Analyzer488

IEEE 488 Bus Analyzer

INSTRUCTION MANUAL

IOtech, Inc. • 25971 Cannon Road • Cleveland, Ohio 44146 • (440) 439-4091

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

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, Cautions, Notes, and Tips

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 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.

Analyzer488

User's Manual

Part No. Analyzer488-901

Revision 1.0

September 1989

IOtech, Inc. • 25971 Cannon Road • Cleveland, Ohio 44146 • (440) 439-4091

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Introduction

1.1 Description

The Analyzer488 IEEE 488 Bus Analyzer is a multifunction IEEE 488 device that can monitor, capture, and analyze IEEE 488 bus events. The Analyzer488 can function as a full-featured IEEE 488 bus controller, providing full IEEE 488-1978 bus implementation. The Analyzer488 can also operate as an instrument simulator, emulating any one of hundreds of IEEE 488 instruments, such as digital oscilloscopes and multimeters. Data for several different waveforms can be retrieved from the Analyzer488.

The Analyzer488 can be operated from the front panel or from a personal computer or terminal connected to the serial port. Analyst488 software is included that runs on an IBM PC to provide all the Analyzer488 functions using a menu driven approach.

As a high level controller, the Analyzer488 interprets simple high level commands sent from the computer's serial port and performs the necessary, and usually complex, bus control and handshaking. The commands and protocol are similar to those used by the Hewlett Packard HP-85 computer. As a low level controller, The Analyzer488 offers complete control of bus management, data, and handshaking lines.

Front panel operation allows a user to configure the Analyzer488, control the IEEE 488 bus, view bus transactions, record bus transactions, and print recorded bus events to a serial or IEEE 488 printer.

1.2 About This Manual

This manual describes the operation of the Analyzer488 IEEE 488 Bus Analyzer, manufactured by IOtech, Inc. It assumes that you are familiar with using the IEEE 488 bus, using a serial terminal, using a personal computer, menus, and using a mouse.

Section 1, Introduction, is an introduction to the Analyzer488. It includes a description, specifications, and service information.

Section 2, Getting Started, presents a step-by-step introduction to using the Analyzer488. It provides an overview of the three operating modes and examples of displaying bus information, storing bus events, and analyzing bus data.

Section 3, Front Panel Operation, gives a detailed explanation of operating the Analyzer488 from the front panel. It includes a description of all the controls and indicators and provides examples of all the features of front panel operation.

Section 4, Serial Controller, provides a description of the Serial Controller Mode of operation of the Analyzer488. It includes an explanation of how to use the Analyzer488 as a serial controller and a detailed description of all the commands.

Section 5, Analyst488 PC Software, shows how to use the Analyzer488 with the Analyst488 PC software running on an IBM PC computer. It includes a description of all menus, screens, and windows and gives detailed examples of all the Analyst488 software features.

Section 6, Instrument Simulator, shows how the use the Analyzer488 as an instrument simulator. It gives a detailed explanation of all the commands that are available when the Analyzer488 is in this mode of operation.

Appendices A to H provide reference material, including an IEEE primer, cable and pinout details, command summaries, a sample program listing, error messages, and a glossary of terms.

1.2.1 Typographic Conventions

This manual uses different typefaces to represent keys, commands, displays, and various other elements important to using the Analyzer488. The following typographical conventions are used in this manual:

[KEYS]	Analyzer488 front panel keys are shown in square brackets.
<K>	Computer keys are shown in angle brackets
COMMANDS	Command keywords are shown in this typeface.
Menus	Menu selections are shown in boldface.
DISPLAYS	Analyzer488 front panel displays are shown this way.
output	Responses are shown like this.
ATN LED	LED indicators are called out this way.
%00000000	Specifies Binary data
&H3F	Specifies Hexadecimal data
122	Specifies decimal data
'A'	Specifies ASCII data

1.2.2 Abbreviations

The following IEEE 488 abbreviations are used throughout this manual.

addr n	IEEE bus address "n"
ACG	Addressed Command Group
ACK	Acknowledge
ATN	Attention
BEL	Bell
BS	Backspace
CAN	Cancel
CR	Carriage Return
data	Data String
DAV	Data Valid bus handshaking line
DC1	Device Control 1
DC2	Device Control 2
DC3	Device Control 3
DC4	Device Control 4
DCL	Device Clear
DIOn	Bus data lines n = 1 to 8
DLE	Data Link Escape
EM	End of Medium

ENQ	Enquiry
EOI	End or Identify
EOT	End of Transmission
ESC	Escape
ETB	End Transmission Block
ETX	End of Text
FF	Form Feed
FS	File Separator
GET	Group Execute Trigger
GS	Group Separator
GTL	Go To Local
HT	Horizontal Tab
IFC	Interface Clear
LAG	Listen Address Group
LF	Line Feed
LLO	Local Lock Out
MLA	My Listen Address
MTA	My Talk Address
NAK	Negative Acknowledge
NDAC	Not Data Accepted
NRFD	Not Ready For Data
NUL	Null
PPC	Parallel Poll Configure
PPU	Parallel Poll Unconfigure
REN	Remote Enable
RS	Record Separator
SCG	Secondary Command Group
SDC	Selected Device Clear
SI	Shift In
SO	Shift Out
SOH	Start Of Heading
SPD	Serial Poll Disable
SPE	Serial Poll Enable
SRQ	Service Request
STX	Start of Text
SUB	Substitute
SYN	Synchronous Idle
TAD	Talker Address
TAG	Talk Address Group
TCT	Take Control

`term`	Terminator
UCG	Universal Command Group
UNL	Unlisten
UNT	Untalk
US	Unit Separator
VT	Vertical Tab
*	Unasserted

1.3 Specifications

IEEE 488-1978 Interface

Front Panel Mode:	SH1, AH1, E1
Serial Controller Mode:	SH1, AH1, E1 with C1, C2, C3, C4 and C28 controller subsets
Instrument Simulator Mode:	SH1, AH1, T6, TE0, L4, LE0, SR1, RL0, PP2, DC1, DT1, E1. Programmable terminators, waveforms, data format, buffer data, display data, keypad response, parallel and serial poll responses, service request and status.
Connectors:	Standard IEEE 488 connector with metric studs. BNC connector for trigger output. Card edge connector for all IEEE 488 bus signals(unbuffered). DB9 RS-232 connector.
Handshake Speeds:	Programmable Single Step, Slow (1 to 10 bytes/second), Fast (up to 1M byte/second) and None.
Capture Trigger:	Bus Management and Data line programmable for low (0), high (1) and don't care (0 or 1). Programmable pattern occurrence from 1 to 65,535, trigger delay from 1 to 99,999,999 bus events and post trigger capture depth from 1 to 32767 transactions.

Serial Interface

EIA RS-232C:	AB, BA, BB, CA, CB
Character Set:	Asynchronous bit serial.
Serial I/O Buffers:	8,000 characters total, dynamically allocated.
Output Voltage:	±5 volts min (RS-232C)
Input Voltage:	±3 volts min.; ±15v max.
Baud Rate:	Selectable 300, 600, 1200, 2400, 4800, 9600, and 19,200.
Data Format:	Selectable 7 or 8 data bits; 1 or 2 stop bits; odd, even, and no parity.
Duplex:	Full with Echo/No Echo.

Serial Control:	Selectable CTS/RTS or XON/XOFF.
Terminators:	Selectable CR, LF, LF-CR and CR-LF.
Connector:	9-pin Sub-D male. Serial cable included to connect to 9 or 25 pin IBM PC, XT and AT.

General

Display:	20 character vacuum fluorescent alphanumeric display with adjustable brightness level.
Indicators:	20 LEDs for various status indications.
Power:	105-125V or 210-250V; 50-60 Hz, 20 VA Max.
Dimensions:	245 mm deep x 280 mm wide x 35 mm high (9.6" x 11" x 1.4").
Weight:	1.8 kg. (4 lbs).
Environment:	0° - 50°C; 0 to 70% R.H. to 35°C. Linearly derate 3% R.H./°C from 35° to 50°C.
Controls:	Power Switch (external), 46 key keypad.

Specifications subject to change without notice.

1.4 Available Accessories

Additional accessories that can be ordered for the Analyzer488 include:

CA-7-1	1.5 foot IEEE 488 Cable
CA-7-2	6 foot IEEE 488 Cable
CA-7-3	6 foot shielded IEEE 488 Cable
CA-7-4	6 foot reverse entry IEEE 488 Cable
CA-47	IBM PC/XT/PS2/AT to Analyzer488 RS-232 Cable
CN-20	Right Angle IEEE 488 adapter, male and female
CN-22	IEEE 488 Multi-tap bus strip, four female connectors in parallel
CN-23	IEEE 488 panel mount feed-through connector, male and female
ABC488	IEEE 488 ABC switch
Analyzer488-901	Analyzer488 Instruction Manual
Analyzer488-301	Analyzer488 Carrying Case

1.5 Inspection

The Analyzer488 was carefully inspected, both mechanically and electrically, prior to shipment. When you receive the Analyzer488, 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 such damage to the shipping agent. Remember to retain all shipping materials in the event that shipment back to the factory becomes necessary.

Every Analyzer488 is shipped with the following:

• **Analyzer488**	IEEE 488 Bus Analyzer
• **Analyzer488 -901**	Instruction Manual
• **CA-47**	Serial Cable
• **Analyst488**	Analyst Software for the IBM PC
• **Analyzer488-311**	Function Key Templates
• **Analyzer488-900**	Quick Reference Card
• **Power Supply**	5 Volt Regulated TR-9; 115V or TR-9E; 220V

When the Analyzer488 is powered-on, it performs various hardware tests and displays the results on the front panel display. If the Analyzer488 detects a hardware failure during these tests, a message will appear on the display showing the failure. The three failure messages that may be displayed are shown below:

Analyzer RAM Error

The Analyzer488 has detected an error while testing system RAM. No further operation will occur.

Analyzer ROM Error

The Analyzer488 has detected an error while testing system ROM. No further operation will occur.

Analyzer NVRAM Error

The Analyzer488 has detected an error while testing the non-volatile RAM. The factory default configuration will be restored, the error will be corrected, and the Analyzer488 will configure itself for Front Panel operation.

After a normal power-up or a configuration operation, the Analyzer488 displays its operating mode on the 20 character front panel display . If configured for the Serial Controlled Bus Analyzer operation the display will show the following message :

Serial Controller

If configured for Instrument Simulator operation the display will show the massage:

Instrument Simulator

These messages will remain displayed until overwritten by a user display command.

If the Analyzer488 is configured as a stand alone bus controller the display will show the current state if the IEEE488 bus as described in the following sections. A typical display when in Front Panel mode is shown below:

BUS 'A' 41 ¤

1.6 Service Information

IOtech maintains a factory service center in Cleveland, Ohio. If problems are encountered in using the Analyzer488, you should first telephone the factory. Many problems can be resolved by discussing the problem with our applications department. If the problem cannot be solved by this method, you will be instructed as to the proper return procedure.

Getting Started

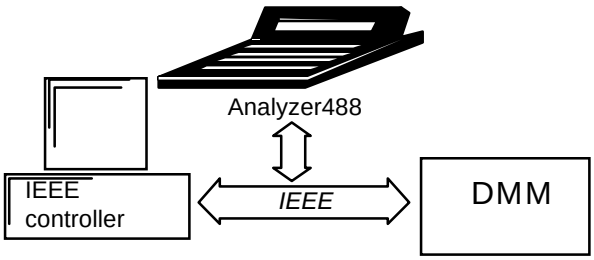
2.1 Introduction

This section contains basic operating instructions for the Analyzer488. A quick introduction to using the Analyzer488 is presented. Then, detailed instructions are given. The explanations include information about configuration, operation, and an explanation of the rear panel connectors.

2.2 Quick Start

The following is a simple example using the Analyzer488 to monitor the IEEE 488 bus, record some bus events, and view the recorded transactions. This example uses the Analyzer488 to monitor and record simple transactions between an IOtech Personal488 PC/IEEE Controller and a Keithley Instruments Model 196 System Digital Multimeter.

Using the Analyzer488 to Monitor and Record Bus Tranactions



Connect the controller to the DMM and set the IEEE address of the DMM to 16. Apply power to the PC and the DMM and run the IOtech keyboard controller program (KYBDCTRL.EXE) that is included with the Personal488 interface.

Plug the Analyzer488 power supply connector into the rear panel power jack.

CAUTION

Never install the power supply into the interface while it is connected to AC line power. Failure to observe this caution may result in damage to the Analyzer488.

WARNING

The power supply provided with the interface is intended for <u>INDOOR USE ONLY</u>. Failure to observe this warning could result in equipment failure, personal injury or death.

After installing the power supply connector into the interface, plug the power supply into AC line power. Place the rear panel power switch in the ON [depressed] position. All of the front panel LEDs will light (with the exception of the TALK LED) while the Analyzer488 performs an internal ROM and RAM self test. At the end of this self test, the front panel display will show the following message:

Analyzer488 Rev 1.0

After a few seconds, the front panel display will show the state of the IEEE 488 bus in the format as shown below:

BUS DCL 14 •

Note that the BUS and LISTEN LEDs will be on at this time. Now, connect an IEEE cable to the rear panel IEEE 488 connector on the Analyzer488 and to the bus. For the following examples, the commands sent to the Personal488 will be shown on the left column and the display of the Analyzer488 will be shown on the right.

Enter the following commands at the keyboard controller prompt (CMD>) and observe the Analyzer488 display:

```
                                   LOCATION    MESSAGE    EOI  ATN SRQ  REN IFC
CMD> output 16;F0R0X               BUS         ACG13      OD    ¤   ¤
CMD> enter 16;eoi                  BUS         TAG16      50    ¤   ¤
NDCV-000.0156E-3
```

The display of the Analyzer488 after each command shows the last byte that was transferred on the bus. Although you only see the last data byte, several bytes were transferred. To view each transaction at at readable rate, set the handshake speed to Slow by pressing the [SLOW] key. Set the rate to 3 bytes per second by pressing the [SHIFT] key followed by the [SLOW] key. Use the curosr keys to select a rate of 3 steps per second then press [ENTER].

Enter the same commands to the keyboard controller. Observe the Anlayzer488 display and you will see the bus activity at 3 bytes per second:

```
                                   LOCATION    MESSAGE    EOI  ATN SRQ  REN IFC
CMD> output 16;F0R0X               BUS         TAG21      55    ¤   ¤
                                   BUS         UNL        3F    ¤   ¤
                                   BUS         LAG21      30    ¤   ¤
                                   BUS         'F'        46        ¤
                                   BUS         '0'        30        ¤
                                   BUS         'R'        52        ¤
                                   BUS         '0'        30        ¤
                                   BUS         'X'        58        ¤
                                   BUS         CR         OD    ¤   ¤
                                   BUS         ACG13      OD    ¤   ¤
```

	LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
`CMD> enter 16;eoi`	BUS	UNL	3F		¤		¤	
	BUS	LAG21	35		¤		¤	
	BUS	TAG16	50		¤		¤	
	BUS	'N'	4E				¤	
	BUS	'D'	4D				¤	
	BUS	'C'	43				¤	
	BUS	'V'	56				¤	
	BUS	'-'	2D				¤	
	BUS	'0'	30				¤	
	BUS	'0'	30				¤	
	BUS	'0'	30				¤	
	BUS	'.'	2E				¤	
	BUS	'0'	30				¤	
	BUS	'1'	31				¤	
	BUS	'5'	35				¤	
	BUS	'6'	36				¤	
	BUS	'E'	45				¤	
	BUS	'-'	2D				¤	
	BUS	'3'	33				¤	
	BUS	CR	OD				¤	
	BUS	TAG16	50		¤		¤	

Now, record the bus events by pressing `[REC]` to enable recording. You may want to set the handshake selection to FAST or NONE to prevent the Analyzer488 from affecting the bus transaction speed by pressing `[FAST]` or

[SHIFT][STEP](which will select NONE).

Enter the same commands as before.

After issuing the commands, press [REC] to disable recording and then press [VIEW] to view the record memory. To scroll through the record memory locations, press [↑] or [↓].

Refer to paragraph 3.7 for more examples of how to use the front panel of the Analyzer488.

2.3 Configuration

Configuration of the Analyzer488 operating mode is accomplished at power-up. The default operating mode and specific parameters for that mode, such as IEEE 488 bus and RS232 serial port parameters and terminators can be set. Once configuration parameters are set, they become the power-on default values for the Analyzer488.

Pressing and holding the [SETUP] key while turning on the rear panel ON/OFF switch invokes menus that use the front panel display and keyboard to set various operating mode parameters. Use the [↑] and [↓] keys to scroll through parameters or use the [0] - [9] keys for numeric data. Press [ENTER] to select a parameter, press [EXIT] when parameters are set, or press [ESC] to abort the configuration session. Pressing [EXIT] will use all parameters that have been selected using [ENTER] and will leave the configuration menus with any remaining parameters unchanged.

The following menus are used for configuration. Press [ENTER] between displays to advance to the next menu.

	LOCATION	MESSAGE	ATN EOI	REN SRQ	REN IFC
Press [ENTER] to skip reset.		Config Reset?			No
Press [ENTER] to reset factory default parameters		Config Reset?			Yes

Press [ENTER] for Front Panel Mode — Config? Front Panel

Press [ENTER] for Serial Controller Mode — Config? Serial

Press [ENTER] for Instrument Simulator Mode — Config? Simulator

If Front Panel Mode is selected, configuration ends. The following displays are used only for Serial Controller Mode and Instrument Simulator Mode.

The default IEEE 488 bus address is displayed in decimal. — IEEE488 Address? 10

Change the bus address using the data entry and cursor keys. — IEEE488 Address? 21

The default IEEE bus terminators are displayed — Bus Terms? CR LF

Change IEEE bus terminators using the cursor keys — Bus Terms? CR / Bus Terms? LF / Bus Terms? LF CR

If Instrument Simulator Mode is selected, configuration ends. The following displays are used only for Serial Controller Mode.

The default serial port baud rate is displayed. — Serial Baud? 9600

Change the baud rate using the cursor keys. — Serial Baud? 300 / Serial Baud? 600 / Serial Baud? 1200 / Serial Baud? 2400 / Serial Baud? 4800 / Serial Baud? 19200

The default selection for data bits is displayed.	Serial Data Bits? 8
Change the number of data bits using the cursor keys.	Serial Data Bits? 7
The default selection for number of stop bits is displayed.	Serial Stop Bits? 2
Change the number of stop bits using the cursor keys.	Serial Stop Bits? 1
The default selection for parity is displayed.	Serial Parity? None
Change the parity using the cursor keys.	Serial Parity? Even
	Serial Parity? Odd
The default handshaking selection is displayed.	Serial Ctrl?XON/XOFF
Change the handshake selection using the cursor keys.	Serial Ctrl? RTS/CTS
The default echo mode is displayed.	Serial Echo? No
Change the echo mode using the cursor keys.	Serial Echo? Yes
The default serial terminator is displayed.	Serial Terms CR
Change serial terminators using the cursor keys.	Serial Terms? LF Serial Terms? LF CR Serial Terms? CR LF
When configuration is finished is displayed.	Serial Controller

2.4 Detailed Operation

This section describes the operation of the Analyzer488 hardware. It includes terminology, a description of the IEEE 488 bus handshaking control, bus recording with and without triggering, time and rate measurement, and analyzing results.

2.4.1 Analyzer488 Terminology

Asserted	The "asserted" state of a signal is the logical true state of that signal regardless of whether high true or low true logic is being used.
Unasserted	The "unasserted" state of a signal is the logical false state of that signal regardless of whether high true or low true logic is being used.
Bus State	The logical state of the IEEE 488 bus data lines, management lines and handshake lines.
Bus Lines	Refers to the five bus management lines, three handshake lines, and eight bus data lines. The management lines are Attention (ATN), Interface Clear (IFC), Remote Enable (REN), End or Identify (EOI), and Service Request (SRQ).
Bus Handshake	Used to refer to the three-wire IEEE 488 bus handshake lines. The handshake lines are Data Available (DAV), Not Ready for Data (NRFD), and Not Data Accepted (NDAC).
Bus Event	A bus event is a transaction on the bus. The term "bus event" indicates an action on the bus that will force the Analyzer488 to record the bus state. These events include asserting DAV, asserting IFC, and asserting EOI after asserting ATN (Parallel Poll). Bus events are sensed in an edge-sensitive manner. If a bus event occurs, subsequent bus events cannot be recorded until the first bus event terminates, i.e. if IFC is asserted, assertion of the DAV signal will not be recognized until IFC is unasserted.

Bus Error — A bus error occurs when a talker on the IEEE 488 bus attempts to transfer data by asserting the DAV signal and finds that the NRFD and NDAC handshake signals are both unasserted, indicating no listeners are present. The Analyzer488 is capable of recording bus errors.

Bus Record — The bus events that the Analyzer488 records are stored as bus records. These records include the state of the bus data lines and bus management lines. The Analyzer488 does not record the state of the bus handshake lines directly, but records a single bit which signifies whether or not a bus error occurred on the event. Additionally, a bit indicating the position of a trigger point and a Not Empty indicator bit are recorded.

2.4.2 Handshaking and Controlling the Speed of the Bus

When connected to an IEEE 488 bus, the Analyzer488 is capable of participating in, or not participating in, the bus three-wire handshake. The Analyzer488 handshaking is used with all Analyzer488 bus operations including viewing, recording, and controlling events on the IEEE 488 bus. When participating, the Analyzer488 acts as either a source (Talker) or acceptor (Listener). When not participating in handshaking (NONE handshake selection), the Analyzer488 monitors the bus without affecting bus transfer rates or appearing to be a listener.

Three general data transfer rates may be used for both source and acceptor handshaking. The three rates are defined as Fast (up to 1 MegaByte/Second), Slow (1 to 10 Bytes/Second) and Step (single-step operation).

The Fast handshake selection allows the Analyzer488 to perform data transfers on the IEEE 488 bus at the maximum transfer rate of the bus, 1 MegaByte/Second. Slow handshaking and Step handshaking allow the Analyzer488 to control the speed of bus transfers on the IEEE 488 bus at rates below 10 Bytes/Second.

When configured as a Talker, the Analyzer488 will assert the DAV line when set to Step handshaking. It will assert DAV from 1 to 10 times a second if Slow handshaking is used. When configured as a Listener, the

Analyzer488 unasserts the bus NDAC line for each Step, or from 1 to 10 times a second if Slow handshaking is used.

If the Analyzer488 is to talk or listen on the bus it must be in one of the handshake modes, Fast, Slow, or Step. To view the data transactions on the bus as they occur, either Slow or Step handshaking speeds may be used. If the Analyzer488 is used to control the flow of data on the bus, then Step handshaking must be used to hold off and step through bus transactions.

If the Analyzer488 is used to monitor an active bus at full speed, either Fast handshaking or no handshaking (None) may be used. If the Analyzer488 is used to detect bus errors, it must be used with None handshake otherwise the Analyzer488 will complete a handshake on the bus when a bus error would normally occur.

2.4.3 Bus Recording System

The bus recording system is used to record bus events for later analysis. When recording is enabled, the Analyzer488 stores the state of the bus data lines and bus management lines, the bus error state, the state of the trigger signal, and a Not Empty flag into record memory when bus events occur on the IEEE 488 bus. The Analyzer488 can record 32,768 bus events at a maximum rate of 1 MegaByte/second.

The record memory in the Analyzer488 is organized as a circular buffer. If more than 32,768 bus events occur while recording, only the latest 32,768 bus events will be stored, the earlier recorded events are overwritten.

The recording system is enabled and disabled by a simple command or keypress and may operate with any of the handshake selections. Bus events cause the Analyzer488 to record regardless of their source. If the Analyzer488 is talking while recording is enabled, these bus events will be recorded.

The recording system includes a trigger mechanism which may be used to automatically disable recording or switch the handshake selection to Step upon satisfying specified conditions. The trigger system is described in more detail in section 2.4.6.

2.4.4 Recording Statistics

While recording, the Analyzer488 maintains various statistics about the number of bus events which have occurred and the position of particular bus events within the Analyzer488 record memory. When recording is first enabled these statistics are reset to zero or set to default values.

The statistics include the following:

Total number of bus events which have occurred	0 to 99,999,999
Total number of bus events in memory	0 to 32,768
Total number of bus events which occurred prior to the trigger event	0 to 99,999,999
Total number of bus events in memory which occurred prior to the trigger point event	0 to 32,767
Total number of bus events which occurred after the trigger signal	0 to 99,999,999
Total number of bus records in memory which occurred after the trigger point event	0 to 32,767
Location of the earliest recorded bus record	0 to 32767 or invalid
Location of the last recorded bus record	0 to 32767 or invalid
Location of the trigger point event	0 to 32767 or invalid

2.4.5 Statistics Formulas

The total number of bus events can exceed the record memory size of the Analyzer488. The Analyzer488 will count up to 99,999,999 bus events before resetting the statistics. The maximum total number of recorded events is 32,768, the size of the record memory. Below are simple equations showing how the Total Event and Total number of recorded events are computed. In each equation, the trigger point counts as one event.

Total events = events prior to trigger + trigger point + total after trigger

Total records = records prior to trigger + trigger point + total after trigger

2.4.6 Trigger System

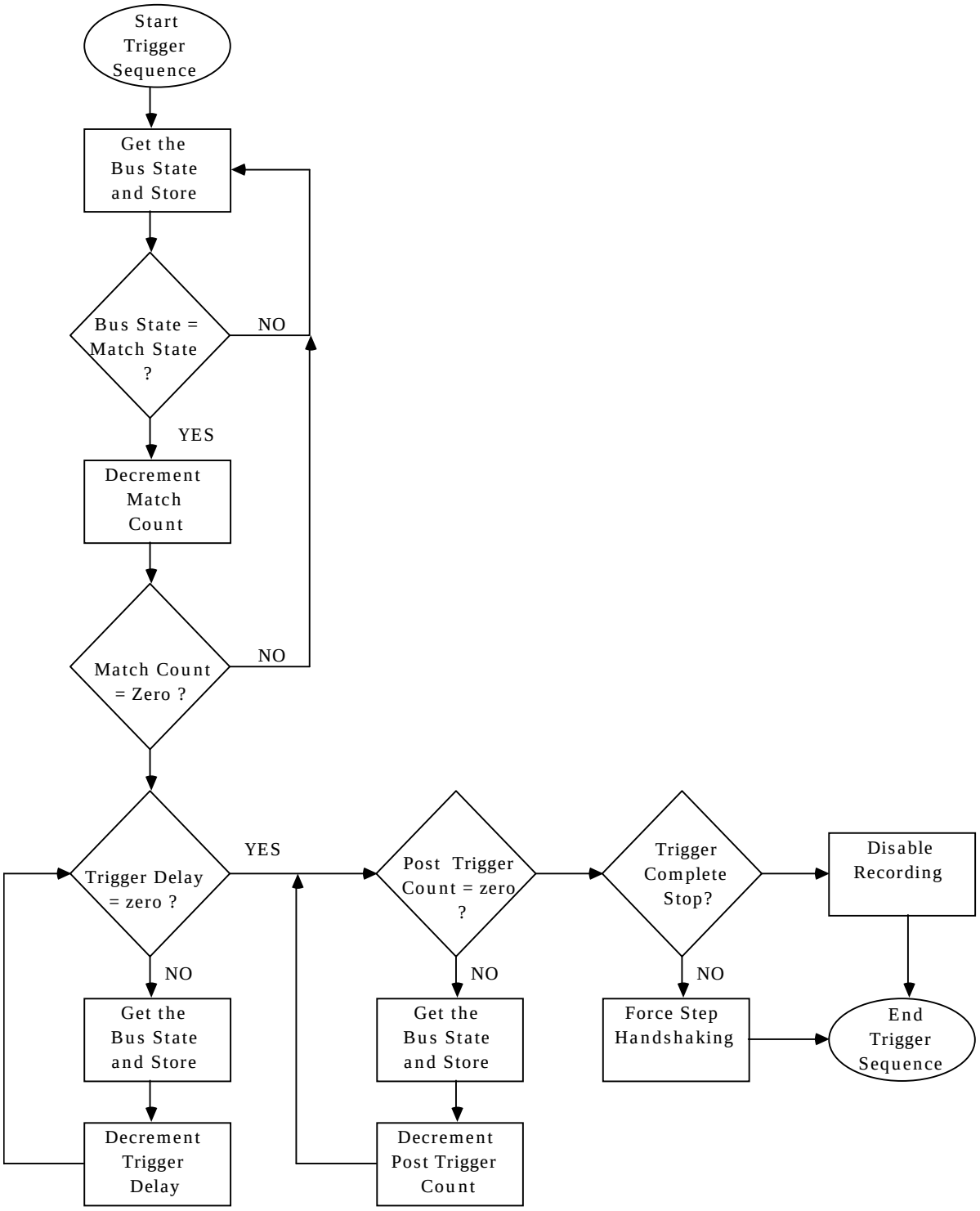
Triggering may be used while recording bus events to either halt the IEEE 488 bus or to stop recording after a specific bus event occurs. This feature is especially useful if more than 32,768 bus events occur while recording. By either stopping the IEEE 488 bus or disabling recording, the trigger system ensures that bus events of interest will not be overwritten by subsequent bus events.

To use the trigger system to record bus events, a trigger point (a specific bus state) must first be defined. Also, the number of matches to skip before recording, the number of events to delay after the specified number of matches, the number of events to record after the delay, and a trigger complete action must be specified.

The trigger system is enabled (armed) and disabled (disarmed) by a command or keypress. The arming mechanism resets the trigger system counters, the trigger output signal, and partially enables the trigger system. However, if the recording system is not also enabled, bus events will be ignored by the trigger system. When recording and triggering are both enabled, the Analyzer488 will record bus events and monitor them for the trigger event.

The trigger system is composed of five stages. These stages are listed below and shown pictorally in the following flowchart:

1. The Match Pattern (State of Data and Bus Management lines)
2. The Number of Occurrences of the trigger Match Pattern (Match Count)
3. The Number of events to count after the specified number of occurrences of the match pattern have been found. (Trigger Delay)
4. The Number of events to count after the Trigger Delay count is decremented to zero. (Post Count)
5. The action of the Analyzer488 after the specified number of events have been recorded (When Complete action)



Trigger Sequence Flow Chart

Figure #1

The first stage of the trigger system is the match comparator stage. This stage compares the bus state to a specified match state for each bus event. If a match is detected, this stage sends a signal to the next stage and also to the CMP output on the Analyzer488 rear panel card edge connector. The match state is composed of 8 bus data bits, 5 bus management bits, and 1 bit which signifies whether or not a bus error occurred during the event. Each bit may be set to 0 (unasserted), 1 (asserted), or x (don't care).

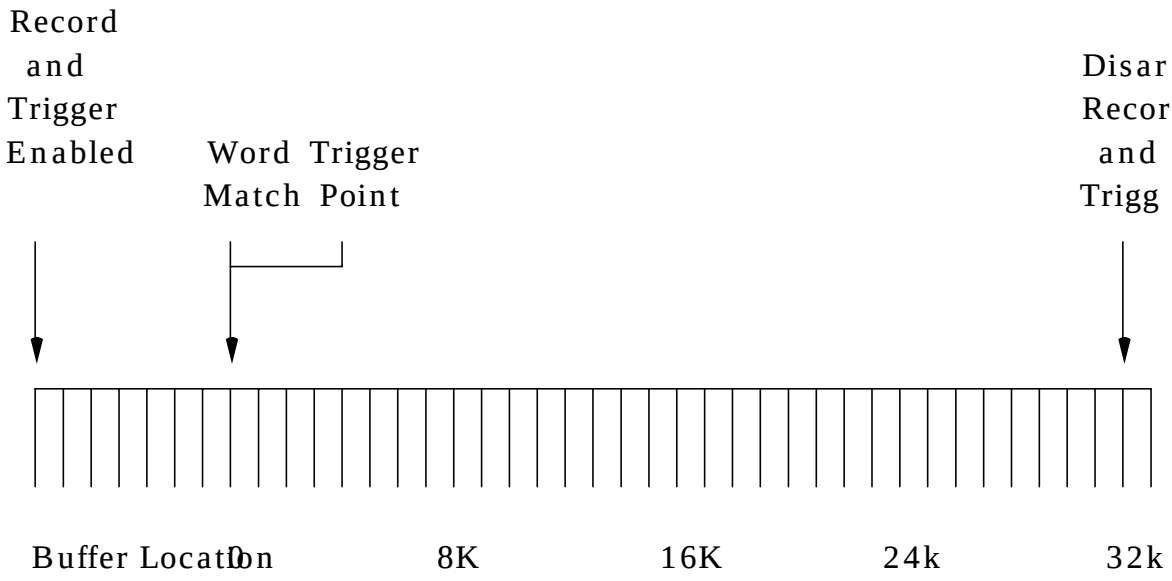
The second stage of the trigger system is the match count stage. This stage counts the number of matches found in the match comparator stage. When a specified number of matches has occurred, this stage sends a signal to the next stage.

The third stage of the trigger system is the delay count stage. After this stage receives a signal from the match count stage, it counts bus events. When a specified number of bus events has occurred, this stage marks the trigger point and sends a signal to the next stage and to the trigger output (the BNC connector) on the rear panel of the Analyzer488.

The fourth stage of the trigger system is the post trigger count stage. This stage counts the number of bus events occurring after the trigger point. When a specified number of post trigger events have occurred, the post trigger count stage generates a signal to the last stage of the trigger system.

The fifth and final stage is the trigger complete stage, which either disarms the recording system or forces handshaking to Step (leaving recording enabled) when the required number of post trigger events have occurred. At this time the trigger system is disabled.

The following illustrations and explanations show three examples using the trigger system.

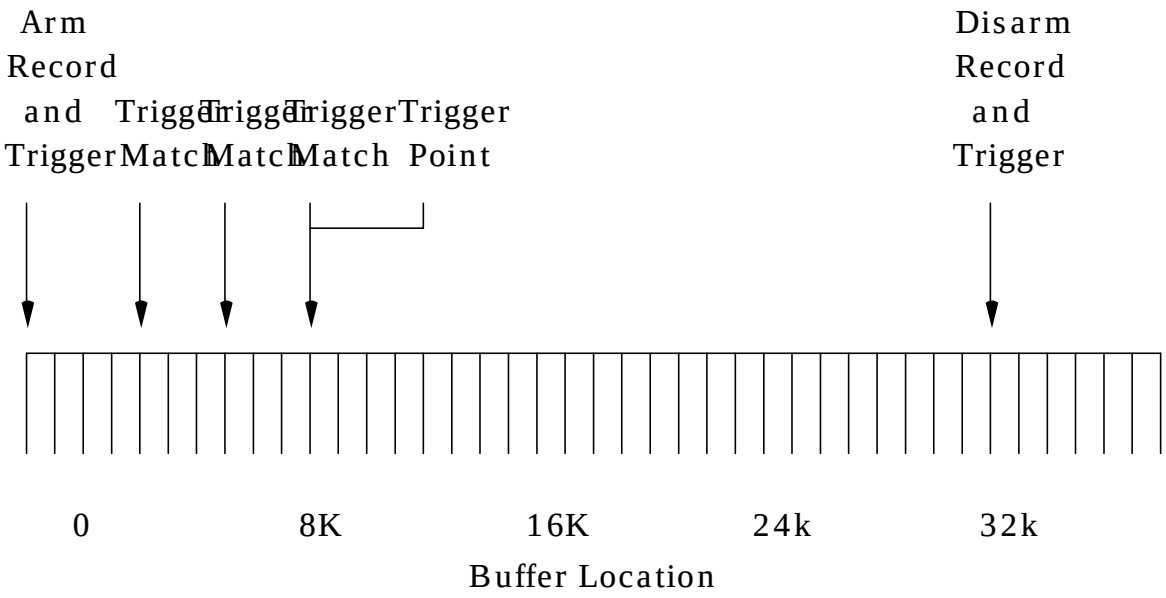


Example #1 Simple trigger word ma

Match Word = LAG10, Match Count = 1, Trigger Delay = 0,
Post Count = 32767 and Trigger Complete Action is Stop

For the diagram shown above, the trigger event was defined as follows: Match Word = LAG10 (Listen Address Group 10), Match Count = 1,Trigger Delay = 0, Post Count = 32767, and Trigger Complete Action = Stop.

Once the trigger event was defined, recording and triggering were enabled. The Analyzer488 began recording bus transactions, comparing each event with the defined match word. As soon as the match word was found, the Analyzer488 began counting and recording bus events until 32767 events were recorded. At that time, recording was disabled. The diagram shows that the event marked as the trigger point is stored in location 0 and the last event recorded is stored in location 32767.

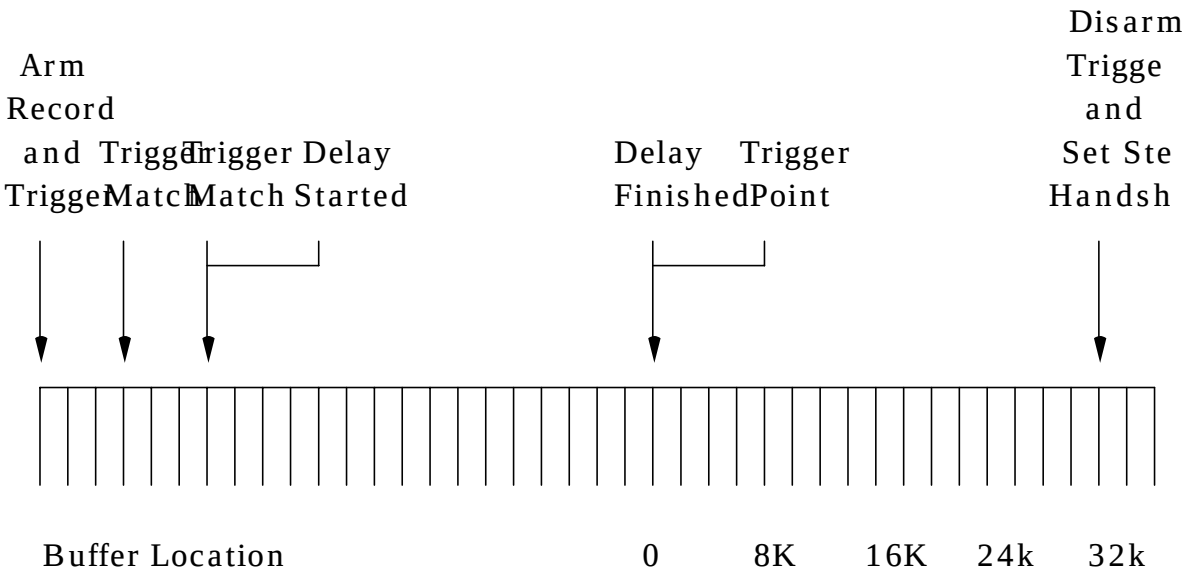


Example # 2 Counted trigger with Pre-trigger

Match Word = LAG10, Match Count = 3, Trigger Delay = 0,
Post Trigger Count = 24576 and Trigger Complete Action is Stop

For the diagram shown above, the trigger event was defined as follows: Match Word = LAG10 (Listen Address Group 10), Match Count = 3,Trigger Delay = 0, Post Count = 24576, and Trigger Complete Action = Stop.

