPUT #2,A$(I,J)
450     NEXT I
460 NEXT J
```

4.12.7 High Speed Direct Memory Access

High Speed Direct Memory Access (DMA) can be used to move data directly to memory. Before DMA can be used, all variables must be established and their locations in memory known. Uncompensated binary format, with either high or low byte first, must be used. BASIC stores bytes in memory that way so they should be put in memory the same way.

The program SAMPLE2.BAS, included on the demo disk shipped with the ADC488, shows how to use high speed DMA. This example uses QuickBASIC because of QuickBASIC's VARSEG and VARPTR functions. They are used to point to the segment and offset address of the array.

First, dimension an integer array (a 32K integer array equals 64kB):

```
DIM  S%(32766)
```

Configure eight differential inputs, two channels (one and two), both at ±1 volt range, 20µs scan interval (50 kHz), uncompensated binary (low byte first), and trigger on GET:

```
PRINT#1,"output14;a2  c1,2  r0,0  i1  g11  t1  x"
```

Send a GET:

```
PRINT#1,"trigger14"
```

Tell the Driver488 to DMA 32766 words (64kB) to array S% (all one line):

```
    PRINT#1, "enter14#65532buffer";VARSEG(S%(0));":";VARTR
(S%(0));
```

The readings are now in S%:

```
FOR  I=0  to  32766  STEP  2
PRINT  S%(I),S%(I+1)
NEXT  I
```

Command Descriptions

5.1 Introduction

Control of the ADC488 is implemented with 26 commands, described here in detail. Examples are given for the commands using a PC functioning as an IEEE 488 controller by using the IOtech Personal488 PC/IEEE 488 board and associated driver software. All examples are given using GW-BASIC, QuickBASIC, or BASICA. The ADC488 IEEE 488 bus address should be set to 14 for all examples.

In order to establish communication with Driver488 from GW-BASIC, QuickBASIC, or BASICA, the following sequence must be used:

```
OPEN  "\DEV\IEEEOUT"  FOR  OUTPUT  AS  #1
IOCTL#1,"BREAK"
PRINT#1,"RESET"
OPEN  "\DEV\IEEEIN"  FOR  INPUT  AS  #2
```

All of the command examples that follow assume that the driver has been properly opened and reset by completing the above sequence. Most of the instructions offer a query (?) option. This option can be used to determine the present configuration or mode of a command previously executed. Any number of these query commands may be combined into one string to allow the user to construct a specialized status command requesting the ADC488 to return only that information which is of interest for a given application.

Attention: It is necessary that the EXECUTE command (X) follow all command strings sent to the ADC488. No commands are executed until an X is received by the ADC488. The only exceptions to this rule are the query command options (?). These query options do not have to be followed by an X.

5.2 ADC488 Command Set Index

This index is categorized by general function of the ADC488, and is intended to aid you in finding a specific command description while executing the functions.

Acquisition Parameter Commands		Page
C	Scan Group	5.9
I	Scan Interval	5.21
N	Scan Count	5.30
R	Range	5.36
Buffer Operation Commands		
B	Scan Buffer Pointer	5.7
F	Buffer Pointer Mode	5.14
P	Buffer Select	5.33
Digital I/O Command		
D	Digital Output	5.11
IEEE 488 Bus Parameter Commands		
J	User Terminator	5.23
K	Command Channel EOI	5.24
O	End of Reading Terminator	5.32
Q	End of Scan Terminator	5.35
Y	Command Channel Terminator	5.46
Trigger Parameter Commands		
L	Trigger Level	5.25
T	Trigger	5.39
Z	Trigger Delay	5.47
Miscellaneous Commands		
A	Input Configuration	5.5
E	Error Query	5.12

G	Output Format	5.15
H	Calibration	5.19
M	SRQ Mask	5.26
S	Setup (Save, Recall)	5.37
U	Status	5.41
V	Version Query	5.43
W	Test	5.44
X	Execute	5.45
?	Query	5.48

Input Configuration An

The Input Configuration command is used to set the analog input connections to be used as differential input channels or single-ended input channels. The An command is also used to select Master/Slave operation. As a master, the ADC488 can control the timing and synchronization of four other ADC488 A/D converters. As a slave, it can be externally controlled and synchronized. Refer to Section 2.7 for further information on ADC488 Master/Slave operation. The available command options for this command are:

A0	Set input configuration to eight differential input channels - Master.
A1	Set input configuration to eight differential input channels - Slave.
A2	Set input configuration to sixteen single-ended input channels - Master.
A3	Set input configuration to sixteen single-ended Input channels - Slave.
A?	Returns the present input configuration setting.

Sixteen analog input connections are provided on the ADC488/16A. These 16 input connections can be arranged to work in one of two configurations: single-ended or differential. Once defined, the input configuration is universal to all input connections, i.e. all input channels are single-ended or all input channels are differential. The ADC488/8SA is configured in hardware for eight differential input channels, only. Issuing an An command to configure it for 16 channels generates an Invalid Device Dependent Command Option (E2) error. In the single-ended input configuration, any of the 16 analog input connections may be measured with respect to analog ground. In this configuration, there are sixteen analog input channels. In the differential input configuration, the 16 input connections are paired, and the measured value is the difference between one analog input connection and another. In this configuration, there are eight analog input channels. Changing the analog input configuration is a major change in the operation of the ADC488. For that reason, issuing this command forces changes in other programmable parameters. These changes include:

1. Scan Group defaults to a single channel (channel 1) (C1).
2. The Scan Interval is set to 10µs (I0).

Example:

`PRINT#1,"CLEAR14"`	Reset the ADC488.
`PRINT#1,"OUTPUT14;A2 C2 R3,1 X"`	Select 16 single-ended inputs - Master, one channel (ch 2), and ±2 volt range.
`PRINT#1,"OUTPUT14;A? C? R?"`	Query configuration, channel and range.

`PRINT#1,"ENTER14"`	Address the ADC488 to talk.
`INPUT#2,A$`	Get the response from the ADC488.
`PRINT A$`	Display shows:

```
A2C02,00,00,00,00,00,00,00,00,00,00,00,00,00,00R3,1,3,3,3,3,3,3
,3,3,3,3,3,3,3,3
```

`PRINT#1,"OUTPUT14;A0 X"`	Select eight differential inputs - Master.
`PRINT#1,"OUTPUT14;A? C? R?"`	Query configuration, channel and range.
`PRINT#1,"ENTER14"`	Address the ADC488 to talk.
`INPUT#2,A$`	Get the response from the ADC488.
`PRINT A$`	Display shows:

```
A0C01,00,00,00,00,00,00,00,00,00,00,00,00,00,00R3,3,3,3,3,3,3,3
```

Scan Buffer Pointer Bn

The Scan Buffer Pointer command is used to set the scan buffer pointer to a group of readings in the scan buffer, all the readings in one scan. The scan buffer is divided into channel buffers, one channel buffer for each channel defined in a Scan Group (Cn,n,n,n) command. The scan buffer can be thought of as an array. Each reading in a scan is stored at the same scan buffer location (name) in a separate channel buffer (index). The Scan Buffer Pointer (Bn) command points to all the readings in one scan. Individual channel buffers are indexed with the Buffer Select (Pn) command.

B±n Set the scan buffer location pointer to n.
(n = ±2048 without memory options)
(n = ±131,072 with MEMX1 RAM Expansion Board)
(n = ±262,144 with MEMX2 RAM Expansion Board)
(n = ±1,048,576 with MEMX5 RAM Expansion Board)
(n = ±2,097,152 with MEMX6 RAM Expansion Board)
(n = ±4,194,304 with MEMX7 RAM Expansion Board)

B? Return the present scan buffer location number.

The number of channel buffers and the number of scan buffer locations in the scan buffer is dependent on the scan buffer memory size, the number of channels defined in the Scan Group command (Cn,n,n,n), and the number of pretrigger scans and

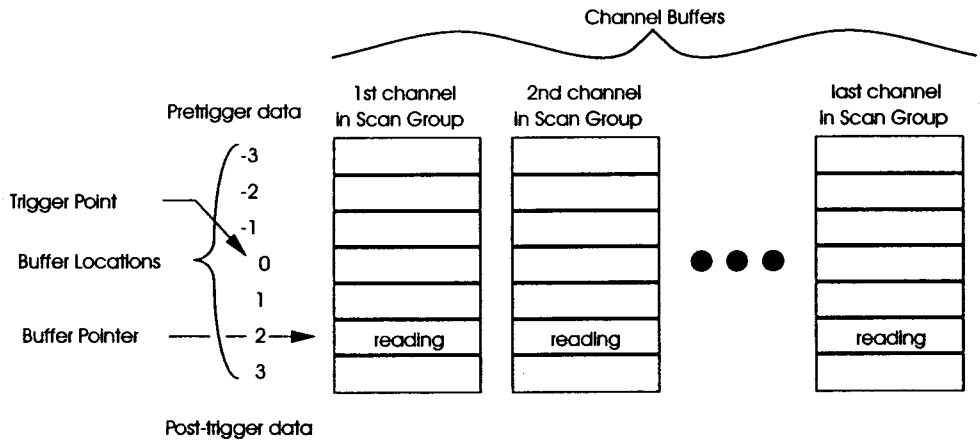


Figure 5-1: ADC488 Scan Buffer Organization

post-trigger scans specified in the Scan Count (Nn,m) command. The maximum number of buffer locations is equal to the size of the scan buffer divided by the number of channels.

The U2 Status command can be used to return the valid range of scan buffer locations once data has been acquired. After data acquisition is complete, trying to set the scan buffer pointer to a value that is out of range of pretrigger or post-trigger data causes a conflict error (E4). Immediately after a data acquisition sequence has been completed, the scan buffer pointer is set to 0. Scan buffer pointer location 0 is always the trigger point, or, in the case of one-shot trigger mode, the first trigger point. Pretrigger data, if available, is accessed with negative buffer locations. The oldest (least recent) pretrigger data has the most negative value. Post-trigger data, if available, is accessed with positive buffer locations. The newest (most recent) post-trigger data has the most positive value. In one-shot trigger mode, only positive values for buffer locations are used.

Example:

```
PRINT#1,"CLEAR14"                    Reset the ADC488.
PRINT#1,"OUTPUT14;C1,2X"             Select channel 1 and channel 2.
PRINT#1,"OUTPUT14;n100,0X"           Get at least 100 pretrigger readings.
PRINT#1,"OUTPUT14;T4X"               Trigger on rising analog level (Channel1).
INPUT#1,OUTPUT14;L+80X"              Set analog level to 80% of range.
....                                 Acquire data.
FOR  I  =  -100  to  0               Get pretrigger readings
    PRINT#1,"OUTPUT14;B";I;"X"       Point to scan buffer location I.
    PRINT#1,"OUTPUT14;P2X"           Point to channel buffer 2.
    PRINT#1,"ENTER14"                Address the ADC488 to TALK.
    INPUT#1,A$                       Put the reading into A$ array
NEXT  I
```

Scan Group Cn,n,n,n

The Scan Group command is used to define the analog input channels to be scanned (when triggered), the order that they are scanned, and the number and order of the channel buffers that store the sampled data in the scan buffer. Channels can be scanned in any order and can be scanned more than once in a scan group. Each channel that is included in a scan group is mapped to an individual channel buffer for that channel. Up to sixteen channel buffers can be defined. The Scan Group command is also used during calibration to specify an analog input channel for calibration. If more than one channel is specified before calibration, only the first channel in the scan group is calibrated.

Cn	One channel defined.
Cn,n	Two channels defined.
Cn,n,n,n	Four channels defined.
Cn,n,n,n,n,n,n,n	Eight channels defined.
Cn,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n	Sixteen channels defined.
C?	Returns the channels that are defined.

Where: **n can be 1 through 8 for differential mode (Model ADC488/16A and Model ADC488/8SA) and 1 through 16 for single-ended mode (Model ADC488/16A only). When using the Scan Group command, some restrictions apply:**

1. Scan groups must be defined in powers of 2. This allows only scan group sizes of 1, 2, 4, 8 and 16.
2. The order the channels are scanned is determined by the order they are set within the Scan Group (Cn,n,n,n) command.
3. The maximum size of each channel buffer is determined by the number of scan buffer locations divided by the number of channels.
4. The first channel specified in a scan group is the channel that is used for the Analog Trigger Level (Lval).

Multiple channels are setup, scanned, and converted sequentially as a group. Since the A/D converter can convert once every 10 microseconds, the time required to do a complete scan is equal to the number of channels in a scan group times 10 microseconds. The maximum sampling frequency is limited by the number of channels to be scanned. One channel may be scanned at up to 100 kHz, two channels may be scanned at up to 50 kHz, four channels at up to 20 kHz, eight channels at up to 10 kHZ, and 16 channels at up to 5 kHZ (ADC488/16A only). Refer to the Scan Interval (In) command to set the time between scans.

Examples:

`PRINT#1,"OUTPUT14;C1,2,3,4X"`	Define four channels to be scanned.
`PRINT#1,"OUTPUT14;C?"`	Query the current scan group.
`PRINT#1,"ENTER14"`	Address the ADC488 to Talk.
`LINE INPUT#2,A$`	Read data from the ADC488.
`PRINT A$`	Returns: C1,2,3,4,0,0,0,0,0,0,0,0,0,0,0,0

Digital Output Dn

The Digital Output command sets the digital output lines on the rear panel eight bit digital output connector to the binary equivalent of the value specified in the command option. This value can range from 000 to 255 (00000000 to 11111111 binary). The query option to this command returns the programmed value of the output lines and not the actual value sensed on the output lines. If a specific bit on the output lines is set to a logic '1', the output is set to a logic high (5 volts typical) level. The outputs are TTL compatible and are set to 000 at power-on or upon detection of an IEEE 488 bus device clear (DCL and SDC). Refer to the U1 Status command for information on reading the digital inputs.

Dn	Set the digital output lines to the binary equivalent of n. (n = 0 to 255).
D?	Returns the present value of the digital outputs.

Examples:

`PRINT#1,"CLEAR14"`	Reset the ADC488.
`PRINT#1,"OUTPUT14;D0X"`	Set the outputs to zero.
`PRINT#1,"OUTPUT14;D?"`	Query the present output.
`PRINT#1,"ENTER14"`	Get response for the ADC488.
`INPUT#1,A$`	
`PRINT A$`	Display shows D000.
`PRINT#1,"OUTPUT14;D127X"`	Set the outputs to 127 (&H7F).
`PRINT#1,"OUTPUT14;D?"`	Query the present output.
`PRINT#1,"ENTER14"`	Get response for the ADC488.
`INPUT#1,A$`	
`PRINT A$`	Display shows D127.

Error Query E?

The Error Query command is used to determine the present error condition of the ADC488. When an error has occurred, the Error indicator light on the front panel of the ADC488 turns on.

E? Return and clear present error condition.

After execution of the Error Query command, the ADC488 returns one, or the numeric sum of more than one, of the following errorout

E00 No Error has occurred.

E01 Invalid Device Dependent Command (@4X).

E02 Invalid Device Dependent Command Option (A7X).

E04 Conflict Error (Parameters inconsistent with mode selected).

E08 NVRAM Setup Error.

E16 Calibration Error.

E32 NVRAM Calibration Error.

After execution of the Error Query command, most error conditions are cleared. Error conditions are also cleared by executing the U0 Status command. The ERROR indicator light stays on until an Error Query (E?) or U0 Status command command is executed.

An Invalid Device Dependent Command (IDDC) error, E01, indicates that an invalid command was issued to the ADC488, such as: 4X. Correct the command string and try again.

An Invalid Device Dependent Command Option (IDDCO) error, E02, indicates that an invalid option was issued for a particular command such as: A27Xd. Correct the command string and try again.

A Conflict Error, E04, indicates that an invalid combination of commands was issued to the ADC488, such as C5,6I0X. Correct the command string, and try again.

An NVRAM setup error, E8, indicates a problem with the Non-Volatile Random Access Memory (NVRAM) used for saving and recalling user configurations. Issue a Setup (Save) (Sn,1) command to clear the error.

A Calibration Error, E16, indicates a problem with the calibration procedure (check the rear panel CAL ENABLE switch). In case a calibration error occurs, an Error Query (E?) or U0 Status command can be used to clear the error condition.

An NVRAM Calibration Error, E32, indicates a problem with the NVRAM used for calibration. Recalibrate the ADC488 if this error occurs.

More than one error can occur at the same time. An error code such as E40 means two error conditions have occurred: E08 and E32.

Examples:

`PRINT#1,"CLEAR14"`	Reset the ADC488.
`PRINT#1,"OUTPUT14;E?"`	Request error status.
`PRINT#1,"ENTER14"`	
`INPUT#2,A$`	
`PRINT A$`	Display shows E00, no errors have occurred.
`PRINT#1,"OUTPUT14;K3 X"`	Send an illegal command option to the ADC488. (Note: ERROR indicator light turns on).
`PRINT#1,"OUTPUT14;E?"`	Request error status (ERROR light turns off).
`PRINT#1,"ENTER14"`	
`INPUT#2,A$`	
`PRINT A$`	Display shows E02 (Invalid Device Dependent Command Option).

Buffer Pointer Mode Fn

The Buffer Pointer Mode command is an obsolete command, that was used to assign one of three actions to the scan buffer pointer after reading data in the scan buffer. Formerly, it could be programmed to auto-increment, auto-decrement or remain unchanged, as shown below.

F0 Buffer Pointer Auto-Increments (default).

F? Returns the value of the present buffer pointer mode.

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;n100t1x"`	One channel, 100 scans, trigger on GET.
`PRINT#1,"trigger14"`	Issue a GET.
`PRINT#1,"output14;q1b0f0g0x"`	End of Scan = CRLF, Buffer pointer=0. Autoincrement pointer, ASCII format.
`PRINT#1,"enter14"`	Ask for the first reading.
`INPUT#2,A$`	Get the reading.
`PRINT A$`	Print the first reading: +4.93600
`PRINT#1, "enter14"`	Ask for the second reading
`INPUT#2,A$`	Get the reading
`PRINT A$`	Print the second reading: +4.93320

Output Format Gn

The Output Format command determines the format of the data returned to the IEEE 488 bus. Six basic data formats are available, each with or without compensation for gain and offset, for a total of twelve formats. The End of Reading Terminator (On) and the End of Scan Terminator (Qn) commands may be used to change the terminator and EOI selections for ASCII, integer, and hexadecimal formats. All binary formats assert EOI on the last reading from the scan buffer.

Each of the formats allow for compensated and uncompensated readings. For high speed data transfers (>200 kB/s), the uncompensated binary modes (G9 or G11) must be used.

G0	Compensated ASCII Fixed Point with leading + sign (default).
G1	Uncompensated ASCII Fixed Point with leading + sign.
G2	Compensated ASCII Fixed Point with + sign implied.
G3	Uncompensated ASCII Fixed Point with + sign implied.
G4	Compensated Integer Decimal with + sign implied.
G5	Uncompensated Integer Decimal with + sign implied.
G6	Compensated Hexadecimal.
G7	Uncompensated Hexadecimal.
G8	Compensated two's complement integer binary -high byte first.
G9	Uncompensated two's complement integer binary - high byte first.
G10	Compensated two's complement integer binary -low byte first.
G11	Uncompensated two's complement integer binary - low byte first.
G?	Responds with present Output Format selection.

The six output formats are:

G0 G1 ASCII Fixed point format returns an ASCII decimal numeric representation of the voltage. This output provides both leading positive and negative signs. The number of characters transferred per reading is kept fixed by padding with spaces. If this format is selected, data returned from the ADC488 is output using the End of Reading Terminator (On), the User Terminator (Jn) and the End of Scan Terminator (Qn) commands.

Typical Output	Range	Typical Resolution
±10.50000	10 volt range	0.00033V
±05.25000	5 volt range	0.00017V
±02.10000	2 volt range	0.00006V

±01.05000	1 volt range	0.00003V

G2 G3 An alternative ASCII fixed point format returns an ASCII decimal numeric representation of the voltage. This output provides data with the positive sign implied and the negative sign expressed. The number of characters transferred per reading is kept fixed by padding with spaces. If this format is selected, data returned from the ADC488 is output using the End of Reading Terminator (On), the User Terminator (Jn) and the End of Scan Terminator (Qn) commands.

Typical Output	Range	Typical Resolution
-10.50000	10 volt range	0.00033V
-05.25000	5 volt range	0.00017V
02.10000	2 volt range	0.00006V
01.05000	1 volt range	0.00003V

G4 G5 A decimal format returns a signed integer decimal representation of each reading. If this format is selected, data returned from the ADC488 is output using the End of Reading Terminator (On), the User Terminator (Jn) and the End of Scan Terminator (Qn) commands.

Decimal Value	Voltage equivalent (10V Range)
32767	+Overflow indication
30000	+10.00000V
15000	+ 5.00000V
00001	+ 0.00033V
00000	+ 0.00000V
-00001	-0.00033V
-15000	-5.00000V
-30000	-10.00000V
-32768	–Overflow indication

G6 G7 Hexadecimal format returns a four character representation of the two's-complement integer binary value of each reading. If this format is selected, data returned from the ADC488 is output using the End of Reading Terminator (On), the User Terminator (Jn) and the End of Scan Terminator (Qn) commands.

HEX Value	Voltage equivalent (10V Range)
7FFF	+ Overflow indication
7B0C	+10.50000V

3A98	+ 5.00000V
0001	+ 0.00033V
HEX Value	**Voltage equivalent (10V Range)**
0000	+ 0.00000V
FFFF	-0.00033V
C568	-5.00000V
84F4	-10.50000V
8000	– Overflow indication

G8 G9 Binary, with high byte first, returns two bytes of two's-complement binary data for each converted value. The End of Reading Delimiter (On), the User Terminator (Jn) or the End of Scan Terminator (Qn) commands have no effect on this output format. EOI is used to signal the last buffer reading of the scan or of the selected data channel.

High Byte	Low Byte	Voltage equivalent (10V Range)
0111 1111	1111 1111	+ Overflow indication
0111 1011	0000 1100	+10.50000V
0011 1010	1001 1000	+5.00000V
0000 0000	0000 0001	+0.00033V
0000 0000	0000 0000	+0.00000V
1111 1111	1111 1111	-0.00033V
1100 0101	0110 1000	-5.00000V
1000 0100	1111 0100	-10.00000V
1000 0000	0000 0000	– Overflow indication

G10 G11 Binary, with low byte first, returns two bytes of two's-complement binary data for each converted value. The End of Reading Delimiter (On), the User Terminator (Jn) or the End of Scan Terminator (Qn) commands have no effect on this output format. EOI is used to signal the last buffer reading of the scan or of the selected data channel.

Low Byte	High Byte	Voltage equivalent (10V Range)
1111 1111	0111 1111	+ Overflow indication
0000 1100	0111 1011	+10.50000V
1001 1000	0011 1010	+5.00000V

0000 0001	0000 0000	+0.00033V
0000 0000	0000 0000	+0.00000V
1111 1111	1111 1111	-0.00033V
Low Byte	**High Byte**	**Voltage equivalent (10V Range)**
0110 1000	1100 0101	-5.00000V
1111 0100	1000 0100	-10.00000V
0000 0000	1000 0000	– Overflow indication

Example:

```
DIM  s%(1000)                          Dimension an integer array.
PRINT#1,"clear14"                      Reset the ADC488.
PRINT#1,"output14;g11n1000t7x"         Use binary for 1000 scans.
PRINT#1,"trigger14"                    Trigger.
SPOLL:                                 Loop.
PRINT#1,"spoll14"                      Serial poll.
INPUT#2,spoll                          Get the spoll value.
IF  (spoll  AND  &H80)  <>  &H80  THEN  GOTO  SPOLL
                                       Wait until done.
                                       Buffer the data into the s% array:
PRINT#1,"enter14,#2000buffer";VARSEG(s%(0));":";VARPTR(s%(0))
PRINT#1,"output14;u4x"                 Ask for the calibration constants.
INPUT#2,UO$                            Get the first offset value (channel 1).
INPUT#2,UC$                            Get the first constant value.
LINE  INPUT#2,U$                       Skip the rest of the values.
OFFSET=VAL(UO$)                        Convert the offset into a number.
GAIN=VAL(UC$)/32768                    Convert the gain.
FOR  I=0  to  999                      Loop.
s%(I)=INT((s%(I)-OFFSET)*GAIN)         Calculate the corrected value.
NEXT  I                                End of loop
```

Calibration Hn

The Calibration command is used to calibrate the analog input channels of the ADC488. It is used to specify the value of the input voltage it is measuring so that it can provide the necessary compensation to the measured voltage in normal operation. This command acts on a selected channel (Cn) and Range (Rn or R#n). Although the ADC488/16A may be calibrated entirely on the channel, the ADC488/8SA requires calibration on every channel. The ADC488 must be recalibrated on a twelve month basis so it can provide the necessary compensation to the measured voltage. The calibration procedure requires the following steps:

1. Prior to calibration, the rear panel CAL switch must be enabled (depressed). This is accomplished by using a small screw driver to depress the switch through the hole in the rear panel of the ADC488.
2. Set the ADC to the differential input configuration (A0 or A1), channel #1 (C1), and the 1 volt range (R#1,0).
3. Apply 0.0000 volts dc to Channel 1 input.
4. Issue the H0 calibration command and wait one second.
5. Issue the H1 calibration command and wait one second.
6. Repeat the above process for each range (both ADC488/16A and ADC488/8SA) and each channel (ADC488/8SA only).
7. H0 Measure 0 volts on selected channel and range.
8. H1 Measure full scale voltage on selected channel and range.
9. H2 Calibrate selected channel and range.
10. H? Returns last Calibrate command issued.
11. Finish this channel and range by issuing the H2 calibration command. This causes the ADC488 to transfer the calibration constants into its protected calibration memory. If the CAL ENABLE switch located on the rear panel is not enabled, the ADC488 generates a calibration error.

Example:

Calibrate channel one for 1 volt:

```
PRINT#1,"clear14"                    Reset the ADC488.
PRINT#1,"output14;a0x"               Configure eight differential inputs.
PRINT#1,"output14;c1x"               Select channel one.
PRINT#1,"output14;r0x"               Select 1 volt range.
INPUT "Apply 0 volts to channel one. Press <Enter>";I
PRINT#1,"output14;h0x"               Set calibration for 0 volts.
```

```
PRINT#1,"output14;e?"                    Query error status.
PRINT#1,"enter14"                        Get error status from ADC488.
INPUT#2,ER$                              Get error status from the bus.
IF ER$<>"E00" THEN PRINT "ERROR!":END
INPUT "Apply 1.00000 volts to channel one. Press
<Enter>";I
PRINT#1,"output14;h1x"                   Set calibration for full scale.
PRINT#1,"output14;e?"                    Query error status.
PRINT#1,"enter14"                        Get error status from ADC488.
INPUT#2,ER$                              Get error status from the bus.
IF ER$<>"E00" THEN PRINT "ERROR!":END
INPUT "Wait one second. Press <Enter>";I
PRINT#1,"output14;h2x"                   Save calibration in NVRAM.
```

Scan Interval In

The Scan Interval command is used to control the amount of time between scans. The scan interval specifies the time between the start of one scan to the start of the next successive scan. The scan interval has a minimum resolution of one conversion time (10µs). Options for the scan interval follow a 1, 2, 5, 10 sequence from a minimum of 10µs up to a maximum of 50 seconds.

Option	Scan Interval	Scan Frequency	Max Channels
I0	10µs	100 kHz	1 channel
I1	20µs	50 kHz	2 channels
I2	50µs	20 kHz	4 channels
I3	100µs	10 kHz	8 channels
I4	200µs	5 kHz	16 channels
I5	500µs	2 kHz	16 channels
I6	1 ms	1 kHz	16 channels
I7	2 ms	500 Hz	16 channels
I8	5 ms	200 Hz	16 channels
I9	10 ms	100 Hz	16 channels
I10	20 ms	50 Hz	16 channels
I11	50 ms	20 Hz	16 channels
I12	100 ms	10 Hz	16 channels
I13	200 ms	5 Hz	16 channels
I14	500 ms	2 Hz	16 channels
I15	1 second	1 Hz	16 channels
I16	2 seconds	500 mHz	16 channels
I17	5 seconds	200 mHz	16 channels
I18	10 seconds	100 mHz	16 channels
I19	20 seconds	50 mHz	16 channels
I20	50 seconds	20 mHz	16 channels
I?	Query the present scan interval.		

The number of available options for this command depends on the number of channels defined with the Scan Group (Cn,n,n,n) command. Issuing a Scan Group command and specifying a scan interval which requires less time than the product of the

number of channels times ten microseconds generates a conflict error (E4) and forces the scan interval to the fastest interval possible with that number of channels. For example, issuing C1,2,3,4 (four channels) and I1 (20µs) generates an E04 error and sets the scan interval to 50µs.

Examples:

`PRINT#1,"TIME OUT 2"`	Set the controllers timeout to two seconds.
`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;i15 t0x"`	Trigger on TALK, one reading a second.
`reading:`	
`PRINT#1,"enter14"`	Demand a reading.
`INPUT#2,R$`	Get the reading.
`PRINT R$`	Print the reading.
`GOTO reading`	Do it again.

User Terminator Jn

The User Terminator command is used to set the User Terminator value to any character whose numeric value (ASCII value) is in the range 0 to 255. The User Terminator can be used in place of the End of Reading Terminator (On) and End of Scan Terminator (Qn) to place a specific character on the IEEE 488 bus after each reading or scan.

J0	Set User Terminator to 'NUL' (default).
Jn	Set User Terminator to n. (n = 0 to 255)
J?	Query the present value of the user terminator.

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;c1,2,3,4x"`	Use four channels.
`PRINT#1,"output14;t6n1x"`	Trigger one-shot on TALK. One scan.
`PRINT#1,"output14;j32o9x"`	Set End of Reading terminator to space.
`PRINT#1,"enter14"`	Address the ADC488 to TALK.
`INPUT#2,DA$`	Get the reading string.
`PRINT DA$`	Print the reading string:

```
+09.89088  +08.89743  +07.54309  +06.63319
```

Command Channel EOI Kn

The Command Channel EOI command is used to enable or disable the End Or Identify (EOI) line on the last character of a query (?) or status (Un) response. This command enables the ADC488 to assert or not assert the EOI line on the last byte of the response. The EOI line is one of five interface management lines on the IEEE 488 bus. It is usually used by a talker to indicate the end of a multiple byte transfer sequence.

K0 EOI enabled (default)
Assert EOI on the last character of a command channel response.

K1 EOI disabled.
Do not assert EOI on the last character of a command channel response.

K? Return present Command Channel EOI selection.

Examples:

```
PRINT#1,"CLEAR14"          Reset the ADC488.
PRINT#1,"OUTPUT14;K1X"     do not assert EOI.
PRINT#1,"OUTPUT14;K?"      query EOI selection from ADC488.
PRINT#1,"ENTER14"
INPUT#2,A$
PRINT  A$                  display shows K1.
```

Trigger Level Lval

The Trigger Level command is used to set the signed percentage (%) value of the analog trigger level. This trigger level value is assigned to the first defined channel within a scan group and is range independent. For example, if the level is set to +80% and the first channel in the scan group is on the ±5 volt range, the analog trigger level is 80% of 5 volts or +4.0000 volts. If the range is changed to the ±10 volt range, this trigger level value changes to +8.0000 volts.

L±n	Set Analog Trigger Level to ±n% of range.
L?	Query the present Analog Trigger value.

The actual trigger level is derived from the upper eight-most significant bits of the uncompensated A/D converter output. This provides a trigger level resolution of 1 part in 128, plus sign. The following table illustrates the trigger sensitivity for each range. The accuracy of the uncompensated A/D conversions are typically better than ±1% of Range.

Range	Trigger Resolution
1	0.008 V
2	0.016 V
5	0.039 V
10	0.078 V

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;c8 r2 175 t4x"`	Trigger on rising analog level of channel 8 at 75% of 5 volts (3.75 V).

SRQ Mask Mn

The SRQ Mask command uses the IEEE 488 Service Request (SRQ) mechanism to inform the IEEE 488 bus controller of the existence of several conditions. These conditions are described below. Multiple conditions can be enabled simultaneously. If multiple conditions are contained within the same command string, each SRQ Mask command should be proceeded by an Execute (X) command. The resulting SRQ Mask register value is the logical ORed value of the individual values sent. Alternately, the entire value of all the desired conditions may be sent within one SRQ Mask command. The programmed SRQ Mask remains enabled until the receipt of a M0 command or the detection of a Device Clear (DCL) or Selected Device Clear (SDC).

M0	SRQ's are disabled, the mask is cleared.
M1	SRQ On Trigger.
M2	SRQ On Trigger Overrun.
M4	SRQ On Buffer Overrun.
M8	SRQ On Bus Error.
M16	SRQ On Trigger Armed
M32	SRQ On Ready.
M128	SRQ on Acquisition Complete.

M0 This is the power-on default mask value. It disables the ADC488 from generating service requests by clearing the entire mask to zero.

M1 Triggered - Sending this command allows the ADC488 to generate a service request when it has detected a valid trigger from the programmed trigger source.

M2 Trigger Overrun - When the ADC488 trigger is selected for one-shot mode, sending this command forces a service request if the selected trigger source tries to issue another trigger while one is being processed. It is an indication that the triggers are being issued too quickly and that the resulting collected data is suspect.

M4 Buffer Overrun - This command allows the ADC488 to request service when the Scan Interval (In) is short and the data transferred from the ADC488 over the IEEE 488 bus does not keep up with the rate at which the data is collected. It is also used in the one-shot trigger mode to indicate that the last triggered data overwrote the first data in the buffer.

M8 Error - This will cause the ADC488 to request service when it detects an Invalid Device Dependent Command Error (IDDC), an Invalid Device Dependent Command Option Error (IDDCO), or a Conflict error.

M16 Trigger Armed - This command causes a service request to be generated whenever theADC488 is activately chacking for a trigger. The Trigger (Tn) and Execute (X) commands must have been performed and if a pretrigger count was specified in the Scan Count (Nn,m), then at least that number of pretrigger scans must have been acquired. When the trigger condition is met, this service request is cleared.

M32 Ready - This command causes a service request to be generated when the ADC488 has completed executing a set of commands from the IEEE 488 bus controller. This is used to allow the bus controller to attend to other bus matters while the ADC488 is changing its internal state.

M128 Acquisition Complete - Using this command will force the ADC488 to issue a service request when data acquisition is complete and all specified scans are stored into the scan buffer.

Serial Poll Response

There is a one-to-one relationship with the Serial Poll Response byte and the SRQ Mask. The Figure below illustrates this relationship.

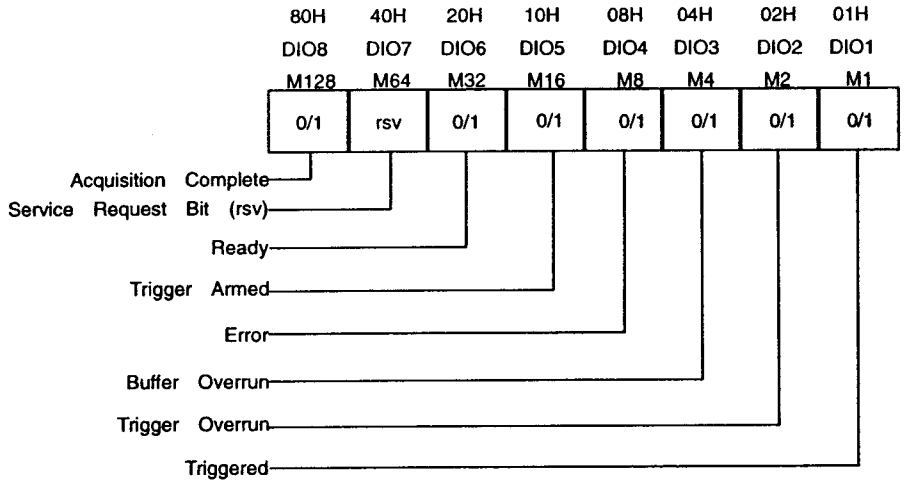


Figure 5-2: Serial Poll Response Byte & SRQ Mask

Serial Poll Bit Descriptions

DIO1 Triggered - This bit is set when the ADC488 detects a valid trigger. In the one-shot trigger mode, it is cleared when all the channels in the scan group have been converted. Although the time this bit is set is very small (10 to 160 s), a service request would be generated if the mask bit is set and the

indication that the ADC488 received a valid trigger would remain until the SRQ was acknowledged by the controller. When the trigger mode is set to continuous, this bit is not cleared until the trigger is rearmed by issuing another trigger (Tn) command.

DIO2 Trigger Overrun - This bit is set when a trigger overrun condition is detected. It is cleared upon reissuing the Trigger (Tn) command. When the ADC488 trigger mode is selected for one-shot, this bit indicates the selected trigger source tried to issue another trigger while one was being processed. It is an indication that the triggers are being issued too quickly and that the resulting collected data is suspect.

DIO3 Buffer Overrun - This bit is set if the data transferred from the ADC488 over the IEEE 488 bus does not keep up with the rate in which the data is collected. It is also used in the one-shot mode to indicate that the last triggered data overwrote the first data in the buffer. It is cleared when the Trigger (Tn) command is reissued.

DIO4 Error - This bit is set when the ADC488 detects an Invalid Device Dependent Command Error (IDDC), an Invalid Device Dependent Command Option Error (IDDCO) or a Conflict error (it cannot do what you requested). It is cleared when the error status is read via the E? or the U0 status commands.

DIO5 Trigger Armed - This bit is set when the ADC488 is ready and able to be triggered. For pre-trigger acquisitions, this bit is not set until after the specified numbers of pre-trigger scans have been acquired. For all other acquisitions, this bit is set when a trigger (Tn) command is issued followed by an Execute (X) command. It is cleared when a trigger is detected.

DIO6 Ready - This bit is set when the command processor of the ADC488 is in an idle state. It is cleared when an Execute (X) command is received.

DIO7 This is the 'rsv' (service request bit) and is set if any of the previously described conditions exist AND the corresponding SRQ Mask bit is programmed to a '1'. This bit indicates to the controller that the ADC488 is the IEEE 488 bus device requesting service. It is cleared upon being serial polled by the controller.

DIO8 Data Acquisition Complete - This bit is set when the last post-trigger scan has been acquired and stored in the scan buffer. It is cleared when a Trigger (Tn) command is issued.

Example:

```
100
. . .                                  Configure the ADC488 as you like it.
400
500  PRINT#1,"OUTPUT14;T1X"            Trigger on GET.
510  PRINT#1,"OUTPUT14;M80X"           Set 'rsv' when Trigger Armed (M16 AND
                                       M64).
```

`520 PRINT#1,"SPOLL 14"`	SPOLL the ADC488 until it is ready.
`530 INPUT#2,POLL$`	Get the Serial Poll Response byte into POLL$.
	Repeat SPOLL until ready:
`540 IF(VAL(POLL$) AND &H10) &H10 GOTO 520`	
`550 PRINT#1,"TRIGGER 14"`	The ADC488 is ready for a trigger, so issue a GET.

Scan Count Nn,m

The Scan Count command controls how many scans will be taken in response to a trigger. This value may range from zero to a maximum of the buffer size divided by the number of channels defined in the scan group for post-trigger scans and from zero to sixteen million (16,000,000) for pretrigger scans.

The Scan Count command accepts one or two parameters. If only one is included, it is considered to be the number of post-trigger (after the trigger) scans. If the pretrigger value is greater than sixteen million or the post-trigger value is greater than the number of scan buffer locations divided by the number of channels in the Scan Group (Cn,n,n,n) command, the ADC488 will generate a Invalid Device Dependent Command Option (IDDCO) error (E02).

If two parameters are included with this command, the first is the pretrigger (before the trigger) scan count value and the second is the post-trigger (after the trigger) scan count. If this sum of these two values is greater than the number of scan buffer locations divided by the number of channels defined in the Scan Group (Cn,n,n,n) command, the ADC488 will generate a Invalid Device Dependent Command Option (IDDCO) error (E02).

Nm Set pretrigger count to 0 and post-trigger count to 'm'.
(m ≤ locations/channels)

Nn,m Set pretrigger count to 'n' and post-trigger count to 'm'.
(m ≤ locations/channels)
(n ≤ 16,000,000)

N? Return the programmed pretrigger and post-trigger values.
locations = 2048 without Ram Expansion
locations = 131,072 for MEMX1 Ram Expansion Board
locations = 262,144 for MEMX2 Ram Expansion Board
locations = 1,048,576 for MEMX5 Ram Expansion Board
locations = 2,097,152 for MEMX6 Ram Expansion Board
locations = 4,194,304 for MEMX7 Ram Expansion Board

In one-shot trigger mode, no pretrigger scans are retained. Any pretrigger scans that were specified using the Nn,m command are ignored.

In continuous mode, a trigger causes the specified number of Scan Count (Nm) scans to be taken at the Scan Interval (In) and stored in the scan buffer. If pretrigger scans (Nn) are specified, at least n scans will be collected before the trigger is armed, guaranteeing a minimum number of pretrigger scans.

Setting the scan count to 0 (N0) results in an infinite number of samples being taken. If the IEEE 488 controller is capable of taking the samples from ADC488 as fast as they are being acquired, data collection may proceed indefinitely. If the controller is not capable of maintaining this rate, or data is not removed from the buffer at all, then data collection will proceed until the scan buffer is full, when a buffer overrun error will be generated. Data collection will not cease, but upon

detection of a buffer overrun error, the ADC488 will continue to output data from the newest data point available. As a result, buffer overruns may generate discontinuities in collected data.

Issuing a Scan Count Query (N?) returns the last values programmed for the pretrigger and post-trigger scans. It does not return the actual number of scans that were taken. Use the U2 Status command to determine the number of readings in the scan buffer after data acquisition is complete.

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;c1,2r3,3i1t6x"`	Set up two channels.
`PRINT#1,"output14;n512,512x"`	Put the trigger point in the middle of the scan buffer.

End of Reading Terminator — On

The End of Reading Terminator command is used to set the terminator that will be sent to the IEEE 488 bus after each reading. A reading is a single data point from either a selected channel or from every channel in a complete scan. When the ADC488 is addressed to Talk, it will append the End of Reading Terminator to each reading.

The End of Reading Terminator may also be set to any character whose numeric (ASCII) value is from 0 to 255 by using the User Terminator (Jn) command and then selecting the O8 or O9 options of the End of Reading Terminator command.

The choices for the End of Reading Terminator are listed below:

O0	CR LF EOI (default)
O1	CR LF
O2	LF CR EOI
O3	LF CR
O4	CR EOI
O5	CR
O6	LF EOI
O7	LF
O8	User Terminator (Jn) with EOI
O9	User Terminator (Jn)
O?	Return current end of reading delimiter selection

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;c1,2,3,4,5,6,7,8x"`	Set up eight channels.
`PRINT#1,"output14;o9j32x"`	Use a space for end-of-reading.
`PRINT#1;"output14;t6x"`	Trigger on TALK.
`PRINT#1;"enter14"`	Ask for the readings.
`INPUT#2,A$`	Get the readings.
`PRINT A$`	Eight readings separated by spaces.

Buffer Select Pn

The Buffer Select command is used to identify which channel buffer to use when outputting data from the scan buffer onto the IEEE 488 bus. The number of channel buffers is determined by the number of channels setup in the Scan Group (Cn,n,n,n) command. One channel buffer is created for each channel in a scan group. Channel buffers are numbered from left to right, corresponding to the channels in the Cn,n,n,n command. One channel can have more than one channel buffer if the same channel is included more than once in a scan group.

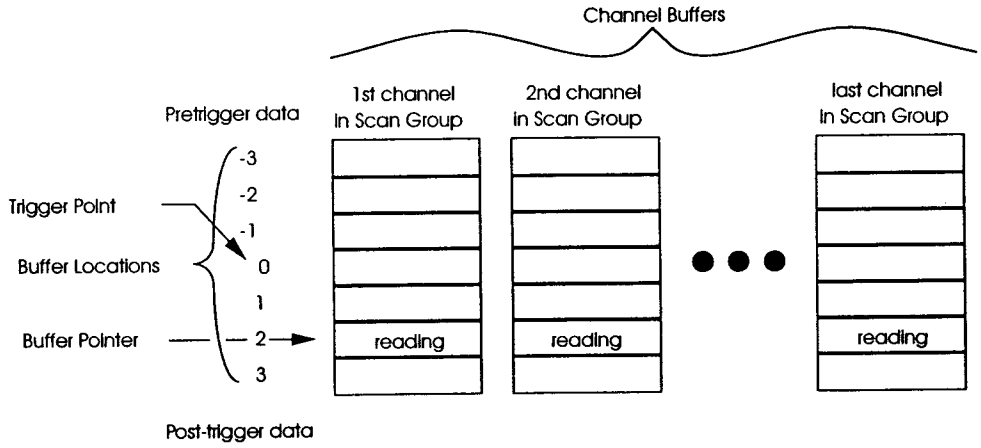


Figure 5-3: ADC488 Scan Buffer Organization

All of the data in the scan buffer may be read by accessing channel 0 (P0) of the ADC488. Or, the data for a specific channel may be read by accessing that particular channel buffer number (P1 through P8, or P1 through P16, depending on the model and input configuration).

Pn	Select channel buffer n. where n = 1 to 16 (the same number of channels used in the Cn,n,n,n command).
P0	Request scan buffer data from all channels in a scan group.
P?	Request presently selected channel buffer.

Examples:

```
PRINT#1,"clear14"                    Reset the ADC488.
                                     Use all sixteen channels (defaults to R3):
PRINT#1,"output14;a2  c1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16x"
PRINT#1,"output14;n1000t6x"          Collect 1000 scans when addressed to talk.
PRINT#1,"enter14"                    Address the ADC488 to Talk.
spoll:                               Wait until data acquisition is complete.
PRINT#1,  "spoll  14"                Serial poll ADC488.
INPUT#2,SPOLL$                       Get the Serial Poll Response.
                                     Check to see if data acquisition is complete:
IF  (VAL(SPOLL$)  AND  &H80)  <>  &H80  GOTO  spoll
FOR  I=  1  TO  16                   Loop through each channel buffer.
PRINT#1,"output14;p";I;"x"           Access the buffer.
FOR  J=1  to  1000                   Loop through each reading.
PRINT#1,"enter14"                    Get a reading.
INPUT#2,A$(I,J)                      Input a reading into array A$.
NEXT  J                              End of J loop.
NEXT  I                              End of I loop.
```

End of Scan Terminator Qn

The End of Scan Terminator command sets the terminator sent by the ADC488 at the end of a scan. When the ADC488 outputs data, it appends the End of Scan Terminator to the final reading in the scan buffer for each scan.
The End of Scan Terminator may be set to any character whose numeric value is from 0 to 255 by using the User Terminator (Jn) command and then selecting the Q8 or Q9 options of the End of Scan Terminator command. The choices for the End of Scan Terminator are listed below:

Q0	CR LF EOI (default)
Q1	CR LF
Q2	LF CR EOI
Q3	LF CR
Q4	CR EOI
Q5	CR
Q6	LF EOI
Q7	LF
Q8	User Terminator (Jn) with EOI
Q9	User Terminator (Jn)
Q?	Return current end of scan terminator selection.

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;c1,2,3,4x`	Set up four channels.
`PRINT#1,"output14;o9j44x"`	Use a comma (,) for end-of-reading.
`PRINT#1,"output14;q4x"`	Use CR EOI for end-of-scan.
`PRINT#1;"output14;t6x"`	Trigger on TALK.
`PRINT#1;"enter14 EOI"`	Ask for the readings.
`LINE INPUT#2,A$`	Get the readings.
`PRINT A$`	Four readings separated by commas.
`PRINT#1;"enter14 EOI"`	Ask for more readings.
`LINE INPUT#2,A$`	Get the readings.
`PRINT A$`	Four more readings on the same line.

Range Rn

The Range command is used to set the input voltage range for each input channel. Ranges of ±1, ±2, ±5, or ±10 volts may be specified. The order of the range values is left to right, from channel one to channel eight (or channel sixteen). This is not the same order as the order of the channels specified using the Scan Group (Cn,n,n,n) command. Ranges can be set for each channel, even if a channel is not used in a scan group. All channels can be set at once or a channel can be set individually using the R# command. If a range for a particular channel is not changed, it is set to the last value programmed for that channel (default value of R3 if never set). The Range command is also used during calibration with the Scan Group (Cn,n,n,n) and Calibration (Hn) commands to specify the input voltage range for the channel under calibration.

Rn,n,n,n,n,n,n,n	Eight channels
Rn,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n	Sixteen channels (ADC488/16A)

where:

n=0	±1 volt Range
n=1	±2 volt Range
n=2	±5 volt Range
n=3	±10 volt Range (default)

R#c,n	Set channel c to range n
R?	Returns the present range of all channels.

Example:

`PRINT#1,"CLEAR14"`	Reset the ADC488 to power on defaults.
`PRINT#1,"OUTPUT14;A0 C1,2 X"`	Define single-ended channels 1 and 2.
`PRINT#1,"OUTPUT14;R2,2 X"`	Select Range 2 (±5 volts) for both channels.
`PRINT#1,"OUTPUT14;R?"`	Query the present ranges.
`PRINT#1,"ENTER14"`	Address the ADC488 to TALK.
`INPUT#2,A$`	Read data from the ADC488.
`PRINT A$`	The display shows R2,2,3,3,3,3,3,3.

Setup (Save,Recall) Sn,m

The Setup command is used to save and recall ADC488 setup configurations. The ADC488 provides non-volatile RAM (NVRAM) storage for up to eight preset configurations. Two operations are available for configuration storage. Save configuration is used to save the present ADC488 settings to NVRAM memory. Recall configuration is used to set ADC488 to a previously stored condition.

When issued, a configuration recall command automatically ARMs the ADC488 trigger for the stored trigger setting and clear the scan buffer.

Sn,0	Recall configuration n.
Sn,1	Save present setup to configuration buffer n.
S?	Request the last Setup operation.

The data stored includes the following ADC488 programmable parameters:

An	Input Configuration
Bn	Scan Buffer Pointer
Cn,n,n,n	Scan Group Definition
Dn	Digital Output
Fn	Buffer Pointer Mode
Gn	Output Format
In	Scan Interval
Jn	User Terminator
Kn	Command Channel EOI
Lval	Trigger Level
Mn	SRQ Mask
Nn,m	Scan Count, pretrigger and post-trigger
On	End of Reading Terminator
Pn	Buffer Select
Qn	End of Scan Terminator
Rn	Range information for all channels
Tn	Trigger source and mode
Yn	Command Channel Terminators
Zn	Trigger Delay

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;c1,2 t4 15x"`	Two channels, trigger at 5% analog level.
`PRINT#1,"output14;s1,1x"`	Save the setup as configuration number 1.
`PRINT#1,"output14;s1,0x"`	Recall configuration 1, ready for trigger (TRIGGER indicator light flashes).

Trigger Tn

The Trigger command is used to setup the ADC488 for triggered operation. It is used to specify both the trigger mode and the source of the trigger. The mode defines how the ADC488 responds to successive triggers on a selected trigger source. Trigger mode may be set to one-shot mode or continuous mode. The available source of trigger events include TALK (IEEE 488 talker address state), GET (IEEE 488 Group Execute Trigger), an analog input level, and an external TTL level signal.

T0	Continuous on TALK (default)
T1	Continuous on GET.
T2	Continuous on rising edge external TTL.
T3	Continuous on falling edge external TTL.
T4	Continuous on rising edge analog level.
T5	Continuous on falling edge analog level.
T6	One-shot on TALK.
T7	One-shot on GET.
T8	One-shot on rising edge external TTL.
T9	One-shot on falling edge external TTL.
T?	Returns the present trigger mode and source information.

In one-shot mode, a trigger causes one scan to be collected and stored into the scan buffer. At the conclusion of the scan, the trigger is rearmed. Once the specified number of Scan Count (Nm) triggers have been accepted and stored in the scan buffer, triggers are disarmed. No pretrigger scans are retained. Any pretrigger scans that were specified using the Nn,m command are ignored.

In continuous mode, a trigger causes the specified number of Scan Count (Nm) scans to be taken at the Scan Interval (In) and stored in the scan buffer. If pretrigger scans (Nn) are specified, then at least n scans will be collected before the trigger is armed, guaranteeing a minimum number of pretrigger scans.

When TALK is selected as the trigger source, data collection is triggered every time the ADC488 is addressed to TALK, and the ADC488 is not being serial polled.

When GET is selected as the trigger source, data collection is triggered whenever the IEEE 488 Group Execute Trigger command is sensed on the IEEE 488 bus and the ADC488 is addressed to listen.

When External TTL is specified as a trigger source, a TTL level signal on the external trigger input BNC connector on the rear panel generates a trigger. The external trigger is edge sensitive and either rising or falling edge may be specified.

When the Analog level is selected as the trigger source, the converted data from the first channel specified in the Scan Group (Cn,n,n,n) command is monitored for the specified level (Lval) and direction (rising or falling) as it is transferred into the scan buffer. Once the level is detected with the proper slope, data collection will

be triggered. Due to the mechanism used for this trigger source, one-shot mode is not available with analog level triggering. Pretrigger scan information is always available.

The response to GET or External TTL trigger has a maximum latency of ten microseconds if trigger mode is selected for one-shot.

The response to GET, Analog Level or External TTL trigger has a maximum latency of the Scan Interval (In) if trigger mode is selected for continuous.

Example:

```
10  PRINT#1,"clear14"          Reset the ADC488.
20  PRINT#1,"output14;t7x"     Trigger on GET (one-shot).
30  PRINT#1,"trigger14"        Trigger the ADC488.
40  PRINT#1,"enter14"          Send a reading from the ADC488.
50  INPUT#2,"A$"               Get the reading from the bus.
60  PRINT  A$                  Print the reading on the screen.
70  goto  30                   Repeat.
```

Status Un

Status commands are used to return information about various conditions of the ADC488. The status of the ADC488 may be read at any time without interfering with normal operation. Any error conditions, except Calibration and Setup Errors, are cleared after the status string is read by the controller. Calibration errors are cleared when the ADC488 is calibrated. Setup errors are cleared when a setup save (Sn,1) command is issued.

Status strings are returned when the ADC488 is next addressed to Talk. The terminators used and the use of the End or Identify (EOI) line can be changed with the Command Channel Terminator (Yn) and Command Channel EOI (Kn) commands.

If the ADC488 is configured to trigger on TALK, requesting a status report will not cause the ADC488 to trigger.

U0 Send the ADC488 command status.

U1 Send the value on the digital input lines.

U2 Send the pretrigger and post-trigger address limits.

U3 Send the value of the internal scan counter.

U4 Send the calibration constants.

U5 Send the number of unread scans.

U? Always returns U0.

The U0 Status command causes the ADC488 to return a fixed-length status message string. The status message shows the current options in use for each command.

The format of the fixed string returned by the ADC488 after receiving a U0 Status command is as follows:

```
*.*A#B#######C##,##,##,##,##,##,##,##,##,##,##,##,##,
##,##
D###E##F#G##H#I##J###K#L###M###N#######,#######O#P##Q#
R#,#,#,#,#,#,#,#,#,#,#,#,#,#,#S#,#T#U#W#Y#Z######
```

where each # shows the current option in use for that command. The leading information *.* is the revision level (V?) of the ADC488 firmware.

The U1 Status command causes the ADC488 to return a decimal number from 000 to 255. This number is the decimal equivalent of the binary information on the digital input lines on the rear panel card edge digital I/O connector.

The U1 Status command returns a string in the form: ###, where ### is the decimal equivalent of the binary value of the eight digital input lines of the rear panel digital I/O connector. For example, 054 would mean 00110110, (from pin 8 [MSB] to pin 1 [LSB]) where a 0 means the line is low (ground), and a 1 means that the line is high (+5 V).

The U2 Status command causes the ADC488 to return the pretrigger and post-trigger limits from the last scan conversion. These are the limits of the scan buffer that contain valid data. U2 returns a string in the form: -#######,+#######

where the ###### indicates the limits of the pretrigger (minus value) and post-trigger (positive value) scans in the scan buffer. The first number is the scan buffer location containing the first recorded sample (oldest reading) and the second number is the scan buffer location containing the newest sample (most recent sample recorded). These values are the limits of the scan buffer pointer that can be specified with the Scan Buffer Pointer (Bn) command.

The U3 Status command causes the ADC488 to return the value of the internal scan counter. This is the next location in the scan buffer that contains unread data. The internal scan counter is used to ensure that data is valid and continuous when reading data before data acquisition is finished. U3 returns a string in the form: ±######

where ###### is the value of the internal scan counter that counts the number of scans in a data acquisition sequence. Refer to Section 4.9 for examples of using the U3 command.

The U4 Status command causes the ADC488 to return the calibration constants from the calibration NVRAM. These calibration constants are the gain and offset values that are calculated when the ADC488 is calibrated. They can be used to correct readings after they have been transfered at full speed (200kB/s per second). Refer to the example listed for the Output Format (Gn) command for an example of how to use calibration constants to correct reading data after readings have been output from the ADC488. Refer to Section 4.7 for more information on the correction algorithm and use of the calibration constants.

The U4 Status command returns a string of values for each channel where the values for each channel are in the form: #####,#####

The first number is the offset correction and the second is the gain correction value. The U5 Status command causes the ADC488 to return the number of acquired scans that are available for reading. This value ranges from zero to the maximum number of scans 9based on the current Scan Group (Cn,n,n,n) command) that can fit in the current memory module. U5 returns a string in the form: # to ######. Refer to section 4.9 for examples of using the U5 command.

Example:

`FOR I=0 to 5`	Loop through all status parameters.
`PRINT#1,"OUTPUT14;U";I;"X"`	Request the status.
`PRINT#1,"ENTER14"`	Address the ADC488 to TALK.
`LINE INPUT#2,S$`	Read data from the ADC488.
`PRINT S$`	Print the status string.
`NEXT I`	

Version Query V?

The Version Query command returns the firmware revision level of the ADC488. In response to this command, the ADC488 returns a number in the following format:

.

Where each * is a one digit number. The Execute (X) command is not needed.

Examples:

`PRINT#1,"CLEAR14"`	Reset the ADC488.
`PRINT#1,"OUTPUT14;V?"`	Request firmware revision level.
`PRINT#1,"ENTER14"`	Address the ADC488 to TALK.
`INPUT#2,A$`	Read data from the ADC488.
`PRINT A$`	Response is 1.0 (or higher).

Test Wn

The Test command allows the user to test the communication with the ADC488 command channel. Two options are included. One causes the TEST indicator light on the front panel to turn on, the other causes it to turn off.

W0	Turn TEST indicator light off.
W1	Turn TEST indicator light on.
W?	Return the on/off indication of the TEST indicator light.

Example:

`PRINT #1,"CLEAR14"`	Reset the ADC488.
`PRINT #1,"OUTPUT14;W1 X"`	Turn on the TEST indicator light.
`PRINT #1,"OUTPUT14;W?"`	Query the state of the indicator light.
`PRINT #1,"ENTER14"`	Address the ADC488 to TALK.
`INPUT #2,A$`	Read the query response.
`PRINT A$`	The display shows: W1
`PRINT #1,"OUTPUT14;W0 X"`	Turn off the TEST indicator light.

Execute X

The Execute command is used to execute ADC488 commands. Commands sent to the ADC488 will result in no action until it is instructed to execute these commands. This is done by sending an X, usually as the last character of a command string. Commands sent without an X are stored in an internal command buffer until an X is received. Any number of Execute commands may be inserted into the same command string.

Examples:

```
PRINT#1,"CLEAR14"
```
Reset the ADC488.