Once the trigger event was defined, recording and triggering were then enabled. The Analyzer488 began recording bus transactions, comparing each event with the defined match word. Each time the match word was found, the Match Count was decremented. When the Match Count was decremented to zero, the Analyzer488 began counting and recording bus events until 24576 events were recorded. At that time, recording was disabled. The diagram shows that the event marked as the trigger point is stored at location 8192 and the last event recorded is stored in location 32767.



Example # 3 Counted trigger with Delay

Match Word = LAG10, Match Count = 2, Trigger Delay = 32768,
Post Trigger Count = 32768 and Trigger Complete Action is Step

For the diagram shown above, the trigger event was defined as follows: Match Word = LAG10 (Listen Address Group 10), Match Count = 2,Trigger Delay = 32768, Post Count = 32768, and Trigger Complete Action = Step.

Once the trigger event was defined, recording and triggering were then enabled. The Analyzer488 began recording bus transactions, comparing each event with the defined match word. Each time the match word was found, the Match Count was decremented. When the Match Count was decremented to zero, the Analyzer488 began counting and recording bus events until 32768 events were counted. At that time, the trigger event was marked and recording continued until 32768 events were recorded. At that time,recording was still enabled and the handshake circuitry was set to step mode, halting bus transactions until a step command was issued. The diagram shows that the event marked as the trigger point is stored at location 0. The last event recorded in this case will be the last event recorded before recording was disabled by issuing a command or by a keypress.

2.4.7 Time and Rate Measurements

The Analyzer488 provides statistics of time and rate measurements of previous bus transactions. It reports the post-trigger time period, the number of bus events that occured during that time period, and the effective transfer rate.

When the Analyzer488 records bus transactions using triggering, it measures the length of time between the trigger point and the last post-count event recorded. This measurement is made with a resolution of two microseconds (2 μsec).

The number of bytes recorded is divided by the post-trigger time period the to calculate an effective transfer rate of bus transactions in bytes per second.

2.4.8 Analyzing Bus Records

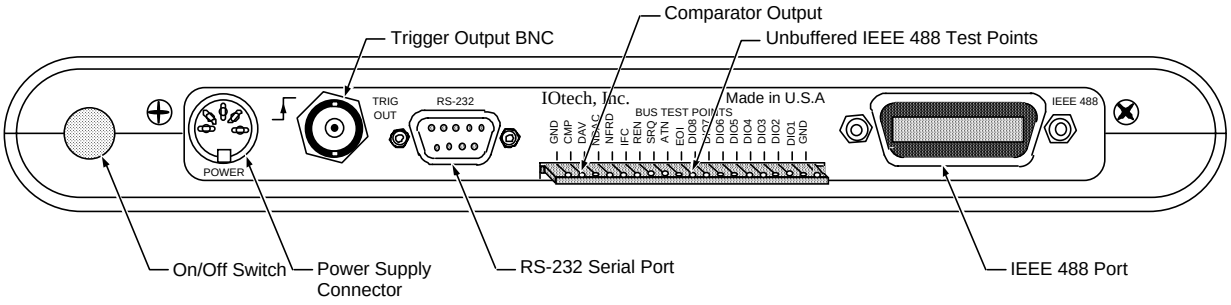
The record memory of the Analyzer488 contains information about the state of the IEEE 488 bus for each bus event that occurred while recording was enabled. For each event, the state of the bus management lines and data lines is preserved. Also, status information is retained, including whether the event data is valid, whether the event is a trigger point, and if the event occurred during a bus error.

The Analyzer488 provides facilities for retrieving the bus records, searching for specific bus records and assigning logical addresses to the bus records. Part or all of the record memory may be printed using a serial or IEEE 488 printer. Additionally, the Analyzer488 provides options for erasing and resetting the record memory.

2.5 Rear Panel Description

The rear panel of the Analyzer488 gives access to bus signals and simple hook-up to peripherals. The following connectors are provided and described below:

View of the Analyzer488 Rear Panel



1. On/Off switch - Depress the power switch to apply power to the Analyzer488.

2. Power Supply connector - Plug the cable from the external power supply into this jack.

3. Trigger Output - When the trigger point event is recorded by the Analyzer488, the trigger output will transition from a low (0) to a high (1) logic level and stay high.

4. RS-232 Port - Connect a 9 pin RS-232 cable to this connector to use the Analyzer488 with a serial host or serial printer.

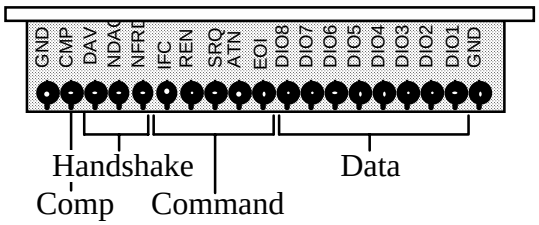
5. Edge Card Connector - All IEEE 488 bus lines are brought out to this connector for use with an oscilloscope or logic analyzer. Also a compare (CMP) output is provided that pulses whenever the trigger match pattern appears on the bus

6. IEEE 488 Connector - Connect a standard IEEE 488 cable to this connector to use the Analyzer488 with other bus devices.

2.5.1 Edge Card Connector

The edge card connector allows easy monitoring of particular IEEE 488 bus lines with an oscilloscope, logic probe, or logic analyzer. Also a compare (CMP) output is provided that pulses whenever the trigger match pattern appears on the bus. These signals are not buffered and are connected directly to the corresponding pins on the IEEE 488 connector.

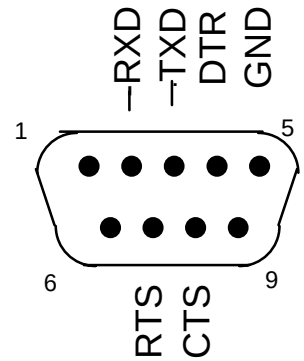
View of Edge Card Connector



2.5.2 Serial Port Description

The Analyzer488 is equipped with a standard DB-9S connector on its rear panel and requires a standard DB-9P mating connector. The Analyzer488 connector is configured as DCE type equipment for RS-232 communications, which means the Analyzer488 always transmits data on Pin 3 and receives data on Pin 2. The following paragraph lists and describes the RS-232 signals provided on the Analyzer488.

View of the Analyzer488 Serial Connector



-RxD Receive Data - Input - Pin 2
This pin accepts serial data sent by the RS-232 host. The serial data is expected with the word length, baud rate, stop bits and parity selected by the internal switches. The signal level is low true.

-TxD Transmit Data - Output - Pin 3
This pin transmits serial data to the RS-232 host. The serial data is sent with the word length, baud rate, stop bits and parity selected by the internal switches. The signal level is low true.

DTR Data Terminal Ready - Ouptut - Pin 4
The DTR output is connected directly to + 9 volts.

Gnd Ground - Pin 5
This pin sets the ground reference point for the other RS-232 inputs and outputs.

RTS Request To Send - Output - Pin 7
The RTS output is used as a hardware handshake line to prevent the RS-232 host from transmitting serial data if the Analyzer488 is not ready to accept it. When RTS/CTS handshake is selected, the Analyzer488 will drive the RTS output high when there are greater than 1000 character locations available in its internal buffer. If the number of available locations drops to less than 1000, the Analyzer488 will un-assert (low) this output. If XON/XOFF

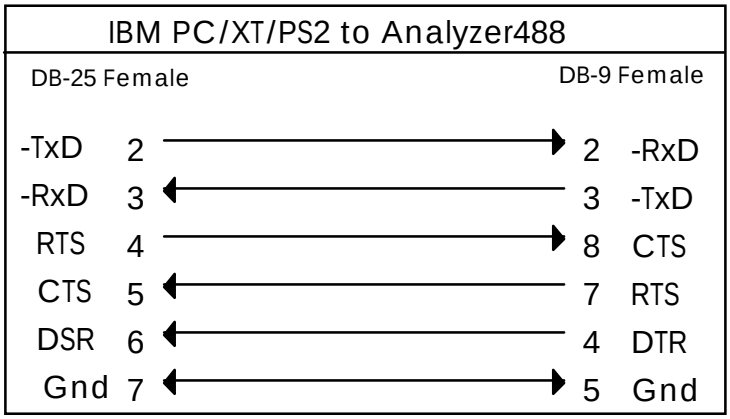
handshake is selected, the RTS line will be permanently driven active high.

CTS Clear To Send - Input - Pin 8
The CTS input is used as a hardware handshake line to prevent the Analyzer488 from transmitting serial data when the RS-232 host is not ready to accept it. When RTS/CTS handshake is selected on the internal switches, the Analyzer488 will not transmit data out - TxD while this line is un-asserted (low). If the RS-232 host is not capable of driving this line it can be connected to the Vtest output (Pin 6) of the Analyzer488. If XON/XOFF handshake is selected, the CTS line is not tested to determine if it can transmit data.

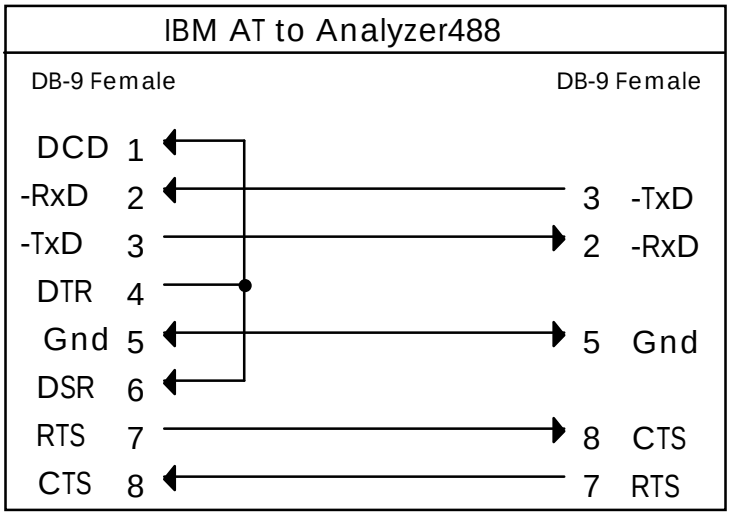
2.5.3 Serial Cable Wiring Diagrams

If a cable was not purchased with the interface, the following diagrams will be helpful in making your own cable. Simple soldering skills and an attention to detail will ensure successful construction. Refer to the Serial Controller section of this manual for complete information on using the Analyzer488 with serial devices.

IBM PC/XT/PS2 to Analyzer488 Wiring Diagram



IBM AT to Analyzer488 Wiring Diagram



Note: Standard AT 9 Pin to 25 Pin adapter cables are not wired as shown above and will not work with the Analyzer488. Order IOtech Part Number CA-47.

WARNING

The Analyzer488 makes its earth ground connection through the serial interface cable. It should only be connected to IEEE bus devices after being first connected to the host. Failure to do so may allow the Analyzer488 to float to a bus device test voltage. This could result in damage to the interface, personal injury or death.

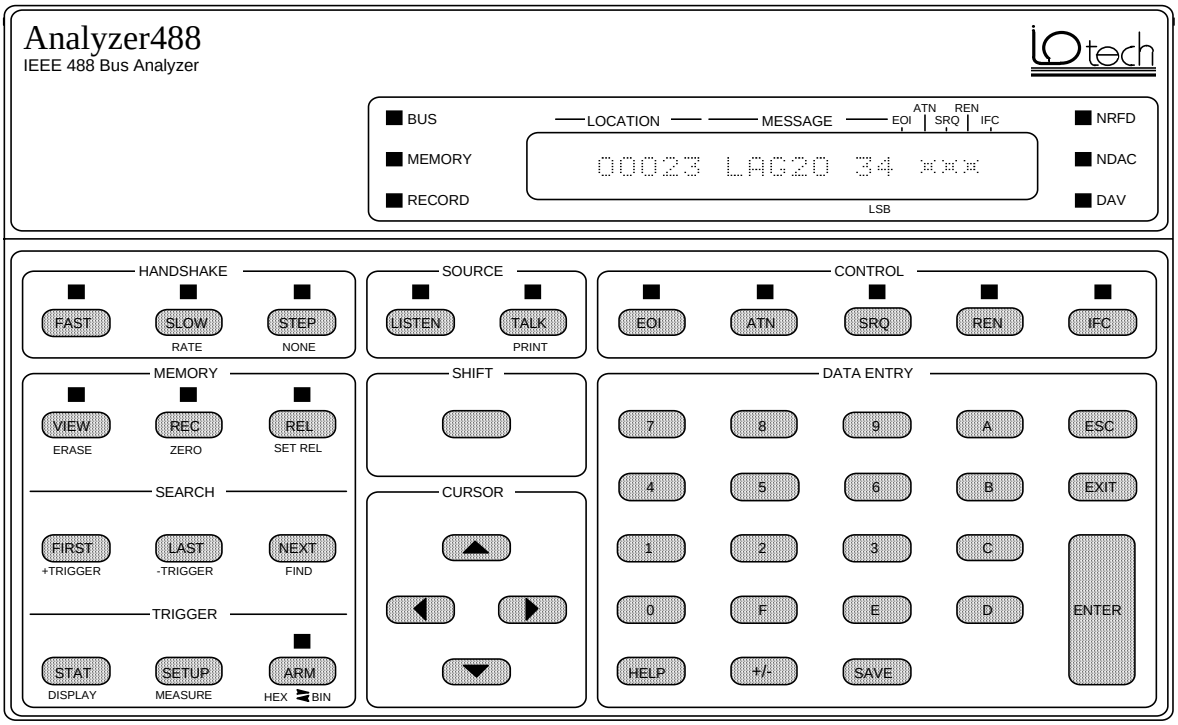
Front Panel Operation

3.1 Introduction

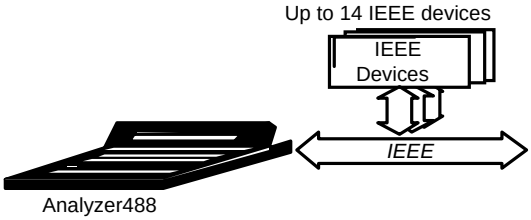
This section contains a detailed explanation of the front panel operation of the Analyzer488 IEEE 488 Bus Analyzer. It contains information regarding setup, configuration, an explanation of how to use the front panel, and a detailed description of all the keys and LED indicators.

3.2 Description

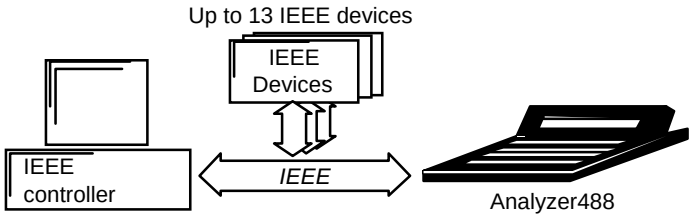
Front panel operation of the Analyzer488 provides simple, portable, bench-top access to all of the Analyzer488's features. Two display modes, bus display mode and memory view mode, show the state of the IEEE 488 bus or the contents of record memory. The front panel display shows bus messages and bus data in either hexadecimal and ASCII format or in binary format. Front panel LED indicators show bus signal states and Analyzer488 states at a glance. Command keys, menu keys, and edit keys provide functions to monitor, record, and analyze bus activity.



3.3 Setup



To setup the Analyzer488 for front panel operation, connect it to the IEEE 488 bus through the IEEE 488 bus connector. In front panel operation, the Analyzer488 can operate as a low-level system controller or as a monitor to view bus events.



3.4 Configuration

Configuration of the Analyzer488 is accomplished at power up. Once configured, the configuration parameters are stored in nonvolatile RAM and used as the default configuration when the Analyzer488 is powered on. To configure the Analyzer488 for front panel operation, press the [SETUP] key while turning on the power. Menus on the front panel display show the configuration options:

Factory Config? No

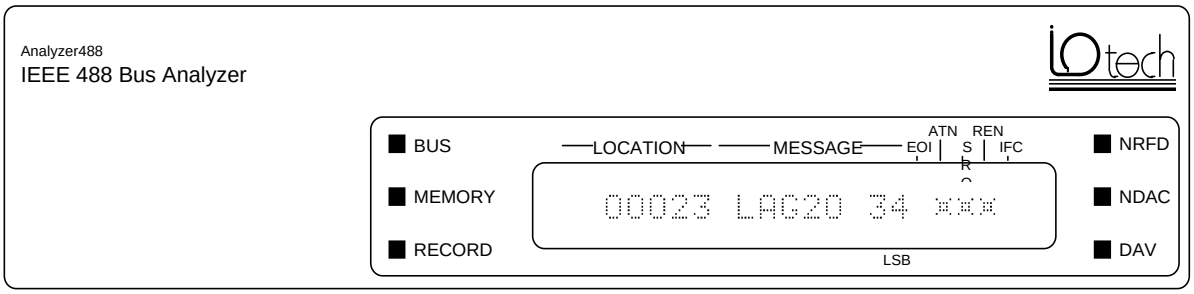
This menu is used to reset factory default values for all parameters or to skip reset to change only a few. Press the [↑] or [↓] cursor keys to select Yes or No. Press [ENTER] to choose.

Config? Front Panel

This menu prompts for the default operating mode of the Analyzer488. Press [ENTER] to select front panel operation.

3.5 Display Panel

The display panel of the Analyzer488 contains a 20 character alphanumeric vacuum fluorescent display and LED indicators. The display panel shows the state of the bus, the contents of record memory, and status, help and other messages. LED indicators show the display mode and the state of the handshaking lines.



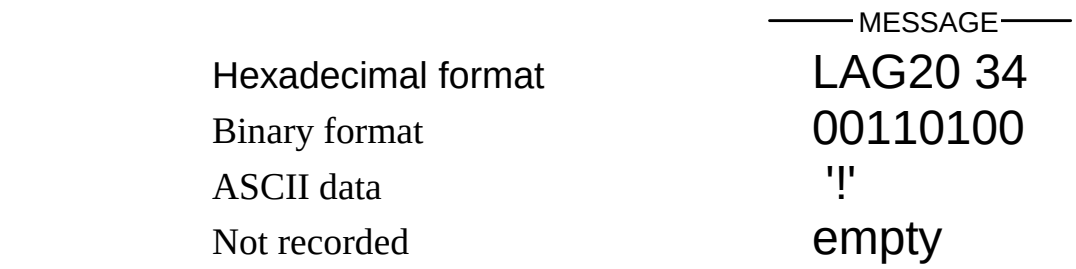
3.5.1 Alphanumeric Display

-LOCATION- The -LOCATION- area shows either record memory location addresses when in the memory view mode (MEMORY LED on), or the message BUS when in the bus display mode (BUS LED on). A T in the first column means that the location is a trigger point. A + or - means that relative addressing format is used. Note that when recording (RECORD LED on), the -LOCATION- area might change. See paragraph 3.7 for examples.

	—LOCATION—
address in the record memory	00008
T indicates a trigger point	T00055
+ indicates relative addressing	+00001
- indicates relative addressing	-00002
bus indicates bus display mode	BUS

-MESSAGE- The -MESSAGE- area displays bus data in bus message

and hexadecimal format, or in binary format. -MESSAGE- also shows ASCII data and if the record memory location has been used. If hexadecimal format is used, the IEEE 488 bus command and hexadecimal value are shown. If binary format is used, the state of each data line is shown with a 1 if asserted, and a 0 if unasserted. Flashing digits mean that the actual state of the data lines is different from the driven state. See paragraph 3.7 for examples.



LSB — The Least Significant Bit marker, LSB, shows the least significant bit of binary data when binary data format is selected.

EOI — The EOI digit indicates the actual state of the End or Identify (EOI) bus management line. It does not necessarily indicate the driven state. A ¤ indicates that the line is asserted, blank means that the line is unasserted.

ATN — The ATN digit indicates the actual state of the Attention (ATN) bus management line. It does not necessarily indicate the driven state. A ¤ indicates that the line is asserted, blank means that the line is unasserted.

SRQ — The SRQ digit indicates the actual state of the Service Request (SRQ) bus management line. It does not necessarily indicate the driven state. A ¤ indicates that the line is asserted, blank means that the line is unasserted.

REN — The REN digit indicates the actual state of the Remote

Enable (REN) bus management line. It does not necessarily indicate the driven state. A ¤ indicates that the line is asserted, blank means that the line is unasserted.

`IFC` The `IFC` digit indicates the actual state of the Interface Clear (IFC) bus management line. It does not necessarily indicate the driven state. A ¤ indicates that the line is asserted, blank means that the line is unasserted.

3.5.2 Front Panel LED Indicators

LED indicators on the front panel show the current display mode, a recording ON LED, and the state of the three handshaking lines.

`BUS` The `BUS` LED indicates that the Analyzer488 is in the bus display mode, displaying information directly from the IEEE 488 bus. Changing the display mode is accomplished by using the `[VIEW]` key.

`MEMORY` The `MEMORY` LED indicates that the Analyzer488 is in the memory view mode, displaying information from the record memory. Changing the display mode is accomplished by using the `[VIEW]` key.

`RECORD` The `RECORD` LED indicates that the Analyzer488 is recording bus events into the record memory.

`NFRD` The `NFRD` LED indicates the logical state of the Not Ready for Data (NRFD) handshaking line. The LED is on when the NFRD line is asserted. This LED indicates the real-time state of the bus, even when viewing recorded data.

`NDAC` The `NDAC` LED indicates the state of the Not Data Accepted (NDAC) handshaking line. The LED is on when the NDAC line is asserted. This LED indicates the real-time state of the bus, even when viewing recorded data.

DAV The DAV LED indicates the state of the Data Valid (DAV) handshaking line. The LED is on when the DAV line is asserted. This LED indicates the real-time state of the bus, even when viewing recorded data.

3.5.3 Bus Display Mode

Bus display mode is the default display mode of the Analyzer488. In normal operation, with the BUS LED on, the -LOCATION- area of the front panel display shows the word: BUS, followed by bus data in the -MESSAGE- area of the display. When recording events (RECORD LED is on), -LOCATION- shows the next record memory location that will be used to record bus events. The -MESSAGE- area still shows the actual state of the bus data lines.

When the Analyzer488 is in the Talk state (TALK LED is on), the actual state of the data lines shown in the -MESSAGE- area may not be the same as the driven state, shown by the keyboard LED indicators. If the actual state of the data lines is different from the driven state, the bus data in the -MESSAGE- area will flash. If hexadecimal format is used, the hexadecimal value will flash. If binary format is used, the digits that represent the individual lines that are different will flash.

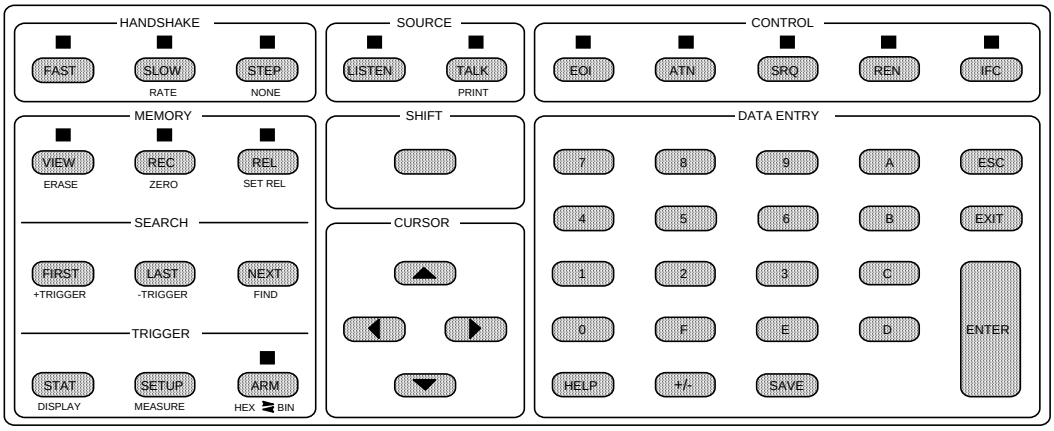
3.5.4 Memory View Mode

Memory view mode is used to examine record memory. In normal operation, with the MEMORY LED on, the -LOCATION- area shows a record memory location address and corresponding bus data in the -MESSAGE- area. When recording events (RECORD LED is on), -LOCATION- does not change, recording goes on in the background. If the current memory location is overwritten, the -MESSAGE- area will change to show the new bus event that was recorded into the location.

3.6 Keyboard

The front panel keyboard contains 46 pushbutton keys and 14 LED indicators. Command keys provide immediate actions, menu keys give menu

selections, and edit keys let you enter and change data selections. Keystroke macros can be saved and recalled. Keyboard LED indicators show handshaking settings, record memory functions, data source state, and the driven state of the bus management lines.



3.6.1 Using the Keyboard

Most keys are labeled with a symbol, `[EOI]`, or a command, `[REC]`. Some keys have labels on the front panel, below the keys `[ZERO]`. To select one of these keys, first press `[SHIFT]`. The front panel display will show the message: Shift?, indicating that the Analyzer488 is waiting for another keypress. Press the key to complete the selection. Pressing `[ESC]`, `[EXIT]`, `[ENTER]`, or `[SHIFT]` will abort the selection. Pressing an unlabeled key will result in an error message: Invalid Key Press.

3.6.2 Using Command Keys

Command keys, like `[EOI]`, `[REC]`, or `[SET REL]`, are immediate action keys, displaying results on the front panel display or with LED indicators. Press these keys for the intended action.

3.6.3 Using Menu Keys

Menu keys, like `[SETUP]` or `[STAT]`, use front panel menus for data input or information display. Menu keys such as `[SETUP]`,

[FIND], [PRINT], [RATE], and [DISPLAY] use a sequence of menus that display information and require data input. [STAT] and [MEASURE] display information menus. The [ERASE] and [ZERO] commands use a confirmation prompt. [SHIFT], [HELP], and [SAVE] are special purpose menu keys. Refer to the Key Description section for a full description of all the keys and menus.

3.6.4 Using Edit Keys

Data input fields use the -DATA ENTRY- keys and the -CURSOR- keys to enter parameters. The hexadecimal data entry keys are used to input numeric data. [↑] and [↓] are used to scroll between list items, such as baud rates, bus management line states, (X - don't care, 0 - unasserted, and 1 - asserted), hexadecimal bytes (automatically changing corresponding messages), and numeric data, such as record memory locations or count parameters. [←] and [→] are used in [SETUP] and [FIND] to move between fields.

Menus can be accepted using [ENTER] or [EXIT] and are canceled using [ESC]. [ENTER] is used to accept data and display the next menu in sequence, if one is used. [EXIT] leaves the menu immediately and accepts data in the displayed menu only. Pressing [EXIT] does not change any parameters in following menus. Pressing [ESC] aborts menu selections and does not change any menu parameters.

3.6.5 Saving and Recalling Keystroke Macros

Up to sixteen keystroke macros can be saved and recalled using the sixteen hexadecimal keys, [0] - [F], and the [SAVE] key. Keystroke macros can be used to save any combination of front panel commands. Keystroke macros can recall repetitive actions, complicated setups, and even simulate IEEE 488 high-level bus commands using the low-level front panel commands. Keystroke macros can call other keystroke macros and can even call itself to loop through repetitive actions.

To save keystrokes into a keystroke macro, press [SAVE]. A message: Save Macro? prompts for a keystroke macro name, one of the

16 hexadecimal keys, [0] to [F]. Press one of the hexadecimal keys to name the keystroke macro. The message: Saving to Macro n means that keystrokes are being saved into macro n. Enter the keystrokes, and press [SAVE] to stop storing keystrokes. The message End Saving Keys indicates that the keystroke macro is saved. Pressing [ESC], [EXIT], or [ENTER] will not abort saving keystrokes. If pressed, those keystrokes will be added to the keystroke macro.

When keystrokes are saved, some of the Analyzer488 control states are also stored. The state of the control bits, the handshaking configuration, and the display mode (bus display or memory view) and format (hexadecimal or binary) are saved.

To recall keystroke macros, press [SHIFT], and at the SHIFT? prompt, enter the hexadecimal key that names the keystroke macro. The message: Recalling from Macro n means that the keystrokes stored in macro n are controlling the Analyzer488. Pressing [ESC] will stop the keystroke macro. A message: End of Macro indicates that all keystrokes have been recalled and the macro is finished.

When a keystroke macro is recalled, the Analyzer488 first sets the display mode, the handshaking configuration, and the state of the control bits that were stored when the macro was saved.

3.6.6 Getting Help

Help messages for each key can be viewed on the front panel display. Press [HELP] followed by a key to see a help message for that key. Context sensitive help is also available. Pressing [HELP] at a menu will show a message that describes each data input or information field. The help message scroll rate can be changed using the [DISPLAY] command.

3.7 Using the Analyzer488 on the IEEE 488 bus

Following are some examples of front panel operation. The first examples use the Analyzer488 with an IBM PC and an IOtech Personal 488 PC/IEEE Controller using the KYBDCTRL keyboard controller program.

The final examples use the Analyzer488 as an active controller connected to a Keithley Instruments Model 196 System Digital Multimeter.

3.7.1 Controlling Bus Management and Bus Data Lines

Run the KYBDCTRL keyboard controller program, and turn the Analyzer488 on. The front panel display will show a typical bus display:

The front panel [EOI], [ATN], [SRQ], [REN], and [IFC] keys are used to assert and unassert bus management lines. The corresponding LED indicators show whether the Analyzer488 is asserting (LED on) or unasserting (LED off) the lines. The EOI, ATN, SRQ, REN, and IFC digits on front panel display show the actual state of the bus management lines. A • indicates that the line is asserted and a blank indicates that it is unasserted.

The driven state, as shown by the keyboard LED indicators, and the actual state, as shown by the digits on the front panel display, might not be the same. If, for instance, the Analyzer488 is trying to unassert a line (ATN LED off) and the controller is asserting the Attention line, the LED indicator for that line will be off, but the front panel digit for ATN will be •, showing that the line really is asserted.

The -MESSAGE- area on the front panel display shows the state of the bus data lines. When the Analyzer488 is in the Talk state (TALK LED is on), bus data lines are driven by the hexadecimal keys, [0] - [F], and the [↑] and [↓] cursor keys. The actual bus state shown in the -MESSAGE- area may not be the same as the driven state. For instance, two devices may be driving the bus at the same time. If the actual bus state is different from the driven state, the bus data in the -MESSAGE- area will flash. If hexadecimal format is used, the hexadecimal value will flash. If binary format is used, the digits that represent the individual lines that are different will flash.

3.7.2 Viewing Bus Events

Press `[LISTEN]` to set Listen state — The `LISTEN` LED will light

Press `[SLOW]` to set slow handshaking — The `SLOW` LED will light.

> (Note that slow handshaking speed is set to 3 bytes/sec at the factory. It can be changed using `[RATE]`.)

Enter the following commands to the keyboard controller:

```
CMD> timeout 0
CMD> term out cr
CMD> output 00;test
```

The display will show:

LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
BUS	TAG21	55	¤			¤	¤
BUS	UNL	3F	¤			¤	¤
BUS	LAG00	20		¤		¤	¤
BUS	't'	74		¤			
BUS	'e'	65		¤			
BUS	's'	73		¤			
BUS	't'	74		¤			
BUS	CR	0D	¤	¤			
BUS	ACG13	0D		¤		¤	¤

Bus events are displayed as they happen; Eight events occurred on the bus. The ninth display shows that the controller asserted Attention after the "test" message. `ACG13` is the bus message for the multiline command: `&h0D` (ASCII CR).

Press [STEP] to set handshaking to step. — The STEP LED will light.

(Note that pressing either [ENTER] or [STEP] will perform a handshake when STEP handshaking is selected.)

```
                                                  ATN  REN
               —LOCATION—   —MESSAGE—   EOI | SRQ |  IFC
CMD> output 00;test   BUS     TAG21  55   ¤    ¤
```

This display shows that one event has taken place. Pressing [STEP] to set handshaking also performs a single handshake. Press [STEP] 8 times to step through and display the same events as shown above.

Press [FAST] to set handshaking to fast. — The FAST LED will light.

```
                                                  ATN  REN
               —LOCATION—   —MESSAGE—   EOI | SRQ |  IFC
CMD> output 00;test   BUS    ACG13 OD   ¤    ¤
```

The same events occurred on the bus, but the the transactions were too fast to view.

Press [NONE] to turn handshaking off. — All handshaking LEDs will be off.

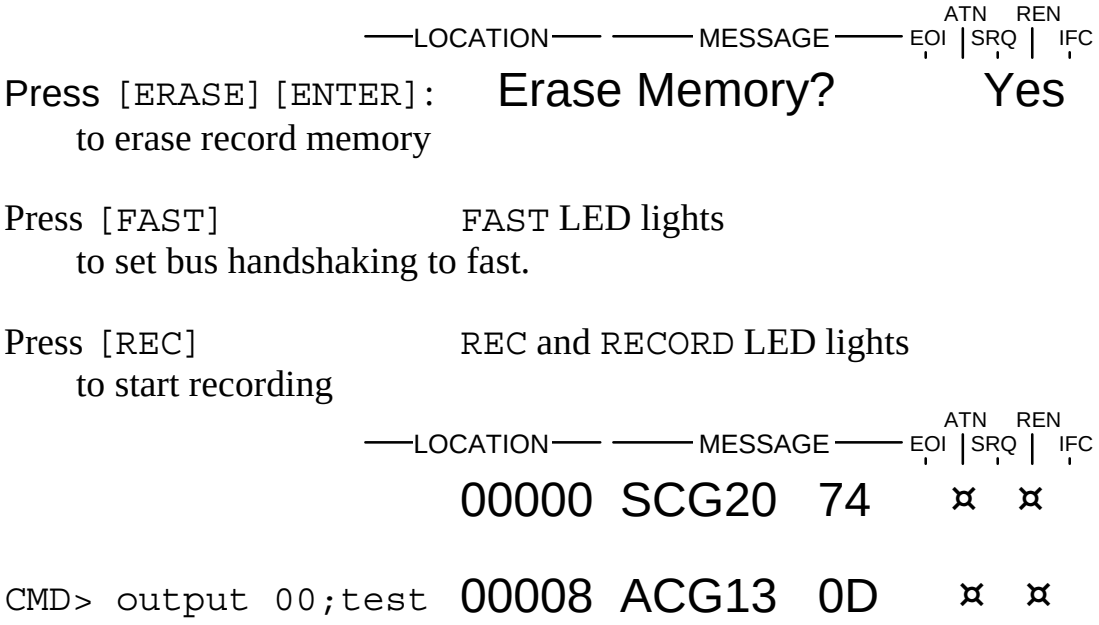
```
                                                  ATN  REN
               —LOCATION—   —MESSAGE—   EOI | SRQ |  IFC
CMD> output 00;test   BUS    SCG20  74    ¤    ¤
```

ERROR 09: TIME OUT OR BUS ERROR ON WRITE

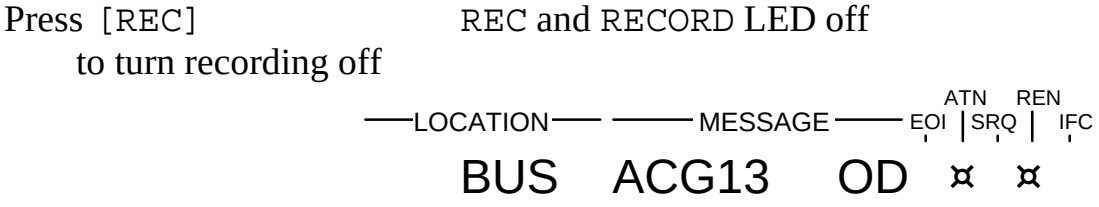
An error occurred because no listener was present. The controller asserted Attention to talk and presented 't' on the bus. The bus message, SCG20, is the multiline command, &h74, the hex value of ASCII 't'.

3.7.3 Recording Bus Events

Use the same keyboard controller program to record a test sequence of events on the bus.



This display means that eight events were recorded. The controller asserted Attention after the "test" message and `ACG13` is the bus message for the multiline command `&h0D` (ASCII CR).



This display shows that the Analyzer488 is again in the bus display mode, `&h0D` is on the bus, and the Attention and Remote Enable bus management lines are asserted.

3.7.4 Viewing Recorded Bus Events

To look at the events that were just recorded,

Press [VIEW] The VIEW and MEMORY LEDs will light

```
                                                    ATN  REN
──LOCATION── ─── MESSAGE ─── EOI │SRQ │ IFC
00000    TAG21   55  ¤   ¤
```

The first controller command, TAG21, is stored in record memory location 0.

```
                                                    ATN  REN
──LOCATION── ─── MESSAGE ─── EOI │SRQ │ IFC
00001  UNL      3F  ¤   ¤
00002 LAG00     20  ¤   ¤
00003  't'      74  ¤   ¤
00004  'e'      65      ¤
00005  's'      73      ¤
00006  't'      74      ¤
00007  CR       OD¤     ¤
00008 empty       - - -  -
```

Press [↑] eight times to see the rest:

3.7.5 Using Triggers

The following example shows how to set up a trigger, record some bus data, view the record memory, search the record memory, and review recording statistics. To set up a trigger point, press [SETUP] to configure the trigger parameters:

```
                                                    ATN  REN
──LOCATION── ─── MESSAGE ─── EOI │SRQ │ IFC
Match? LAG00 20X1XXX
```

Press [→] [2] [0] [→] [1] [ENTER] to match &h20 (LAG00)

with ATN asserted,
and continue

	LOCATION	MESSAGE	EOI	ATN	SRQ	REN	IFC
Press [2][ENTER]	# of Matches?	00002					
to trigger on the second match							
Press [6][ENTER]	Trig Delay?	0000006					
to delay 6 events after the trigger point							
Press [2][2][ENTER]	Post Count?	00022					
to capture 22 events after the trigger point							
Press [ENTER]	When Completed? Stop						
to stop recording when the trigger sequence is completed							

Press [FAST] — FAST LED lights
to set handshaking to fast

Press [ARM] — ARM LED lights
to enable the trigger system

Press [REC] — REC LED lights, ARM LED starts to flash
to start recording

Enter the following controller commands at the PC keyboard:

```
cmd> output 01;Analyzer488 trigger test
cmd> output 00;Pretrigger data
cmd> output 00;Here !
cmd> output 00;Post trigger data
```

The ARM LED stops flashing when the trigger point is found (!). It remains on until the post trigger sequence is complete. When the trigger sequence is finished, both the REC and ARM LED turn off.

3.7.6 Viewing the Triggered Bus Events

	LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
Press [VIEW] to view record memory	00000	TAG21	55	¤			¤	
Press [FIRST] to view the first location	00000	TAG21	55	¤			¤	
Press [+TRIGGER] to view the trigger point	T00055	'!'	21	¤				
Press [LAST] to view the last location	00077	CR	0D¤	¤				
Press [FIRST] to return to the first location	00000	TAG21	55	¤			¤	

Press [↑] and [↓] to view the bus events:

3.7.7 Using Relative Addressing

	LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
Press [LAST] to view the last location	00077	CR	0D¤	¤				
Press [-TRIGGER] to find the trigger point	T00055	'!'	21	¤				
Press [SET REL] to set the location pointer	Reference Set							
Press [REL] to select relative addressing	T00000	'!'	21	¤				
Press [↑] to advance one location	+00001	CR	0D¤	¤				
Press [↓] [↓] to retreat two locations	-00001	' '	20	¤				

3.7.8 Reviewing Recording Status

	LOCATION	MESSAGE	ATN EOI	SRQ	REN IFC
Press [STAT]	Total Count	00000078			
Then press [ENTER]	Recorded	00078			
7 times to see all the	Pre Count	00000055			
statistics	Pre Recorded	00055			
	Post Count	00000022			
	Post Recorded	00022			
	Trigger	00055			
	First	00000			
	Last	00077			

3.7.9 Searching Record Memory

	LOCATION	MESSAGE	ATN EOI	SRQ	REN IFC
Press [FIRST] to find the first location	-00055	TAG21 55	¤		¤
Press [FIND] [→] [→] [3] [F] [→] [↑] [↑] to setup to find UNL with ATN asserted	Find? +	UNL 3FX1XXX			
Press [ENTER] to find the first UNL	-00054	UNL 3F	¤		¤
Press [NEXT] to find the second UNL	-00026	UNL 3F	¤		¤
Press [NEXT] to find the third UNL	-00007	UNL 3F	¤		¤

	LOCATION	MESSAGE	ATN EOI	SRQ	REN IFC
Press [LAST] to find the last location	+00022	CR	0D¤	¤	
Press [FIND][+/-] to search backwards	Find? -	UNL	3FX1XXX		
Press [ENTER] to find the last UNL	+00003	UNL	3F ¤	¤	
Press [NEXT] to continue the backwards search	-00007	UNL	3F ¤	¤	

3.7.10 Printing Record Memory

Record memory can be printed to a serial or IEEE 488 printer. Refer to the [PRINT] key description and configure the Analyzer488 for your printer.

To print the last sequence of events:

Press [PRINT][EXIT] Printing...
to print the record memory

3.7.11 Using Low-Level Controller Commands

The following two examples use the Analyzer488 as an active controller connected to a Keithley Instruments Model 196 System Digital Multimeter (DMM). The Analyzer488 will address the DMM to listen, issue a setup command, take a reading, and record it.

Set the IEEE 488 bus address of the DMM to 16. Turn the DMM on and short out the voltage input leads. Connect the DMM to the Analyzer488 rear panel IEEE 488 connector. Turn the Analyzer488 on.

	LOCATION	MESSAGE		ATN EOI	SRQ	REN IFC
LISTEN LED on	BUS	NUL	00			
Press [STEP][ATN]	BUS	ACG00	00	¤		
STEP and ATN LED on						
[REN][TALK]	BUS	ACG00	00	¤	¤	
TALK and REN LED on						
to make the Analyzer488 the active controller						
Press [IFC]	BUS	IFC	00	¤	¤¤	
to clear the DMM bus interface						
the IFC LED and front panel digit will flash						
Press [3][0][ENTER]	BUS	LAG16	30	¤	¤	
or [3][0][STEP]						
to address the DMM to listen						
The RMT and LSN LEDs on the DMM will light						

(Note that pressing either [ENTER] or [STEP] will perform a handshake when STEP handshaking is selected.)

	LOCATION	MESSAGE		ATN EOI	SRQ	REN IFC
Press [ATN]	BUS	'0'	30		¤	
to release the Attention line						
Press [4][6][ENTER]	BUS	'F'	46		¤	
to send 'F'						
Press [3][0][ENTER]	BUS	'0'	30		¤	
to send '0'						
Press [5][2][ENTER]	BUS	'R'	52		¤	
to send 'R'						

	LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
Press [3][0][ENTER]	BUS	'0'	30	¤				
to send '0'								
Press [5][8][ENTER]	BUS	'X'	58	¤				
to send 'X'								

The AUTO and DCV LEDs on the DMM will light.

Press [ATN]	BUS	TAG24	58	¤	¤		
to assert the Attention line							
and become the active controller							
Press [5][0][ENTER]	BUS	TAG16	50	¤	¤		
to address the DMM to talk							

The LSN LED will go off and the TLK LED will light on the DMM.

Press [LISTEN][ATN]	BUS	'N'	4E	¤			
to become a listener							

The DMM will present data on the bus and all handshake LEDs will be lit.

Press [STEP] or [ENTER]	BUS	'D'	44	¤			
18 times to see the reading	BUS	'C'	43	¤			
	BUS	'V'	56	¤			
	BUS	'-'	2D	¤			
	BUS	'0'	30	¤			
	BUS	'0'	30	¤			
	BUS	'0'	30	¤			
	BUS	'.'	2E	¤			
	BUS	'0'	30	¤			

LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
BUS	'1'	31					¤
BUS	'2'	32					¤
BUS	'9'	39					¤
BUS	'E'	45					¤
BUS	'-'	2D					¤
BUS	'3'	33					¤
BUS	CR	0D					¤
BUS	LF	0A¤					¤

Your actual voltage reading may vary. On the last step, EOI is asserted to indicate the end of reading, and the DMM continues to present data on the bus. Press `[SLOW]` to read the data at 3 bytes per second.

	LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
Press `[ERASE][ENTER]` to erase record memory	Erase Memory?							Yes
Press `[REC]` to start recording	00000	'-'	2D					¤
Wait a few seconds and								
Press `[REC]` to stop recording	BUS	'-'	2D					¤
Press `[ATN]` to regain control	BUS	ACG00	00	¤				¤
Press `[STEP][TALK]` to become a talker	BUS	TAG16	50	¤				¤
Press `[5][F][ENTER]` to stop the DMM from talking `TLK` LED goes off on the DMM	BUS	UNT	5F	¤				¤

	LOCATION	MESSAGE	EOI	ATN	SRQ	REN	IFC
Press [VIEW]	00000	'-'	2D		¤		
to view the record memory							
Press [↑] and [↓]	00001	'0'	30		¤		
to scroll through the record memory							

3.7.12 Using Keypress Macros

The following example uses the [SAVE] key to save and recall a keypress macro to perform the previous sequence of operations. Use the same setup configuration as the last example.

	LOCATION	MESSAGE	EOI	ATN	SRQ	REN	IFC
LISTEN LED on	BUS	NUL 00					
Press [SAVE][7]	Save Macro?						
to start saving macro 7	Saving to Macro						7
After each key is pressed,	Key 001 in Macro						7
a message is displayed							
Press [STEP][ATN]	BUS	ACG00 00			¤		
STEP and ATN LED on							
Press [REN][TALK]	BUS	ACG00 00		¤	¤		
TALK and REN LED on							
to make the Analyzer488 the active controller							
Press [IFC]	BUS	IFC 00		¤	¤	¤	
to clear the DMM bus interface							
The IFC LED and front panel digit will flash							

	LOCATION	MESSAGE		ATN EOI	SRQ	REN IFC
Press [3][0][ENTER] or [3][0][STEP] to address the DMM to listen	BUS	LAG16	30	¤		¤
The RMT and LSN LEDs on the DMM will light						
(Note that pressing either [ENTER] or [STEP] will perform a handshake when STEP handshaking is selected.)						
Press [ATN] to release the Attention line and become a talker	BUS	'0'	30	¤		
Press [4][6][ENTER] to send 'F'	BUS	'F'	46	¤		
Press [3][0][ENTER] to send '0'	BUS	'0'	30	¤		
Press [5][2][ENTER] to send 'R'	BUS	'R'	52	¤		
Press [3][0][ENTER] to send '0'	BUS	'0'	30	¤		
Press [5][8][ENTER] to send 'X'	BUS	'X'	58	¤		
The AUTO and DCV LEDs on the DMM will light.						
Press [ATN] to assert the Attention line and become the active controller	BUS	TAG24	58	¤		¤
Press [5][0][ENTER] to address the DMM to talk	BUS	TAG16	50	¤		¤

The LSN LED will go off and the TLK LED will light on the DMM.

	LOCATION	MESSAGE		EOI	ATN	SRQ	REN	IFC
Press [LISTEN] [ATN] to become a listener	BUS	'N'	4E				¤	

The DMM will present data on the bus and all handshake LEDs will be lit.

Press [STEP] or [ENTER]	BUS	'D'	44				¤	
18 times	BUS	'C'	43				¤	
	BUS	'V'	56				¤	
	BUS	'-'	2D				¤	
	BUS	'0'	30				¤	
	BUS	'0'	30				¤	
	BUS	'0'	30				¤	
	BUS	'.'	2E				¤	
	BUS	'0'	30				¤	
	BUS	'1'	31				¤	
	BUS	'2'	32				¤	
	BUS	'9'	39				¤	
	BUS	'E'	45				¤	
	BUS	'-'	2D				¤	
	BUS	'3'	33				¤	
	BUS	CR	0D				¤	
	BUS	LF	0A¤				¤	
Press [ATN] [TALK] to become the active controller	BUS	TAG16	50	¤			¤	

	LOCATION	MESSAGE	EOI	ATN	SRQ	REN	IFC
Press [5][F][ENTER] to unaddress the DMM	BUS	UNT 5F	¤			¤	
Press [SAVE] to end the macro definition	End Saving Keys						
Press [ERASE][ENTER] to erase record memory	Erase Memory?	Yes					
Press [REC] to start recording	00000	UNT 5F	¤			¤	
Press [SHIFT][7] to recall the macro	Recall From Macro	7					
When the macro is finished a message is flashed	End Macro						
Press [REC] to stop recording	00027	UNT 5F	¤			¤	
Press [VIEW] to view the record memory	00000	'-' 2D	¤				
Press [↑] and [↓] to scroll through the record memory	00001	'0' 30	¤				

3.8 Key Descriptions

This section contains a detailed explanation of each of the keys and LED indicators on the Analyzer488 front panel keyboard.

3.8.1 Keyboard functions

The keys on the front panel keyboard are divided into 7 groups: Data Entry, Control, Cursor, Shift, Source, Memory, and Handshake.

-DATA ENTRY- The Data Entry keys are used for making menu selections, entering numeric data into display fields, saving keystroke macros, and getting help.

-CURSOR- The Cursor keys are used to scroll through menu selections and display fields.

-SHIFT- The Shift key is used with other keys to select shift keys and to recall keystroke macros.

-MEMORY- The Memory keys are used for IEEE 488 bus event recording, analysis, and display functions.

-HANDSHAKE- The Handshake keys are used to control the speed of the IEEE 488 bus data transfers.

-SOURCE- The Source keys are used to toggle the Analyzer488 between the Listen state and Talk state.

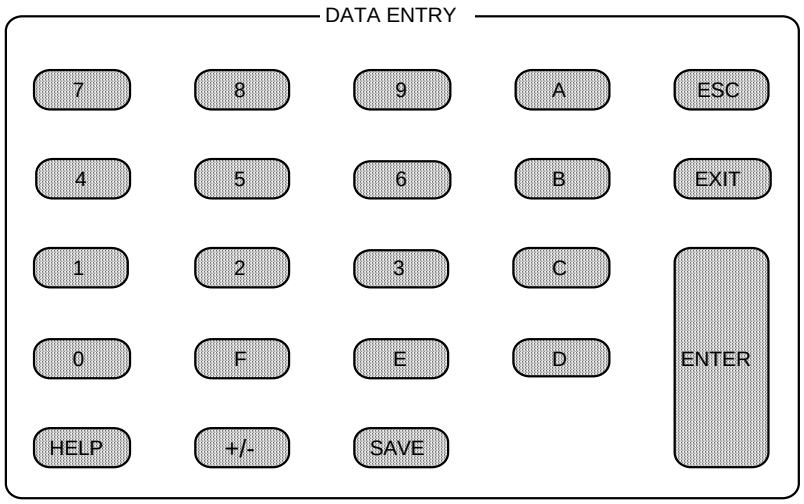
-CONTROL- The Control keys are used to control and monitor the IEEE 488 bus management lines.

3.8.2 Keyboard Keys and LEDs

The following pages contain a description of each key and LED indicator on the front panel of the Analyzer488.

Data Entry Keys

The Data Entry keys are used for data entry and other miscellaneous operations.



`[0] - [F]` The sixteen hexadecimal keys are used as general purpose data entry keys. These keys allow for the direct entry of numeric values into numeric display fields. When in the memory view mode, `[0] - [9]` are used to specify record memory locations. When in the bus display mode and in the talk state, `[0] - [F]` output data on the bus data lines. When in the bus display mode and in the listen state, `[0] - [F]` store data that will be output when the Analyzer488 is put into the talk state.

`[ENTER]` The Enter key, `[ENTER]`, is used within menus to choose or accept menu selections and, if more menus are provided, advance to the next menu. When in the memory view mode, `[ENTER]` is used to enter record memory location addresses. When in the bus display mode and `[STEP]` handshaking is selected, `[ENTER]` is used to force a source (talk state) or acceptor (listen state) handshake.

`[EXIT]` The Exit key, `[EXIT]`, is used like `[ENTER]` to choose menu selections but, if more menus are available, `[EXIT]` leaves the current menu with unselected menus unchanged. Changes made in menus are accepted when `[EXIT]` is used.

[ESC] The Escape key, [ESC], is used to abort or cancel menu selections. [ESC] escapes a menu immediately without proceeding to the menu's conclusion. Changes within the menu are not accepted when [ESC] is used.

[+/-] The Sign key, [+/-], is used as a general purpose data entry key. The [+/-] key is used to change the sign of numeric values, to indicate a search direction using [FIND], or to set a "don't care" (X) condition in [SETUP] and [FIND] menus.

[SAVE] The Save key, [SAVE] is used to record up to 126 key presses into one of sixteen keystroke macros. [SAVE], followed by a numeric key from [0] - [F], stores key presses until [SAVE] is pressed again, ending the macro. Recall key presses with [SHIFT] followed by the associated numeric key. Pressing [SAVE] displays a data input menu:

Save Macro? X

For each key press, a status message is displayed:

Saving to Macro X

Once [SAVE] is pressed again, ending the macro, a status message is displayed:

End Saving Keys

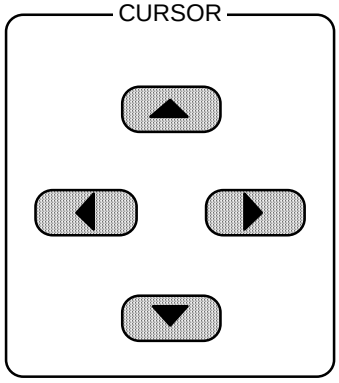
[HELP] The Help key, [HELP], is used to display information on the front panel display about individual keys or to give context sensitive help when the Analyzer488 is displaying menu screens. Pressing [HELP] shows a menu:

Select Key for Help

Press any key to see a help message for that key. Or press [HELP] while viewing a menu to see context sensitive help messages.

Cursor Keys

The cursor keys are used as edit keys to scroll through data or display fields and select or change functions and/or parameters.



[←] The left cursor key, [←], is used in menus to select display fields by moving the cursor to the left. It logically wraps if used on the first field of a display line. [←] also provides a backspace key for editing numeric entries.

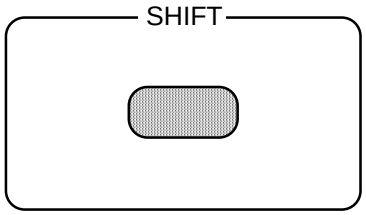
[→] The right cursor key, [→], is used in menus to select display fields by moving the cursor to the right. [→] logically wraps if used on the last field of a display line.

[↑] The up cursor key, [↑], when in the memory view mode, increments the record memory location on the front panel display. When in the bus display mode and in the talk state, [↑] increments the data value on the bus data lines. When in the bus display mode and in the listen state, [↑] increments the data value that will be output when the Analyzer488 is put into the talk state.

[↓] The down cursor key, [↓], when in the memory view mode, decrements the record memory location on the front panel display. When in the bus display mode and in the talk state, [↓] decrements the data value on the bus data lines. When in the bus display mode and in the listen state, [↓] decrements the data value that will be output when the Analyzer488 is put into the talk state.

Shift

The Shift key, [SHIFT], is used in conjunction with other keys to select commands. [SHIFT] is also used to perform a sequence of key presses previously saved in one of the sixteen keystroke macros.



[SHIFT][*key*] To select a command labeled below a key, first press [SHIFT]. The display will show a menu indicating that the Analyzer488 is waiting for a choice of commands:

Shift?

Press a key to complete the selection. Pressing [ESC] will abort the selection. If an invalid key is pressed, an error message is shown on the front panel display:

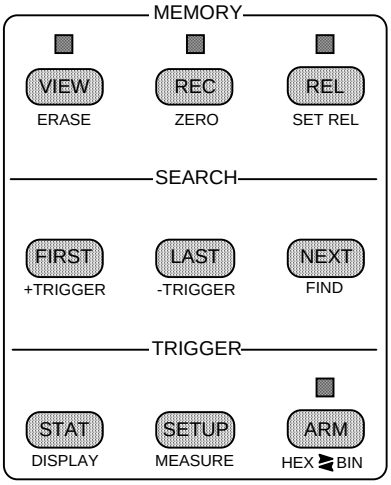
Invalid Key Press

When a menu is used, pressing [EXIT] will end menu selection for that command and accept any current parameters. Pressing [ENTER] proceeds to any following menus until all selections are made.

[SHIFT][*0-F*] To recall a keystroke macro, first press [SHIFT] and at the Shift? prompt, enter the hexadecimal key naming the keystroke macro that was saved. Refer to the [SAVE] key description for instructions on how to save keystroke macros.

Memory

The -MEMORY- keys are used for IEEE 488 bus event recording and analysis. They are used to setup a trigger point, arm the system, start recording, and to analyze the record memory. Menus provide recorded event statistics, time and rate measurements, and display functions.



The -MEMORY- keys are grouped into three areas:

-MEMORY-

[VIEW] The View key, [VIEW] is used to toggle the display format between bus display mode and memory view mode. To view record memory locations, use the -DATA ENTRY- keys or the -CURSOR- keys. See also the [REC] key description.

VIEW LED The VIEW LED indicates that the Analyzer488 is in the memory view mode, displaying information from the record memory. Changing the display mode is accomplished by pressing [VIEW].

[ERASE] The Erase command, [SHIFT][VIEW], erases the record memory and resets the reference pointer to 0. [ERASE] uses a confirmation menu:

Erase Memory? Yes

Verifies that the record memory should be erased, Yes or No. Use [↑] and [↓] to select. Accept value with [ENTER] and [EXIT] or cancel with [ESC]. A status message confirms that the record memory is erased:

Record Memory Erased

[REC] The Record key, [REC], is used to start or stop recording bus events. When used to start recording, [REC] resets the recording statistics seen using [STAT]. See also the descriptions of the [VIEW], [FIND], and [PRINT] keys.

REC LED The REC LED indicates the Analyzer488 is recording bus events. Recording may be disabled by pressing [REC].

[ZERO] The Zero command, [SHIFT][REC], resets the reference pointer to location 0 without erasing the record memory. Further recording will overwrite previous record memory locations. [ZERO] uses a confirmation menu:

Reset Memory? Yes

Verifies that the record memory should be reset, Yes or No. Use [↑] and [↓] to select. Accept value with [ENTER] and [EXIT] or cancel with [ESC]. A status message confirms that the record memory is reset:

Record Memory Reset

[REL] The Relative key, [REL], is used to toggle record memory location addressing format between absolute memory addressing and relative memory addressing. [REL] is valid only when viewing record memory. To use relative addressing, set the reference pointer with the Set Relative, [SET REL], command.

REL LED The REL LED indicates the Analyzer488 is in relative addressing

format. The REL LED can only be on when viewing record memory. Changing the addressing format is accomplished by pressing [REL]. Changing the record memory reference pointer is accomplished by using the Set Relative, [SHIFT][REL], command.

[SET REL] The Set Relative command, [SHIFT][REL], is used to set the current reference pointer. [SET REL] resets the reference pointer to the currently displayed record memory location, in either relative addressing or absolute addressing format. A status message confirms the action:

Reference Set

-SEARCH-

[FIRST] The First key, [FIRST], is used to search the record memory for the first location written in the record memory during the most resent recording operation. The record memory location, bus message, and management line status are displayed on the front panel display. [FIRST] is valid only when viewing record memory. If nothing was recorded, a status message is displayed:

Not Available

[+TRIGGER] The Next Trigger command, [SHIFT][FIRST], searches the record memory for the next trigger point. The record memory location, bus message, and management line status are displayed on the front panel display. [+TRIGGER] is valid only when viewing record memory. A status message is displayed while searching:

+Searching...

If no trigger is found, a status message is displayed:

Trigger Not Found

[LAST] The Last key, [LAST], is used to search the record memory for the

last location written in the record memory during the most resent recording operation. The record memory location, bus message, and management line status are displayed on the front panel display. [LAST] is valid only when viewing record memory. If nothing was recorded, a status message is displayed:

Not Available

[-TRIGGER] The Last Trigger command, [SHIFT][LAST], searches the record memory for the previous trigger point. The record memory location, bus message, and management line status are displayed on the front panel display. [-TRIGGER] is valid only when viewing record memory. A status message is displayed while searching:

+Searching...

If no trigger is found, a status message is displayed:

Trigger Not Found

[NEXT] The Next key, [NEXT], is used to search the record memory for the next occurrence of an event specified using [FIND]. The parameters specified for the Find command will be used in the search. If the event is found, the record memory location containing that event is displayed. [FIND] and [NEXT] operations will search the entire record memory until the event is found. A status message is displayed while searching:

+Searching...

If no match is found, a status message is displayed and the previously displayed location is displayed again:

Pattern Not Found

[FIND] The Find menu, [SHIFT][NEXT], is used to search the record

memory for the first occurrence of a specific bus event or bus error. Pressing [FIND] displays a menu that is used to specify search parameters and to start the search. The record memory is searched and the location, bus message, and management line status are displayed on the front panel display.

Find? + NUL XXXXXXX (hex)
Find? +XXXXXXXXXXXXX (bin)

Requests the bus pattern to find. Bus data and signal line patterns are specified as true (1), false (0) or don't care (X). When displaying hexadecimal data, changing the hexadecimal data byte automatically changes the corresponding bus message. Use the [→] and [←] cursor keys to move between input fields. Use the [↑] and [↓] cursor keys to set values. Pressing [↑] at the NUL bus message selects a bus error (ERROR). Accept entry with [ENTER] and [EXIT] or cancel with [ESC]. A status message is displayed while searching:

+Searching...

If no match is found, a status message is displayed:

Pattern Not Found

-TRIGGER-

[STAT] The Statistics key, [STAT], is used to display statistics of the last recording operation. When [STAT] is pressed, statistics are displayed in data menus. Use the [↑] and [↓] cursor keys or [ENTER] to step between statistics. Press [ESC] or [EXIT] to exit the information display menus. The following statistics are presented:

Total Count 00000000

The total number of bus events which occurred since [REC] was enabled. Total count can be from 0 to 99,999,999 events.

Recorded 00000

Number of bus events which have been recorded since [REC] was enabled. Recorded count can be from 0 to 32,768.

Pre Count 00000000

Total number of bus events that occurred prior to the trigger since [REC] was enabled. Pre count can be from 0 to 99,999,999.

Pre Recorded 00000

Number of bus events currently in record memory that occurred prior to the trigger. Pre recorded count can be from 0 to 32,768.

Post Count 00000000

Number of bus events that occurred after the trigger since [REC] was enabled. Post count can be from 0 to 99,999,999.

Post Recorded 00000

Number of bus events currently in record memory that occurred after the trigger. Post recorded can be from 0 to 32,768.

Trigger 00000

Record memory location of the trigger point. Trigger can be EMPTY or be from 0 to 32,767.

First 00000

Record memory location of the first available recorded bus event. The first location can be EMPTY or from 0 to 32,767.

Last 00000

Record memory location of the last available recorded bus event. The last location can be EMPTY or be from 0 to 32767.

[DISPLAY] The Display function [SHIFT][STAT], is used to adjust the brightness of the front panel alphanumeric display and to set the scroll rate of help messages. A data input menu prompts for input. Use the [↑] and [↓] cursor keys to vary the display brightness and scroll rate. Leave the menu with [ENTER] or [EXIT] and cancel with [ESC].

Display Intensity 32

Requests a display intensity. Intensity can be from 1 to 60.

Help Scroll Speed 6

Request a scroll speed for help messages. Scroll speed can be from 1 to 10.

[SETUP] The Setup key, [SETUP], is used to configure one or more triggering parameters used in recording bus events. See also [ARM] and [MEASURE]. [SETUP] uses menus to input setup parameters. Use the cursor keys to step between configuration parameters. The configuration session is ended by pressing [ENTER] to set every parameter or pressing [EXIT] when the parameters of interest have been changed. The operation may be aborted by pressing [ESC]. If [SETUP] is pressed during power-up, it is used to configure the Analyzer488 operating mode, serial port parameters and IEEE 488 bus address of the Analyzer488. Refer to paragraph 3.4 for configuration details.

Match? Nul XXXXXXX (hex)

Match? XXXXXXXXXXXX (bin)

Requests the bus pattern for match. Bus data and signal line patterns are specified as true (1), false (0) or don't care (X). When displaying hexadecimal data, changing the hexadecimal data byte automatically changes the corresponding bus message. Use the [→] and [←] cursor keys to move between input fields. Use the [↑] and [↓] cursor keys to set values. Pressing [↑] at the NUL bus message selects a bus error (ERROR). Accept entry with [ENTER] and [EXIT] or cancel with [ESC].

of Matches? 00001

Requests the number of times the bus pattern should match before starting the trigger delay. Number of matches can be from 1 to 65,535. Accept entry with [ENTER] and [EXIT] or cancel with [ESC].

Trig Delay? 0000000

Requests the number of events to delay before starting the post count. Trig delay can be from 1 to 9,999,999. Accept entry with [ENTER] and [EXIT] or cancel with [ESC].

Post Count? 00000

Requests the number of events in the trigger post count. Post count can be from 1 to 32,767. Accept entry with [ENTER] and [EXIT] or cancel with [ESC].

When Complete? Stop

Requests the recording action to be taken following the completion of the post trigger count. The two choices are: Stop or Step. Stop allows the activity on the bus to continue at the selected handshake speed. Step continues recording at single-step handshake rate (use

[STEP] to perform handshaking). Select with [↑] and [↓] cursor keys. Accept entry with [ENTER] and [EXIT] or cancel with [ESC].

[MEASURE] The Measure command, [SHIFT] [SETUP], displays post trigger bus event performance. Data menus show the time, number of bytes, and transfer rate of the last triggered recording operation.

Time 0.000 mSeconds

Amount of time that elapsed during the post trigger recording. Time can be from 0.001 milliseconds to 999.9 seconds. Use the [↑] and [↓] cursor keys or [ENTER] to display further measure information, [EXIT] or [ESC] will exit.

Count 00000 Bytes

Number of events that occurred during the post trigger recording. Count will be from 0 to 32767. Use the [↑] and [↓] cursor keys or [ENTER] to display further measure information. [EXIT] or [ESC] will exit.

Rate 0.000 B/sec

Speed of the transfer which occurred during the post trigger recording. Rate will be from 0.001 Bytes per second (B/sec) to 1.100 megabytes per second (Mbyte/sec). Use the [↑] and [↓] cursor keys to display further measure information. [ENTER], [EXIT] or [ESC] will exit.

[ARM] The Arm key, [ARM], is used to enable or disable the triggering system while recording data. When recording, [ARM] will cause the Arm LED to flash until the trigger condition is satisfied then remain on until the trigger sequence is complete. Also see [SETUP] and [MEASURE]. While the Analyzer488 is NOT recording data, [ARM] has no effect.

ARM LED The Arm LED, when flashing, indicates the triggering system is

armed and actively looking for a trigger point. When on continuously, the Arm LED indicates the triggering system is armed but not looking for a trigger. This may occur because either the trigger has been found or recording is disabled. Triggering may be disabled by pressing [ARM].

[HEX↔BIN] The display format toggle, [SHIFT][ARM], toggles the format of the front panel display -MESSAGE- area between hexadecimal data with bus messages and binary display of data. A status message will confirm the -MESSAGE- format:

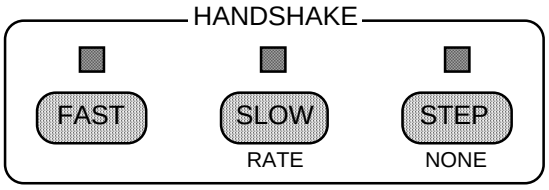
Display Hex Data

or

Display Binary Data

Handshake

The Handshake keys are used to control the speed of IEEE 488 bus data transfers.



`[FAST]` The Fast key, `[FAST]`, is used to set the IEEE 488 bus handshake control to fast operation. It enables participation in high speed bus source or acceptor handshaking at speeds up to 1 Mbyte/sec. Source (Talk state) or acceptor (Listen state) handshaking, is dependent on the `-SOURCE-` keys.

`FAST LED` The `FAST` LED indicates that the IEEE 488 bus handshake control is currently set to fast operation. Changing the handshake control may be accomplished by pressing `[SLOW]`, `[STEP]`, or `[NONE]`.

`[SLOW]` The Slow key, `[SLOW]`, is used to set the IEEE 488 bus handshake control to slow. It enables participation in low speed source or acceptor bus handshaking at speeds from 1 to 10 bytes/sec. The handshake speed is set using the `[RATE]` command. Source (Talk state) or acceptor (Listen state) handshaking, is dependent on the `-SOURCE-` keys.

`SLOW LED` The `SLOW` LED indicates that the IEEE 488 bus handshake control is currently set to slow operation, recording and/or viewing bus events at a rate between 1 to 10 events per second while participating in the bus three wire handshake. Changing the handshake speed may be accomplished using the `[FAST]`, `[STEP]`, or `[NONE]` keys.

`[RATE]` The Rate command, `[SHIFT][SLOW]`, is used to adjust the slow handshake rate. A menu requests the speed of the slow handshake:

Steps Per second 03

Rate can be set from 1 to 10 bytes/sec. Use [↑] and [↓] to select. Accept value with [ENTER] and [EXIT] or cancel with [ESC].

[STEP] The Step Key, [STEP], is used to set the IEEE 488 bus handshake speed to step or perform a bus handshake operation if the speed is already set to step. The first press of [STEP] lights the SLOW LED and sets the handshake circuitry to Slow operation. Subsequent presses of [STEP] causes a single bus handshake to occur. [STEP] enables STEP bus source or acceptor handshaking. Source (Talk) or acceptor (Listen) handshaking is dependent on -SOURCE- keys. Subsequent [STEP] or [ENTER] key presses initiates (Talk state) or completes (Listen state) one handshake sequence if in bus display mode.

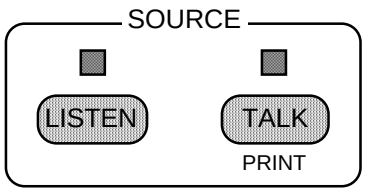
STEP LED The STEP LED indicates that the IEEE 488 bus handshake speed is currently set to single step operation to record and/or view bus events individually in a single-step fashion. [STEP] must be pressed to complete each event. Changing the handshake speed may be accomplished using the [FAST], [SLOW], or [NONE] keys.

[NONE] The None handshake key, [SHIFT][STEP], disables the Analyzer488 from handshaking on the IEEE 488 bus. Pressing [NONE] disables handshaking and displays a status message:

Handshaking Off

Source

The Source keys are used to toggle the Analyzer488 between the Listen state and Talk state.



`[LISTEN]` The Listen key, `[LISTEN]` switches the Analyzer488 from the Talk state to the Listen state, reading the data lines. The acceptor handshake, selected by this key, is enabled by the `[FAST]`, `[SLOW]` and `[STEP]` keys and disabled by the `[NONE]` key.

`LISTEN LED` The Listen LED indicates that the Analyzer488 is currently in the Listen state.

`[TALK]` The Talk key, `[TALK]` switches the Analyzer488 from the Listen state to the Talk state. While in the Talk state the Analyzer488 drives the bus data lines and ásserts the EOI control signal. The source handshake, selected by this key, is enabled by the `[FAST]`, `[SLOW]` and `[STEP]` keys and disable by the `[NONE]` key. Pressing `[TALK]` asserts the Data Available (DAV) handshake line and lights the `DAV` LED.

`TALK LED` The Talk LED, when on, indicates the Analyzer488 is currently in the Talk state.

`[PRINT]` The Print menu `[SHIFT][TALK]` prints all or a portion of the record memory to a serial (RS-232) or IEEE 488 printer. Pressing `[PRINT]` starts a sequence of menus that prompt for the `first` and `last` record memory locations to be printed and for printer parameters. Once a printer is configured, it becomes the default printer. The settings remain in non-volatile RAM until they are reset. Default values for `first` and `last` are the first and last record memory locations used in the latest recording session. Once a printer is configured and a sequence of bus events is recorded,

pressing [PRINT] followed by [EXIT] will print the record memory from first to last on the default printer. A status message is displayed while the Analyzer488 is printing:

Printing...

Print First? 00000

Requests the first record memory location to be printed. Physical locations, 0 to 32767, or relative locations, -32767 to 32767, may be used depending on the record memory location address format. Use [ENTER] or [EXIT] to proceed, [ESC] to abort printing.

Print Last? 00000

Requests the last record memory location to be printed. Physical locations, 0 to 32767, or relative locations, -32767 to 32767, may be used depending on the record memory location address format. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Print to? Serial Port

Requests which printer to use, Serial Port, or IEEE 488 Bus. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

If a Serial Port printer is selected:

Printer Baud? 9600

Requests the serial printer baud rate. Baud rate can be 300, 600, 1200, 2400, 4800, 9600, or 19200. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Printer Data Bits? 8

Requests serial printer data bits, 7 or 8. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Printer Stop Bits? 2

Requests serial printer stop bits, 1 or 2. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Printer Parity? None

Requests serial printer parity. Parity can be None, Even, or Odd. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Printer Ctrl? RTS/CTS

Requests serial printer handshake method, XON/XOFF or RTS/CTS. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Print Terms? CR

Requests serial printer terminators. Terminators can be CR, LF, LF CR, or CR LF. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

If an IEEE 488 bus printer is selected:

Printer Address? 16

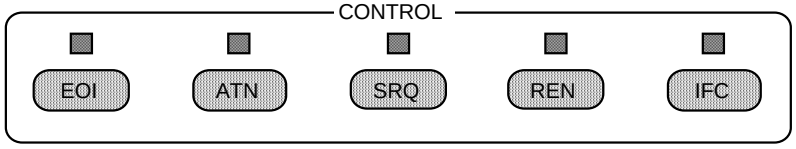
Requests the address of the IEEE 488 printer, from 0 to 30. Use [↑] and [↓] cursor keys or numeric keys to enter address. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Print Terms? CR

Requests IEEE 488 printer terminators. Terminators can be CR, LF, LF CR, or CR LF. Use [↑] and [↓] cursor keys to select. Use [ENTER] or [EXIT] to proceed, [ESC] to abort print operation.

Control

The Control keys are used to control the bus management lines. The Control LED indicators show the driven state of the bus management lines, not the actual state. The actual state of the bus management lines can be seen on the front panel alphanumeric display.



`[EOI]` The End or Identify key, `[EOI]`, is used to assert or unassert the End or Identify (EOI) bus management line.

`EOI LED` The `EOI` LED indicates that the Analyzer488 is asserting the EOI line. It does not display the state of the bus EOI signal.

`[ATN]` The Attention key, `[ATN]`, is used to assert or unassert the Attention (ATN) bus management line.

`ATN LED` The `ATN` LED indicates that the Analyzer488 is asserting the ATN line. It does not display the state of the bus ATN signal.

`[SRQ]` The Service Request key, `[SRQ]`, is used to assert or unassert the Service Request (SRQ) bus management line.

`SRQ LED` The `SRQ` LED indicates that the Analyzer488 is asserting the SRQ line. It does not display the state of the bus SRQ signal.

`[REN]` The Remote Enable key, `[REN]`, is used to assert or unassert the Remote Enable (REN) bus management line.

`REN LED` The `REN` LED indicates that the Analyzer488 is asserting the REN line. It does not display the state of the bus REN line.

`[IFC]` The Interface Clear key, `[IFC]`, is used to pulse the Interface Clear (IFC) bus management line.

`IFC LED` The `IFC` LED indicates that the Analyzer488 is asserting the IFC line. It does not display the state of the bus IFC line.

Serial Controller

4.1 Introduction

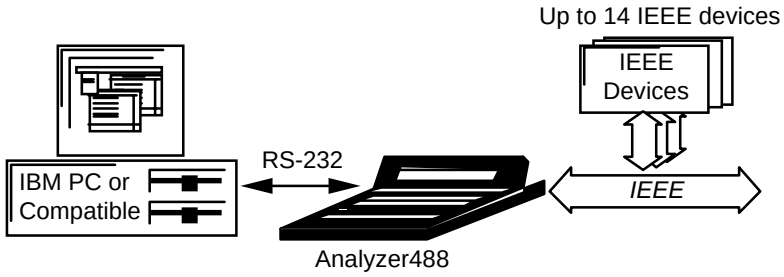
This section contains a detailed explanation of the Serial Controller Mode of operation of the Analyzer488 IEEE 488 Bus Analyzer. The following paragraphs contain a description of the Serial Controller Mode, information regarding setup, configuration, an explanation of how to use the Analyzer488 as a serial controller, and a detailed explanation of all the commands.