```
PRINT#1,"OUTPUT14;A2C4I12X"
```
Configure one channel, single-ended.

is the same as:

```
PRINT#1,"OUTPUT14;A2"
PRINT#1,"OUTPUT14;C4"
PRINT#1,"OUTPUT14;I12"
PRINT#1,"OUTPUT14;X"
```

is the same as:

```
PRINT#1,"OUTPUT14;A2XC4XI12X"
```

is the same as:

```
PRINT#1,"OUTPUT14;A2X"
PRINT#1,"OUTPUT14;C4X"
PRINT#1,"OUTPUT14;I12X"
```

Command Channel Terminator Yn

The Command Channel Terminator command is used to set the terminator that will be sent by the ADC488 at the end of a response from a Query (?) or Status (Un) command. When the ADC488 is addressed to Talk, it will append the Command Channel Terminator to the response.

The Command Channel EOI mode can be changed by using the Command Channel EOI (Kn) command.

The choices for the Command Channel Terminator are listed below:

Y0	CR LF (default)
Y1	LF CR
Y2	CR
Y3	LF
Y?	Return current Command Channel Terminator selection

Example:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;y3x"`	Change the terminator to LF only.

Trigger Delay Zn

The Trigger Delay command is used to delay storing the trigger point in the scan buffer. Although there may be a convenient trigger source available, the region of interest may actually occur some time later than the trigger point. The Trigger Delay is used to shift the "real" trigger point to some later time. This "effective" trigger point, which is treated as the "real" trigger point in the scan buffer, has the effect of extending the ADC488 scan buffer. Any data collected during the trigger delay is stored as pretrigger data.

Trigger Delay is specified as a number of scans and may be a value from one scan to 16,000,000 scans. Issuing a Trigger Delay command will clear the scan buffer and reset the scan buffer pointer to location 0.

Trigger delays are not used in one-shot trigger modes. Any trigger delay values are ignored when one-shot trigger modes are specified.

Examples:

`PRINT#1,"clear14"`	Reset the ADC488.
`PRINT#1,"output14;i6t2x"`	1 ms scan interval, trigger on external TTL.
`PRINT#1,"output14;z3600000x"`	Delay the trigger for one hour.

Query ?

Most ADC488 commands offer a Query (?) option, which can be used to determine the present configuration or mode of a command previously executed. To use this option, the first letter of the command is used followed by a question mark (?). Any number of these query commands may be combined into one string to allow the user to construct a specialized status command requesting the ADC488 to return only that information which is of interest for a given application. The Execute (X) command is not needed when using the Query option.

? Return the current configuration or mode of the command preceding the ?.

Examples:

```
PRINT#1,"OUTPUT14;U0X"          Request the status of the ADC488.
PRINT#1,"ENTER14"               Address the ADC488 to TALK.
INPUT#2,A$                      Read data from the ADC488.
PRINT  A$                       Display shows status message:
1.0A0.B+0000000C01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
D000E00F0G00H0I00J000K0L+000M00N0000000,0000000O0P01Q0
R3,3,3,3,3,3,3,3S0,0T0U0W0Y0Z00000000
```

Using the Query command to return the same status message:

```
PRINT#1,"OUTPUT14;V?A?B?C?D?E?F?G?H?I?J?K?L?M?N?O?P?Q?R?S?T?U?W?Y?Z?"
                                Request the status of the ADC488
PRINT#1,"ENTER14"               Address the ADC488 to TALK.
INPUT#2,A$                      Read data from the ADC488.
PRINT  A$                       Display shows status message:
1.0A0.B+0000000C01,00,00,00,00,00,00,00,00,00,00,00,00,00,00   ,00
D000E00F0G00H0I00J000K0L+000M00N0000000,0000000O0P01Q0
R3,3,3,3,3,3,3,3S0,0T0U0W0Y0Z00000000
```

Appendix A

A.1 Command Summary

Command	Code	Description
Input Configuration	A0	Set 8 Differential Input channels - Master
	A2	Set 16 Single-ended Input channels - Master
	A3	Set 16 Single-ended Input channels - Slave
	A?	Return present Input Configuration setting
Scan Buffer Pointer	B±n	Set the Buffer Location pointer to '±n'.
	B?	Return present Buffer Location number.
Scan Group	Cn	One channel defined
	Cn,n	Two channels defined
	Cn,n,n,n	Four channels defined
	Cn,n,n,n,n,n,n,n	Eight channels defined
	Cn,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n	Sixteen channels defined
	C?	Returns the defined channels
Digital Output	Dn	Set digital lines to binary equivalent of 'n'.
	D?	Returns value of the present digital outputs.

Error Query	E?	Return and clear present error condition
	E0	No Error has occurred
	E1	Invalid Device Dependent Command
	E2	Invalid Device Dependent Command Option
	E4	Conflict Error (Mode/parameter conflict)
	E8	NVRAM Setup Error
	E16	Calibration Error
	E32	NVRAM Calibration Error
Buffer Pointer Mode	F0	Buffer Pointer Auto-Increments (default)
	F?	Returns present buffer pointer mode.
Output Format	G0	Compensated ASCII Fixed Point with leading '+' sign (default)
	G1	Uncompensated ASCII Fixed Point with leading '+' sign
	G2	Compensated ASCII Fixed Point with '+' sign implied
	G3	Uncompensated ASCII Fixed Point with '+' sign implied
	G4	Compensated Integer Decimal with '+' sign implied
	G5	Uncompensated Integer Decimal with '+' sign implied
	G6	Compensated Hexadecimal
	G7	Uncompensated Hexadecimal
	G8	Compensated 2's Complement integer binary - high byte first
	G9	Uncompensated 2's Complement integer binary - high byte first
	G10	Compensated 2's Complement integer binary - low byte first

	G11	Uncompensated 2's Complement integer binary - low byte first
	G?	Return present Format selection
Calibration	H0	Measure 0 volts on selected channel and range
	H1	Measure full scale voltage on selected channel and range
	H2	Calibrate selected channel and range
	H?	Returns last Calibrate command issued
Scan Interval	I0	10 µsec (100 kHz frequency)
	I1	20 µsec (50 kHz frequency)
	I2	50 µsec (20 kHz frequency)
	I3	100 µsec (10 kHz frequency)
	I4	200 µsec (5 kHz frequency)
	I5	500 µsec (2 kHz frequency)
	I6	1 msec (1 kHz frequency)
	I7	2 msec (500 Hz frequency)
	I8	5 msec (200 Hz frequency)
	I9	10 msec (100 Hz frequency)
	I10	20 msec (50 Hz frequency)
	I11	50 msec (20 Hz frequency)
	I12	100 msec (10 Hz frequency)
	I13	200 msec (5 Hz frequency)
	I14	500 msec (2 Hz frequency)
	I15	1 sec (1 Hz frequency)
	I16	2 sec (500 mHz frequency)
	I17	5 sec (200 mHz frequency)
	I18	10 sec (100 mHz frequency)
	I19	20 sec (50 mHz frequency)
	I20	50 sec (20 mHz frequency)

	I?	Query the present Scan Interval
User Terminator	J0	Set User Terminator to 'NUL' (default)
	Jn	Set User Terminator to 'n'. (0 to 255)
	J?	Query present value of User Terminator
Command Channel EOI	K0	EOI enabled (default)
	K1	EOI disabled
	K?	Return Command Channel EOI selection
Trigger Level	L±n	Set Analog Trigger Level to +n% of range
	L?	Query the present Analog Trigger value.
SRQ Mask	M0	SRQ's are disabled
	M1	SRQ On Trigger
	M2	SRQ On Trigger Overrun
	M4	SRQ On Buffer Overrun
	M8	SRQ On Error
	M16	SRQ On Trigger Armed
	M32	SRQ On Ready
	M64	*rsv* Service Request Bit
	M128	SRQ on Acquisition Complete
Scan Count	Nm	Set pretrigger count to 0 and post-trigger count to 'm'. (m ≤ scan buffer size/#channels)
	Nn,m	Set pretrigger count to 'n' and post-trigger count to 'm'. (m scan buffer size/#channels)
	N?	Query the present Scan Count pretrigger and post-trigger values.

End of Reading Terminator	O0	CR LF EOI (default)
	O1	CR LF
	O2	LF CR EOI
	O3	LF CR
	O4	CR EOI
	O5	CR
	O6	LF EOI
	O7	LF
	O8	User Terminator (Jn) with EOI
	O9	User Terminator (Jn)
	O?	Return current reading delimiter selection
Buffer Select	Pn	Select buffer 'n'. (n = 1 to 16)
	P0	Select channel '0'. Requests scan data.
	P?	Request presently select channel.
End of Scan Terminator	Q0	CR LF EOI (default)
	Q1	CR LF
	Q2	LF CR EOI
	Q3	LF CR
	Q4	CR EOI
	Q5	CR
	Q6	LF EOI
	Q7	LF
	Q8	User Reading Delimiter(Jn) with EOI
	Q9	User Reading Delimiter(Jn)
	Q?	Return current End of Scan Terminator
Range	Rn,n,n,n,n,n,n,n	(Differential Inputs)

	Rn,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n (ADC488/16)	n = 0 for ±1 volt range n = 1 for +2 volt range n = 2 for ±5 volt range n = 3 for ±10 volt range (default)
	R#c,n	Set channel c to range n
	R?	Returns the present ranges of all channels.
Setup (Save,Recall)	Sn,0	Recall configuration 'n'.
	Sn,1	Save present setup in buffer 'n'.
	S?	Request the last Setup operation.
Trigger	T0	Continuous on TALK (default)
	T1	Continuous on GET
	T2	Continuous on rising edge external TTL
	T3	Continuous on falling edge external TTL
	T4	Continuous on rising edge analog level
	T5	Continuous on falling edge analog level
	T6	One-shot on TALK
	T7	One-shot on GET
	T8	One-shot on rising edge external TTL
	T9	One-shot on falling edge external TTL
	T?	Returns the present trigger configuration
Status	U0	Send the ADC488 status when next addressed to talk
	U1	Send the value on the digital input lines when next addressed to talk

	U2	Send pretrigger and post-trigger address limits when next addressed to talk
	U3	Send the internal scan counter value
	U4	Send calibration gain and offset values
	U5	Send the number of unread scans
	U?	Always returns U0
Version Query	V?	Return firmware revision level
Test	W0	Turn TEST LED off
	W1	Turn TEST LED on
	W?	Return on/off indication of TEST LED.
Execute	X	Execute command string
Command Channel Terminator	Y0	CR LF (default)
	Y1	LF CR
	Y2	CR
	Y3	LF
	Y?	Return current Command Channel Terminator selection.
Trigger Delay	Z0	Set Trigger Scan Delay to zero (default)
	Zn	Set Trigger Scan Delay to 'n'. (n = 0 to 16,000,000)
Query	?	Return the present configuration or mode of the command preceding the ?

Character Codes and IEEE Multiline Messages

$00 0 NUL	$10 16 DLE	$20 32 SP 00	$30 48 0 16	$40 64 @ 00	$50 80 P 16	$60 96 ‘ SCG	$70 112 p SCG
$01 1 SOH GTL	$11 17 DC1 LLO	$21 33 ! 01	$31 49 1 17	$41 65 A 01	$51 81 Q 17	$61 97 a SCG	$71 113 q SCG
$02 2 STX	$12 18 DC2	$22 34 “ 02	$32 50 2 18	$42 66 B 02	$52 82 R 18	$62 98 b SCG	$72 114 r SCG
$03 3 ETX	$13 19 DC3	$23 35 # 03	$33 51 3 19	$43 67 C 03	$53 83 S 19	$63 99 c SCG	$73 115 s SCG
$04 4 EOT SDC	$14 20 DC4 DCL	$24 36 $ 04	$34 52 4 20	$44 68 D 04	$54 84 T 20	$64 100 d SCG	$74 116 t SCG
$05 5 ENQ PPC	$15 21 NAK PPU	$25 37 % 05	$35 53 5 21	$45 69 E 05	$55 85 U 21	$65 101 e SCG	$75 117 u SCG
$06 6 ACK	$16 22 SYN	$26 38 & 06	$36 54 6 22	$46 70 F 06	$56 86 V 22	$66 102 f SCG	$76 118 v SCG
$07 7 BEL	$17 23 ETB	$27 39 ‘ 07	$37 55 7 23	$47 71 G 07	$57 87 W 23	$67 103 g SCG	$77 119 w SCG
$08 8 BS GET	$18 24 CAN SPE	$28 40 (08	$38 56 8 24	$48 72 H 08	$58 88 X 24	$68 104 h SCG	$78 120 x SCG
$09 9 HT TCT	$19 25 EM SPD	$29 41) 09	$39 57 9 25	$49 73 I 09	$59 89 Y 25	$69 105 i SCG	$79 121 y SCG

$0A 10 LF	$1A 26 SUB	$2A 42 * 10	$3A 58 : 26	$4A 74 J 10	$5A 90 Z 26	$6A 106 j SCG	$7A 122 z SCG
$0B 11 VT	$1B 27 ESC	$2B 43 + 11	$3B 59 ; 27	$4B 75 K 11	$5B 91 [27	$6B 107 k SCG	$7B 123 { SCG
$0C 12 FF	$1C 28 FS	$2C 44 , 12	$3C 60 < 28	$4C 76 L 12	$5C 92 \ 28	$6C 108 l SCG	$7C 124 | SCG
$0D 13 CR	$1D 29 GS	$2D 45 – 13	$3D 61 = 29	$4D 77 M 13	$5D 93] 29	$6D 109 m SCG	$7D 125 } SCG
$0E 14 SO	$1E 30 RS	$2E 46 . 14	$3E 62 > 30	$4E 78 N 14	$5E 94 ^ 30	$6E 110 n SCG	$7E 126 ~ SCG
$0F 15 SI	$1F 31 US	$2F 47 / 15	$3F 63 ? UNL	$4F 79 O 15	$5F 95 _ UNT	$6F 111 o SCG	$7F 127 DEL SCG
ACG	**UCG**	**LAG**		**TAG**		**SCG**	

ACG = Addressed Command Group
UCG = Universal Command Group
LAG = Listen Command Group
TAG = Talk Address Group
SCG = Sencondary Command Group

Appendix B

B.1 Example Programs