4.2 Description

When in the Serial Controller Mode, a serial host computer or terminal may control the Analyzer488 through a serial port. Once the Analyzer488 is attached to a computer and the IEEE 488 bus, it becomes a full-featured IEEE 488 bus controller. The Analyzer488 may be operated as a High Level System controller or as a Low Level System Controller. High Level Controller mode allows control of the IEEE 488 bus using High Level commands. Low Level Controller mode allows direct control over all data and bus management lines using an alternative set of commands. The Analyzer488 can record bus events and send the recorded data back to the host computer. Triggering may be used to stop the recording of bus data. The Analyzer488 front panel keyboard can be read and messages can be sent to the front panel 20 character display.

4.3 Serial Interface

To set up the Analyzer488 for use as a serial controller, connect it to the host computer through the RS-232 connectors and to the IEEE 488 bus through the IEEE 488 bus connector on the rear panel.



4.4 Configuration

Configuration of the Analyzer488 is accomplished using the front panel keyboard. Once configured, the configuration parameters will be stored in nonvolatile RAM and used as the default configuration when the Analyzer488 is powered on. Refer to Section 1 for configuration details.

4.5 Using the Analyzer488 with a Personal Computer

Commands may be sent to the Analyzer488 from a host computer using the IBM BASIC `PRINT#` command, or it's equivalent in other languages. The Analyzer488 interprets the commands, performs the necessary bus control and handshaking, and returns data from the bus. Data acquired from the bus and responses from the Analyzer488 are sent back to the host computer and read using the BASIC `INPUT#` command, or its equivalent.

Listed below is a keyboard controller program that allows direct interaction between a personal computer and the IEEE 488 bus using the Serial Controller Mode of the Analyzer488.

```
10 OPEN "COM1:  9600,n,8,2,cs,ds" AS 1
20 IF LOC(1) THEN PRINT INPUT$(LOC(1),1);
30 K$=INKEY$
40 PRINT #1,K$;:PRINT K$;
50 GOTO 20
```

Refer to Section 1 and configure the Analyzer488 as a Serial Controller with the following serial port parameters:

9600 baud
8 data bits
2 stop bits
No parity
XON/XOFF handshaking

Run the program and type: `hello` The Analyzer488 will respond with:

```
Analyzer488 Revision N.N Copyright (C) 1989 IOtech Inc.
```

If you get the above response, the Analyzer488 is alive and ready to operate as a serial controller. If you do not receive the hello response, check for proper connection and fit of the cables. If the cables are connected properly and you still do not obtain the hello response, then refer to the Service Section of this manual.

Refer to the Command Descriptions section for a complete listing of all the Serial Controller commands and examples of their use.

4.6 Using the Analyzer488 with a Terminal

The Analyzer488 may be controlled from an RS-232 terminal or a personal computer running a terminal program. Commands typed at the keyboard are sent directly to the Analyzer488 and any responses are then displayed on the terminal screen.

To use the Analyzer488 with a terminal, obtain a cable wired for your particular terminal and the Analyzer488. Refer to the Getting Started section for wiring diagrams and pinout information. Next, configure the Analyzer488 serial port parameters to match those of your terminal using the power on configuration procedure in Section 1. The parameters that must be set are:

Serial Baud Rate
Serial Data Bits (7 or 8)
Serial Stop Bits (1 or 2)
Serial Parity (None, Even, or Odd)
Serial Ctrl (RTS/CTS or XON/XOFF)
Serial Echo (Enabled or Disabled)
Serial Terminators

If Full Duplex operation is selected for your terminal, Echo should be enabled on the Analyzer488 in order for commands sent to the Analyzer488 to be displayed on the terminal screen. If Half Duplex operation is selected, Echo should be disabled. In either case, commands sent to the Analyzer488 and responses received will appear on the terminal screen.

Once the Analyzer488 has been connected to the terminal and the serial port parameters are set, type the command `hello` at the keyboard. The following response should be displayed on the terminal screen:

```
Analyzer488 Revision N.N Copyright (c)  1989 Iotech Inc.
```

If you get the above response, the Analyzer488 is alive and ready to operate as a serial controller. you may find it useful to send the command `ERROR MESSAGE` which will cause the Analyzer488 to always send a response after every command it has received and processed. If a command is sent which causes an error, an appropriate error message will be returned. If no errors have occurred, the message `OK` will be sent to the terminal.

If you do not receive the hello response, check for proper wiring and fit of the cables and the serial port configurations of the terminal and the Analyzer488. If all is well and you still do not obtain the hello command response, refer to the Service Section of the Manual.

Refer to the Command Description section for a complete listing of all the Serial controller commands and examples of their usage.

4.7 Serial Controller Commands

There are four groups of Serial Controller commands: system commands, record commands, high level controller commands, and low level controller commands. System and record commands configure or request information from the Analyzer488. High level and low level controller commands communicate with IEEE 488 bus devices.

4.7.1 System Commands:

@	DISARM	HELLO	STATUS
@@	DISPLAY	ID	STERM
ARM	ERROR	KEY	TIMEOUT
CONTROLLER	HANDSHAKE	MASK	

4.7.2 Record Commands:

DUMP	MEASURE	RECORD	SET TRIGGER
ERASE	NEXT	RELATIVE	VIEW
FIND	PRINT (?)	SET RELATIVE	ZERO

4.7.3 High Level Controller Commands:

ABORT	LOCAL LOCKOUT	REMOTE	SPOLL
ADDRESS	OUTPUT	RESET	PPOLL
CLEAR	PPOLL CONFIG	RESUME	TRIGGER
ENTER	PPOLL DISABLE	SEND	TERM
LOCAL	PPOLL UNCONFIG		

4.7.4 Low Level Controller commands:

ASSERT	READ	TALK	WRITE
LISTEN	STEP	UNASSERT	

4.8 Bus State Format

The state of the IEEE 488 bus management and data lines can be returned using the VIEW or READ commands. The VIEW command returns the state of the bus without performing a handshake. The READ command returns the state of the bus and performs an acceptor handshake. Bus data is returned in two hexadecimal bytes, formatted as shown below. If a line is asserted, the bit for that line will be a 1.

Hex Byte 1 (bit positions) (Control and Handshake Lines)								Hex Byte 2 (bit positions) (Data Lines)							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
REN	IFC	SRQ	EOI	DAV	NRFD	NDAC	ATN	D8							D1

A response of 88F4 is interpreted as follows:

Byte 1, (88) Control and Handshake Line
The upper nibble (8) (%1000) = REN asserted.
The lower nibble (8) (%1000) = DAV asserted

Byte 2, (F4) Data Lines
F4= (%11110100)
Lines DIO8 through DIO5 are asserted, DIO3 is asserted.

4.9 Record Memory Format

The bus events stored in the record memory can be seen using the `VIEW` or `DUMP` commands. The `VIEW` Record command returns events from the record memory in blocks. The `DUMP` Record command returns the contents of the record memory in S1S9 record format (See the `DUMP` command description for details of S1S9 format). Record memory events are returned in two hex bytes and are formatted as follows:

Byte 1 (bit positions) (Control and Status Lines)								Byte 2 (bit positions) (Data Lines)							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
REN	IFC	SRQ	EOI	BERR	VALID	TRIG	ATN	D8							D1

A response of 84F4 is interpreted as follows:
Byte 1, (88) Control and Handshake Lines
Upper nibble (8) (%1000)=REN asserted
Lower nibble (4) (%0100)=this data is valid.

Byte 2, (F4) Data Lines
F4=(%11110100)
Lines DIO8 through DIO5 are asserted, DIO3 is asserted.

4.10 Memory Usage

Memory in the Analyzer488 is dynamically allocated for the serial input and serial output. This allows for the most efficient partitioning of memory for any given application.

At power on, each serial buffer is allocated a 127 byte mini-buffer or queue. When the serial input [or output] requires more buffer space, additional queues are allocated. When a queue is empty, it is released from the input buffers so that it may be reallocated when, and where, required.

There are approximately 32 available queues for a total of 4096 bytes of buffer (character) space. Queues are continually allocated and released as required. Of the 32 available queues, 22 are issued without regard to controlling the receipt of additional serial input data.

When the serial input buffer requests one of the last 10 queues (1270 character locations left), it signals the serial host that it should stop sending data. This is accomplished by either unasserting RTS or issuing "XOFF", depending on which serial handshake control has been switch selected. When more than 10 queues become available, it asserts RTS or issues "XON".

4.11 Command Descriptions

The commands for the Serial Controller Mode of the Analyzer488 are described on the following pages. Commands are listed in alphabetical order. Each command description is divided into several areas:

4.11.1 Type

The type section of the command description lists the command group, one of: system command, record command, high level controller command, or low level controller command.

4.11.2 Syntax

The syntax section of the command description describes the proper command syntax which must be sent to the Analyzer488 using the IBM BASIC `PRINT#` command, or its equivalent in other languages, to the COM port. The following conventions are used in the syntax descriptions:

No command, along with its options, may be more than 127 characters long. The data part of the `OUTPUT` command is not constrained by this length. The `OUTPUT #count; data` command may be as long as necessary. Refer to the `OUTPUT` command description for more information on this.

Items shown in capital letters, such as ENTER or OUTPUT must be used exactly as stated but may be entered in upper or lower case.

Items in lower case, such as addr or count represent parameters which must be substituted with an appropriate value.

Blank spaces in commands are generally ignored. Thus, LOCAL LOCK OUT is the same as LOCALLOCKOUT. Spaces are not ignored in four places: the data part of an OUTPUT command, within quoted strings in a SEND command, after an apostrophe (') in a terminator specification (term), and after the semi-colon following the ID command.

The number sign or hash character (#) and the semi-colon (;) must be present exactly as shown. A comma (,) represents an address separator. The oblique or slash character (/) or period (.) may be used in its place as the address separator.

Optional semicolons ([;]) may be used, if desired, for consistency with other IOtech products.

Items enclosed in the square brackets ([item]) are optional. Multiple items enclosed in square brackets separated by vertical lines ([item1|item2|item]) are optional, any one or none may be chosen. No more than one item may be selected.

Ellipses(,,,) within square brackets mean that the items in the brackets may be repeated as many times as desired. For example [,addr,,,] means that any number of address separator-address combinations may be used, up to a maximum of fifteen.

Items enclosed in braces or curly brackets ({}) are required. {item1|item2} means that exactly one of the enclosed items is required.

Combinations of brackets are possible. For example, {term[term][EOI]|EOI} allows the choice of term, term EOI, term term, term term EOI, or just EOI, but does not allow the choice of "nothing."

Numeric parameters (those that are given as numbers) are decimal unless preceded by &H in which case they are considered to be hexadecimal or by % in which case they are considered to be binary. Thus 100 is decimal 100, &H64 is hexadecimal 64 (decimal 100), %00000100 is binary (decimal 4), &HFF is decimal 255, and OFF is invalid because F is not a valid decimal digit. The only exception to the rule is the IEEE 488 bus addresses, both primary and secondary, must be specified as two digit decimal numbers. Hexadecimal bus address are not allowed. When specifying binary numbers, eight digits must follow the % sign.

Several of the commands require additional or optional parameters and keywords. These are further described with each command, but discussion of the more common ones follow. If no optional parameters are specified with a command that requires them, the command will be treated as invalid.

4.11.2.1 Bus Addressing

`pri-addr` A two digit primary device IEEE 488 bus address in the range of `00` to `30`.

`sec-addr` An optional two digit secondary IEEE 488 bus device address in the range of `00` to `31`.

`addr` An IEEE 488 bus address. A numeric primary address optionally followed by a secondary address. Thus, `addr` is of the form... **{**`pri-addr`**[**`sec-addr`**]}**
where `pri-addr` is a two digit primary IEEE 488 bus device address from `00` through `30` and `sec-addr` is a two digit secondary IEEE 488 bus device address from `00` through `31`. Numeric addresses must be given as two digit decimal numbers, e.g. `05` for address `5`, and `1601` for primary address `16`, secondary address `1`

`[,addr,,,]` An optional list of bus addresses, each one preceded by an address separator; either a comma (`,`), a slash (`/`) or a period (`.`).

No more than 15 bus addresses are allowed in any single command.

4.11.2.2 Record Memory Addressing

`location` An address of the record memory. Absolute record memory locations are from 0 to 32767, and relative locations are from -32767 to +32767.

4.11.2.3 Character Count

`#count` The number of characters to be transferred. A pound sign (#) followed by an integer in the range of 1 to 32767. May be specified in hexadecimal by preceding it with &H. The hexadecimal range is &H1 to &H7FFF. A character count of zero in invalid.

4.11.2.4 ASCII Characters

`$char` A single character whose ASCII value is the number `char`, a decimal number in the range `0` to `255` or a hexadecimal number in the range of `&H0` to `&HFF`. For example, `$65` is the letter "`A`", as is `&H41`.

`CR` The carriage return character (`$13`, `&H0D`).

`LF` The line feed character (`$10`, `&H0A`).

`'X` Any printable character. The apostrophe, (`'`), is immediately followed, without any intervening spaces, by a single character which is taken to be the character specified.

4.11.2.5 ASCII Character Strings

`data` An arbitrary string of characters. None of the special forms given above (`$char`, `CR`, `LF`, or `'X`) are used. For example, `CRLF` as data is taken as the letters, "`C`", "`R`", "`L`", and "`F`", not as carriage return line feed.

`'data'` An arbitrary string of characters enclosed in apostrophes (`'`) or quotes (`"`).

4.11.2.6 Terminators

`term` Any single character, specified as `CR`, `LF`, `'X`, or `$char` as described previously. Part of the terminator sequence used to mark the end of lines of data and commands.

`[term]` An optional term character. `term[term]` means that one or two terminators may be specified.

`EOI` The IEEE 488 bus End Or Identify (EOI) signal. When asserted during the transfer of a character, EOI signals that that character is the last in the transfer. On input, `EOI`, if specified, causes the input to stop. On output, `EOI` causes the bus EOI signal to be asserted during transmission of the last character transferred.

`NONE` The no end-of-line character indicator. When `STERM NONE` is specified, the Analyzer488 does not append any serial output terminator(s) to serially transmitted data.

4.11.2.7 Bus Management and Data Lines

`line` A bus management line. May be one or more of: `SRQ`, `ATN`, `REN`, `IFC`, or `ALL`.

`hsline` A state of the handshake lines. May be `ERROR`. `ERROR` is used to specify the state of the handshake lines when a bus error occurs.

`state` A bus management or handshake line state. A slash, (`[/]`), indicates unasserted, `[x]` means "don't care". The absence of a slash `[/]` or `[x]` means true (asserted).

`<byte>` A data byte specifying the state of the data lines. `byte` may be specified in Binary (`%01001010`), Decimal (`122`), Hexadecimal (`&H3F`), or ASCII (`'A`). If Binary is used, all eight digits must be specified. If Hexadecimal is used, both digits must be specified.

4.11.2.8 Event Conditions

`event` A parameter that specifies a bus management line condition or an Analyzer488 circuit event. One or more of: `SRQ`, `RECORD`, `STEP`, or `TRIGGER`.

4.11.3 Response

The response section of the command description describes the response that your program should read from the serial host COM port after sending the command. If a response is provided, it must be read to maintain proper program sequence.

4.11.4 Remarks

Remarks, if included, give general comments and explanations of the commands. The remarks section includes detailed descriptions of command options.

4.11.5 Mode

The Analyzer488 may be configured as a High Level System Controller or as a Low Level System Controller. High Level Controller

mode allows control of the IEEE 488 bus using high level commands. Low Level Controller mode allows direct control over all data and bus management lines using an alternate command set.

4.11.6 Bus States

This section describes the bus command and data transfers using IEEE 488 bus mnemonics abbreviated as follows:

		DIO lines							
		8	7	6	5	4	3	2	1
ATN	Attention								
data	Data String								
DCL	Device Clear	x	0	0	1	0	1	0	0
EOI	End Or Identify								
GET	Group Execute Trigger	x	0	0	0	1	0	0	0
GTL	Go To Local	x	0	0	0	0	0	0	1
IFC	Interface Clear								
LAG	Listen Address Group	x	0	1	a	d	d	r	n
LLO	Local Lock Out	x	0	0	1	0	0	0	1
MLA	My Listen Address	x	0	1	a	d	d	r	n
MTA	My Talk Address	x	1	0	a	d	d	r	n
PPC	Parallel Poll Configure	x	0	0	0	0	1	0	1
PPD	Parallel Poll Disable	x	1	1	1	0	0	0	0
PPE	Parallel Poll Enable	x	1	1	0	S	P3	P2	P1
PPU	Parallel Poll Unconfigure	x	0	0	1	0	1	0	1
REN	Remote Enable								
SDC	Selected Device Clear	x	0	0	0	0	1	0	0
SPD	Serial Poll Disable	x	0	0	1	1	0	0	1
SPE	Serial Poll Enable	x	0	0	1	1	0	0	0
SRQ	Service Request								
TAG	Talker Address Group	x	1	0	a	d	d	r	n
TCT	Take Control	x	0	0	0	1	0	0	1
UNL	Unlisten	x	0	1	1	1	1	1	1
UNT	Untalk	x	1	0	1	1	1	1	1

(x = "don't care")

If a command is preceded by an asterisk then that command is unasserted. For example, *REN states that the remote enable line is unasserted. Conversely, REN without the asterisk states that the line becomes asserted.

4.11.7 Query Response

Many of the Analyzer488 commands offer a Query (`?`) option. This option may be used to examine the current state of a given command, or the parameters specified when the command was last used. The query option is specified by sending the command keyword followed by a question mark. For example, the Talk/Listen State of the Analyzer488 when in a low level controller mode may be returned by sending the `TALK` command followed by a question mark (`TALK?`). In response, the Analyzer488 will return the string `TALK` in in the Talk State or `LISTEN` if in the Listen State. All query responses strings are formatted such that the string may be sent back to the Analyzer488 as a command. One use for the Query option is to save the present configuration of the Analyzer488 for later use.

4.11.8 Examples

This section gives programming examples written in the BASIC language.

4.12 Command Descriptions

The following pages describe in detail all the commands used in the Serial Controller Mode of operation of the Analyzer488.

@

The system ID command @, followed by a CR and/or LF, is used to unlock the Analyzer488 from an inappropriate command. When the @ command is received, the serial port handshake line, Request To Send (RTS), is unasserted. It is asserted when the Analyzer488 is capable of buffering commands. If XON/XOFF handshake is selected, the software handshake state is not modified.

The @ character, referred to as the ID character, can be changed or disabled by using the ID command. If it is anticipated that the ID character may be part of the data within an OUTPUT or SEND command, it should be changed or disabled.

Issuing the @ command clears the serial input buffer (pending commands) and the serial output buffer (pending data). It is equivalent to issuing the following commands:

```
DISARM
ERROR OFF
ID;@
MASK OFF
TIME OUT 10
```

This command does not have a query option.

TYPE	System Command
SYNTAX	@
RESPONSE	None
MODE	Any
BUS STATES	None
QUERY RESPONSE	None
EXAMPLES	PRINT #1,"@"

@@

Sending the system ID command @@ causes the Analyzer488 to return to power-on conditions. All data buffers are cleared, the ID character is reset to @, and any software programmable terminators are returned to the power-on defaults.

This is the only command which does not require a serial terminator to execute. Reset is executed upon receipt of the second @.

When the @@ command is received, the serial port handshake line Request To Send (RTS) is unasserted. It is asserted when the Analyzer488 is capable of buffering commands. If XON/XOFF handshake is selected, the software handshake state is reset.

The @ character, referred to as the 'ID' character, can be changed or disabled by using the ID command. If it is anticipated that the ID character may be part of the data within an OUTPUT or SEND command, it should be changed or disabled.

This command does not have a query option.

TYPE	System Command
SYNTAX	@@
RESPONSE	None
MODE	Any
BUS STATES:	IFC,*IFC
QUERY RESPONSE	None
EXAMPLES	PRINT #1,"@@"

ABORT

The ABORT command causes the Interface Clear (IFC) bus management line to be asserted for at least 500 microseconds. By asserting IFC, the Analyzer488 regains control of the bus even if one of the devices has locked it up during a data transfer. ABORT forces all IEEE 488 bus device interfaces into a quiescent idle state.

This command does not have a query option.

TYPE	High Level Controller command	
SYNTAX	ABORT	
RESPONSE	None	
MODE	High Level Controller	
BUS STATES	IFC, *IFC	
QUERY RESPONSE	None	
EXAMPLES	PRINT#1,"ABORT"	pulse IFC line

ADDRESS

The ADDRESS command sets the IEEE 488 bus address of the Analyzer488 when functioning as a high level controller.

The query option, ADDRESS?, returns the word ADDRESS followed by a two digit number between 00 and 30, specifying the IEEE 488 bus address of the Analyzer488.

TYPE High Level Controller command

SYNTAX ADDRESS {[?]|[addr]}

addr is an IEEE 488 bus device address.

RESPONSE None

MODE High Level Controller

BUS STATES None

QUERY RESPONSE ADDRESS nn

EXAMPLES

```
PRINT#1,"CONTROLLER ON"      select high level mode
PRINT#1,"ADDRESS 24"         set IEEE address to 24
PRINT#1,"ADDRESS?"           query the address
INPUT#1,A$                   read address message
PRINT A$                     display response: ADDRESS 24
```

ARM

The ARM command programs the Analyzer488 to send event messages to the serial host when one or more specified events (SRQ, TRIGGER, RECORD, or STEP) occur. The event messages returned are the same strings used to program the events.

Once a condition is ARMed it remains ARMed until the event specified has occurred, it is DISARMed, or the Analyzer488 is reset. The ARM command must be resent after an event message is reported to reactivate the ARMed condition.

There are two types of events, level sensitive and edge sensitive. The level sensitive event, SRQ, will be reported immediately if the SRQ line is asserted when ARM is issued. Usually, some action must be taken, such as issuing a SPOLL command to clear SRQ, prior to reissuing the ARM command. The edge sensitive events, TRIGGER, RECORD, and STEP, are cleared when reported.

The query option, ARM?, returns the conditions that are presently ARMed.

TYPE System Command

SYNTAX ARM **[[?]|[**event**[**event…**]]]**

event is one or more of SRQ, TRIGGER, RECORD, or STEP

If no event is specified, ARM SRQ is assumed.

RESPONSE Event string sent when the event occurs

REMARKS The following conditions are available using the ARM command:

SRQ The event message 'SRQ' is returned to the serial host when the state of the Service Request (SRQ) bus line is detected to be in the asserted state. This event is level sensitive. If the condition exists at the time the ARM SRQ command is issued, the Analyzer488 will return the event message immediately. If the ARM command is issued without any specified events, ARM SRQ is assumed.

TRIGGER The event message 'TRIGGER' is returned to the serial host when the Analyzer488 trigger circuitry has found the trigger point. This event is edge sensitive.

RECORD The event message 'RECORD' is returned to the serial host when the Analyzer488 recording sequence is finished. This event is edge sensitive.

STEP The event message 'STEP' is returned to the serial host when the bus is ready for a source (talker) or acceptor (listener) handshake. This event is edge sensitive.

MODE any

BUS STATES None

QUERY RESPONSE String of events in the following format:
`ARM SRQ TRIGGER STEP RECORD`

EXAMPLES

```
100 PRINT#1,"ARM SRQ"   Enable SRQ Condition
110 T = TIMER
110 IF LOC(1) THEN PRINT INPUT$(LOC(1),1);
120 IF TIMER - T < 5 GOTO 110
```

SRQ message will be received and printed if an SRQ occurs within 5 seconds after receipt of the ARM command.

```
PRINT #1,"ARM SRQ STEP"     Arm SRQ and STEP.
PRINT #1,"ARM?"             Query armed status.
INPUT #1, A$                Get response.
PRINT A$                    Display results
```

The returned string would be:
`'ARM SRQ STEP'`

Refer to the demonstration program listed in appendix F for further examples of the ARM command.

ASSERT

The ASSERT command asserts one or more bus management and/or data lines when the Analyzer488 is in the Low Level Controller Mode. Individual lines or ALL lines may be specified. Data is specified after the keyword DATA in Binary, Decimal, Hexadecimal, or ASCII format

The ASSERT command will only assert data lines while the Analyzer488 is in the Talk State. If ASSERT is issued while the Analyzer488 is in Listen State, the specified management lines and data lines will be asserted when the Analyzer488 is changed to the Talk State (by issuing the TALK command).

Lines remain asserted until they are unasserted, the Analyzer488 is reset, or the CONTROLLER OFF command is used to put in the Analyzer488 into the High Level controller Mode.

The query option, ASSERT?, returns the management and data lines that have been asserted.

TYPE	Low Level Controller command
SYNTAX	ASSERT {[line][ALL][DATA<data>]} line is one or more of SRQ, ATN, REN, EOI, or IFC data may be specified in Binary (%01001010) (all eight digits must be specified), Decimal (122), Hexadecimal (&H3F) (both digits must be specified), or ASCII ('A).
RESPONSE	None
MODE	Low Level Controller
BUS STATES	Bus states are defined by the usage of the commands.
QUERY RESPONSE	The string: ASSERT, followed by the management and data lines that were asserted.

EXAMPLES

```
PRINT#1,"ASSERT SRQ"
```

assert the SRQ line

```
PRINT#1,"ASSERT ATN REN DATA &H54"
```

send the bus command TAG20 (Talk Address Group 20)

```
PRINT#1,"ASSERT DATA &H1F"
```

assert data lines 5 through 1, leaving lines 6, 7, and 8 unchanged

CLEAR

The CLEAR command causes the Analyzer488 to issue a Device Clear (DCL) bus command. If optional addresses are included, the Selected Device Clear (SDC) command is issued to all specified devices. IEEE 488 bus devices which receive a DCL or SDC command normally reset to their power-on state.

This command does not have a query option.

TYPE	High Level Controller command	
SYNTAX	CLEAR [addr[,addr…]]	
	addr is an IEEE 488 bus device address (primary address with optional secondary address). , is the address separator, either a comma, a slash [/] or a period [.].	
RESPONSE	None	
MODE	High Level Controller	
BUS STATES	ATN•DCL ATN•UNL,MTA,LAG,SDC	(all devices) (selected devices)
QUERY RESPONSE	None	
EXAMPLES	PRINT #1,"CLEAR"	Issue a Device Clear to all devices.
	PRINT #1,"CLEAR 12,18"	Issue a Selected Device Clear to devices 12 and 18.

CONTROLLER

The CONTROLLER command is used to configure the Analyzer488 to operate as a High Level System Controller (CONTROLLER ON) or a Low Level System Controller (CONTROLLER OFF). High Level Controller commands may only be used when the Analyzer488 is in the High Level Controller Mode. Low Level Controller commands may only be used when the Analyzer488 is in the Low Level Controller Mode.

Issuing a CONTROLLER OFF command unasserts all bus management and data lines. Issuing a CONTROLLER ON command pulses the IFC bus management line.

The query option, CONTROLLER?, returns the operational mode of the Analyzer488.

TYPE	System command
SYNTAX	CONTROLLER {[?]|[ON|OFF]}
RESPONSE	None
MODE	Any
BUS STATES	All data and management lines unasserted (CONTROLLER OFF) IFC, *IFC (CONTROLLER ON)
QUERY RESPONSE	CONTROLLER ON or CONTROLLER OFF

EXAMPLES Using High Level Control

```
PRINT #1,"CONTROLLER ON"    Select high level control
PRINT #1,"CLEAR 12"         Issue a Selected Device Clear
                            to device  at address 12
```

The same example using Low Level Control

```
PRINT #1,"CONTROLLER OFF"     Select low level control
PRINT #1,"HANDSHAKE STEP"     Select Step speed
handshaking
PRINT #1,"ASSERT REN ATN"     Assert ATN and REN to
                              cause devices to interpret data
                              as bus commands
PRINT #1,"WRITE &H3F"         Send UNL command
PRINT #1,"WRITE &H50"         Send TAG16 command
PRINT #1,"WRITE &H2C"         Send LAG12 command
PRINT #1,"WRITE &H04"         Send SDC command
```

DISARM

The DISARM command prevents the Analyzer488 from sending an event status message to the serial host, even when an ARMed condition occurs. A user's program can still check for the events by using the STATUS command.

If the DISARM command is issued without specifying any events, then all event message reporting will be disabled.

The ARM command must be used to re-enable the event responses after the event occurs.

This command does not have a query option although ARM? may be used to see which events have been armed.

TYPE	System Command	
SYNTAX	DISARM [;][event[event…]]	
	event is one or more of SRQ, RECORD, STEP, or TRIGGER.	
RESPONSE	None	
MODE	Any	
BUS STATES	None	
QUERY RESPONSE	None	
EXAMPLES	PRINT#1,"DISARM"	Disable all conditions
	PRINT#1,"DISARM SRQ"	Do not send 'SRQ' string if SRQ line is asserted.

DISPLAY

The DISPLAY command allows the user to display an alphanumeric message on the 20 character front panel display of the Analyzer488.

A message may be up to 20 characters long. If a message string is longer than 20 characters, only the first 20 characters will be displayed. A space ' ' or null string '' will clear the display.

This command does not have a query option.

TYPE	System command
SYNTAX	DISPLAY **{ [**'string'**]}**
	string is any alphanumeric character string.
RESPONSE	The first 20 characters of string are displayed on the front panel display of the Analyzer488.
MODE	any
BUS STATES	None
QUERY RESPONSE	None
EXAMPLES	PRINT#1,"DISPLAY'Analyzer488'"
	PRINT#1,"DISPLAY' '" Clear the display.

DUMP

The record command, DUMP, returns bus event data from the record memory. Blocks of data may be dumped by specifying a starting and ending memory location or a starting location and a length. Issuing DUMP without any options will dump the entire record memory.

DUMP returns blocks of data in Motorola S record (S1S9) format, up to 32 data bytes per line. Any range of locations, from 0 to 32767, may be dumped.

This command does not have a query option.

TYPE Record command

SYNTAX
```
DUMP
or
DUMP {location} TO {location}
or
DUMP {location} LENGTH {count}
```

location is an absolute address of the record memory, from 0 to 32767.
count is the number of record memory locations to be dumped from 0 to 32767.

RESPONSE Data encoded in S1S9 file format: S1 data records with a maximum of 32 data bytes per line, followed by an S9 termination record.

REMARKS S1 data record format:
```
<type> <record length> <address> <data>
<checksum>
```

where (all numbers are in hexadecimal):
type is the record header: S1
record length is the byte count (two hex characters).
address is the record memory starting address (four hex characters). The address is incremented twice for each location that is dumped. (address is always twice the starting record memory location.)

`data` are the management line and data line status (two hex bytes for each event) . Data is dependent on the contents of the record memory.

`checksum` is the one's complement of `record length` plus `address` plus all `data` bytes.

`The last line sent is the S9 record which is always: S9030000FC`

Format of `data` bytes is:

Byte 1 (bit positions) (Control and Status Lines)								Byte 2 (bit positions) (Data Lines)							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
REN	IFC	SRQ	EOI	BERR	VALID	TRIG	ATN	D8							D1

If a bus management or data line is asserted, its corresponding bit will be true (1).

MODE Any

BUS STATES None

QUERY RESPONSE None

EXAMPLES

```
10 PRINT#1,"ERASE"            Clear the record memory.
20 PRINT#1,"CONTROLLER ON"    Select High level control
30 PRINT#1,"SET TRIGGER OFF"  Do not use triggering
40 PRINT#1,"RECORD ON"        Start recording bus events
50 PRINT#1,"OUTPUT 20;W3X"    Send the command W3X
60 PRINT#1,"RECORD OFF"       Stop recording bus events
70 PRINT#1,"DUMP 0 TO 8"      Dump the first 9 locations
80 IF LOC(1) THEN PRINT INPUT$(LOC(1),1);
90 GOTO 80
```

The following response will be returned:

```
S1150000854A853F85348477843384788484 0D940A0000C1
S9030000FC
```

ENTER

The ENTER command reads data from the IEEE 488 bus.

If a device address (with optional secondary address) is specified, that device will be addressed to talk. If no address is specified, the Analyzer488 must already be configured to receive data, either as a result of an immediately preceding ENTER and ADDRESS command, or as a result of a SEND sub-command.

If time-out is enabled, a time-out error will occur if the Analyzer488 does not receive a data byte within the time-out period.

If character count, count, is specified, then exactly count characters will be read from the device. If count is not specified, ENTER terminates reception on detection of the first line feed (LF) terminator character. Serial output terminators are appended after count characters are read. The terminator may be overridden by specifying a different terminator, term, in the ENTER command.

If a terminator, term, option is specified, all CR and LF characters in the input data are unconditionally discarded. When the specified terminator is detected, it is discarded and replaced with serial output terminator(s) before control is returned to the serial host. The optional terminator applies ONLY to the ENTER command it is sent with. The terminator returns to a Line Feed on subsequent ENTER commands.

If the EOI option is specified, all characters are returned to the host until the EOI line is detected. The character sent with EOI asserted is also returned followed by the serial output terminator(s).

This command does not have a query option.

TYPE High Level Controller command

SYNTAX `ENTER[addr][#count|term|EOI|;count|;term|;EOI]`

addr is an IEEE 488 bus device address.
count is the number of characters to ENTER.
term and EOI override the normal IEEE bus input LF terminator.

RESPONSE Device-dependent data. If `count` is specified, then `count` characters will be returned followed by the serial output terminators. Otherwise the response ends when the IEEE bus input terminator is detected. The serial output terminators are then appended to the returned data.

MODE High Level Controller

BUS STATES ATN•UNL,MLA,TAG,*ATN,data…,ATN (With `addr`)
*ATN, data…, ATN (Without `addr`)

QUERY RESPONSE None

EXAMPLES

```
PRINT#1,"ENTER16"
INPUT#1,A$
```
Read data from device 16.

```
PRINT#1,"ENTER16"
LINE INPUT#1,A$
```
Read an entire line of data from device 16 even if it contains commas or other punctuation.

```
PRINT#1,"ENTER16;CR"
INPUT#1,A$
```
Read data from device 16 until CR is encountered.

```
PRINT#1,"ENTER16 EOI"
INPUT#1,A$
```
Read data until EOI is detected.

```
PRINT#1,"ENTER 0702"
INPUT#1,A$
```
Read data from device 7, secondary address 2.

```
PRINT#1,"ENTER #20"
A$=INPUT$(20,#1)
```
Read 20 more bytes.

```
PRINT#1,"ENTER ;20"
A$=INPUT$(20,#1)
```
Read 20 more bytes.

ERASE

The ERASE command erases the entire record memory. ERASE also resets the current location pointer to 0.

This command does not have a query option.

TYPE	Record command	
SYNTAX	ERASE	
RESPONSE	None	
MODE	Any	
BUS STATES	None	
QUERY RESPONSE	None	
EXAMPLES	PRINT#1,"ERASE"	Erase the record memory and reset the current location pointer to 0

ERROR

The ERROR command enables or disables automatic reporting of Analyzer488 error messages. When enabled, error messages are returned when commands are completed. ERROR MESSAGE enables error message string reporting, ERROR NUMBER enables error message number reporting, and ERROR OFF disables error reporting. ERROR OFF is the factory default condition.

The query option, ERROR?, returns the ERROR command option currently in use.

TYPE System command

SYNTAX ERROR[;]{[?]|[MESSAGE|NUMBER|OFF]}

MODE Any

BUS STATES None

QUERY RESPONSE String in the following format:
ERROR MESSAGE NUMBER OFF

EXAMPLES

```
PRINT#1,"ERROR OFF"          Disable error message reporting.

PRINT#1,"ERROR MESSAGE"      Enable error message reporting.
INPUT#1,A$:PRINT A$          Response is error message or: OK
PRINT#1,"ERROR?"                   Query current error mode.
INPUT#1,A$                   Get response
PRINT A$                     "ERROR MESSAGE"
```

FIND

The FIND command searches the record memory for a specific bus event, trigger point, first recorded location, or last recorded location. FIND returns the record memory location containing the bus event that was specified. If the event is not found, the number 65535 is returned.

Bus management and data line states are specified following the keyword MATCH. Bus management lines and data lines can be specified as asserted, unasserted, or "don't care". All bus management and data lines are "don't care" unless specified. Data lines may be specified in Binary, Decimal, Hexadecimal, or ASCII. The bus ERROR condition can be specified to find events which caused a bus error.

The trigger point is specified using the keyword TRIGGER. TRIGGER and MATCH can not be used in the same command line.

The starting location for a search may be specified using the keyword START followed by a record memory location. If no memory location is specified, the search starts at the current location pointer.

Search direction is specified with a plus sign [+] for a forward search or a minus sign [-] for a backward search. If no sign is given, search direction is forward.

The keyword FIRST specifies the location of the first recorded bus event of the most recently recorded sequence. The keyword LAST specifies the location of the last recorded bus event of the most recently recorded sequence. Only one of these may be used in a command line and the keyword MATCH may not be used.

The query option, FIND?, returns the last bus event specified using the FIND command. Response is a fixed-field string with the match pattern returned.

TYPE Record command

SYNTAX
```
FIND {[?]|MATCH[state][line]<byte>[ERROR]|
[TRIGGER][START{location}][+|-]|[FIRST|LAST]}
```

state is the state of a bus management line. A slash **[**/**]** indicates unasserted, **[**X**]** means "don't care". The absence of a slash **[**/**]** or **[**X**]** means true (asserted).

`line` is a bus management line, one or more of: `ATN`, `REN`, `EOI`, `IFC`, or `SRQ`.

`byte` is a data byte specifying the state of the data lines. `byte` may be specified in Binary (%01001010), Decimal (122), Hexadecimal (&H3F), or ASCII ('A). In binary or hexadecmal format, an `X` may be used to specify "don't care".

`location` is an address of the record memory, from 0 to 32767

`+` or `-` indicates a search direction.

`ERROR` specifies a bus error condition.

RESPONSE A five digit number, 0 to 32767, specifying a location in the record memory, or the number 65535 if no match is found.

MODE Any

BUS STATES None

QUERY RESPONSE A fixed-field string in the following format:

```
FIND MATCH[[ ][/][X]]ATN [[ ][/][X]]REN
[[ ][/][X]]EOI [[ ][/][X]]IFC [[ ][/][X]]SRQ
byte [[ ][/][X]]ERROR
```

EXAMPLES

```
PRINT#1,"FIND MATCH &H0D"
```
Find first location containing a Carriage Return.

```
INPUT#1,A$
```
Get response

```
PRINT A$
```
Response: `6`

Location 6 contains a Carriage Return character

HANDSHAKE

The HANDSHAKE command sets the Analyzer488 bus handshaking speed when functioning as either a high or low level controller. The handshake selection is used to control the speed of bus transfers when participating in bus events. Handshaking may be set to FAST, SLOW (with a specified rate), NONE, or STEP. When in the High Level Controller Mode, set handshake to FAST or NONE to prevent the Analyzer488 handshake circuitry from affecting the speed of bus events.

The query command, HANDSHAKE? returns the present handshake speed.

TYPE System

SYNTAX HANDSHAKE **{[?]|[**FAST|STEP|NONE|SLOW**{<**rate**>}]}**

rate is the number of bus events per second, from 1 to 10

RESPONSE None

REMARKS HANDSHAKE FAST is used is used for participating in bus events without limiting the speed. This selection may not be useful when viewing bus events because the display rate cannot keep up with a fast bus event rate.

HANDSHAKE SLOW, with a specified rate is useful for viewing bus data as it is transferred. The Analyzer488 participates in bus handshaking and limits the speed to 1 to 10 events per second. Selection of the exact speed is accomplished by following the command keyword SLOW with a number from 1 to 10. HANDSHAKE SLOW may also be used when recording bus events.

HANDSHAKING NONE is used to record or view bus data without the Analyzer488 appearing to be a listener. The Analyzer488 does not participate in bus handshaking and has no control of the bus event speed. HANDSHAKING NONE cannot be used when participating in bus events as an active talker or listener.

HANDSHAKING STEP allows the user to participate in bus events one event at a time. The Analyzer488 performs one bus handshake upon receipt of the STEP command. The STEP command must be used to complete each bus event when using the ASSERT and VIEW commands. However, when using the READ or WRITE commands, the STEP command is not required. STEP may be used when recording, or participating in bus events.

MODE High or Low Level Controller

BUS STATES Dependent on the current state of the bus.

QUERY RESPONSE Present handshake speed.

EXAMPLES

```
PRINT #1,"CONTROLLER OFF"    Select low level control
PRINT #1,"HANDSHAKE NONE"    Do not participate in bus
                             events
PRINT #1,"RECORD ON"         Record bus events without
                             appearing to be an active
                             listener.
```

```
PRINT #1,"CONTROLLER OFF"    Select low level control
PRINT #1,"HANDSHAKE STEP"    Participate in bus events in
                             single step speed
PRINT #1,"ASSERT REN ATN"    Assert ATN and REN to
                             cause devices to interpret data
                             as bus commands
PRINT #1,"WRITE &H3F"        Send UNL command
PRINT #1,"WRITE &H50"        Send TAG16 command
PRINT #1,"VIEW BUS"          Read current bus status
INPUT #1,A$ : PRINT A$       Print response
PRINT #1,"STEP"              Perform 1 handshake
PRINT #1,"VIEW BUS"          Read current bus status
INPUT #1,A$ : PRINT A$       Print response
```

HELLO

The HELLO command is used to verify communication with the Analyzer488, and to read the firmware revision level. When the command is sent, the Analyzer488 returns a string similar to the following:

```
Analyzer488 Revision N.N Copyright (C) 1989 IOtech Inc.
```

where N.N is the revision level of the firmware.

This command does not have a query option.

TYPE	System command
SYNTAX	HELLO
RESPONSE	See response message shown above
MODE	Any
BUS STATES	None
QUERY RESPONSE	None

EXAMPLE The following program shows a message on the front panel display of the Analyzer488 stating which key was last pressed.

```
PRINT#1,"HELLO"          Get the HELLO response
INPUT#1,A$
PRINT A$                 and display it.
```

ID

The ID command allows the user to change the @ or @@ ID command character to any printable ASCII character. The new ID command character must immediately follow the semi-colon without any intervening spaces. If the double ID character command is issued, the ID character will default back to '@'.

The @ and @@ ID command can be disabled by not including a character following the semicolon. It can be re-enabled again by issuing the ID command with a valid character.

If you anticipate that the data part of an OUTPUT or SEND command may contain the presently programmed ID character, it should be disabled.

TYPE	System command
SYNTAX	ID;**[**ascii**]** ascii is any printable ASCII character immediately following the semi-colon (;)
RESPONSE	None
MODE	Any
BUS STATES	None
QUERY RESPONSE	None
EXAMPLES	PRINT #1,"ID;#" change the ID character to # PRINT #1,"ID;" disable the ID commands PRINT #1,"ID;@" re-enable the ID character to @

KEY

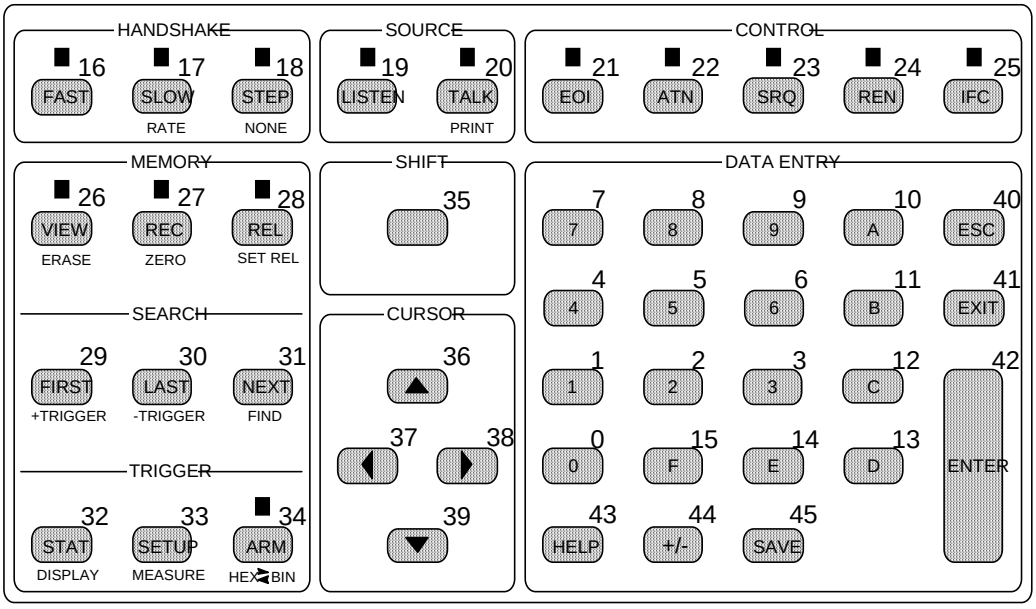
The KEY command is used to read the Analyzer488 keyboard. Upon receipt of the KEY command, the Analyzer488 waits for the current TIMEOUT period for a keypress. If a key is pressed before the TIMEOUT period, a three-digit key number from 000 to 046 will be returned. If no key is pressed before TIMEOUT, the number 255 will be returned. If another command is sent before a key is pressed or a TIMEOUT has occurred, the new command will not be executed until after a key is pressed or the TIMEOUT period has passed.

The query option, KEY?, is used to check if a key is pressed on the Analyzer488 keyboard. If a key is pressed, the Analyzer488 returns a three-digit key number, from 000 to 046. If no key is pressed, 255 is returned. The response to the KEY? command is immediate since timeouts are not used.

TYPE System command

SYNTAX KEY[?]

RESPONSE Three digit key number, from 000 to 046, or 255 if no key is pressed. The following illustration shows the key numbers for the Analyzer488 front panel keyboard:



MODE Any

BUS STATES None

QUERY RESPONSE Three digit key number, from 000 to 046, or 255 if no key is pressed.

EXAMPLES

```
10 PRINT#1,"KEY"
20 INPUT#1,A$
30 PRINT#1,"DISPLAY 'Key pressed is ";A$;"'"
40 GOTO 10
```

LISTEN

The LISTEN command sets the Analyzer488 to the Listen State when in the Low Level Controller Mode (CONTROLLER OFF). Once the LISTEN command has been executed, the READ command may be used to accept data from the bus, one event at a time.

The query option, LISTEN?, returns the current Listen or Talk State of the Analyzer488. LISTEN is returned if it is functioning in the Listen State, TALK is returned if functioning in the Talk State.

TYPE Low Level Controller command

SYNTAX LISTEN[?]

RESPONSE None

MODE Low Level Controller

BUS STATES Dependent on data sent from active talker

QUERY RESPONSE LISTEN if in Listen State, TALK if in Talk State.

EXAMPLES

```
PRINT#1,"CONTROLLER OFF"
PRINT#1,"ASSERT IFC ATN REN"
PRINT#1,"UNASSERT IFC"
PRINT#1,"TALK"
PRINT#1,"HANDSHAKE STEP"
PRINT#1,"WRITE &H54"
PRINT#1,"UNASSERT ATN"
PRINT#1,"LISTEN"
PRINT#1,"READ"
INPUT#1,A$;PRINT A$
...
...
```

LOCAL

The LOCAL command causes the Analyzer488 to unassert the Remote Enable line (REN). This causes devices on the bus to return to manual operation. If optional bus addresses are specified, bus devices at those addresses are placed in the local mode by the Go To Local (GTL) bus command. If addresses are specified, the state of the Remote Enable line is not affected.

This command does not have a query option.

TYPE	High Level Controller command	
SYNTAX	`LOCAL [addr[,addr...]]`	
	`addr` is an IEEE 488 bus device address.	
RESPONSE	None	
MODE	High Level Controller	
BUS STATES	*REN (without addresses specified) ATN•UNL, MTA, LAG,GTL (with addresses specified)	
QUERY RESPONSE	None	
EXAMPLE	`PRINT#1,"LOCAL"`	Unassert the REN Line
	`PRINT#1,"LOCAL 12,16"`	Send Go To Local to devices 12 and 16

LOCAL LOCKOUT

The `LOCAL LOCKOUT` command causes the Analyzer488 to issue a Local Lockout bus command (LLO). Bus devices that support this command are thereby inhibited from being controlled manually from their front panels.

This command does not have a query option.

TYPE	High Level Controller command
SYNTAX	`LOCAL LOCKOUT`
RESPONSE	None
MODE	High Level Controller
BUS STATES	ATN•LLO
QUERY RESPONSE	None
EXAMPLES	`PRINT#1,"LOCAL LOCKOUT"` Send Local Lockout command.

MASK

The MASK command is used to mask the most significant bit (MSB) of serial input data. Some serial host computers set the MSB (high bit) when using eight bit serial data lengths. When the MASK ON command is issued, each serial character received is logically ANDed with &H7F (127 decimal).

MASK OFF is the factory default. In default operation, all serial input data is automatically masked with &H7F. The exception to this is any data which follows a semi-colon (;), an apostrophe (') or a quotation mark ("). After a MASK ON command, all characters are masked.

The query option MASK? returns the state of the MASK command.

TYPE	System command
SYNTAX	MASK **{[**ON | OFF**]}**
RESPONSE	None
MODE	Any
BUS STATES	None
QUERY RESPONSE	The string MASK ON or MASK OFF
EXAMPLE	PRINT#1,"MASK ON"

MEASURE

The MEASURE command is used to return the speed of previous bus events. Responses may be returned in either a string format or a machine readable format.

If the command MEASURE or MEASURE0 is issued, a string in the following format will be returned: Time n.nnn m Seconds Count nnnnn Bytes Rate n.nnn Bytes/Sec.

If the command MEASURE1 is issued a machine readable string in the following format will be returned: nnnnnnnn mmmm, where nnnnnnnn is the time in microseconds and mmmm is the number of bytes transferred during that time.

Triggering, using the SET TRIGGER command, must be used to obtain correct results when using the MEASURE command. Recording must also be on (by issuing the RECORD ON command). Measuring starts when the trigger point has been found and is stopped when the record sequence has completed. Refer to the SET TRIGGER command for further information on specifying trigger conditions.

In order to ensure accurate readings, HANDSHAKE must be set to NONE or FAST during the bus events being measured.

This command does not have a query option.

TYPE Record command

SYNTAX MEASURE [number]
number is 0 or 1. If not specified, 0 is assumed.

RESPONSE Time n.nnn m Seconds Count nnnnn Bytes Rate n.nnn Bytes/Sec. (MEASURE0)
or
nnnnnnnn mmmm (MEASURE1)

REMARKS To measure the response time of a particular instrument, (the time between the device being addressed to TALK and when it sends out the first byte), set the trigger conditions as follows: MATCH = the talk address of the instrument, TRIG DELAY = 0, POST COUNT =

1. In this manner, value for `Time` will be the response time of the instrument.

To measure the data transfer rate of a particular instrument, (the time needed by the device to transfer `n` number of bytes), set the trigger conditions as follows: `MATCH` = the talk address of the instrument, `TRIG DELAY` = 1, `POST COUNT` = the larger the number here, the more accurate the reading. In this manner, the value for `Bytes/sec.` will be the desired information.

MODE Any

BUS STATES None

QUERY RESPONSE None

EXAMPLE

```
PRINT#1,"MEASURE"
INPUT#1,A$:PRINT A$
```

Read and display data transfer rate of last triggered record sequence

NEXT

The NEXT command searches the record memory and locates the next occurrence of a event specified using the FIND command. NEXT returns a five digit number which is the record memory location containing the event specified in the FIND command. If the event is not found, the number 65535 is returned.

Search direction is that which was specified when the FIND command was last issued.

This command does not have a query option.

TYPE Record command

SYNTAX NEXT

RESPONSE 5 digit number, 0-32767, specifying a location in the record memory, or 65535 if no match is found.

MODE Any

BUS STATES None

QUERY RESPONSE None

EXAMPLES

```
PRINT#1,"FIND MATCH &H0D"      Find first location containing a
                               carriage return.
INPUT#1,A$                     Get query response
PRINT A$                       Response: 6
                               Location 6 contains a carriage
                               return character

PRINT#1,"NEXT"                 Find next location containing a
INPUT#1,A$:PRINT A$            carriage return and print the
                               location.
```

OUTPUT

The OUTPUT command sends data to the IEEE 488 bus. The Remote Enable line (REN) is first asserted. If optional device addresses are specified, those devices will then be addressed to listen. If addresses are not specified, the Analyzer488 must already be configured to send data, either as a result of an immediately preceding OUTPUT command or as the result of a SEND command.

If the character count, count, is specified then count number of characters will be sent to the bus devices. Otherwise, OUTPUT terminates data transfer upon detection of a serial CR of LF terminator from the serial input. The serial input terminator(s) are replaced with the bus output terminator(s) before being sent to the bus devices.

The IEEE 488 bus output terminators can be modified with the TERM command. Refer to this command description for complete information.

TYPE High Level Controller command

SYNTAX `OUTPUT[addr[,addr…]][#count];data`

addr is a bus device address. Up to 15 addresses may be specified.
count is the number of characters to OUTPUT.
data is a string of characters to OUTPUT terminated by the serial terminator(s), unless count is specified in which case no terminator is needed.

RESPONSE None

MODE High Level Controller

BUS STATES REN (if SC), *ATN, data (without addr)
REN (if SC), ATN•MTA, UNL, LAG, *ATN, data (with addr)

QUERY RESPONSE None

EXAMPLES

```
PRINT#1,"OUTPUT 22;R0C0T1X"
```
Send "R0C0T1X" to device 22.

```
PRINT#1,"OUTPUT 06,12;ABC"
```
Send "ABC" to devices 6 and 12.

```
PRINT#1,"OUTPUT;XYZ"
```
And send them "XYZ".

```
PRINT#1,"OUTPUT 0602;DEF"
```
Send "DEF" to device 6, sec addr 2.

```
PRINT#1,"OUTPUT06#26;abcdefghijklmnopqrstuvwxyz"
```
Send the 26 letters of the alphabet without terminators to device 6.

PPOLL

The Parallel Poll command, PPOLL, causes the Analyzer488 to conduct a Parallel Poll of bus devices. It is used to request status information from many bus devices simultaneously. If a device requires service then it will respond to a Parallel Poll by asserting one of the eight IEEE 488 bus data lines (DIO1 through DIO8, with DIO1 being the least significant). In this manner, up to eight devices may be polled simultaneously by the controller. More than one device can share any particular DIO line. In this case it is necessary to perform a Serial Poll command, SPOLL, to determine which device actually requires service.

Parallel polling is often used upon detection of a Service Request (SRQ), though it may also be performed periodically by the controller. In either case, PPOLL will respond with a number from 0 to 255 corresponding to the eight binary DIO lines.

Not every device supports parallel polling. Refer to the manufacturer's documentation for each bus device to determine if Parallel Poll capabilities are supported.

This command does not have a query option.

TYPE	High Level Controller command
SYNTAX	PPOLL
RESPONSE	Number in the range of 0 to 255
MODE	High Level Controller
BUS STATES	ATN•EOI,<parallel poll response>, *EOI
QUERY RESPONSE	None

EXAMPLE

```
PRINT#1,"PPOLL"          Conduct a Parallel Poll
INPUT#1,PPSTAT           Receive the PPOLL status
PRINT PPSTAT             and print it.
```

PPOLL CONFIG

The Parallel Poll Configure command, `PPOLL CONFIG`, configures the Parallel Poll response of a specified bus device. Not all devices support Parallel Polling and, among those that do, not all support software control of their Parallel Poll response. Some devices are configured by internal switches.

The Parallel Poll response is set by `response`, a four-bit binary number (`S P2 P1 P0`). The most significant bit of `response` is the Sense (`S`) bit. The Sense bit is used to determine when the device will assert its Parallel Poll response. Each bus device has an internal individual status (*ist*). The Parallel Poll response will be asserted when this *ist* equals the Sense bit value. *ist* is normally a logic "1" when the device requires attention, so the S bit should normally also be a logic "1". If the S bit is "0" then the device will assert its Parallel Poll response when its *ist* is a logic "0", i.e. it does not require attention. However, the meaning of *ist* can vary between devices, so refer to your IEEE 488 bus device documentation.

The remaining 3 least significant bits of `response`, `P2`, `P1`, and `P0`, specify which DIO bus data line will be asserted by the device in response to a Parallel Poll. These bits form a binary number with a value from 0 through 7, specifying data lines DIO1 through DIO8, respectively.

TYPE High Level Controller command

SYNTAX `PPOLL CONFIG addr;response`

`addr` is an IEEE 488 bus address.
`response` is the decimal equivalent of the four binary bits `S`, `P2`, `P1`, and `P0`.

RESPONSE None

MODE High Level Controller

BUS STATES ATN•UNL, MTA, LAG, PPC, PPE

QUERY RESPONSE None

EXAMPLES

```
PRINT #1,"PPOLL CONFIG23;&H0D"
```

Configure device 23 to assert DIO6 when it desires service and it is Parallel Polled (&H0D = 1101 binary; S = 1, P2P1P0 = 101 = 5 decimal = DIO6).

PPOLL DISABLE

The Parallel Poll Disable command, `PPOLL DISABLE,` disables the Parallel Poll response of selected bus devices.

This command does not have a query option.

TYPE High Level Controller command

SYNTAX `PPOLL DISABLE addr[,addr…]`

`addr` is an IEEE 488 bus device address

RESPONSE None

MODE High Level Controller

BUS STATES ATN•UNL, MTA, LAG, PPC, PPD

QUERY RESPONSE None

EXAMPLE
```
PRINT#1,"PPOLL DISABLE18,06,13"
```
Disable Parallel Poll response of devices 18, 6, and 13.

PPOLL UNCONFIG

The Parallel Poll Unconfigure command, `PPOLL UNCONFIG`, disables the Parallel Poll response of all bus devices.

This command does not have a query option.

TYPE	High Level Controller command
SYNTAX	`PPOLL UNCONFIG`
RESPONSE	None
MODE	High Level Controller
BUS STATES	ATN•PPU
QUERY RESPONSE	None
EXAMPLE	`PRINT #1,"PPOLL UNCONFIG` Disable the Parallel Poll response of all bus devices.

? (PRINT)

The ? (PRINT) command returns formatted bus data or record memory data. PRINT returns bus messages for data lines and a three-letter abbreviation for each bus management line if it is asserted.

Blocks of record memory data can be printed by specifying a starting and an ending memory location, or a starting location and a length. Sending the ? command without an option will return bus data if BUS was the most recent option used, or record memory data from the location following the last location printed if location was the last option used.

The state of all bus lines is shown using the BUS option. When requesting the state of the bus using the ? BUS command, the state of the handshake lines is returned as follows: ND if Not Data Accepted (NDAC) is asserted, NR if Not Ready for Data (NRFD) is asserted, and DA if Data Available (DAV) is asserted.

If relative addressing is used (see RELATIVE), a sign, + or -, will always be printed in front of the record memory location. If relative addressing is not used, a space will be printed instead of a sign. If the memory location printed contains the trigger point, a T will be printed in the first position. If the memory location contains a bus error, the word ERROR will be printed at the end of the line.

This command does not have a query option.

TYPE Record command

SYNTAX
```
?
```
or
```
? [location] TO [location]
```
or
```
? [location] LENGTH [count]
```
or
```
? BUS
```

location is an address of the record memory. location is from 0 to 32767 if absolute addressing is used, and from -32767 to 32767 if relative addressing is used.

`count` is the number of record memory locations to be printed, from 0 to 32768.

RESPONSE Response from `? [location]` is :

<location> <bus command or ASCII data> < data (hex)> < data (decimal)> < data (binary) > <EOI state> <ATN state> <SRQ state> <REN state> <IFC state> <ERROR (if bus error)>

Response from `? BUS` is :

< BUS > <bus command or ASCII data> < data (hex)> < data (decimal)> < data (binary) > <EOI state> <ATN state> <SRQ state> <REN state> <IFC state> <DA ND NR (if all handshake lines are asserted)>

MODE Any

BUS STATES None

QUERY RESPONSE None

EXAMPLE

```
PRINT#1,"ERASE"
PRINT#1,"CONTROLLER ON"
PRINT#1,"RECORD ON"
PRINT#1,"CLEAR 20"
PRINT#1,"RECORD OFF"
PRINT#1,"?3"
```

Response is fixed field:

```
00003     SDC       04   004   0000 0100    ATN
```

PRINT#1,"?BUS" returns:

```
 BUS      SDC       04   004   0000 0100    ATN ND
```

READ

The READ command is used to accept data from the bus when in the Low Level Controller Mode and return a machine readable bus status string.

READ may only be used when the Analyzer488 is in the Listen State (see LISTEN) and handshaking is set to STEP (see HANDSHAKE). READ performs an acceptor handshake. The STEP command is not required when using the READ command since the handshake is automatically performed when READ is executed.

This command does not have a query option.

TYPE Low Level Controller command

SYNTAX READ

RESPONSE Response is dependent on the data currently on the bus. Two hexadecimal bytes are returned in the following format:

Hex Byte 1 (bit positions) (Control and Handshake Lines)								Hex Byte 2 (bit positions) (Data Lines)							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
REN	IFC	SRQ	EOI	DAV	NRFD	NDAC	ATN	D8							D1

MODE Low level controller

BUS STATES None

QUERY RESPONSE None

EXAMPLES
```
PRINT#1,"CONTROLLER OFF"
PRINT#1,"ASSERT IFC ATN REN"
PRINT#1,"UNASSERT IFC"
PRINT#1,"TALK"
PRINT#1,"HANDSHAKE STEP"
```

PRINT#1,"WRITE &H54"	Send TAG20 command
`PRINT#1,"UNASSERT ATN"`	Allow data transfer to take place
`PRINT#1,"LISTEN"`	
PRINT#1,"READ"	Read the first byte from
`INPUT#1,A$;PRINT A$`	device 20.
...	
...	

RECORD

The RECORD command enables or disables the recording of bus data. RECORD ON causes the Analyzer488 to begin recording bus events into the record memory. RECORD OFF stops recording bus events.

There is an interaction between the RECORD and the SET TRIGGER commands. If Triggering is disabled (by issuing the SET TRIGGER OFF command), recording will begin upon receipt of the RECORD ON command and will stop upon receipt of the RECORD OFF command.

If Triggering is enabled (by issuing the SET TRIGGER ON command), recording will begin upon receipt of the RECORD ON command. If WHEN COMPLETE STOP is selected, recording will stop when match count occurrences of match have occurred, trig delay events have occurred, and post count events have occurred.

If WHEN COMPLETE STEP is selected, recording will remain on until the RECORD OFF command is received. Each time the STEP command is sent, the post count value returned by the STATUS 4 and STATUS 5 commands will be incremented.

The query option, RECORD?, returns the string, RECORD ON, if record is enabled or RECORD OFF if recording is disabled.

TYPE	Record command
SYNTAX	RECORD {[?]|[ON|OFF]}
RESPONSE	None
MODE	Any
BUS STATES	None
QUERY RESPONSE	RECORD ON if record is enabled or RECORD OFF if record is disabled.

```
EXAMPLES    PRINT#1,"ERASE"
            PRINT#1,"CONTROLLER ON"
            PRINT#1,"RECORD ON"
            PRINT#1,"CLEAR 20"
            PRINT#1,"RECORD OFF"
            PRINT#1,"VIEW 2"
            INPUT#1,A$;PRINT A$     Response: 00002  0534
```

RELATIVE

The RELATIVE command enables or disables the Relative Addressing Format for use in referencing data in the record memory. RELATIVE ON puts the Analyzer488 in the Relative Addressing Format. RELATIVE OFF puts the Analyzer488 in the Absolute Addressing Format.

The query option, RELATIVE?, returns RELATIVE ON if Relative Addressing Format is enabled or RELATIVE OFF if Absolute Addressing Format is enabled.

TYPE Record command

SYNTAX RELATIVE{[?]|[ON|OFF]}

RESPONSE None

REMARKS Relative addressing is used to allow easier referencing of bus data in the record memory. For example, assume bus events have been recorded and the trigger point is at location 24356. To see the bus event that occurred 24 locations before the trigger point, you would need to subtract 24 from 24356 and use this new address (24332) for the desired location. An simpler approach would be to use relative addressing and set the relative address to the address of the trigger point. To do this, set the relative address to 24356 by issuing the SET RELATIVE 24356 command. Enable relative addressing by using the RELATIVE ON command. To specify the same location as before, use -24 as the address rather than 24332. When addresses are printed using the PRINT or VIEW commands, the relative addresses (not the physical addresses) are printed. For this example, the printed address of the trigger point would be 00000 if Relative Addressing Format is enabled.

MODE Any

BUS STATES None

QUERY RESPONSE Returns RELATIVE ON if Relative Addressing is enabled or RELATIVE OFF if Absolute Addressing is enabled.

EXAMPLES

```
PRINT#1,"ERASE"
PRINT#1,"RELATIVE OFF"
PRINT#1,"CONTROLLER ON"
PRINT#1,"RECORD ON"
PRINT#1,"CLEAR 20"
PRINT#1,"RECORD OFF"
PRINT #1,"VIEW 2"
INPUT#1,A$;PRINT A$     Returns:  00002  0534

PRINT#1,"SET RELATIVE 2"
PRINT#1,"RELATIVE ON"
PRINT#1,"VIEW 0"
INPUT#1,A$;PRINT A$     Returns: +00000  0534

PRINT#1,"SET RELATIVE 10"
PRINT#1,"VIEW -10"
INPUT#1,A$;PRINT A$     Returns: -00010  0534
```

REMOTE

The REMOTE command asserts the Remote Enable (REN) bus management line. If optional bus addresses are specified, then REMOTE addresses those devices to listen, placing them in the Remote Addressed state.

This command does not have a query option.

TYPE High Level Controller command

SYNTAX `REMOTE [addr[,addr...]]`

`addr` is an IEEE 488 bus device address

RESPONSE None

MODE High Level Controller

BUS STATES REN (without addresses)
REN, ATN•UNL, MTA, LAG (with addresses)

QUERY RESPONSE None

EXAMPLES
```
PRINT #1,"REMOTE"
```
Assert Remote Enable

```
PRINT #1,"REMOTE16,28"
```
Assert Remote Enable and address devices 16 and 28 to listen.

RESET

The High Level Controller Mode command RESET provides a warm start of the Analyzer488 interface. Issuing the RESET command clears the serial input (pending commands) and output (pending data) buffers and reinitializes the internal IEEE controller hardware. It is equivalent to issuing the following commands:

```
ABORT
DISARM
ERROR OFF
LOCAL
TIME OUT 10
```

The RESET command provides a warm start of the interface as well as clearing all error conditions. Upon detection of the RESET command, the Analyzer488 unasserts its serial output handshake line (RTS). It reasserts RTS when it is capable of accepting serial input data. If XON/XOFF handshake is selected, the handshake state is not effected by the RESET command.

This command does not have a query option.

TYPE	High Level Controller
SYNTAX	RESET
RESPONSE	None
MODE	High Level Controller
BUS STATES	IFC,*IFC,*REN
QUERY RESPONSE	None
EXAMPLE	PRINT#1,"RESET"

RESUME

The RESUME command unasserts the Attention (ATN) bus management line. As the Active Controller, ATN is normally kept asserted by the Analyzer488, but, it must be unasserted to allow transfers to take place between two peripheral devices. To allow transfers, the Analyzer488 SENDs the appropriate talk and listen addresses, and ATN must be unasserted with the RESUME command.

This command does not have a query option.

TYPE	High Level Controller command	
SYNTAX	RESUME	
RESPONSE	None	
MODE	High Level Controller	
BUS STATES	*ATN	
QUERY RESPONSE	None	
EXAMPLE	PRINT#1,"RESUME"	unassert ATTENTION line.

SEND

The SEND command is used when the Analyzer488 is in the High Level Controller Mode to send low level bus sequences. SEND provides byte-by-byte control of data and control transfers on the bus and can specify exactly which operations will be executed by the Analyzer488.

This command does not have a query option.

TYPE High Level Controller command

SYNTAX `SEND [;] sub-command[sub-command…]`

RESPONSE None or device data

REMARKS The following sub-commands are available within the SEND command:

`UNT` Send the multiline Untalk command. ATN is asserted.

`UNL` Send the multiline Unlisten command. ATN is asserted.

`MTA` Send My Talk Address (Analyzer488). ATN is asserted.

`MLA` Send My Listen Address (Analyzer488). ATN is asserted.

`TALK addr`
Send Talk Address `addr` device (TAG). ATN is asserted.

`LISTEN addr[,addr…]`
Send Listen Addresses (LAG). ATN is asserted.

`DATA {'data'|char[,char…]}`
Send character strings, `data`, or characters with numeric ASCII values, `char`. ATN is unasserted.

`EOI {'data'|char[,char…]}`
Send character strings, `data,` or characters with numeric ASCII values, `char.` ATN is unasserted. EOI is asserted on the last character.

`CMD {'data'|char[,char…]}`
Send character strings, `data,` or characters with numeric ASCII values, `char.` ATN is asserted.

`ENTER` Request data from a device terminating on LF. ATN is unasserted.

The `DATA`, `EOI` and, `CMD` sub-commands send data bytes or characters over the bus. The characters to be sent are specified either as a quoted string (`'data'`) or as individual ASCII values (`char[,char…]`). For example, `DATA'R0X'` sends the characters `R`, `0`, and `X` to the active listeners, and `DATA 13,&H0A` sends carriage-return and line-feed. Multiple ASCII `char` bytes may be specified by separating them with commas.

The `EOI` sub-command is identical to the `DATA` sub-command except that the End Or Identify (EOI) signal is asserted on the transfer of the last character.

The `CMD` sub-command sends the data bytes with Attention (ATN) is asserted. This tells the bus devices that the characters are to be interpreted as IEEE 488 bus commands, rather than as data. EOI is not asserted during `CMD` transfers. For example `CMD &H3F` is the same as Unlisten (`UNL`). Note that it is not possible to assert EOI during the transfer of a command byte because EOI and ATN together specify an IEEE parallel poll command.

The `ENTER` sub-command inputs data from the active talker, which has been set to Talk from either a `TALK addr` sub-command, or a previous `ENTER` command. Addresses are not allowed to be specified as options to this sub-command. The `ENTER` sub-command will terminate upon detection of a Line Feed (LF) character.

Note that the maximum length of the `SEND` command, including any sub-commands, is 127 characters. If large amounts of

data must be transferred using the SEND command, then multiple SEND commands must be used so that they are each less than 127 characters long. For example…

```
PRINT#1,"SEND UNT UNL MTA LISTEN 16 DATA
1,2,3,4,5,6"
```

is equivalent to…

```
PRINT#1,"SEND UNT UNL MTA LISTEN 16"
PRINT#1,"SEND DATA 1,2,3"
PRINT#1,"SEND DATA 4,5,6"
```

This way, a long SEND command can be broken up into shorter commands.

MODE High Level Controller

BUS STATES User defined

QUERY RESPONSE None

EXAMPLES

```
PRINT#1,"SEND MTA UNL LISTEN 16 DATA
'T1S0R2X'"
```

is the same as:

```
PRINT#1,"OUTPUT16;T1S0R2X"
```

```
PRINT#1,"SEND CMD128,0,10 DATA156,35
EOI'ABC'"
```

sends the following byte sequence:

Data	ATN	EOI
10000000	ATN	*EOI
00000000	ATN	*EOI
00001010	ATN	*EOI
10011100	*ATN	*EOI
00100011	*ATN	*EOI
01000001	*ATN	*EOI
01000010	*ATN	*EOI
01000011	*ATN	EOI

SET RELATIVE

The SET RELATIVE command is used to specify the reference pointer (location 00000) of the record memory to be used in the Relative Addressing Format (see RELATIVE ON). The record memory reference pointer must be a number from 0 to 32767 when using absolute addressing and from -32767 to +32767 when using relative addressing.

The query option, SET RELATIVE?, returns the current reference pointer.

TYPE Record command

SYNTAX `SET RELATIVE{[?]|[location]}`
`location` is a memory address of the record memory

RESPONSE None

MODE Any

BUS STATES None

QUERY RESPONSE The string SET RELATIVE nnnnn
nnnnn is a five digit decimal number, from -32767 to +32767.

EXAMPLES

```
PRINT#1,"ERASE"
PRINT#1,"RELATIVE OFF"
PRINT#1,"CONTROLLER ON"
PRINT#1,"RECORD ON"
PRINT#1,"CLEAR 20"
PRINT#1,"RECORD OFF"
PRINT #1,"VIEW 2"
INPUT#1,A$;PRINT A$     Returns:  00002  0534
PRINT#1,"SET RELATIVE 2"
PRINT#1,"RELATIVE ON"
PRINT#1,"VIEW 0"
INPUT#1,A$;PRINT A$     Returns: +00000  0534
PRINT#1,"SET RELATIVE 10"
PRINT#1,"VIEW -10"
INPUT#1,A$;PRINT A$     Returns: -00010  0534
```

SET TRIGGER

The SET TRIGGER command is used to configure one or more triggering parameters used in recording bus events.

There is an interaction between the RECORD and the SET TRIGGER commands. If Triggering is disabled (by issuing the SET TRIGGER OFF command), recording will begin upon receipt of the RECORD ON command and will stop upon receipt of the RECORD OFF command.

If Triggering is enabled (by issuing the SET TRIGGER ON command), recording will begin upon receipt of the RECORD ON command. If WHEN COMPLETE STOP is selected, recording will stop when match count occurrences of match have occurred, trig delay events have occurred, and post count events have occurred.

If WHEN COMPLETE STEP is selected, recording will remain on until the RECORD OFF command is received. For each bus event, the post count value returned by the STATUS 4 and STATUS 5 commands will be incremented.

The query option, SET TRIGGER? returns all current trigger parameters.

TYPE Record command

SYNTAX
```
SET
TRIGGER{[?]|[ON|OFF][MATCH[state][line][byte]]
  [# OF MATCHES[match count]][TRIG DELAY[delay]]
  [POST COUNT[post count]]
  [WHEN COMPLETE [STEP|STOP]]}
```

RESPONSE None

REMARKS The following parameters may be set:

[ON|OFF**]** Enables or disables triggering.

[state**]** state is the state of a bus management line. A slash **[/]** indicates unasserted, **[**X**]** means "don't care". The absence of a slash **[/]** or **[**X**]** means true (asserted). Any bus management line is "don't care" if not specified.

[line**]** line are bus management lines, one or more of: ATN, REN, EOI, IFC, SRQ, or ALL.

[byte**]** Selects a bus data byte trigger condition. byte may be specified in Binary (%01001010) (all eight digits must be specified), Decimal (122), Hexadecimal (&H3F) (both digits must be specified), or ASCII ('A). In binary or hexadecimal format, an X may be used to specify "don't care".

OF MATCHES **[**match count**]**
Skip match count occurrences of the trigger pattern before recording data. match count is from 1 to 65,535.

TRIG DELAY **[**delay count**]**
Skip delay count bus events after the trigger pattern is matched before recording data. delay count is from 0 to 9,999,999.

POST COUNT **[**post count**]**
Record post count bus events after match count occurrences of the trigger pattern and delay count events have occurred. post count is from 0 to 32,767.

WHEN COMPLETE **[**STEP | STOP**]**
Select the action taken at the end of the recording sequence. If STEP is selected, HANDSHAKE will be changed to STEP at the end of the recording sequence. If STOP is selected, data recording will stop at the end of the recording sequence and the bus will run at the current HANDSHAKE speed.

MODE Any

BUS STATES None

QUERY RESPONSE Returns all current trigger parameters.

EXAMPLES
```
PRINT#1,"SET TRIGGER ON # OF MATCHES 1  MATCH
SRQ &HXX TRIG DELAY 15000 POST COUNT 15000
WHEN COMPLETE STOP"
```

```
PRINT#1,"SET TRIGGER?"      Request trigger selections
INPUT#1,A$:PRINT A$         Print response
```

To trigger on DIO1 and DIO2 asserted, DIO3 and DIO4 don't cares, and DIO5 through DIO8 unasserted (using binary format):

```
PRINT#1,"SET TRIGGER ON MATCH %0000 XX11 # OF
         MATCHES 1 TRIG DELAY 0 POST COUNT 10
         WHEN COMPLETE STEP"
```

SPOLL

The Serial Poll command, SPOLL, performs a Serial Poll of bus devices and responds with number from 0 to 255 representing the decimal equivalent of the eight-bit device response. Serial Polls are normally performed in response to assertion of the Service Request (SRQ) bus signal by some bus device. If *rsv* (DIO7, decimal value 64) is set, then that device is signaling that it requires service. The meanings of the other bits are device-specific.

If SPOLL is issued with multiple addresses specified, the Analyzer488 will perform a serial poll on each device and send the output of each response to the serial port with the serial output terminator(s) appended.

With no bus address specified, the SPOLL command returns the external SRQ status. If the SRQ line is asserted, the Analyzer488 will return a "64". If it is not asserted, the Analyzer488 will return a "0".

This command does not have a query option.