```
10      'This program demonstrates simple data acquisition of audio
        'input.
20      '
30      'The ADC488 is setup for two channels (differential inputs),
        'one volt range,
40      '50 kHz scan rate, trigger on GET, and take 1000 scans.
50      'Data is output for either DADiSP, Asyst, or spreadsheet
        '(Lotus or Quatro).
60      'Connect your stero up to channels one and two and run
        'this program.
70      '
80      OPEN "\dev\ieeeout" FOR OUTPUT AS #1
90      IOCTL#1,"BREAK"
100     PRINT#1,"RESET"
110     OPEN "\dev\ieeein" FOR INPUT AS #2
120     '
130     DIM CH1!(1000) : DIM CH2!(1000)
140     ADR$ ="14"
150     '
160     ON ERROR GOTO 1960
170     PRINT#1,"ERROR OFF"
180     '
190     PRINT"Current ADC488 Address is ";ADR$;" enter a 2 digit
        address"
200     PRINT"or hit any key to continue"
210     INPUT A$
220     IF A$ ="" THEN GOTO 240
230     IF A$  ADR$ THEN ADR$ = A$
240     IF VAL(ADR$)  30 GOTO 190
250     PRINT#1,"abort"
260     PRINT#1,"clear";ADR$
270     '
280     PRINT#1,"output";ADR$;";a0"   'set 8 dffrntl input channels
290     PRINT#1,"output";ADR$;";c1,2" 'use channels one and two
300     PRINT#1,"output";ADR$;";r0,0" 'set both channels to ± 1 volt
                                      'range
310     PRINT#1,"output";ADR$;";i1"   'set scan interval to 20 usec
                                      '50kHz rate)
```

```
320     PRINT#1,"output";ADR$;";n1000"  'put 1000 post-trigger scans in
                                        'the scan buffer
330     PRINT#1,"output";ADR$;";t1"     'trigger on GET (continuous mode)
340     PRINT#1,"output";ADR$;";x"      'execute the command string
350     '
360     'note that the above command could have been in one string
370     'as:  "output14;a0c1,2r0,0i1n1000t1x"
380     '
390     'trigger the ADC and get some data
400     '
410     PRINT#1,"trigger";ADR$              'send a GET
420     PRINT "ADC Triggered, now acquiring data"
430     '
440     'Wait until data acquisition is complete
450     '
460     PRINT#1,"spoll";ADR$
470     INPUT#2, STAT
480     IF (STAT AND &H80)    &H80 GOTO 460
490     '
495     PRINT "Data acquisition done, now making spread.dat"
500     ' Get the data from scan buffer and write in Lotus Format
510     '
520     OPEN "spread.dat" FOR OUTPUT AS #3 'Lotus format
530     '
540     PRINT#1,"output";ADR$;";g0"   'select ASCII data format
550     PRINT#1,"output";ADR$;";p0"   'select all channels
560     PRINT#1,"output";ADR$;";b0"   'point to the trigger location
570     PRINT#1,"output";ADR$;";f0"     'auto-increment the pointer
580     PRINT#1,"output";ADR$;";j44o9"'set end-of-rdng trmntr to ","
590     PRINT#1,"output";ADR$;";q1"   'set end-of-scan trmntr to CR LF
600     PRINT#1,"output";ADR$;";X"    'execute the command string
610     '
620     FOR I = 1 TO 1000
630     PRINT#1,"ENTER";ADR$
640     LINE INPUT#2,A$
650     CH1$ = MID$(A$,1,9)
660     CH2$ = MID$(A$,11,9)
670     PRINT#3,I;" ";CH1$;" ";CH2$
680     NEXT I
690     '
695     PRINT "now do DADiSP files"
700     PRINT#1,"output";ADR$;";b0X"  'point to trigger point
710     FOR I = 1 TO 1000
720     PRINT#1,"ENTER";ADR$
730     LINE INPUT#2,A$
740     CH1$ = MID$(A$,1,9)
750     CH2$ = MID$(A$,11,9)
760     CH1!(I) = VAL(CH1$) 'lst fill arrays with buffer data
770     CH2!(I) = VAL(CH2$)
780     NEXT I
```

```
790     '
800     'find  max  and  min  values  in  channel  1  array
810     '
820     MIN  =  CH1!(1)
830     MAX  =  CH1!(1)
840     FOR  J  =  1  TO  1000
850     IF  CH1!(J)    MAX  THEN  MAX1  =  CH1!(J)
860     IF  CH1!(J)    MIN  THEN  MIN1  =  CH1!(J)
870     NEXT
880     '
890     MAXVAL  =  MAX1  :  MINVAL  =  MIN1
900     '
905     PRINT  "open  dadispl.dat"
910     OPEN  "dadispl.dat"  FOR  OUTPUT  AS  #4
920     '
930     GOSUB  1130  'Write  DADisp  header  info  to  disk
940     '
950     GOSUB  2070  'Create  DADisp  files
960     MIN  =  CH2!(1)
970     MAX  =  CH2!(1)
980     FOR  J  =  1  TO  SIZE
990     IF  CH2!(J)  MAX  THEN  MAX2  =  CH2!(J)
1000    IF  CH2!(J)  MIN  THEN  MIN2  =  CH2!(J)
1010    NEXT
1020    MAXVAL  =  MAX2  :  MINVAL  =  MIN2
1030    PRINT:PRINT"The  following  header  info  is  for  Channel  2
        data":PRINT
1035    PRINT  "open  dadisp2.dat"
1040    OPEN  "dadisp2.dat"  FOR  OUTPUT  AS  #9
1045    PRINT  "do  header"
1050    GOSUB  1500  'Write  DADisp  header  info  to  disk
1060    '
1065    PRINT  "write  data"
1070    GOSUB  2120  'Write  DADisp  data  to  disk
1075    PRINT  "create  Asyst"
1080    GOSUB  1840  'Create  Asyst  format  file
1090    CLOSE
1100    END
1110    '
1120    '
1130    'Start  subroutine  to  format  data  for  DADISP  II
1140    '
1150    O$="DATASET  Channel_1"
1160    WRITE  #4,O$
1170    O$="VERSION  "+"1"
1180    WRITE  #4,O$
1190    O$="CHAN_NAME  ADC488"
1200    WRITE  #4,O$
1210    O$="MODULE  "+"NONE"
1220    WRITE#4,O$
```

```
1230    O$="CHAN_NUM  One"
1240    WRITE#4,O$
1250    O$="DATE  "+DATE$
1260    WRITE#4,O$
1270    O$="TIME  "+TIME$+".00"
1280    WRITE#4,O$
1290    O$="DATA_TYPE  "+"VOLTS"
1300    WRITE#4,O$
1310    O$="NUM_SAMPS  1000"  ]
1320    WRITE#4,O$
1330    O$="INTERVAL  20  usec"
1340    WRITE#4,O  $
1350    O$="MAX_VAL  "+STR$(MAXVAL)
1360    WRITE#4,O$
1370    O$="MIN_VAL  "+STR$(MINVAL)
1380    WRITE#4,O$
1390    O$="VERT_UNITS  Volts"
1400    WRITE#4,O$
1410    O$="HORZ_UNITS  Samples"
1420    WRITE#4,O$
1430    O$="COMMENT  "+"***  ADC488  Demonstration  ***"
1440    WRITE#4,O$
1450    O$="DATA"
1460    WRITE#4,O$
1470    RETURN
1480    '  End  of  DADiSP  II  routine
1490    '
1500    'Routine  to  write  data  from  channel  2
1510    O$="DATASET  Channel_2"
1520    WRITE  #9,O$
1530    O$="VERSION  "+"1"
1540    WRITE  #9,O$
1550    O$="CHAN_NAME  ADC"
1560    WRITE  #9,O$
1570    O$="MODULE  "+"NONE"
1580    WRITE#9,O$
1590    O$="CHAN_NUM  Two"
1600    WRITE#9,O$
1610    O$="DATE  "+DATE$
1620    WRITE#9,O$
1630    O$="TIME  "+TIME$+".00"
1640    WRITE#9,O$
1650    O$="DATA_TYPE  "+"VOLTS"
1660    WRITE#9,O$
1670    O$="NUM_SAMPS  1000"
1680    WRITE#9,O$
1690    O$="INTERVAL  20  usec"
1700    WRITE#9,O$
1710    O$="MAX_VAL  "+STR$(MAXVAL)
1720    WRITE#9,O$
```

```
1730    O$="MIN_VAL  "+STR$(MINVAL)
1740    WRITE#9,O$
1750    O$="VERT_UNITS  Volts"
1760    WRITE#9,O$
1770    O$="HORZ_UNITS  Samples"
1780    WRITE#9,O$
1790    O$="COMMENT  "+"***  ADC488  Demonstration  ***"
1800    WRITE#9,O$
1810    O$="DATA"
1820    WRITE#9,O$
1830    RETURN
1840    '
1850    '  Routine  to  creat  Asyst  files  of  Channel  1  and  2  data
1860    'Data  point  output  for  ASYST
1870    OPEN  "asyst1.dat"  FOR  OUTPUT  AS  #6       'Asyst  format
1880    PRINT
1890    OPEN  "asyst2.dat"  FOR  OUTPUT  AS  #8       'Asyst  format
1900    FOR  I=1  TO  1000
1910    PRINT#6,CH1!(I)
1920    PRINT#8,CH2!(I)
1930    NEXT  I
1940    RETURN
1950    '
1960    '
1970    '  Error  Handler
1980    '
1990    IOCTL#1,"BREAK"
2000    PRINT#1,"STATUS"
2010    INPUT#2,ST$
2020    PRINT  "Error  #";MID$(ST$,15,2);":  ";MID$(ST$,27)
2030    RESUME  NEXT
2040    '
2050    '
2060    '
2070    'Routine  to  write  data  to  file  in  DADisp  format
2080    FOR  I  =  1  TO  1000
2090    PRINT  #4,  "";CH1!(I)
2100    NEXT  I
2110    RETURN
2120    'Routine  to  write  data  to  file  in  DADisp  format  Chan  2'
2130    FOR  I  =  1  TO  1000
2140    PRINT  #9,  "";CH2!(I)
2150    NEXT  I
2160    RETURN

REM     Demonstration  Program  for  ADC488
REM     This  program  uses  DMA  to  show  high-speed  data  transfer