TYPE	High Level Controller command	
SYNTAX	SPOLL [addr[,addr…]]	
	addr is an IEEE 488 bus device address	
RESPONSE	0 or 64	(without addr)
	0 to 255	(with addr)
MODE	High Level Controller	
BUS STATES	ATN•UNL, MLA, TAG, SPE, *ATN, data, ATN•SPD, UNT	
QUERY RESPONSE	None	

EXAMPLES

```
PRINT#1,"SPOLL 16"          Serial Poll  device 16
INPUT#1,SPSTAT              Receive the Spoll status
IF SPSTAT AND 64 THEN…      Test rsv…

PRINT#1,"SPOLL"             Check the SRQ status
INPUT#1,SRQ
IF SRQ<>0 THEN…             If SRQ is asserted then …

PRINT#1,"SPOLL 10,12,16"
INPUT#1,SP10,SP12,SP16      Get SPOLL response from devices
                            10, 12 and 16.
```

STATUS

STATUS is a system command used to read the Analyzer488 operational status and data recording conditions. STATUS may be used to return the functional mode, current IEEE 488 bus address, extended internal status, error status, progress of the current recording sequence, and information about the most recent record sequence.

This command does not have a query option.

TYPE System command

SYNTAX STATUS [;][number]

number is 0 to 5. If not specified, 0 is assumed.

RESPONSE Character string as described below.

REMARKS The following STATUS information is available:

STATUS Returns CONTROLLER <addr> if in High Level Controller Mode, where addr is the IEEE 488 bus device address of the Analyzer488. Returns the string: "LOW LEVEL", if in Low Level Controller Mode.

STATUS 1 Returns the string: "LOW LEVEL" if in Low Level Controller Mode. If in High Level Controller Mode, returns an extended internal status string in the following format:

STATUS1 string format: m nn Gn s Sn Enn Tn Cn msg

Item		Starting Col.	# Cols.	Value
m	Operating mode	1	1	C: Controller
nn	Bus address	3	2	Two-digit decimal number 00 to 30
Gn	Address change	6	2	G0: Address status change has not occurred. G1: Address status change has occurred.

S	Addressed state	9	1	T: Talker L: Listener I: Idle
Sn	Service Request	11	2	S0: SRQ is not asserted. S1: SRQ is asserted.
Enn	Error code	14	3	Enn: Letter 'E' followed by two-digit error code. Refer to Appendix B for Error Codes.
Tn	Triggered	18	2	T0: No IEEE Trigger command received.
Cn	Cleared	21	2	C0: No IEEE Clear command received.
Msg	Error description	24	17	Text of error message

`m` The Operating Mode (C) indicates that the Analyzer488 is the Active Controller.

`nn` The Bus Address is the IEEE 488 bus device address assigned to the Analyzer488 by the `ADDRESS` command or the front panel configuration procedure.

`Gn` The Address Change (G0, G1) indicator is set whenever the Analyzer488 changes from the idle state to a Talker or Listener, or from a Talker or Listener state to an idle state. It will not indicate when a change is made from a listener to a talker or a talker to a listener. The address change is reset when `STATUS 1` is read.

`S` The Addressed State is the current talker/listener state of the Analyzer488.

`Sn` The Service Request indicator reflects the external SRQ status. If the SRQ line is asserted, S1 will be reported. If it is unasserted, S0 will be reported.

`Enn` The Error Code is `00` when no error has occurred. If it is non-zero, then the appropriate error message is appended to the `STATUS 1` response. For more details about the individual errors, refer to Appendix B. The Error Code is reset to `00` when the `STATUS` response is read.

`Tn` The Triggered (T0) and Cleared (C0) indicators never change.
`Cn` Since the Analyzer488 is always a system controller, it will never receive a GET (Group Execute Trigger) or SDC/DCL (Selected Device Clear/Device Clear) bus command.

`STATUS 2` If an ERROR has occurred, issuing `STATUS 2` will cause a two digit error code to be returned. If no error has occurred, the value 0 is returned. Once the error message is sent by the Analyzer488, the error condition is cleared. Refer to Appendix B for error message explanations.

`STATUS 3` Returns the state of the Trigger and Record circuitry and the progress of the recording sequence. A five digit binary number will be returned with each bit representing the following:

`Format : ABCDE`

Digit A - Trigger status, 1 = On
Digit B - Trigger Found status, 1 = Trigger pattern found
Digit C - Record status, 1 = On
Digit D - Sequence status, 1 = Sequence complete
Digit E - At End selection, 1 = Step

`STATUS 4` Returns information about the most recent recording sequence. A two-word description, followed by a numeric value will be returned for each parameter. Each description and number is a fixed field string with leading zeros for numeric values. The entire response is sent as one string.

CONDITION	DESCRIPTION	RANGE
LAST TRIGGER LOC	Last Trigger Location	0 to 32767 *
LAST LOC WRITTEN	Last Location Written	0 to 32767 *
PRE TRIG COUNT	Pretrigger Count	0 to 99,999,999
PRE AVAILABLE	Pretrigger Recorded	0 to 32676
POST COUNT	Post Count	0 to 99,999,999
POST TRIG AVAILABLE	Post Trigger Recorded	0 to 32767
FIRST SEQ AVAILABLE	First Location Recorded	0 to 32767 *
TOTAL COUNT	Total Count	0 to 99,999,999
TOTAL AVAILABLE	Total Recorded	0 to 32767

* Returns 65535 if invalid

```
For example:   LAST TRIG LOC 14251 LAST LOC WRITTEN
           31362 PRE TRIG COUNT 00043234 PRE AVAILABLE
           30234 POST COUNT 07287434 POST TRIG AVAILABLE
           26234 FIRST SEQ AVAILABLE 30437 TOTAL COUNT
           00030437 TOTAL AVAILABLE 30437
```

`STATUS 5` Returns information about the most recent recording sequence. The response is the same as for `STATUS 4`, except that only numbers are returned.

```
For example:   14251 31362  00043234 30243    07287434
               26234    30437    00030437 30437
```

MODE Any

BUS STATES None

QUERY RESPONSE None

EXAMPLES

```
PRINT#1,"STATUS"
INPUT#1,A$
PRINT A$
```

Read the Analyzer488 status and display it.

```
CONTROLLER 10
```

Example of displayed `STATUS 0`

```
PRINT#1,"STATUS1"
INPUT#1,A$
PRINT A$
```

Read the Analyzer488 extended status and display it.

```
C 10 G0 I S0 E00 T0 C0 OK
```

Example of `STATUS 1`

```
PRINT#1,"STATUS2"
INPUT#1,A
PRINT A
```

Read the Analyzer488 error status and display it.

```
0
```

Example of displayed `STATUS 2`

STEP

The STEP command causes the Analyzer488 to perform one source or acceptor handshake when STEP handshake is selected. The STEP command is used to complete a handshake and allow new data to be placed on the bus (in Talk State), or received from the bus (in Listen State).

The query option, STEP?, returns 1 (True) if there is data on the bus to be read or if a bus device is waiting for data. STEP? returns 0 (False) otherwise.

TYPE	Low Level Controller command
SYNTAX	STEP [?]
RESPONSE	None
MODE	Low Level Controller
BUS STATES	Dependent on data on the bus
QUERY RESPONSE	Returns 1 (True) if there is data on the bus to be read or if a bus device is waiting for data. Returns 0 (False) otherwise.

EXAMPLES

```
PRINT#1,"CONTROLLER OFF"
PRINT#1,"ASSERT IFC ATN REN"
PRINT#1,"UNASSERT IFC"
PRINT#1,"TALK"
PRINT#1,"HANDSHAKE STEP"
PRINT#1,"WRITE &H54"        Send TAG20 command
PRINT#1,"UNASSERT ATN"
PRINT#1,"LISTEN"
PRINT#1,"VIEW BUS"          View data from device 20
INPUT#1,A$:PRINT A$         one byte at a time
PRINT#1,"STEP"
PRINT#1,"VIEW BUS"
INPUT#1,A$:PRINT A$
PRINT#1,"STEP"
PRINT#1,"VIEW BUS"
```

STERM

The STERM command sets the end-of-line terminators for output to the serial host. All output from the Analyzer488 through the serial port is terminated by the serial output terminator(s). All input from the serial host must be terminated by either a Line Feed (LF) or Carriage Return (CR) except OUTPUT #count.

During INPUT, the Analyzer488 takes the data it receives from the bus device until it detects the LF or other optionally specified input terminator. It strips all CR and LF from the input data and appends the serial output terminator(s) before sending it to the serial host. The default serial terminators for output are factory set for CR LF.

The query command STERM? returns the current serial output terminators.

TYPE	System command
SYNTAX	STERM[;]{[?]|term[term]|[NONE]} term is one of CR, LF, $char, or 'X, specifying a terminator character.
RESPONSE	None
MODE	Any
BUS STATES	None
QUERY RESPONSE	Returns the string STERM followed by the current serial output terminators.

EXAMPLES

```
PRINT#1,"STERM CR"
```
Set the serial output terminator to carriage-return.

```
PRINT#1,"STERM NONE"
```
Disable sending any serial output terminators.

```
PRINT#1,"STERM $&H0D"
```
Set the serial output terminator to carriage-return.

```
PRINT#1,"STERM $0"
```
Set the serial output terminator to NULL.

TALK

The TALK command is used to allow the Analyzer488 to place data on the bus. The TALK command can only be used in the Low Level Controller Mode (CONTROLLER OFF).

The query command, TALK?, returns the current status of the Analyzer488. TALK is returned if it is in Talk State, LISTEN is returned if it is in Listen State.

TYPE Low Level Controller command

SYNTAX TALK [?]

RESPONSE None

MODE Low Level Controller

BUS STATES None

QUERY RESPONSE TALK if in Talk State, LISTEN if in Listen State.

EXAMPLES
```
PRINT#1,"CONTROLLER OFF"
PRINT#1,"ASSERT REN ATN"
PRINT#1,"HANDSHAKE STEP"
PRINT#1,"TALK"
PRINT#1,"WRITE &H54"        Send TAG20 command
```

TERM

The TERM command sets the end-of-line terminators for output from the Analyzer488 to IEEE 488 bus devices. All output to bus devices, except `OUTPUT #count`, is terminated by the IEEE 488 bus output terminator. All `ENTER` input from bus devices is terminated on a Line Feed (LF) or optionally specified with the `ENTER` command.

During `OUTPUT`, the Analyzer488 takes the data it receives from the user's program, strips all CR and LF characters from it (except `OUTPUT #count`) and appends the IEEE 488 bus output terminator before sending it to the bus device. The default terminators for output are factory set to `CR LF`, which is appropriate for most bus devices.

`EOI` has a different meaning when specified for input than when it is specified for output. During input, `EOI` specifies that input will be terminated upon detection of the End Or Identify (EOI) bus signal, regardless of which characters have been received. During output, `EOI` specifies that the EOI bus signal is to be asserted during the last byte transferred.

The query command `TERM?` returns the current IEEE 488 bus terminators.

TYPE	High Level Controller command
SYNTAX	`TERM[;]{[?]|[term[term][EOI]|[EOI]|[NONE]]}` `term` is one of `CR`, `LF`, `$char`, or `'X`, specifying a terminator character.
RESPONSE	None
MODE	High Level Controller
BUS STATES	None
QUERY RESPONSE	The string `TERM` followed by the current IEEE 488 bus terminators.

EXAMPLES

```
PRINT#1,"TERM CR LF EOI"
```
Set output bus terminators to carriage-return, line-feed, with EOI asserted on line-feed.

```
PRINT#1,"TERM LF EOI"
```
Set output term to LF with EOI.

```
PRINT#1,"TERM 'Z"
```
Set bus term to the letter "Z".

```
PRINT#1,"TERM $0 EOI"
```
Set output term to NULL with EOI.

TIMEOUT

The TIMEOUT command sets the number of seconds that the Analyzer488 will wait for a bus event before declaring a time out error. The Analyzer488 checks for time out errors on every byte it transfers, including command bytes. When using the KEY command, the Analyzer488 will wait the TIMEOUT period for a keypress.

Time out checking may not be suppressed however it may be set to a maximum time of about 18 hours.

The query command TIMEOUT? returns the current timeout value.

TYPE	System command
SYNTAX	TIMEOUT[;][n]
	n is the number of seconds to allow in the range of 1 to 65535.
RESPONSE	None
MODE	Any
BUS STATES	None
QUERY RESPONSE	The string TIMEOUT followed by the timeout value.
EXAMPLES	PRINT#1,"TIME OUT 10" Wait 10 sec before time out
	PRINT#1,"TIME OUT 3600" Wait an hour before time out

TRIGGER

The TRIGGER command issues a Group Execute Trigger (GET) bus command to specified devices. If no addresses are specified, then the TRIGGER command issues a GET to only those devices that are already in the listen state as a result of a previous OUTPUT or SEND command.

This command does not have a query option.

TYPE	High Level Controller command
SYNTAX	TRIGGER[addr[,addr…]]
	addr is an IEEE 488 bus device address to be triggered.
RESPONSE	None
MODE	High Level Controller
BUS STATES	ATN•GET (without addr) ATN•UNL, MTA, LAG, GET (with addr)
QUERY RESPONSE	None
EXAMPLES	PRINT#1,"TRIGGER02,04,16" Issue Group Execute Trigger to devices 2, 4, and 16.
	PRINT#1,"TRIGGER" Trigger all current listeners.

UNASSERT

The UNASSERT command unasserts one or more bus management and/or data lines. Data line parameters may be specified following the keyword DATA in Binary (%01001010) (all eight digits must be specified), Decimal (122), Hexadecimal (&H3F) (both digits must be specified), or ASCII ('A). If UNASSERT ALL is issued while the Analyzer488 is in the TALK state, all data and management lines will be unasserted.

This command does not have a query option.

TYPE	Low Level Controller command
SYNTAX	UNASSERT {[line][ALL]DATA[<data>]} ALL unasserts all lines line is one of: SRQ, ATN, EOI, IFC, or REN data is numeric data in Binary, Decimal, Hexadecimal, or ASCII
RESPONSE	None
MODE	Low Level Controller
BUS STATES	Bus states are defined by the usage of the commands.
QUERY RESPONSE	None

EXAMPLES

```
PRINT#1,"ASSERT IFC"      Pulse IFC line
PRINT#1,"UNASSERT IFC"    Similar to ABORT command
```

VIEW

The VIEW command returns data from the bus or the record memory in machine readable format. Blocks of data can be viewed by specifying a memory location, a starting and ending memory location, or a starting location and a length. If VIEW [location] is sent, issuing the VIEW command without an option returns record memory data from the next record memory location (location + 1).

The VIEW BUS command returns data from the bus without performing a handshake.

This command does not have a query option

TYPE Record command

SYNTAX
```
VIEW
or
VIEW [location]
or
VIEW [location] TO [location]
or
VIEW [location] LENGTH [count]
or
VIEW BUS
```

location is an address of the record memory from 0 to 32767 if absolute addressing is used, and from -32767 to 32767 if relative addressing is used.
count is the number of locations to VIEW, from 1 to 32768.

RESPONSE If VIEW BUS is issued, response is the word BUS followed by a four digit hex number. The first two digits show the state of bus management and handshake lines. The second two digits show the state of bus data lines.

If VIEW location is issued, response is record memory location followed by a four digit hex number. The first two digits show the state of bus management lines and whether the location contains

valid data, a bus error, or the trigger point. The second two digits show the state of the bus data lines.

REMARKS For `VIEW BUS`, format is as follows:

Byte 1 (bit positions) (Control and Handshake Lines)								Byte 2 (bit positions) (Data Lines)							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
REN	IFC	SRQ	EOI	DAV	NRFD	NDAC	ATN	D8							D1

For `VIEW location`, format is as follows:

Byte 1 (bit positions) (Control and Status Lines)								Byte 2 (bit positions) (Data Lines)							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
REN	IFC	SRQ	EOI	BERR	VALID	TRIG	ATN	D8							D1

MODE Any

BUS STATES None

QUERY RESPONSE None

EXAMPLES

```
PRINT#1,"CONTROLLER ON"
PRINT#1,"CLEAR 20"
PRINT#1,"VIEW BUS"
INPUT#1,A$;PRINT A$
```

Response: `BUS 0304`

```
PRINT#1,"ERASE"
PRINT#1,"CONTROLLER ON"
PRINT#1,"RECORD ON"
PRINT#1,"CLEAR 20"
PRINT#1,"RECORD OFF"
PRINT#1,"VIEW 2"
INPUT#1,A$;PRINT A$
```

Response: `00002 0534`

WRITE

The WRITE command is used to put a data byte on the bus and initiate a source handshake. WRITE may only be issued when the Analyzer488 is in the Talk State. Handshaking must be set to STEP before using the WRITE command. The STEP command is not required when using the WRITE command since the handshake is automatically performed when the WRITE command is executed.

This command does not have a query option.

TYPE Low Level Controller command

SYNTAX `WRITE [data]`

data may be Decimal or Hexadecimal data

RESPONSE None

MODE Low Level Controller

BUS STATES Determined by data written

QUERY RESPONSE None

EXAMPLES
```
PRINT#1,"CONTROLLER OFF"
PRINT#1,"ASSERT REN ATN"
PRINT#1,"HANDSHAKE STEP"
PRINT#1,"TALK"
PRINT#1,"WRITE &H54"        Send TAG20 bus command
```

ZERO

The ZERO command sets the record memory reference pointer to location 0 without erasing the record memory. When the next recording sequence is started, the first bus event will be stored at location 0 in the record memory.

This command does not have a query option.

TYPE	Record command
SYNTAX	ZERO
RESPONSE	None
MODE	Any
BUS STATES	None
QUERY RESPONSE	None
EXAMPLES	PRINT#1,"ZERO"

Analyst488 Software

5.1 Introduction

This section contains a detailed explanation of the Analyst488 software that is provided with the Analyzer488 IEEE 488 Bus Analyzer. The following paragraphs contain a description of the software, information regarding configuration and setup, an explanation of how to use the menus, screens, and windows, and a detailed explanation of all the commands.

5.2 Description

The Analyst488 software program, ANALYST, that is provided with the Analyzer488 let's you operate the Analyst488 from an IBM PC, PS/2, or compatible personal computer using a simple menu-driven interface. Pull-down menus and Pop-up windows can be activated by using a standard Microsoft mouse, cursor keys, or shortcut keys.

5.3 Setup

To setup the Analyzer488 for use with Analyst488 software, connect it to the PC through the RS-232 connectors and to the IEEE 488 bus through the connector on the rear panel.

5.4 Configuration

5.4.1 Analyzer488 Configuration

To use the Analyzer488 with Analyst488 software, The Analyzer488 must be configured to operate in the Serial Controller Mode using identical configuration parameters on the PC and the Analyzer488. Refer to the Getting Started section of this manual for configuration details.

5.4.2 PC Configuration

To use your personal computer with the Analyzer488, the PC must be configured. The Analyst488 software automatically configures the program to your system, but you can change the configuration if necessary. Using the File Setup menu, you can change the monitor type (color or

mono), color preferences, communications port, and communications parameters.

5.5 Using Analyst488 Software

The Analyst488 program has five pull-down menus on the main screen: File, Display, Record, Search and Handshake. Two windows are used to show bus data: Analyzer Window and File Window. These windows show the contents of record memory and the data from previous recording session that have been saved. Dialog boxes are used with certain commands to provide more information. Shortcut keys are also provided to quickly step through Analyst488 functions.

5.5.1 Using Menus

Drop-down menus drop down over the current screen to provide a list of functions or displays. The highlighted menu bar can be scrolled up or down to choose selected functions. To pull down a menu and display its contents, place the cursor anywhere on the menu title, then click the mouse button. The menu contents will remain visible until you move the cursor and click the mouse button off of the menu area.. If you are not using a mouse, pull down a menu by pressing and holding the <Alt> key followed by the first letter of each menu heading.

To select a command, press the letter corresponding to the highlighted letter of the desired command. For example, to pull down the File menu, press and hold <Alt> then press <F>. To select Open File, press the <O> key.

To select a menu option, place the mouse cursor on the selection you want, then press the left mouse button. If you are not using a mouse, use the cursor keys to move between selections and menus then press <Return> when the cursor is on the desired selection.

You will not be able to choose a selection if it is unavailable. For example, many selections under the Display menu will not be available unless a File or Analyzer Window has been previously opened.

5.5.2 Using Dialog Boxes

When you select certain commands, a dialog box is displayed on the screen. Dialog boxes are similar to menus in that they pop-up over the existing screen. Dialog boxes provide several options at once, rather than asking for one of several choices like a menu does. They may ask for confirmation, let you cancel a command or present further options or parameters. Press <Return> to confirm a command, <Esc> to cancel a command, or select an option using the <tab> or cursor keys.

5.5.3 Using Windows

The primary method of displaying information is through windows. These are regions of the screen which can be thought of like papers on a desktop - you may only be working on one at a time, but the others are available for reference. When large amounts of data are to be viewed and compared, windows provide the ability to view as much or as little information as desired, within the limits of the screen size.

There are two windows used to show data: the Analyzer Window and the File Window. The Analyzer Window displays data which is currently in the record memory of the Analyzer488. The File Window displays data from a file, or data which was transferred from the Analyzer Window. Data in the Analyzer Window may change if recording is used, however data in the File Window cannot be altered. All window operations such as marking, displaying and searching may be used with either window.

5.5.4 Using Shortcut Keys

Analyst488 has several shortcuts for menu commands. These shortcuts are listed under the menu in parenthesis. To use the shortcut keys, press that key from any screen.

5.6 Command Descriptions

The following paragraphs provide a detailed description of the commands listed under each of the five pull-down menus. Shortcut keys and dialog boxes used with each command are also explained.

5.7 File Menu

The file menu is used to open data windows and to manipulate blocks of data. The file menu also opens windows that let you operate the Analyzer488 as if you were using the Analyzer488 front panel controls or sending Analyzer488 serial controller commands. A setup menu is also provided to let you tailor the system configuration to meet your specific needs.

5.7.1 Open Analyzer

This command allows you to open the Analyzer Window and view bus events stored in the Analyzer488 record memory. The bus events are displayed in the format chosen by the Format commands in the Display menu. Recording bus events will change the information displayed in this window. If you wish to save data from this window for future use, mark the data to be saved using the Mark Block command then transfer the marked block to the File Window using the Transfer command.

If a File Window is also displayed, the user is prompted for the orientation of the Analyzer Window, either horizontal or vertical.

This command may also be selected without pulling down the File menu by pressing <F2>.

5.7.2 Open File

This command is used to open a File Window and load it with information that was previously saved to a file using the Write Block command. Any previous data in the window is lost. The Analyst488 will prompt for a file name or file specification (wildcards are accepted). If more than one file matches the file specification, a list of matching filenames is presented.

Once a filename is entered, its contents are loaded into the File Window. The user can then display, compare, and search the data for particular IEEE 488 bus events. This command cannot be used to open files that were saved with the Print Block command.

This command may also be selected without pulling down the File menu by pressing <F3>.

5.7.3 Close

This command closes the active window. If two windows are open, using Close will close the active window and make the previously inactive window the new active window.

This command may also be selected without pulling down the File menu by pressing <Ctrl C>.

5.7.4 Virtual Front Panel

This command opens a window that gives you direct control over the bus data and management lines. The control functions are separated into logical areas for easy identification. The right and left cursor keys are used to move between fields.

5.7.4.1 Source Control

The Talk/Listen state of the Analyzer488 may be changed by pressing the <T> and <L> keys respectively while this field is active. The <up> and <down> cursor keys may also be used to change the state.

5.7.4.2 Message Entry

The state of the data lines may be set by entering a Bus or ASCII message in this field followed by a <CR>. The state of the data and ATN lines will be set automatically.

If this field is not active, this position displays the message corresponding to the current state of the data lines.

5.7.4.3 Hexadecimal Data Entry

The state of the data lines may also be set by entering a hexadecimal digit while this field is active. Valid digits are 0-9 and A-F. They are shifted in from the right when entered. Also, the <up> and <down> cursor keys may be used to increment and decrement this value.

5.7.4.4 Character Data Entry

The state of the data lines may be set by entering any of the printable characters from the keyboard while this field is active.

When this field is inactive, this position displays the state of the data lines with Attention (ATN) unasserted, regardless of its actual state.

5.7.4.5 Binary Data Entry

The individual data lines may be asserted or unasserted from this field. Each data bit has a corresponding key on the keyboard which toggles its current state. The data lines DIO8 through DIO1 are activated by keys <A>, <S>, <D>, <F>, <G>, <H>, <J>, and <K> respectively.

5.7.4.6 Control Line Control

The individual control lines may be asserted or unasserted from this field. Each data bit has a corresponding key on the keyboard which toggles its current state. The control lines EOI, ATN, SRQ, REN, and IFC are activated by keys <E>, <A>, <S>, <R>, and <I> respectively.

5.7.4.7 Handshake Speed Control

This field allows the handshake speed of the Analyzer488 to be set when this field is active. The keys <N>, <F>, <S>, and <P> can be used to choose None, Fast, Slow, and Step respectively. In addition, the <up> and <down> cursor keys may be used to select the handshake speeds.

5.7.5 Command Line Controller

The Command Line Controller command opens a window that allows you to directly issue a subset of the Analyzer488 Serial Controller Mode bus commands.

At the CMD> prompt, you may type any of the Serial Controller commands listed below. If a particular command returns a response, the response will be printed on the screen.

The following commands may be issued when using the Command Line Controller window:

ADDRESS	LOCAL LOCKOUT	PPOLLU	TRIGGER
ABORT	OUTPUT	REMOTE	TERM
CLEAR	PPOLL	RESUME	
ENTER	PPOLLC	SEND	
LOCAL	PPOLLD	SPOLL	

This command may also be selected without pulling down the File menu by pressing <F5>.

For further information on the above commands and syntax, refer to the Serial Controller section (Section 4) of this manual.

5.7.6 Mark Block

This command sets the present cursor position as the beginning of a block of data in a window. The cursor keys may then be used to select the size of the block to be marked. The marked block is displayed in reverse video for easy identification.

This command is used to select a portion of the data in a window for writing, printing, searching, or transferring. This command is also used to remove the current block marker from the active window.

This command may also be selected without pulling down the File menu by pressing <F6>.

5.7.7 Mark All

This command is used to mark an entire block of data in a window for writing, printing, searching or transferring.

This command may also be selected without pulling down the File menu by pressing <Ctrl B>.

5.7.8 Write Block

This command is used to save a marked block of data to disk for later use. Data can be saved from either the Analyzer Window or the File Window. You may then display, compare, and search the saved data for particular IEEE 488 bus events at a later time.

When this command is selected, a dialog box will appear asking for a filename. If the named file does not exist, it is created. If a file extension is not entered, the default is .ANA. If the named file already exists, you will be asked if you want to overwrite the existing file.

This command may also be selected without pulling down the File menu by pressing <Ctrl W>.

5.7.9 Print Block

This command is used to make a printed copy of data displayed in a marked block. Data can be printed from either the Analyzer Window or the File Window. Data is printed in the current display format. The default printer port can be selected using the Setup command.

When this command is selected, a dialog box will ask for the destination of the print operation (disk or printer). If printing to an existing disk file, you will be asked if you want to overwrite the file.

If data is being printed to disk, the file will be in the form of a sequential ASCII text output file. The formatted file data can then be used by a word processing program. Data saved using this command cannot be loaded using the Open File command.

This command may also be selected without pulling down the File menu by pressing <Ctrl P>.

5.7.10 Transfer

This command is used to transfer data from a marked block in the Analyzer Window to the File Window. In this way, the information can be held for later reference without the possibility of modification by further recording of bus events.

If a File Window is not open, using the Transfer command will open a new File Window with the name "Untitled" and load it with the block of transferred data.

If a File Window is already open, the Transfer command may be used to replace the old data with the transferred data.

5.7.11 DOS Shell

This command is used to temporarily exit to DOS. You may then perform DOS functions including running another program. To return to the Analyst488 program, type '`exit`' at the DOS prompt. No data is lost in the Analyst488 program while in the shell.

5.7.12 Setup

The Setup command allows the user to setup miscellaneous program options. A dialog box will appear allowing you to setup one of the following: serial port configuration, program options (type of monitor, printer form length). If Serial Port is selected, a dialog box will prompt for the Com port, baud rate, number of stop bits, number of data bits, and parity.

Using the cursor keys, select the serial port communication settings to match the settings of the Analyzer488. The Serial port configuration of the Analyzer488 may be changed by using the power-on configuration procedure. To do this, press and hold the `[SETUP]` key on the front panel of the Analyzer488 and turn the unit on. Refer to section 2 of this manual for further information.

If factory default setup is selected the serial port of the PC will be set to the factory default configuration which is: COM1, 9600 Baud, 8 data bits, 2 stop bits, no parity. If the Analyzer488 is also set to its factory default, communication with the Analyzer488 may be established.

If program options is selected, a window will appear. Using this window, you can select the type of monitor and printer form length for your system.

5.7.13 Quit

The Quit command is used to quit the Analyst488 program and exit to DOS. This action signifies that you are finished with the Analyst488 program. All data in open windows that were not previously saved are lost. A dialog box will appear asking for confirmation before exiting the Analyst488 program.

This command may also be selected without pulling down the File menu by pressing <Ctrl Q>.

5.8 Display Menu

The display menu is used to let you change the way you look at the data on the screen. Message and addressing formats can be set and the orientation of the screens can be changed.

5.8.1 Format

This command opens a dialog box which allows you to select the format of data in the active window.

5.8.1.1 Message

Selecting Message will display bus events in the format shown below:

```
00021    TAG21    55   085   0101 0101    ATN   REN
```

The leftmost field is the location of the data in the Analyzer488 record memory. If this location contained the trigger point, a 'T' would be displayed in front of the location.

The next field will show the IEEE 488 bus command (if ATN is asserted) corresponding to the data on the bus. If Attention is not asserted, the equivalent ASCII character or code for data is shown.

The next three fields show the Hexadecimal, Decimal and Binary equivalent of the data at that location.

The last five fields show the state of the IEEE 488 bus management lines. If a line is asserted, the abbreviation for that line will be displayed. The abbreviations for each line are: ATN (Attention), REN (Remote Enable), IFC (Interface Clear), SRQ (Service Request), and EOI (End or Identify).

If a Bus Error occurred during the data transfer, the word ERROR will appear on the far right of the display line.

5.8.1.2 Hexadecimal

Selecting Hexadecimal will display data in hexadecimal format. Bus events occurring while a particular bus management line was asserted can be highlighted by using the Hex Highlight command.

The leftmost number is the starting location of the data displayed on that line. Following that location are 16 hexadecimal bytes corresponding to the state of the data lines for each bus event. After the last hexadecimal byte is a text line showing the equivalent ASCII representation for each bus event.

5.8.1.3 Hex Highlight Mode

When this command is selected, a menu will appear asking for the management line to use for highlighting of data bytes. Hex Highlight Mode allows for easy identification of bus events that occurred when a particular management line was in the asserted state.

Hex Highlight Mode may be used at any time, but its effect will not be apparent except with data displayed in Hexadecimal format.

The default Highlight is Attention (ATN). If Trigger is selected, the data transfer that was the trigger point will be highlighted.

5.8.1.4 Mixed

Selecting Mixed Format displays bus events in a composite message/hexadecimal format. Information is shown in two columns: bus commands and ASCII data in the left column and the hexadecimal

equivalent of the bus events in the other. Each bus command or data string is separated by commas. Printable data are shown as ASCII characters enclosed by quotation marks.

5.8.2 Absolute

This command displays memory locations in the active window as Absolute locations (data stored at location 50 will have an address of 50). In this manner, memory locations are numbered from 0 to 32767 and correspond to the actual memory locations in the Analyzer488. This is the default display format.

5.8.3 Relative

This command displays memory locations in the active window as relative locations (data stored at location 50 could be at any address depending on the reference). Locations are numbered relative to the reference location which was set by the Set Reference command. Negative addresses indicate events that have occurred prior to the reference location and positive addresses indicate events occurring after the reference location.

The default reference location is set to the first data byte available in the most recent record sequence.

5.8.4 Set Reference

This command sets the reference location to the current location of the cursor in the active window.

When using the relative display format, this reference location will be labeled '0'. All locations prior to the cursor location will be negative and all locations after the cursor will be positive.

5.8.5 Horizontal

This command allows you to change the orientation of the two windows on the screen. If the File Window and the Analyzer Window are displayed vertically, you can display them horizontally by selecting Horizontal.

5.8.6 Vertical

This command allows you to change the orientation of the two windows on the screen. If the File Window and the Analyzer Window are displayed horizontally, the user can display them vertically by selecting Vertical.

5.9 Record Menu

The record menu is used to setup recording and triggering. This menu also lets you erase and reset the record memory.

5.9.1 Record Enable

This command enables or disables recording of bus events. Each time the command is issued, the recording state will change. When recording is enabled, a checkmark will appear in front of the word Record and the message 'Recording...' will appear on the top row of the screen. When recording is disabled, nothing will appear in front of the word Record.

This command may also be selected without pulling down the Record menu by pressing <F8>.

5.9.2 Record Status

This command opens a dialog box that indicates the current status of the Analyzer488 record system. The meaning of each field in this dialog box is explained below.

`Record`: A dot will appear showing the on/off status of recording.

`Trigger`: A dot will appear showing if Triggering is being used.

`Pre-Trigger Transactions`: Indicates the number of bus events that have occurred since the Record command was issued but prior to the detection of the trigger point. This number may be in the range 0 to 99,999,999.

`Available`: Indicates the number of bus events still in record memory that have been recorded prior to the detection of the trigger point. This number may be in the range 0 to 32,767.

`Post-Trigger Transactions:` Indicates the number of bus events that have occurred since the detection of the trigger point. This number may be in the range 0 to 99,999,999.

`Available`: Indicates the number of bus events still in record memory that have been recorded since the detection of the trigger point. This number may be in the range 0 to 32,767.

`Total Transactions`: The sum of Pre-Trigger Transactions and Post-Trigger Transactions. This number is in the range 0 to 99,999,999.

`Available`: Indicates the total number of bus events still in record memory that were recorded since recording was enabled. This number may be in the range 0 to 32,767.

`Post-Trigger Transaction Time`: This is the time between the occurrence of the trigger word and the completion of the record sequence. This time is measured in microseconds, with 2 microsecond resolution. This time is displayed only at the end of a capture sequence that used triggering. If triggering is not used, then bus event time is not measured.

`Post-Trigger Bus Speed`: The ratio of Post-Trigger Transactions to Post-Trigger Transaction Time. This reading is the average bus data transfer rate during the most recent record sequence. If the time is zero (no post-trigger events), the speed is undefined and displayed as 0. The speed is accurate only at the end of a capture sequence that used triggering.

`Record Complete`: A dot will appear showing the Record Complete selection in use.

`Current Status`: This shows the current state of the recording mechanism. If the capture system is waiting for the trigger point, the message, "Waiting for Trigger" will appear in the window. When the capture sequence is complete, "Capture Sequence Completed" is displayed.

`OK` and `Cancel`: Pressing the `<Return>` key or clicking on these boxes will close the capture status dialog box and return the cursor to its former location in the active window.

5.9.3 Trigger

This command opens a dialog box that allows you to specify the trigger parameters for recording specific IEEE 488 bus events. Use the cursor keys or mouse to move between the various fields.

If Triggering is used, a trigger match pattern must be specified.

The Match Pattern dialog box allows you to specify the state of all data lines and bus management lines. The state of the data lines may be specified as an IEEE 488 bus command, a hexadecimal byte, an ASCII character, or a binary byte. Depending on the format used, a message will appear in the Options box specifying which entries are valid.

If Message (`MSG`) format is selected, the state of the data lines may be specified as an IEEE 488 bus command (refer to the ASCII table in Appendix H which lists IEEE 488 bus commands and their ASCII equivalent). If not specified, 'don't care' is the default.

If Hexadecimal (`HEX`) format is selected, the state of the data lines may be specified as a hexadecimal number between 00 and FF or as 'don't care'.

If Character (`CH`) format is selected, the state of the data lines may be specified as any printable ASCII character (refer to the ASCII table in Appendix H which lists IEEE 488 bus commands and their ASCII equivalent). If not specified, 'don't care' is the default.

If Binary format is selected, the state of the data lines may be specified as a binary number including 'don't cares'. If a line is specified as a don't care, it is not used by the trigger comparison circuitry.

The state of any bus management line may be specified as asserted (`1`), unasserted (`0`), or 'don't care' (`X`).

The Match Count specifies the number of times the selected match pattern must be presented on the bus before the Delay Count (if it is not zero) will be decremented. The match count may be a number between 1 and 65,535. If the Delay Count is zero, the Match Count specifies the number of times the selected match pattern must be presented on the bus before the Post Trigger Count (if it is not zero) will be decremented.

The Delay Count specifies the number of events that must occur before the Post Trigger Count is decremented. (1 to 9,999,999)

The Post Trigger Count specifies the number of events that must occur after the trigger point has been found (1 to 32,767). When the Post Trigger Count has be decremented to Zero, the Record Complete Action will occur.

When the Analyzer488 has finished recording the specified amount of bus events, one of two things will occur depending on the Record Complete selection.

If Stop is selected, the Analyzer488 will stop recording data and allow further bus events to occur. The Handshake selection is not changed.

If Step is selected, the Analyzer488 will automatically go into Step Handshake speed, recording will remain on, and one additional bus event will be recorded each time the Step command is issued.

The Trigger Description box will describe the current trigger setup in text format.

Pressing the OK button enters the current trigger setup and closes the trigger setup window.

Pressing the Cancel button removes any changes to the current trigger setup and closes the trigger setup window.

5.9.4 Erase

This command erases the entire record memory of the Analyzer488 and sets the record pointer to zero. Once this command is issued, the next bus events recorded will be saved starting at location zero.

This command may also be selected without pulling down the Record menu by pressing <Ctrl E>.

5.9.5 Zero

This command sets the record pointer to zero but does not affect any data currently in the Analyzer488 record memory. Once this command is issued, the next bus events recorded will be saved starting at location zero. Any data currently in the record memory will be overwritten.

This command may also be selected without pulling down the Record menu by pressing <Ctrl Z>.

5.10 Search Menu

The search menu lets you look through the data in windows to find specific bus events.

5.10.1 First

This command places the cursor on the first recorded event in the most recent recorded sequence. When a File Window is first opened, this will be the default cursor position and the default reference location.

For an extremely long capture sequence, it is possible that the record memory may be overwritten one or more times. In this case, the first recorded event may be at a higher location than the last recorded event.

5.10.2 Last

This command places the cursor on the last recorded event in the most recent recorded sequence.

For an extremely long capture sequence, it is possible that the record memory may be overwritten one or more times. In this case, the last recorded event may be at a lower location than the first recorded event.

5.10.3 Next Trigger

When more than one record sequence is performed without first erasing the record memory, it is possible to have more than one trigger point in the record memory of the Analyzer488. This command places the cursor on the next location (a location recorded after the present location) that was marked as a trigger point.

If multiple record sequences were performed, it is possible that the record memory may be overwritten one or more times. In this case, the next trigger location may be at a lower location than the current trigger location.

5.10.4 Previous Trigger

When more than one record sequence is performed without first erasing the record memory, it is possible to have more than one trigger point in the record memory. This command places the cursor on the previous location (a location recorded before the present location) that was marked as a trigger point.

If multiple record sequences were performed, it is possible that the record memory may be overwritten one or more times. In this case, the previous trigger location may be at a higher location than the current trigger location.

5.10.5 Block Find

This command allows you to search a marked block for a particular event. This command may be used with either the Analyzer Window or the File Window. A dialog box similar to the one shown below will appear and prompt for a event to find. Use the cursor keys or mouse to move between the various fields.

The Match Pattern dialog box allows you to specify the state of all data lines and bus management lines. The state of the data lines may be specified as an IEEE 488 bus command, a hexadecimal byte, an ASCII character, or a binary byte. Depending on the format used, a message will appear in the Options box specifying which entries are valid.

If Message (`MSG`) format is selected, the state of the data lines may be specified as an IEEE 488 bus command (refer to the ASCII table in Appendix H which lists IEEE 488 bus commands and their ASCII equivalent). If not specified, 'don't care' is the default.

If Hexadecimal (`HEX`) format is selected, the state of the data lines may be specified as a hexadecimal number between 00 and FF. If not specified, 'don't care' is the default.

If Character (`CH`) format is selected, the state of the data lines may be specified as any printable ASCII character (refer to the ASCII table in Appendix H which lists IEEE 488 bus commands and their ASCII equivalent). If not specified, 'don't care' is the default.

If Binary format is selected, the state of the data lines may be specified as a binary number including 'don't cares'. If a line is specified as a don't care, it is not used in the comparison.

The state of any bus management line may be specified as asserted (`1`), unasserted (`0`), or 'don't care' (`X`).

5.10.6 Block Find Next

This command is used to search for the next data item that was specified while using the Block Find command. Data can be found in either the Analyzer Window or the File Window.

This command may also be selected without pulling down the Search Menu by pressing <F9>.

5.10.7 Compare Blocks

The Compare Blocks command allows you to compare a marked block in one window with a marked block in another window. The comparison stops on the first discrepancy and the cursor is positioned to that spot in the active window. This command is useful for comparing a previously recorded event sequence to a current one.

5.10.8 Goto Address

When this command is selected, a dialog box will appear prompting for a location. The cursor is then moved to the specified location in the active window.

This command may also be selected without pulling down the Search menu by pressing <Ctrl G>.

5.10.9 Goto Offset

This command moves the cursor in the active window to a location relative to the current location. When this command is selected, a dialog box will appear prompting for an offset (a number between -32,767 and 32,767). An effective location is computed which is equal to the present location plus the offset. The cursor is then moved to the new location in the active window.

This command may also be selected without pulling down the Search menu by pressing <Ctrl O>.

5.10.10 Goto Reference

This command moves the cursor in the active window to the reference location (relative location 0). The reference location is set by the Set Reference command.

5.11 Handshake Menu

The handshake menu lets you change the handshaking speed of the Analyzer488. It also lets you change the Talk and Listen states.

5.11.1 Listen

This command configures the Analyzer488 handshake circuitry to perform an acceptor handshake and accept data from the bus. The handshake speed may be set to Fast, Slow, Step, or None.

This command may also be selected without pulling down the Handshake menu by pressing `<Ctrl L>`.

5.11.2 Talk

This command configures the Analyzer488 handshake circuitry to perform a source handshake and place data on the bus. The handshake speed may be set to Fast, Slow, Step, or None.

This command may also be selected without pulling down the Handshake menu by pressing `<Ctrl T>`.

5.11.3 Fast

This command allows the Analyzer488 to participate in bus events without limiting the event speed. The Fast handshake speed may not be useful in viewing bus events because the display rate cannot keep up with a fast bus event rate.

5.11.4 Slow

This command configures the Analyzer488 handshake circuitry to perform a handshake at a slow rate of 1 to 10 events per second. When this command is issued, a menu will appear showing the valid speed selections.

5.11.5 Step

This command configures the handshake circuitry to perform one handshake each time the Step command is issued. When the command is first issued, the handshake speed is set to Step and 1 handshake is performed. In this way, each bus events may be easily viewed. If you are using handshake Step speed, the Step command must be used to complete each bus event whether talking or listening on the bus.

This command may also be selected without pulling down the Handshake menu by pressing `<F10>`. Each step is also performed by pressing `<F10>`.

5.11.6 None

This command disables the handshake circuitry and allows the Analyzer488 to record bus data without appearing to be an active listener. The Analyzer488 does not participate in bus handshaking and has no control of the bus event speed. This selection is useful for recording events such as bus errors and measuring the speed of other devices on the bus. It also shows such errors as no active listeners.

Instrument Simulator

6.1 Introduction

This section contains a detailed explanation of the Instrument Simulator Mode of operation of the Analyzer488 IEEE 488 Bus Analyzer. This section contains an description of the Instrument Simulator Mode of operation, information regarding setup, configuration, an explanation of how to use the Instrument Simulator Mode, and a detailed description of all the commands.

6.2 Description

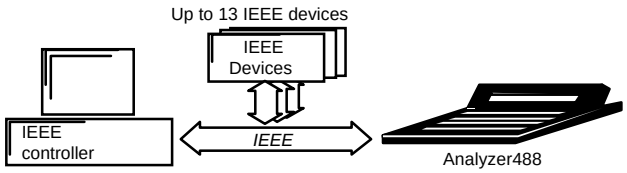
The Instrument Simulator mode of the Analyzer488 can simulate a bus device, accept input from the front panel keyboard, and display messages on the front panel display. The Analyzer488 can provide an IEEE controller with known data derived from various waveforms. Programmable data formats make the Analyzer488 a source of data for a wide range of software packages.

Any of 12 waveforms may be selected. Data for a user defined waveform may also be loaded into the Analyzer488. User waveform data is retained in non-volatile RAM. If the Analyzer488 operating mode is changed, the user waveform data is retained and will be available when the Instrument Simulator Mode is reselected.

Waveform data may be returned in ASCII or Binary format. The terminators for each reading and the entire waveform can be programmed. Parallel and serial poll responses can be programmed. A timed Service Request command allows the Analyzer488 to be programmed to generate timed service requests.

6.3 Setup

To setup the Analyzer488 an an Instrument Simulator, connect it to the IEEE 488 bus through the IEEE 488 bus connector on the rear panel. Plug the external power supply into the rear panel power supply connector.



6.4 Configuration

Configuration of the Instrument Simulator Mode of the Analyzer488 is accomplished using the front panel keyboard during power-up. Once set, the configuration will be stored in nonvolatile RAM and used as the default configuration when the Analyzer488 is powered on. Refer to Section 1 for configuration details.

6.5 Using the Instrument Simulator Mode

To begin operating the Analyzer488, turn the power ON. The last configuration of the Analyzer488 is saved in non-volatile RAM. If the Analyzer488 was last used as an instrument simulator, it be in the same mode when power is restored.

Below are some program examples using the Analyzer488. Theses examples use the IOtech Personal488™ and the KYBDCTRL keyboard controller program. For other controllers, refer to the manufacturer's manual for the proper syntax to input and output data on the IEEE 488 bus. Refer to the Command Description section for a complete listing of all commands and further examples.

Run the IOtech KYBDCTRL keyboard controller program and verify that the `CMD>` prompt appears on the screen. The following is a listing of commands given to the Analyzer488 and the response received. Responses are separated by extra vertical space for clarity although these spaces will not actually appear when using the program.

`CMD> OUTPUT10;C0X`	Clear the Analyzer488 data buffer
`CMD> OUTPUT10;B/+1.4534E+0/`	Send the string `+1.4534E+0` to the user waveform buffer
`CMD> OUTPUT10;B13,10`	Send ASCII codes for a Carriage Return and Line Feed to the user waveform buffer
`CMD> OUTPUT10;W13X`	Select waveform 13 (the user waveform)

`CMD> ENTER16`	Take a reading from the Analyzer488
	`Analyzer488 returns:`
	`+1.4534E+0`

CMD> OUTPUT10;T5000X	Command the Analyzer488 to Generate a Service Request after 5 seconds
`CMD> SPOLL10`	Serial Poll the Analyzer488 `Analyzer488 returns: 64`

6.6 Instrument Simulator Mode Bus Commands.

Control of the Instrument Simulator Mode of the Analyzer488 is implemented with 18 bus commands. There are six groups of Instrument Simulator commands: data buffer commands, keyboard and display commands, waveform control commands, poll response commands, and miscellaneous commands

6.6.1 Data Buffer Commands

The Buffer Data commands allow control over the user-definable data buffer inside the Analyzer488.

User Buffer Data	`B/string/`	Send a string to the data buffer.
	`Bn`	Send binary data to the data buffer. `n` = 0 to 255.
	`B?`	Always returns `B0`.
Clear Buffer	`C0`	Clear the entire data buffer.
	`C?`	Always returns `C0`.

6.6.2 Keyboard and Display Commands

The keyboard and display commands allow the user to input and output data on the Analyzer488 console. This can be used as part of a program to perform interactive data logging or give operator instructions.

Panel Display	`D/string/`	Display string on 20 column display of Analyzer488. All writes to the display are left justified and destructive. Strings of length greater than 20 are truncated.
	`D//`	Clears the display.
	`D?`	Always returns `D0`.
Keyboard Status	`U1`	Test for a key press.
	`U2`	Wait for a key press.

6.6.3 Waveform Control Commands

The following commands provide control over the type of waveform which is output over the data bus in response to an input command. This

allows the Analyzer488 to simulate input from many different types of measurement devices.

Format	`G0`	ASCII text with leading '+' sign `G1` ASCII text without leading '+' sign
	`G2`	2's complement integer binary - high byte first
	`G3`	2's complement integer binary - low byte first
	`G?`	Return current format selection.
Reading Delimiter	`H0`	CR LF
	`H1`	LF CR
	`H2`	CR Only
	`H3`	LF Only
	`H4`	Value set by User Reading Delimiter (`In`) command
	`H?`	Return current reading delimiter selection.
User Reading Delimiter	`I0`	Set user terminator to 'Null'
	`In`	Set user terminator to `n`. `n` = 0 to 255.
	`I?`	Return current user reading delimiter selection.
End of Reading EOI	`J0`	Assert EOI on end of reading
	`J1`	Do not assert EOI on end of reading
	`J?`	Return current End of Reading EOI selection.
End of Waveform EOI	`K0`	Assert EOI on end of Waveform
	`K1`	Do not assert EOI on end of Waveform.
	`K?`	Return current End of Waveform EOI selection.
Waveform	`Wn`	Select waveform `n`. `n` = 0 to 12 - factory waveforms.
		`n` = 13 - User waveform.
	`W?`	Return current waveform selection.
Waveform Terminator	`Y0`	CR LF
	`Y1`	LF CR
	`Y2`	CR only
	`Y3`	LF only

	Y4	Value set by User Waveform Terminator (Zn) command
	Y?	Return current Terminator selection
User Waveform Terminator	Z0	Set User waveform terminator to 'null'
	Zn	Set User waveform terminator to n. n = 0 to 255
	Z?	Return current User waveform terminator selection

6.6.4 Poll Response Commands

The following commands define the response which the Analyzer488 will transmit when polled.

Parallel Poll Response	P0	Clear Parallel Poll register.
	Pn	Set Parallel Poll register to n. n = 0 to 255.
	P?	Return current Parallel Poll register contents.
Serial Poll Response	S0	Clear Serial Poll register.
	Sn	Set Serial Poll register to n. n = 0 to 255.
	S?	Return current Serial Poll register contents.

6.6.5 Miscellaneous Commands

The following commands provide timing and status information about the Analyzer488.

Error Query	E?	Return error code and clear error condition.
Timed SRQs	T0	Disable SRQ on timeout.
	Tn	Generate an SRQ at the end of n milliseconds. n = 0 to 65535
	T?	Return number of milliseconds remaining
Status	U0	Send the Analyzer488 status string.
	U?	Returns last status command
Version Query	V?	Return revision level of the Analyzer488 firmware

Execute	X	Execute all commands in command buffer
Query	?	Return current selection or mode of the command preceding the ?

6.7 IEEE 488 Bus Implementation

The Analyzer488 implements many of the capabilities defined by the IEEE 488 1978 specification. These are discussed in the following sections. Those bus uniline and multiline commands that the Analyzer488 does not support or respond to include:

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

6.7.1 My Talk Address (MTA)

When the Analyzer488 is addressed to talk, it outputs either waveform data or buffer data to the IEEE 488 bus in the format as specified by the Gn, Hn, Wn, Jn, Kn, Yn, Zn, and In commands.

The Analyzer488 can send Analyzer488 status (U0), or keyboard status (U1 and U2), without affecting any timing functions. After the requested status is output, the presently programmed Gn format will be the selected format.

6.7.2 My Listen Address (MLA)

When the Analyzer488 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 paragraph 6-8.

6.7.3 **Device Clear** (DCL and SDC)

Device Clear resets the Analyzer488 to the power on defaults. Any pending Service Requests (SRQ) and errors are cleared.

6.7.4 **Interface Clear** (IFC)

IFC places the Analyzer488 in the Talker/Listener Idle State. It clears any pending SRQs. The condition which caused the SRQ remains unmodified.

6.7.5 **Serial Poll Enable** (SPE)

When Serial Poll Enabled, the Analyzer488 sets itself to respond to a serial poll with its serial poll register byte if addressed to talk. The serial poll response is set by using the Serial Poll Response (`Sn`) command. When the serial poll byte is accepted by the controller, any pending SRQs are cleared. The Analyzer488 will continue to try to output its serial poll response until it is 'Serial Poll Disabled' by the controller.

6.7.6 **Serial Poll Disable** (SPD)

Disables the Analyzer488 from responding to serial polls by the controller.

6.7.7 **Unlisten** (UNL)

UNL places the Analyzer488 in the Listener Idle State.

6.7.8 **Untalk** (UNT)

UNT places the Analyzer488 in the Talker Idle State.

6.7.9 **Parallel Poll** (PP)

When Parallel Polled, the Analyzer488 outputs its parallel poll register byte. The parallel poll response is set by using the Parallel Poll Response (`Pn`) command.

6.8 Command Descriptions

Control of the Instrument Simulator Mode of the Analyzer488 is implemented with 18 bus commands, described here in detail. Examples are given for the commands using a PC functioning as an IEEE 488 controller using the IOtech Personal488 PC/IEEE 488 board and Driver488 software. All examples are given using GWBASIC or BASICA. The Analyzer488 bus address should be set to 10 for all examples.

In order to establish communication with the Analyzer488 from GWBASIC 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 the above sequence.

Most of the instructions offer a query (`?`) option. This option can be used to determine the current 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 Analyzer488 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 Analyzer488. No commands are executed until an `X` is received by the Analyzer488. The only exceptions to this rule are the Buffer Data (`Bn`), Panel Display (`D/string/`), and the Query (`?`) commands. These commands do not have to be followed by an `X`.

Buffer Data Bn

The Buffer Data command, Bn, allows the user to write data to the data buffer. The data buffer consists of 4096 locations. Each location can store one character (a binary number from 0 to 255). Data sent to the buffer is stored as it is received. When the Analyzer488 is commanded to output data from the data buffer, all data currently in the buffer is output in the order that it was received. The Execute command (X) is not required when using the Buffer Data command.

The data sent with this command may be a delimited string or a number in the range 0 to 255. Once the buffer is full, any excess data sent to the buffer will be ignored. The entire buffer may be cleared by executing the Clear Buffer (C0) command.

A string is a list of alphanumeric characters delimited at the beginning and end by single character. The delimiter can be a Slash (/), Backslash (\), Single Quote ('), or Double Quote ("). The delimiter is taken as the first of these characters after the B and the Analyzer488 will accept all input up to the next occurrence of that character as the string to be displayed. It is up to the user to insure that the opening and closing delimiters match.

SYNTAX:

B/string/	Send a string to the data buffer. A string may be one or more characters.
Bn,m,z	Send a binary value to the data buffer (n,m,z can be 0 to 255)
B?	Always returns B0

Note: Any number of data values may be sent when specifying data as binary values.

EXAMPLES:

PRINT#1,"CLEAR10"	reset the Analyzer488
PRINT#1,"OUTPUT10;C0X"	clear the Analyzer488 data buffer
PRINT#1,"OUTPUT10;B/NDCV+0.0000E+0/"	send the string NDCV+0.0000E+0 to the data buffer
PRINT#1,"OUTPUT10;B13,10"	send ASCII codes for a Carriage Return and Line Feed
PRINT#1,"OUTPUT10;W21X"	select Waveform 21 (the user waveform)

```
PRINT#1,"ENTER10"
INPUT#2,A$
PRINT A$
```

request data from the Analyzer488

display shows `NDCV+0.0000E+0`

Clear Buffer C0

The Clear Buffer command, `C0`, is used to clear the entire data buffer. Any buffer data in the data buffer will be deleted after executing this command. Before loading new waveform data into the data buffer, the Clear Buffer command should first be executed to delete any data existing in the buffer.

SYNTAX:

`C0`	Clear the entire data buffer (default)
`C?`	Always returns `C0`

EXAMPLES:

```
PRINT#1,"OUTPUT10;C0X"          clear the Analyzer488 data buffer
PRINT#1,"OUTPUT10;B/ABC/"       output the string "ABC"
PRINT#1,"OUTPUT10;B13,10"       send ASCII codes for a
                                Carriage Return and Line Feed
PRINT#1,"OUTPUT10;W21X"         select Waveform 21 (the user
                                waveform)
PRINT#1,"ENTER10"               request data from Analyzer488
INPUT#2,A$                      get data
PRINT A$                        display shows "ABC"\
PRINT#1,"OUTPUT10;C?"           request the number of bytes used
                                in the buffer
PRINT#1,"ENTER10"               request data from Analyzer488
INPUT#2,A$                      get data
PRINT A$                        display shows C0005 (5 bytes
                                used)
PRINT#1,"OUTPUT10;C0X"          clear the Analyzer488 data buffer
PRINT#1,"OUTPUT10;B/123/"       send the string 123 to the buffer
PRINT#1,"OUTPUT10;B13,10"       send ASCII codes for a Carriage
                                Return and Line Feed
PRINT#1,"OUTPUT10;W21X"         select Waveform 21 (the user
                                waveform)
PRINT#1,"ENTER10"               request data from Analyzer488
INPUT#2,A$                      get data
PRINT A$                        display shows 123
```

Panel Display D/string/

The Panel Display command, D/string/, allows the user to display data or message information on the 20 character display of the Analyzer488. The display buffer consists of 20 locations, each of which can store one character. Each write to the display buffer erases any information previously displayed. The Execute command (X) is not required when using the Panel Display command.

The information is displayed left justified. Once the display buffer is full, any excess data sent to the buffer will be ignored. The entire display may be cleared by writing a null (zero length) string to the display.

The string is a list of alphanumeric characters delimited at the beginning and end by single character. The delimiter can be a Slash (/), Backslash (\), Single Quote ('), or Double Quote ("). The delimiter is taken as the first of these characters after the D and the Analyzer488 will accept all input up to occurrence of that character as the string to be displayed. It is up to the user to insure that the opening and closing delimiters match.

SYNTAX:

D/string/	Send a string to the display. A string may be one or more characters.
D//	Clear the panel display.
D?	Always returns D0

EXAMPLES:

```
PRINT#1,"OUTPUT10;D/Press ENTER/"
```

The Analyzer488 will display: Press ENTER

```
PRINT#1,"OUTPUT10;D/I hope you're in Helvetica
font./"
```

The Analyzer488 will display: I hope you're in Hel

```
PRINT#1,"OUTPUT10;D\Increase by 1/2\"
```

The Analyzer488 will display: Increase by 1/2

PRINT#1,"OUTPUT10;D''" front panel display is blank.

Error Query E?

The Error Query command, E?, is used to determine the current error condition of the Analyzer488. After execution of the Error Query command, the current error condition is cleared. The error condition is also cleared by executing the Status (U0) command.

SYNTAX: E? Return and clear current error condition

After execution of the Error command, the Analyzer488 will return one of the following error codes:

E0	No Error has occurred
E1	Invalid Device Dependent Command (A4X, D5 etc.)
E2	Invalid Device Dependent Command Option (G5X, K3X etc.)

EXAMPLES:

```
PRINT#1,"CLEAR10"          reset the Analyzer488
PRINT#1,"OUTPUT10;E?"      request error status
PRINT#1,"ENTER10"
INPUT#2,A$
PRINT A$                   display shows E0, no errors have
                           occurred

PRINT#1,"OUTPUT10;A5X"     send an illegal command to the
                           Analyzer488
PRINT#1,"OUTPUT10;E?"      request error status
PRINT#1,"ENTER10"
INPUT#2,A$
PRINT A$                   display shows E1-Invalid Device
                           Dependent Command
PRINT#1,"OUTPUT10;E?"      request error status
PRINT#1,"ENTER10"
INPUT#2,A$
```

PRINT A$	display shows `E0` since error condition has been cleared after reading error status
`PRINT#1,"OUTPUT10;K3X"`	send a command with an invalid option
`PRINT#1,"OUTPUT10;E?"`	request error status
`PRINT#1,"ENTER10"`	
`INPUT#2,A$`	
PRINT A$	display shows `E2`-Invalid Device Dependent Command Option

Format Gn

The Format command, Gn, determines the form of the waveform data returned to the IEEE controller. Four data formats are available. Depending on the Format selection, the Reading Delimiter (Hn), User Reading Delimiter (In), End of Reading EOI (Jn), End of Waveform EOI (Kn), Waveform Terminator (Yn), and User Waveform Terminator (Zn) commands may be used to change the terminator and EOI selections. A reading is defined as one data point in a waveform. Each waveform consists of 256 readings.

If ASCII format is selected (Format G0 or G1), data returned from the Analyzer488 is formatted by using the Hn, In, Jn, Kn, Yn and Zn commands.

If binary format is selected (Format G2 or G3), the Hn, In, Jn, Yn, and Zn commands will have no effect on the data sent by the Analyzer488. The Kn command is used with binary data to change the EOI on the End of Waveform selection.

If ASCII format (Format G0 or G1) and waveform 21 (the user waveform) are selected, the Hn, In, Jn, Yn, and Zn commands will have no effect on the data sent by the Analyzer488. The Kn command may be used to change the EOI on the End of Waveform selection.

SYNTAX:

G0	ASCII Text with leading '+' sign (default)
G1	ASCII Text without leading '+' sign
G2	2's Complement integer binary - high byte first
G3	2's Complement integer binary - low byte first
G?	Return current Format selection

EXAMPLES:

```
PRINT#1,"CLEAR10"          Reset the Analyzer488
PRINT#1,"OUTPUT10;G0W10J0H0X"
                           select ASCII format with + sign,
                           select waveform 10, select EOI
                           asserted at the end of each
                           reading, select CR LF reading
                           delimiters
PRINT#1,"ENTER10"          request data from the
                           Analyzer488
```

```
INPUT#2,A$
```
PRINT A$ display shows `+1.990` for first reading

```
PRINT#1,"OUTPUT10;G1W10J0H0X"
```
select ASCII format without + sign, select waveform 10, select EOI asserted at the end of each reading, select CR LF reading delimiters

```
PRINT#1,"ENTER10"
```
request data from the Analyzer488

```
INPUT#2,A$
```
PRINT A$ display shows `1.990` for first reading

Reading Delimiter Hn

The Reading Delimiter command, `Hn`, is used to set the terminator that will be sent by the Analyzer488 after each reading. A reading is a single data point and a waveform (any factory waveform from 0 to 20) is made up of 256 readings. When the Analyzer488 is addressed to talk, it will append the reading delimiter to each reading.

The reading delimiters which may be used are shown below. The Reading Delimiter may also be set to any character whose numeric value is from 0 to 255 by using the User Reading Delimiter (`In`) command and then selecting the `H4` option of the Reading Delimiter command. The choices for the reading delimiter are listed below:

SYNTAX:	`H0`	CR LF
	`H1`	LF CR
	`H2`	CR Only (default)
	`H3`	LF Only
	`H4`	Value set by the User Reading Delimiter (`In`) command
	`H?`	Return current reading delimiter selection

EXAMPLES:

```
PRINT#1,"OUTPUT10;H0X"
```
set the Reading Delimiter to Carriage Return Line Feed

```
PRINT#1,"TERM IN CR LF"
```
set the IEEE 488 input terminators for the Personal488 to Carriage Return Line Feed

User Reading Delimiter In

The User Reading Delimiter command, `In`, is used to set the User Reading Delimiter to any character whose numeric value is in the range 0 to 255. When addressed to talk, the Analyzer488 will insert the character between each reading of the waveform.

The User Reading Delimiter is enabled by the Reading Delimiter, `H4`, command. The User Reading Delimiter command can only be used with the Format commands `G0` and `G1`.

SYNTAX:		
	`I0`	Set User terminator to 'Null' (default)
	`In`	Set User terminator to '`n`'. (`n` = 0 to 255)
	`I?`	Returns current User Reading Terminator

EXAMPLES:		
	`PRINT#1,"OUTPUT10;W0X"`	select waveform 0
	`PRINT#1,"OUTPUT10;G0X"`	select ASCII format with + sign
	`PRINT#1,"OUTPUT10;I44X"`	set the User Reading Delimiter to comma (`,`)
	PRINT#1,"OUTPUT10;H4X"	select user's reading delimiter
	`PRINT#1,"ENTER10"`	request data
	`INPUT#2,A$`	get data
	`PRINT A$`	data shows:

```
+0.000,+0.131,+0.241,+0.316,+0.347,...etc...
```

End of Reading EOI Jn

The End Or Identify (EOI) line is one of five bus management lines on the IEEE 488 Bus. It is used by a talker to indicate the end of a multiple byte transfer sequence. The controller can change the End of Reading EOI mode by programming the Analyzer488 from the bus. A reading is a single data point and a waveform (any factory waveform from 0 to 20) is made up of 256 readings. The End of Reading EOI command allows the Analyzer488 to assert or not assert the EOI line when the last byte of each reading is sent to the controller.

In the `J0` mode, the EOI line is asserted by the Analyzer488 on the last byte of every reading. In the `J1` mode the EOI line is not asserted by the Analyzer488.

SYNTAX:	`J0`	EOI enabled, assert EOI on end of reading
	`J1`	EOI disabled, do not assert EOI on end of reading (default)
	`J?`	Return current End of Reading EOI selection

EXAMPLES:

```
PRINT#1,"OUTPUT10;J0X"  assert EOI on end of reading

PRINT#1,"OUTPUT10;J?"   query EOI mode
PRINT#1,"ENTER10"       request data
INPUT#2,A$              get data
PRINT A$                J0 is displayed on the front panel
```

End of Waveform EOI Kn

The End Or Identify (EOI) line is one of five bus management lines on the IEEE 488 Bus. It is used by a talker to indicate the end of a multiple byte transfer sequence. A waveform is made up of 256 readings. The End of Waveform EOI command, `Kn`, will enable the Analyzer488 to assert or not assert the EOI line on the last byte of the 256th reading (the end of the waveform).

In the `K0` mode, the EOI line is asserted by the Analyzer488 on the last byte of the waveform. In the `K1` mode, the EOI line is not asserted by the Analyzer488.

SYNTAX:

`K0`	EOI enabled, assert EOI on end of waveform (default)
`K1`	EOI disabled, do not assert EOI on end of waveform
`K?`	Return current End of Waveform EOI selection

EXAMPLES:

```
PRINT#1,"OUTPUT10;K1X"      do not assert EOI on End of
                            Waveform
PRINT#1,"OUTPUT10;K?"       request EOI selection from
                            Analyzer488
PRINT#1,"ENTER10"
INPUT#2,A$
PRINT A$                    display shows K1
```

Parallel Poll Response Pn

The Parallel Poll Response command, `Pn`, is used to set the eight-bit parallel poll register to any value from 0 to 255. The value in the parallel poll register will be returned when the IEEE controller issues a parallel poll command (PPOLL) on the bus. If Pn is used to change the parallel poll register, it replaces any previous register data.

SYNTAX:

`P0`	Clear the parallel poll register (default)
`Pn`	Set the Parallel Poll Register to `n`. (new data replaces any previous data)
`P?`	Return Parallel Poll Register contents

EXAMPLES:

```
PRINT#1,"OUTPUT10;P65X"     set the parallel poll register to 65
PRINT#1,"PPOLL"             parallel poll the bus
INPUT#2,C$
PRINT C$                    display shows 65

PRINT#1,"OUTPUT10;P?X"      request parallel poll register
                            contents
PRINT#1,"ENTER10"
INPUT#2,C$
PRINT C$                    display shows P065
```

Serial Poll Response Sn

The Serial Poll Response command, `Sn`, is used to set the eight-bit serial poll register to any value from 0 to 255. The value in the serial poll register will be returned when the IEEE controller issues a serial poll command (SPOLL) on the bus. All data sent to the serial poll register replaces any previous register data. If the seventh data line is set true (x1xxxxxx), the Analyzer488 will assert the Service Request (SRQ) bus line. After the Analyzer488 is serial polled, bit 7 of the serial poll response register will be cleared.

SYNTAX:		
	`S0`	Clear the serial poll register (default)
	`Sn`	Set the serial poll register to `n`. `n` = 0 to 255 (new data replaces any previous data)
	`S?`	Returns current Serial Poll register contents

EXAMPLES:		
	`PRINT#1,"OUTPUT10;S65X"`	set the serial poll register to 65, the Analyzer488 will generate an SRQ
	`PRINT#1,"SPOLL10"` `INPUT#2,C$`	serial poll the Analyzer488
	PRINT C$	display shows 65
	`PRINT#1,"SPOLL10"` `INPUT#2,C$`	serial poll the Analyzer488 again
	PRINT C$	display shows 1 since the Analyzer488 was Serial Polled above which cleared bit 7

Timed SRQs Tn

The Timed SRQs command, `Tn`, enables the Analyzer488 to generate a Service Request (SRQ) after a specified amount of time has passed. The time period may be between 0 and 65535 milliseconds (0 to 65 seconds) in increments of 1 millisecond.

SYNTAX:

`T0`	Disable SRQ on timeout (default)
`Tn`	Generate an SRQ at the end of `n` milliseconds (`n` = 0 to 65535)
`T?`	Return the amount of time (in milliseconds) remaining before a timed SRQ will be generated

EXAMPLES:

```
PRINT#1,"CLEAR10"                reset the Analyzer488
PRINT#1,"OUTPUT10;T5000X"
                                 generate an SRQ after 5 seconds

                                 after 5 seconds, the Analyzer488
                                 will generate an SRQ.
PRINT#1,"SPOLL10"                serial poll the Analyzer488
INPUT#2,C$
PRINT C$                         display shows 64 since the
                                 Analyzer488 was the source of the
                                 Service Request.
```

Status Un

The Status command, `Un`, is used to return a command status message, or a keyboard status message.

The Status command `U0` will cause the Analyzer488 to send a command status message when next addressed to talk. The command status message shows the current options which are in use for each command. The command status of the Analyzer488 may be read at any time without interfering with normal operation. Any error conditions are cleared after the status string is read by the controller.

The format of the status byte returned by the Analyzer488 after receiving a `U0` command is as follows:

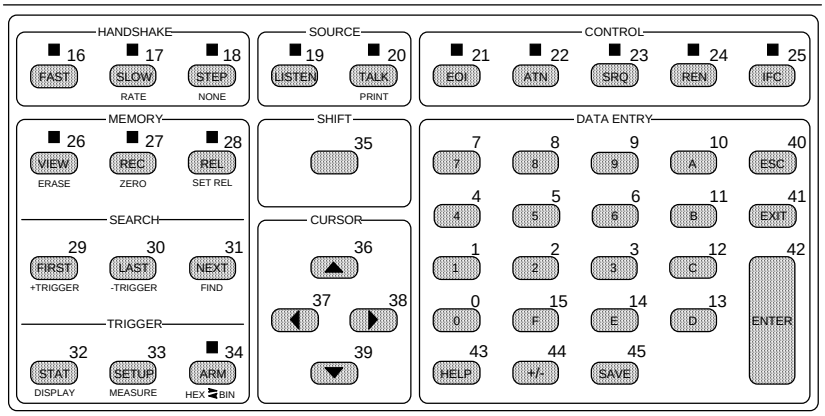
```
*.*C#####D#E#G#H#I###J#K#P###S###T#####U#W##Y#Z###
```

where each # shows the current option in use for that command. The leading information `*.*` is the revision level of the Analyzer488 firmware.

The Status command `U1` returns the key code for a key pressed on the Analyzer488 front panel keyboard. A key code, 0 to 44, is returned if a key is pressed, the value 255 is returned if no key is pressed.

The Status command `U2` waits up to 10 seconds for a key press from the Analyzer488 front panel keyboard. A key code, 0 to 44, is returned if a key is pressed, the value 255 is returned if no key is pressed before 10 seconds have elapsed.

The following illustration shows the key codes for the Analyzer488 front panel keyboard:



SYNTAX:

`U0`	Send the Analyzer488 status when next addressed to talk (default)
`U1`	Tests for a key press
`U2`	Get a key press - 10 second timeout
`U?`	Returns the last Status command issued.

EXAMPLES:

```
PRINT#1,"CLEAR10"          reset the Analyzer488
PRINT#1,"OUTPUT10;C0U0X"   clear the data buffer and request
                           the status of the Analyzer488
PRINT#1,"ENTER10"
INPUT#2,A$                 read the status byte
PRINT A$                   display  will show:

1.1C0000D0E0G0H2I000J1K0P000S000T00000U0W00Y2Z000
```

```
PRINT#1,"U1X"              Request keyboard input buffer
INPUT #2,C$
PRINT C$                   Display shows 255
```

Version Query V?

The Version Query command, V?, is used to look at the firmware revision level of the Analyzer488. In response to this command, the Analyzer488 will return a number in the following format *.* where each * is a 1 digit number. The Execute (X) command is not needed.

SYNTAX: V? Returns firmware revision level

EXAMPLES:

```
PRINT#1,"CLEAR10"          reset the Analyzer488
PRINT#1,"OUTPUT10;V?"      request firmware revision level
PRINT#1,"ENTER10"          read data from the Analyzer488
INPUT#2,A$
PRINT A$                   display shows 1.0 (or higher)
```

Waveform Wn

The Waveform command, Wn, is used to select the waveform data which will be sent when the Analyzer488 is addressed to talk. There are 12 possible waveforms which may be chosen (W0 to W11). Each waveform consists of 256 readings. A reading is defined as one data point in the waveform. The format of the readings may be specified by using the Format (Gn), Reading Delimiter (Hn), User Reading Delimiter (In), End of Reading EOI (Jn), End of Waveform EOI (Kn), Waveform Terminator (Yn), and User Waveform Terminator (Zn) commands. If waveform 12 is selected, data from the user waveform buffer will be returned. The graph of each waveform is shown on the following pages.

SYNTAX:	Wn	Select waveform n. (n=0 to 12)
	W?	Returns current waveform selection

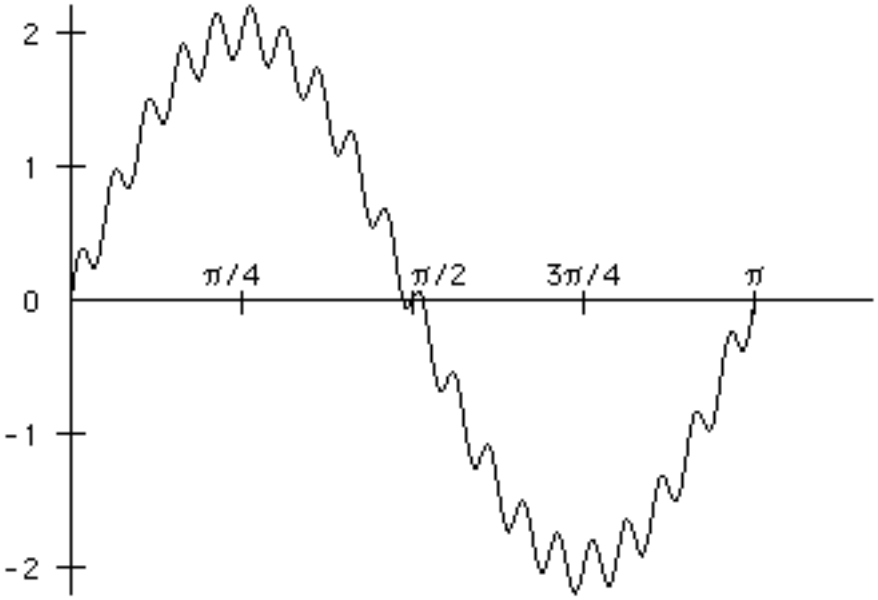
Analyzer488 Waveforms

W0	f(t) = Sin(wt) + 0.1Sin(20wt)
W1	Step Response of a First Order Filter
W2	Half Wave Rectified Sinewave
W3	Noise
W4	Sinewave with Crossover Distortion
W5	Sinewave with clipping on positive and negative peaks
W6	Ramp Waveform centered at zero
W7	10% Duty Cycle Pulse centered at zero
W8	20% Duty Cycle Pulse centered at zero
W9	30% Duty Cycle Pulse centered at zero
W10	Square Wave centered at zero
W11	f(t)=Sin(wt)
W12	User Waveform Buffer

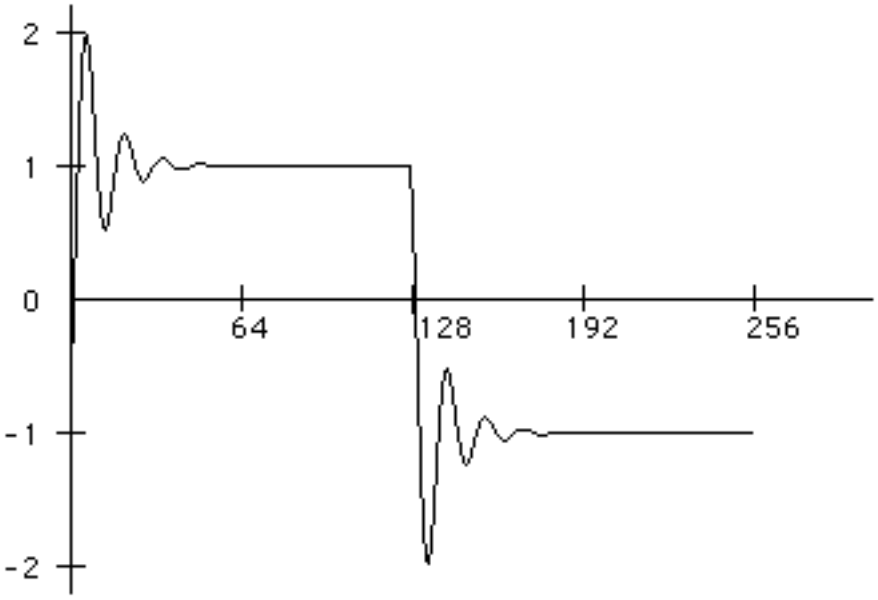
EXAMPLES:
```
PRINT#1,"CLEAR10"          reset the Analyzer488
PRINT#1,"OUTPUT10;W3 J1 K0 I44 H4 Y3 G1 X"
                           Select Noise waveform, EOI
                           unasserted at the end of each
```

reading, EOI asserted at the end of waveform, commas inserted between each reading (`I44` and `H4` set the reading delimiter to comma), waveform terminator Line Feed, ASCII text format without the leading + sign.