OPEN    "\dev\ieeeout"  FOR  OUTPUT  AS  #1
```

```
IOCTL  #1,  "break"
PRINT  #1,  "reset"
OPEN   "\dev\ieeein"  FOR  INPUT  AS  #2

REM    Setup  memory  buffer
DIM    stero%(32766)

PRINT  #1,  "abort"
PRINT  #1,  "clear14"

PRINT  #1,  "output14;a0  c1,2  r0,0  i1  g11  n131064  t1  x"
    'where:
    'a0  =  configure  8  differential  inputs
    'c1,2  =  use  two  channels  (channels  1  and  2)
    'r0,0  =  set  +/-1  volt  range  for  both  channels
    'i1  =  20  us  scan  interval  (50  KHz)
    'n131064  =  get  128k  samples
    'g11  =  uncompensated  binary  -  low  byte  first
    't1  =  trigger  on  GET,  continuous  mode

PRINT  "armed  and  ready  for  a  GET"
GOSUB  pause
PRINT  #1,  "trigger14"        '  send  a  GET

PRINT  "Wait  until  data  is  acquired"
spoll:
PRINT  #1,  "spoll14"
INPUT  #2,  spoll$
IF  (VAL(spoll$)  AND  &H80)    &H80  THEN  GOTO  spoll

REM  Get  the  first  half  of  the  readings
PRINT  #1,  "output14;b0x"
GOSUB  dma

PRINT  "done"
BEEP
REM  Graphing  Routine  is  for  a  Hercules  Graphics  Adapter  card

SCREEN  3
pass  =  1
graph:
dot  =  1
CLS
FOR  i  =  1  TO  32766  STEP  2
    IF  ((i  +  1)  MOD  1400)  =  0  THEN  CLS  :  dot  =  1
    yspot%  =  (261  -  (stero%(i)  *  .0027))
    xspot%  =  dot
    PSET  (dot,  261)
    PSET  (xspot%,  yspot%)
    yspot%  =  (87  -  (stero%(i  +  1)  *  .0027))
```

```
    xspot% = dot
    PSET (dot, 87)
    PSET (xspot%, yspot%)
    dot = dot + 1
NEXT i
IF pass = 1 THEN GOSUB dma ELSE END
pass = 2
GOTO graph
END
pause:
    PRINT "Hit any key to continue"
    WHILE "" = INKEY$: WEND
    RETURN

dma:
    PRINT #1, "enter14 #65532 buffer"; VARSEG(stero%(0)); ":";
    VARPTR(stero%(0)); "DMA"
    RETURN
```

```
10      ' GWBASIC Demonstration Program Number 3 -'Soap Box Derby
20      '
30      ' Uses two channels to determine who wins the race.
40      ' Uses pretrigger, post-trigger, level, digital I/O
50      '
60      OPEN "\dev\ieeeout" FOR OUTPUT AS #1
70      IOCTL#1,"BREAK"
80      PRINT#1,"RESET"
90      OPEN "\dev\ieeein" FOR INPUT AS #2
100     PRINT#1,"FILL ERROR"
110     '
120     ON ERROR GOTO 730
130     PRINT#1,"ERROR OFF"
140     ADR$="09"
150     '
220     SINT = 3        'Sample interval
230     PRE = 200       'Pretrigger scans
240     POST = 50       'Posttrigger scans
250     PER = 70        'Trigger level
260     PRINT#1,"abort"
270     PRINT#1,"clear";ADR$
280     '
290     PRINT#1,"output";ADR$;";a0x"          'set 8 differential channels
300     PRINT#1,"output";ADR$;";c1,2x"        'use channels one and two
310     PRINT#1,"output";ADR$;";r3,3x"        'set to +/- 5 volt range
320     PRINT#1,"output";ADR$;";i";SINT;"x"   'set scan interval to 20
                                              'usec (50kHz rate)
330       PRINT#1,"output";ADR$;";n";PRE;",";POST;"X"
340     PRINT#1,"output";ADR$;";l+";PER;"X"   'trigger on level PER
350     PRINT#1,"output";ADR$;";t4"
360     PRINT#1,"output";ADR$;";x"              'execute the command string
370     '
380     PRINT:PRINT "Waiting for a Trigger"
390     PRINT#1,"output";ADR$;";D0X"
400     PRINT#1,"output";ADR$;";U0X"
410     PRINT#1,"ENTER";ADR$
420     LINE INPUT#2,A$  ':PRINT A$
430     '
440     PRINT#1,"spoll";ADR$
450     INPUT#2, STAT
460     IF STAT 128 GOTO 440              'loop until acqustn is cmplt
470     '
480     PRINT"Aquisition Complete"
490     '
500     PRINT#1,"output";ADR$;";G0X"          'select ASCII data
510     PRINT#1,"output";ADR$;";P0B0X"        'select all channels and poi
                                              'to buffer start
520     PRINT#1,"output";ADR$;";F0X"            'auto-increment the pointer
530     PRINT#1,"output";ADR$;";J4409Q1X"     'rdng trmntr is a comma
540     PRINT"Determining the winner ..."
```

```
550     PRINT#1,"output";ADR$;";P0B-";PRE;"X" 'select  all  channels
560     '
570     PRINT#1,"ENTER";ADR$
580     LINE  INPUT#2,A$
590     CH1$  =  MID$(A$,1,9)
600     CH2$  =  MID$(A$,11,9)
610     CAR1  =  VAL(MID$(A$,1,9))
620     CAR2  =  VAL(MID$(A$,11,9))
640     IF  CAR2  >7  THEN  GOTO  680
650     IF  CAR1  >7  THEN  GOTO  690
660     GOTO  560  'Continue  until  you  find  a  peak
670     '
680     PRINT"Winner  is  Channel  2":PRINT#1,"OUTPUT";ADR$;";d2x":GOTO  700
690     PRINT"Winner  is  Channel  1":PRINT#1,"OUTPUT";ADR$;";d8x"
700     PRINT"Digital  output  shows  the  winner,  hit  any  key  to
        continue"
710     IF  INKEY$=""  GOTO  710
720     GOTO  260
730     '
740     '  Error  Handler
750     '
760     IOCTL#1,"BREAK"
770     PRINT#1,"STATUS"
780     INPUT#2,ST$
790     PRINT  "Error  #";MID$(ST$,15,2);":  ";MID$(ST$,27)
800     RESUME  NEXT
```

```
10      ' GWBASIC demonstration program number 4
20      '
30      ' This program shows how an ADC488 can be used as an
40      ' eight-channel voltmeter.
50      '
60      OPEN "\dev\ieeeout" FOR OUTPUT AS #1
70      IOCTL#1,"BREAK"
80      PRINT#1,"RESET"
90      OPEN "\dev\ieeein" FOR INPUT AS #2
100     '
110     PRINT#1,"abort"
120     PRINT#1,"clear14"
130     '
140     PRINT#1,"output14;a0" 'set 8 differential input channels
150     PRINT#1,"output14;c1,2,3,4,5,6,7,8" 'use eight channels
160     PRINT#1,"output14;r3,3,3,3,3,3,3,3" 'range to +/- 10 volts
170     PRINT#1,"output14;t6" 'trigger on TALK (one-shot mode)
180     PRINT#1,"output14;x" 'execute the command string
190     '
200     CLS
210     LOCATE 10,1,0
220     '
230     ' get the readings
240     PRINT#1,"send talk14 mla"     'address to TALK (trigger)
250     '
260     FOR I=1 TO 8
270     PRINT#1,"enter" 'get a reading
280     INPUT#2,DA$ 'and input it'
290     PRINT USING "&##    &  &";"Channel";I;DA$;"volts"
300     NEXT I
310     '
320     GOTO 210
```

```
10      ' Calibration Program
20      '
30      ' This program shows how to calibrate all channels of the
40      ' ADC488/8S and ADC488/16
50      '
60      ' The ADC488 is assumed to be set to IEEE 488 address 14
70      '
80      OPEN "\dev\ieeeout" FOR OUTPUT AS #1
90      IOCTL#1,"BREAK"
100     PRINT#1,"RESET"
110     OPEN "\dev\ieeein" FOR INPUT AS #2
120     '
130     ' ***** Get ready to calibrate *****
140     '
150     PRINT "Make sure the rear panel CAL switch is enabled
        (depressed)."
160     GOSUB 670 'pause
170     '
180     PRINT#1,"output14;h2x" 'send a "calibration complete" command
190                            'to make sure the switch is enabled.
200     GOSUB 980              'check for calibration error
210                            'exit if switch isn't ready
220     PRINT#1,"output14;a0x" 'configure the ADC488 for 8
                               'differential inputs
230     '
240     ' ***** Calibrate one volt range *****
250     '
260     PRINT "Start calibration of 1 volt range"
270     RANGE=0          'Set RANGE to 0 for Rn command (1 volt)
280     GOSUB 750        'calibrate 0 volts
290     PRINT "Apply 1.00000 volts to the ADC488"
300     GOSUB 870        'calibrate full scale
310     '
320     ' ***** Calibrate two volt range *****
330     '
340     PRINT "Start calibration of 2 volt range"
350     RANGE=1          'Set RANGE to 1 for Rn command (2 volts)
360     GOSUB 750        'calibrate 0 volts
370     PRINT "Apply 2.00000 volts to the ADC488"
380     GOSUB 870        'calibrate full scale
390     '
400     ' ***** Calibrate five volt range *****
410     '
420     PRINT "Start calibration of 5 volt range"
430     RANGE=2          'Set RANGE to 2 for Rn command (5 volts)
440     GOSUB 750        'calibrate 0 volts
450     PRINT "Apply 5.00000 volts to the ADC488"
460     GOSUB 870        'calibrate full scale
470     '
480     ' ***** Calibrate ten volt range *****
```

```
490     '
500     PRINT  "Start  calibration  of  10  volt  range"
510     RANGE=3         'Set  RANGE  to  3  for  Rn  command  (10  volts)
520     GOSUB  750      'calibrate  0  volts
530     PRINT  "Apply  10.00000  volts  to  the  ADC488"
540     GOSUB  870      'calibrate  full  scale
550     '
560     '  *****  Calibration  finished  for  each  range  *****
570     '
580     PRINT  "Wait  one  second,  then"
590     GOSUB  670  'pause  before  the  "h2"  command
600     PRINT#1,"output14;h2x"
610     PRINT  "Calib.  is  complete,  remove  all  calibration  equipment"
620     '
630     '  **********  Done  **********
640     '
650     END
660     '
670     '  Pause  subroutine
680     '
690     PRINT
700     PRINT  "Hit  any  key  to  continue"
710     PRINT
720     WHILE  ""=INKEY$:WEND
730     RETURN
740     '
750     '  Zero  volts  routine  (use  "h0"  to  calibrate  0  volts)
760     '
770     PRINT  "Apply  0.00000  volts  to  the  ADC488
780     GOSUB  670              'pause
790     FOR  CHANNEL=1  TO  8
800       PRINT#1,"output14;c";STR$(CHANNEL);"x"
810       PRINT#1,"output14;r#";STR$(CHANNEL);",";STR$(RANGE);"x"
820     PRINT#1,"output14;h0x"
830     GOSUB  980              'Check  for  a  calibration  error
840     NEXT  CHANNEL
850     RETURN
860     '
870     '  Full  range  routine  (use  "h1"  to  calibrate  full  range)
880     '
890     GOSUB  670              'pauuse
900     FOR  CHANNEL=1  TO  8
910       PRINT#1,"output14;c";STR$(CHANNEL);"x"
920       PRINT#1,"output14;r#";STR$(CHANNEL);",";STR$(RANGE);"x"
930     PRINT#1,"output14;h1x"
940     GOSUB  980              'Check  for  a  calibration  error
950     NEXT  CHANNEL
960     RETURN
970     '
980     'Calibration  Error  Checker
```

```
990     '
1000    PRINT#1,"output14;e?"          'query error status
1010    PRINT#1,"enter14"              'address ADC488 to talk
1020    INPUT#2,E$                     'get status from ADC488
1030    IF E$="E16" THEN PRINT "ADC488 Calibration Error! Refer to
        the ADC488 manual":BEEP:END
1040    RETURN
```

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