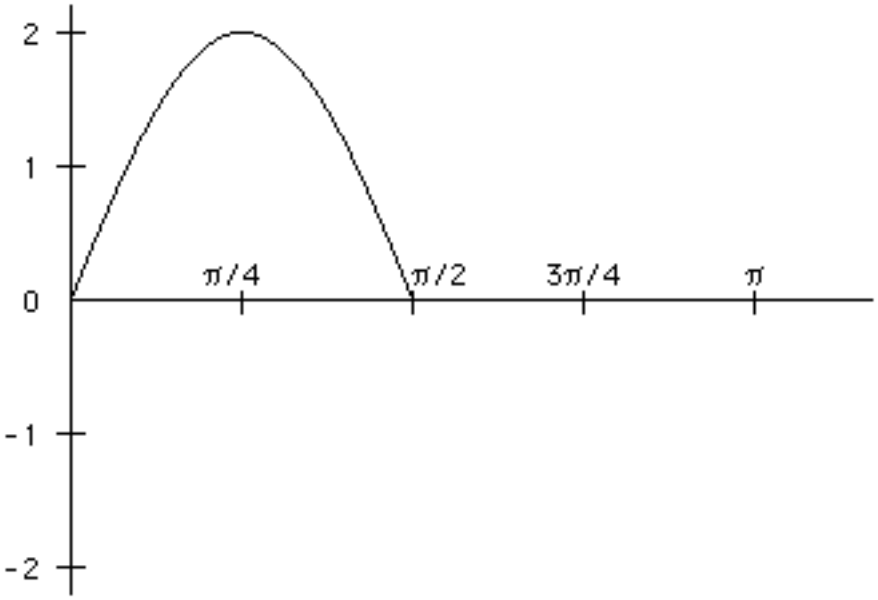
W0 f(t) = Sin(wt) + 0.1Sin(20wt)



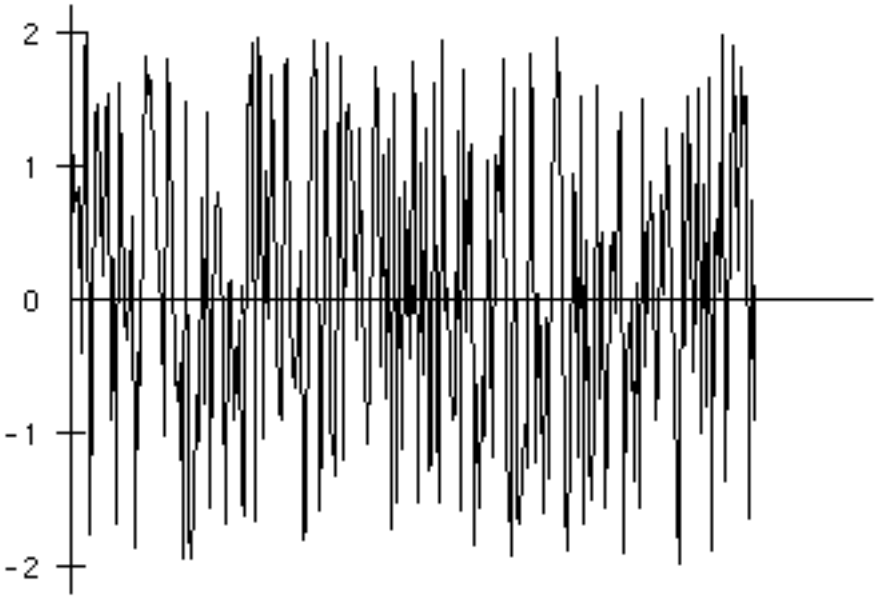
W1 Step Response of a First Order Filter



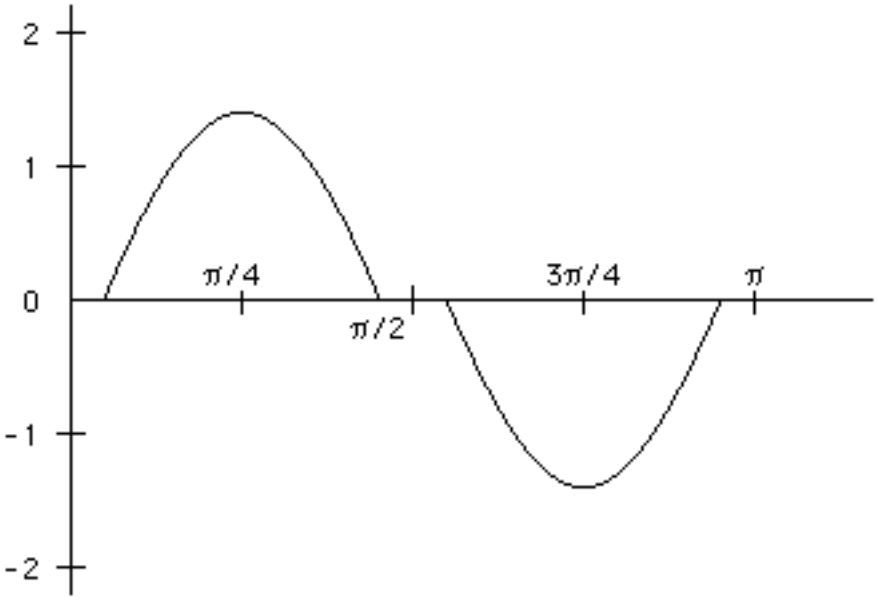
W2 Half Wave Rectified Sinewave



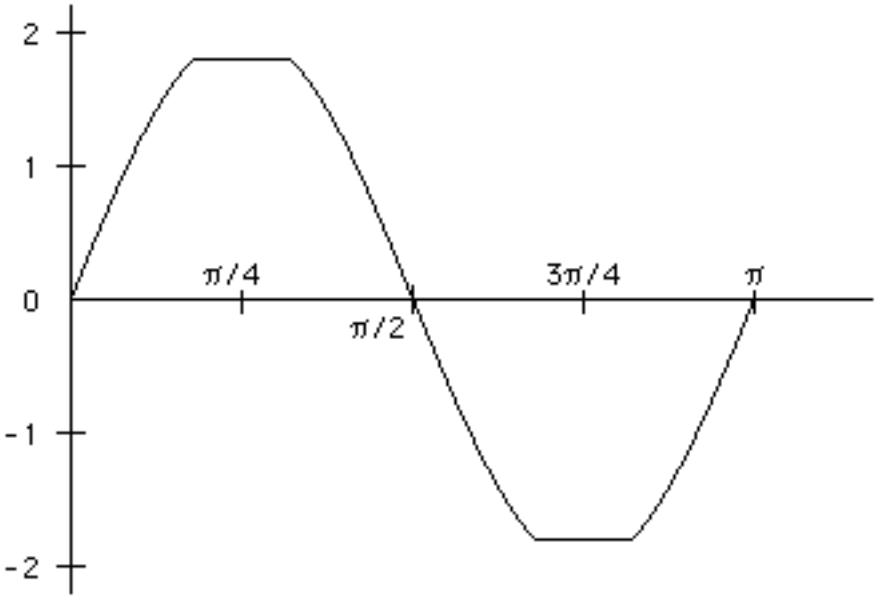
W3 Noise



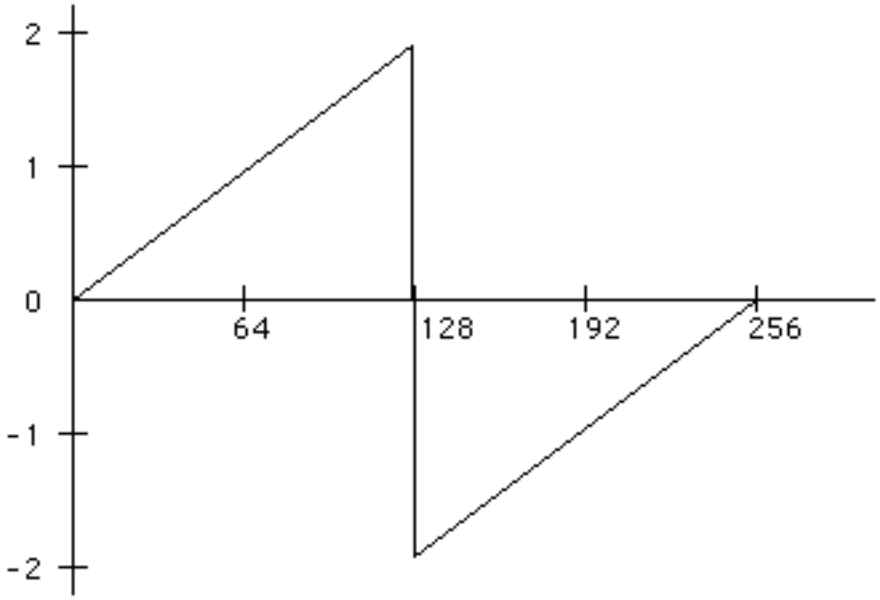
W4 Sinewave with Crossover Distortion



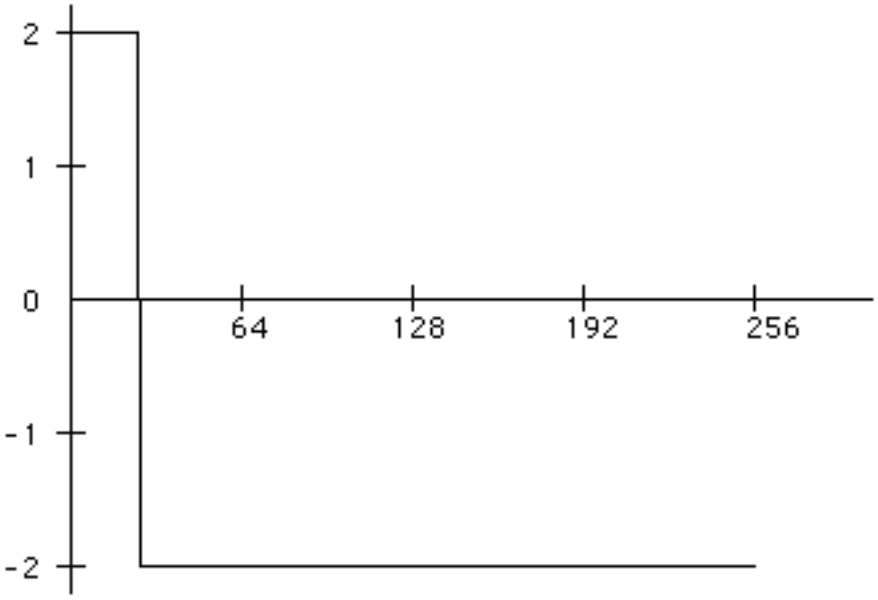
W5 Sinewave with clipping on positive and negative peaks



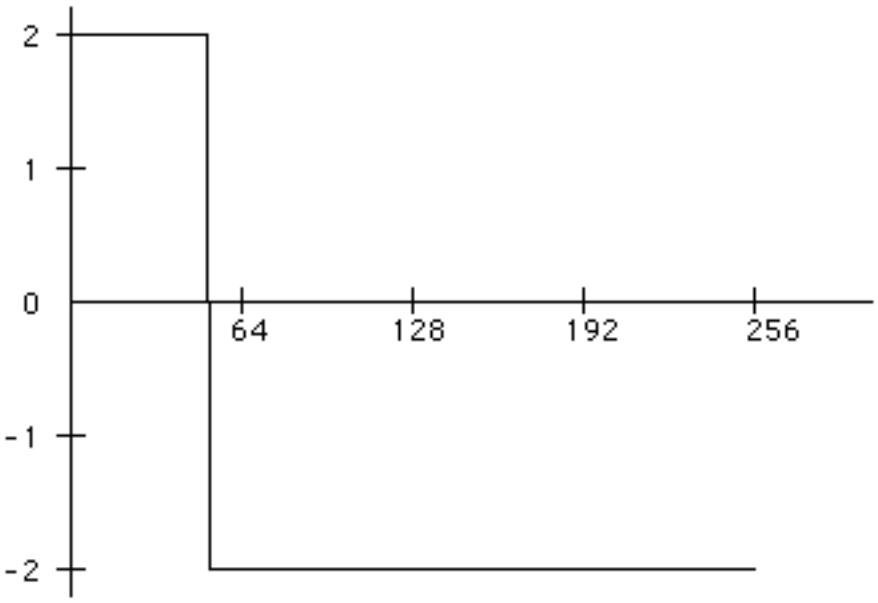
W6 Ramp Waveform centered at zero



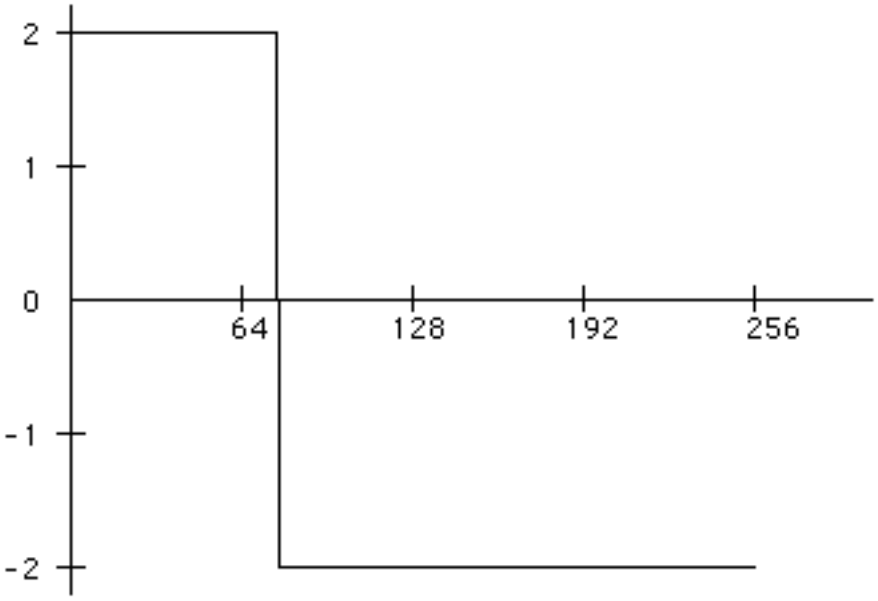
W7 10% Duty Cycle Pulse centered at zero



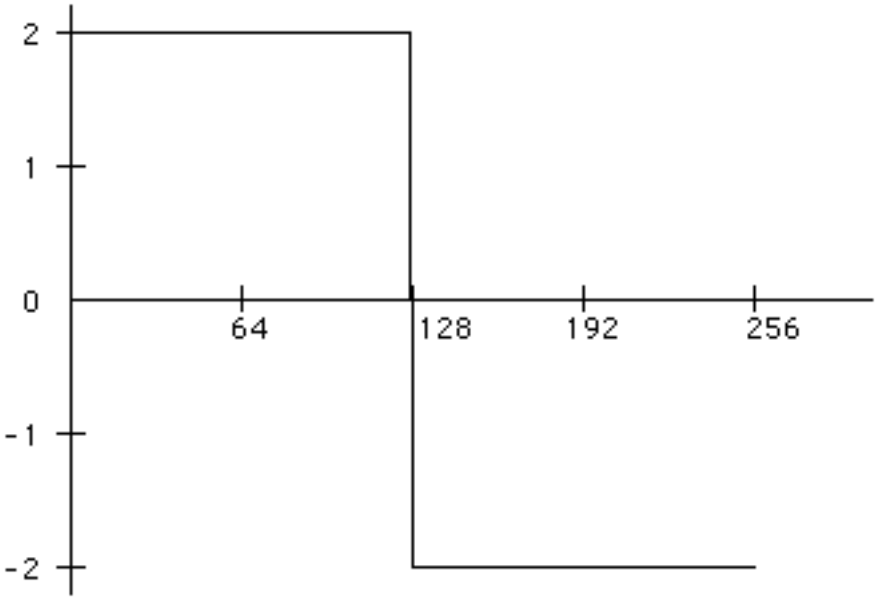
W8 20% Duty Cycle Pulse centered at zero



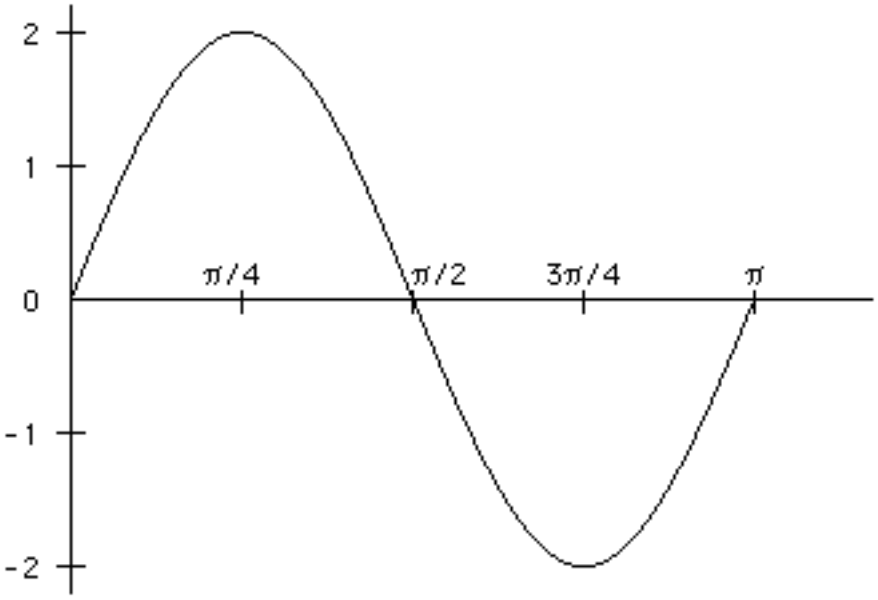
W9 30% Duty Cycle Pulse centered at zero



W10 Square Wave centered at zero



W11 f(t)=Sin(wt)



Execute X

Commands sent to the Analyzer488 will result in no action until the instrument is instructed to execute these commands. This is done by sending an `X`, usually as the last character of a command string. Most commands sent without an `X` are stored in an internal buffer until an `X` is received. The `Bn`, `D/string/`, and `?` (query) commands do not need an `Execute` command. Any number of `Execute` commands may be inserted into the same command string.

SYNTAX: *command*`X` Execute *command*

EXAMPLES:

```
PRINT#1,"CLEAR10"
```
reset the Analyzer488

```
PRINT#1,"OUTPUT10;T2000"
```
send `T2000` to the Analyzer488 command input buffer. This will cause the Analyzer488 to generate a Service Request after 2 seconds

```
PRINT#1,"OUTPUT10;X"
```
instruct the Analyzer488 to execute its command input buffer. The Analyzer488 will start timing upon receipt of the `X`.

Waveform Terminator Yn

The waveform terminator command, `Yn`, is used to set the terminator character that will be sent by the Analyzer488 after a waveform is sent. A waveform consists of 256 readings and each reading is one data point in a waveform. When the Analyzer488 is addressed to talk, it will append the waveform delimiter to each waveform.

The `Y0` mode is the most commonly accepted terminator, CR-LF. Command `Y1` reverses the sequence to send LF-CR. Command `Y2`, the default value, sends CR only and `Y3` sends LF only. The command `Y4` sets the waveform terminator to the value set by using the User Waveform Terminator command, `Zn`.

SYNTAX:	`Y0`	CR LF
	`Y1`	LF CR
	`Y2`	CR only (default)
	`Y3`	LF only
	`Y4`	Value set by User Waveform Terminator (`Zn`)
	`Y?`	Return current Waveform Terminator selection

EXAMPLES: `PRINT#1,"OUTPUT10;Y2X"` select Carriage Return terminator at end of waveform

User Waveform Terminator Zn

The User Waveform Terminator command, `Zn`, is used to set the terminator character that will be sent by the Analyzer488 after a user's waveform is sent. It may be programmed to any value between 0 and 255.

SYNTAX:	`Z0`	Set User Waveform terminator to 'Null' (default)
	`Zn`	Set User Waveform terminator to '`n`'. (`n` = 0 to 255)
	`Z?`	Returns current User Waveform Terminator

EXAMPLES:

```
PRINT#1,"OUTPUT10;W21X"
```
select user waveform (waveform 21)

```
PRINT#1,"OUTPUT10;Z44 Y4 X"
```
set user waveform terminator to comma

Query ?

Most of the Analyzer488 commands offer a Query (?) option which may be used to determine the current 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 Analyzer488 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.

SYNTAX: *command*? Return the current configuration or mode of the command, *command*.

EXAMPLES:

```
PRINT#1,"CLEAR10"              reset the Analyzer488
PRINT#1,"OUTPUT10;W12X"        select waveform 12
PRINT#1,"OUTPUT10;W?"          request current waveform
                               selection
PRINT#1,"ENTER10"              read data from the Analyzer488
INPUT#2,A$
PRINT A$                       display shows: W12


PRINT#1,"OUTPUT10;U0X"         request the status of the
                               Analyzer488
PRINT#1,"ENTER10"              read data from the Analyzer488
INPUT#2,A$
PRINT A$                       display shows status message:
```

```
1.1C0000D0E0G0H2I000J1K0P000S000T00000U0W00Y2Z000
```

Using the Query command to return the same status message:

```
PRINT#1,"OUTPUT10;V?C?D?E?G?H?I?J?K?P?S?T?U?W?Y?
Z?"
                                    request the status of  the
                                    Analyzer488
PRINT#1,"ENTER10"                   read data from the Analyzer488
INPUT#2,A$
PRINT A$                            display shows  status message:
```

```
1.1C0000D0E0G0H2I000J1K0P000S000T00000U0W00Y2Z000
```

Front Panel Keys

-HANDSHAKE-

[FAST]	Participate in handshaking at speeds up to 1 Mbyte/sec.
[SLOW]	Participate in handshaking at speeds from 1 to 10 bytes/sec.
[STEP]	Enable Step handshaking and initiate (Talk state) or complete (Listen state) one handshake sequence.
[RATE]	Adjust Slow handshake speed from 1 to 10 bytes / sec.
[NONE]	Disable participation in handshaking.

-SOURCE-

[LISTEN]	Place Analyzer488 into the Listen state.
[TALK]	Place Analyzer488 into the TALK state.
[PRINT]	Print data contained in the record memory.

-CONTROL-

[EOI]	Toggle the state of the End Or Identify (EOI) bus management line. The LED indicates the driven state of this line.
[ATN]	Toggle the state of the Attention (ATN) bus management line. The LED indicates the driven state of this line.
[SRQ]	Toggle the state of the Service Request (SRQ) management line. The LED indicates the driven state of this line.
[REN]	Toggle the state of the Remote Enable (REN) management line. The LED indicates the driven state of this line.
[IFC]	Pulse the Interface Clear (IFC) management line. The LED indicates the driven state of this line.

-MEMORY-

[VIEW]	Toggle display between bus display mode and memory view mode.
[ERASE]	Erase record memory and reset location pointer to zero.
[REC]	Enable recording.
[ZERO]	Reset location pointer to zero.
[REL]	Toggle addressing mode between relative addressing and absolute addressing.
[SET REL]	Set the location pointer.

-SEARCH-

[FIRST]	Display the first bus event recorded.
[+TRIGGER]	Search for the next trigger point.
[LAST]	Display the last bus event recorded.
[-TRIGGER]	Search for the previous trigger point.
[NEXT]	Find the next location that matches pattern defined with [FIND] key.
[FIND]	Find the first occurrence of a selected bus event.

-TRIGGER-

[STAT]	Provide status information about the last triggered bus recording.
[DISPLAY]	Set the brightness of the front panel display.
[SETUP]	Set up parameters of the trigger command.
[MEASURE]	Display bus event performance data.
[ARM]	Enable triggering.
[HEX↔BIN]	Toggle display format between hexadecimal and binary data.

-CURSOR-

[↑]	Increment the displayed memory location or the data value placed on the bus.
[↓]	Decrement the displayed memory location or the data value placed on the bus.
[→]	Select parameter fields by moving the cursor to the right.
[←]	Select parameter fields by moving the cursor to the left. Also provides a backspace key for numeric entries.

-DATA ENTRY-

[0]-[F]	Specify record memory locations, output data on the data lines, or play back a series of key presses saved with the [SAVE] key.
[HELP]	General key information or context sensitive help.
[+/-]	Change the polarity of a numeric entry, indicate a search direction, or set a "don't care" condition.
[SAVE]	Save or recall keypress macros.
[ENTER]	Enter selections, location addresses, or force a source (Talk state) or acceptor (Listen state) handshake.
[EXIT]	Leave a menu immediately and accept changes.
[ESC]	Escape a menu immediately without accepting changes.

Serial Controller Commands

@	@[CR|LF] Unlock the Analyzer488 from an inappropriate command.
@@	@@ Return the Analyzer488 to power-on conditions.
ABORT	ABORT Send IFC (SC) or MTA (*SC•CA). Stops bus activity.
ADDRESS	ADDRESS{[?]|[addr]} Set the IEEE 488 bus address of the Analyzer488.
ARM	ARM[[?]|[event[event...]]] Send a message at the occurence of a particular event.
ASSERT	ASSERT{[line][ALL][DATA<data>]} Assert one or more bus control and/or data lines
CLEAR	CLEAR[addr[,addr...]] Issue Device Clear (DCL) or Selected Device Clear (SDC).
CONTROLLER	CONTROLLER{[?]|[ON|OFF]} Select the operating mode of the Analyzer488.
DISARM	DISARM[;][event[event...]] Disable selected ARM events.
DISPLAY	DISPLAY{['string']} Send the message ‘string’ to the front panel display.
DUMP	DUMP[{location}TO{location}|{location} LENGTH{count}] Return bus events from record memory in S19 format.
ENTER	ENTER[addr][#count|term|EOI|;count|; term|;EOI] Read data from the IEEE 488 bus.
ERASE	ERASE Erase the entire record memory.

ERROR	ERROR[;]{[?]|MESSAGE|NUMBER|OFF]} Enable or disable automatic error reporting.
FIND	FIND{[?]|[MATCH][state][line][<byte>] [ERROR]|[TRIGGER][START{location}][+|-]} Search the record memory for a specific bus event
HANDSHAKE	HANDSHAKE{[?]|[FAST|STEP|NONE|SLOW{<rate>}]} Set the bus handshaking speed.
HELLO	HELLO Read the Analyzer488 revision identification.
ID	ID;[ASCII] Change the ID character.
KEY	KEY[?] Read the Analyzer488 Keyboard.
LISTEN	LISTEN[?] Go to Listen state and perform an acceptor handshake
LOCAL	LOCAL[addr[,addr...]] Unassert the Remote Enable (REN) line.
LOCAL LOCKOUT	LOCAL LOCKOUT Issue the Local Lockout (LLO) bus command.
MASK	MASK{[ON|OFF]} Mask the high bit of serial input data.
MEASURE	MEASURE[0|1] Return the speed of bus events in bytes per second.
NEXT	NEXT Find the next targeted bus event in record memory.
OUTPUT	OUTPUT[addr[,addr...]][#count];data Send data to IEEE 488 bus device.
PPOLL	PPOLL Read the Parallel Poll response from all bus devices.

PPOLL CONFIG	PPOLL CONFIG addr;response Set the Parallel Poll response of a bus device.
PPOLL DISABLE	PPOLL DISABLE addr[,addr...] Disable a bus device from responding to parallel polls.
PPOLL UNCONFIG	PPOLL UNCONFIG Prevent all bus devices from responding to a Parallel Poll.
PRINT	?[[location]TO[location]|[location]LENGTH [count]|BUS] Returns bus data or record memory data.
READ	READ Accept data from the bus while in low-level mode.
RECORD	RECORD{[?]|[ON|OFF]} Start or stop recording bus events.
RELATIVE	RELATIVE{[?]|[ON|OFF]} Toggles Relative Addressing Format.
REMOTE	REMOTE[addr[,addr...]] Assert Remote Enable line or address devices to listen.
RESET	RESET Analyzer488 warm start.
RESUME	RESUME Un-assert Attention (ATN) line.
SEND	SEND[;]sub-command[sub-command...] Send low level sequences in High Level Controller Mode.
SET RELATIVE	SET RELATIVE{[?]|[location]} Specify record memory reference pointer.
SET TRIGGER	SET TRIGGER{[?]|ON|OFF][MATCH[state] [line][byte]][# OF MATCHES[match count]] [TRIG DELAY[delay]][POST COUNT[post count]] [WHEN COMPLETE[STEP|STOP]]} Configure triggering parameters.

SPOLL	SPOLL[addr[,addr...]] Perform a Serial Poll.
STATUS	STATUS[;][0|1|2|3|4|5] Return Analyzer488 operational status.
STEP	STEP[?] Perform one source or acceptor handshake.
STERM	STERM[;]{[?]|term[term]|[NONE]} Set the output terminators for serial transfers.
TALK	TALK[?] Place data on the bus.
TERM	TERM[;]{[?]|[term[term][EOI]|[EOI]|[NONE]]} Set the output terminators for IEEE bus transfers.
TIMEOUT	TIMEOUT[;][n] Set timeout from 1 to 65535 seconds.
TRIGGER	TRIGGER[addr[,addr...]] Issue the bus Group Execute Trigger (GET) command.
UNASSERT	UNASSERT{[line][ALL]DATA[<data>]} Unassert one or more bus lines.
VIEW	VIEW[[location]|[location]TO[location]|[location]LENGTH[count]|BUS] Return record memory events or bus data.
WRITE	WRITE[data] Put data on the bus and initiate a source handshake.
ZERO	ZERO Reset record memory reference pointer to location 0.

Instrument Simulator Commands

Command	Code	Description
User Buffer Data	B/string/	Send a string to the user data buffer.
	Bn	Send binary data to the user data buffer, n. n = 0 to 255.
	B?	Always outputs B0.
Clear Buffer	C0	Clear the entire user data buffer.
	C?	Always outputs C0.
Panel Display	D/string/	Send a string to the display panel.
	D?	Always outputs D0.
Error Query	E?	Output error code and clear error condition.
Format	G0	ASCII text with leading '+' sign.
	G1	ASCII text without leading '+' sign.
	G2	2's complement integer binary - high byte first.
	G3	2's complement integer binary - low byte first.
	G?	Output current format selection.
Reading Delimiter	H0	CR LF.
	H1	LF CR.
	H2	CR Only.
	H3	LF Only.
	H4	Use value set by User Reading Delimiter (In) command.
	H?	Output current Reading Delimiter selection.
User Reading Delimiter	I0	Set User Reading Delimiter to 'Null'.
	In	Set User Reading Delimiter to n. n = 0 to 255.
	I?	Output User Reading Delimiter selection.

Command	Code	Description
End of Reading EOI	J0	Assert EOI on end of reading.
	J1	Do not assert EOI on end of reading.
	J?	Output current End of Reading EOI selection.
End of Waveform EOI	K0	Assert EOI on end of waveform.
	K1	Do not assert EOI on end of waveform.
	K?	Output current End of Waveform EOI selection.
Parallel Poll Response	P0	Clear Parallel Poll register.
	Pn	Set Parallel Poll register to n. n = 0 to 255.
	P?	Output present Parallel Poll register contents.
Serial Poll Response	S0	Clear Serial Poll register.
	Sn	Set Serial Poll register to n. n = 0 to 255.
	S?	Output present Serial Poll register contents.
Timed SRQ	T0	Disable SRQ on timeout.
	Tn	Generate an SRQ after n milliseconds. n = 0 to 65535.
	T?	Output number of milliseconds remaining before a timed SRQ.
Status	U0	Send the Analyzer488 status string.
	U1	Test for a key press.
	U2	Wait for a key press.
	U?	Always outputs U0.
Version Query	V?	Output Analyzer488 firmware revision level.
Waveform	Wn	Select waveform n. n = 0 to 20 (factory waveforms) n = 21 (User waveform).
	W?	Output current waveform selection.

Command	**Code**	**Description**
Execute	X	Execute all commands in command buffer.
Waveform Terminator	Y0	CR LF.
	Y1	LF CR.
	Y2	CR only.
	Y3	LF only.
	Y4	Use value set by User Waveform (Zn) command.
Terminator selection.	Y?	Output present Waveform Terminator
User Waveform Terminator	Z0	Set User Waveform Terminator to 'null'.
	Zn	Set User Waveform Terminator to n. n = 0 to 255.
	Z?	Output present User Waveform Terminator selection.
Query	?	Output present selection or mode of the command preceding the ?.

Analyst488 Software Menus

Menu	Shortcut Key	Description
File	(Alt F)	
Open Analyzer	(F2)	Open an Analyzer488 window
open File	(F3)	Open a file window
Virtual Front Panel	(F4)	Open a low-level window
coMmand Line Controller	(F5)	Open a keyboard controller
Write Block...	(^W)	Write a block of data
Print Block...		Opens a print menu
Printer	(^P)	Make a hardcopy of data
Disk	(^D)	Make a file-print of data
Transfer Block		Transfer data between windows
DOS Shell		Temporarily exit to DOS
Setup...		Open a setup menu
Serial Port		Seup serial port parameters
Program Options		Setup PC configuration
Restore Default Setup		Resets serial port and program
Quit	(^Q)	Quit and exit to DOS
Display	(Alt D)	
Format...		Open a format menu
Message		Set format to one-line IEEE codes
Hexadecimal		Data in hexadecimal format
miXed		Data in composit format
hEx Mode Highlight...		Highlight bus commands
Absolute	(^A)	Display absolute addresses
Relative	(^R)	Display relative addresses
Set relative	(F7)	Mark cursor for relative addresses
Horizontal	(^H)	Orient screens side-by-side
Vertical	(^V)	Orient screens up and down

Record	(Alt R)	
Record Enable	(F8)	Start or stop recording bus events
record Status...		Get recording statistics
Trigger setup...		Configure triggering parameters
triGger enable		Start or stop triggering
Erase	(^E)	Erase the record memory
Zero	(^Z)	Reset record memory pointer
Search	(Alt S)	
First		Display first recorded bus event
Last		Display the last event recorded
Next Trigger		Search for the next trigger point
Previous Trigger		Search for the previous trigger point
mark/Unmark Block	(F6)	Mark a block of data
Mark all	(^M)	Mark all data
Block find...		Search a marked block for data
block find Next	(F9)	Find next data in a marked block
block Compare		Compare two marked blocks
goto Address	(^G)	Cursor to an absolute address
goto Offset	(^O)	Cursor to a relative address
goto Reference		Cursor to relative address 0
Handshake	(Alt H)	
Listen	(^L)	Put Analyzer488 in listen state
Talk	(^T)	Put Analyzer488 in talk state
Fast		Set handshake to high speed
Slow...		Set handshake to 1 to 10 bps
steP	(F10)	Enable and step one handshake
None		Disable handshake participation

Analyzer488 Error Messages

The following error messages are returned if an error condition exists and the STATUS command is executed. The error condition is reset after the message is sent. Only the most recent error is maintained.

Error No.	Error Text and Description
00	OK
01	INVALID ADDRESS Caused by an invalid address outside the allowable IEEE 488 bus range of 00 to 30 for primary addresses and 00 to 31 for secondary addresses.
02	INVALID COMMAND Caused by an unrecognized command or invalid parameter.
03	WRONG MODE Caused by trying to execute a command not allowed within the present state of the interface.
04	Unassigned - Reserved
05	Unassigned - Reserved
06	Unassigned - Reserved
07	Unassigned - Reserved
08	COMMAND OVERFLOW More than 127 characters were received and interpreted as a command.
09	ADDRESS OVERFLOW More than 15 primary address/secondary address pairs were received.
10	MESSAGE OVERFLOW No memory is available to buffer the received data of the OUTPUT command.

11 NOT A TALKER
An unaddressed OUTPUT, a SEND DATA or a SEND CMD was received and the Analyzer488 was not in the Talk Addressed State.

12 NOT A LISTENER
An unaddressed ENTER or a SEND ENTER was received and the Analyzer488 was not in the Listen Addressed State.

13 BUS ERROR
The Analyzer488 tried to output data to the bus but there was no active listener to accept it.

14 TIMEOUT - WRITE
The specified TIME OUT time has elapsed before the last command or data byte sent by the Analyzer488 was accepted by an external device.

15 TIMEOUT - READ
The specified TIME OUT time has elapsed while the Analyzer488 was waiting for a data byte from an external device.

16 Unassigned - Reserved

17 Unassigned - Reserved

Sample BASIC Program

```
10  'Program to illustrate use of the ARM command
20  'This program uses the Wave488 as a bus device at
30  'address 20 to Generate a Service Request
35  'and provide data
40  '
50  PRINT"ARM SRQ, TRIGGER ,CAPTURE Demo Program"
55  '
60  OPEN "COM2: 9600,N,8,2,CS,DS" AS 1
70  PRINT#1,"@@" 'Reset the Analyzer488
80  GOSUB 440 'Delay
90  PRINT#1,"ERASE" 'Clear the Analyzer488 capture buffer
100 GOSUB 440 'Delay
105 'Put the Analyzer488 in High Level Mode
110 PRINT#1,"CONTROLLER ON"
120 PRINT#1,"abort"   'Clear the Wave488
130 'Configure the Analyzer488 to trigger on
140 'the SRQ line asserted during one data transfer,
145 'wait for 11,000 bytes to be transfered,
150 'capture 15,000 bytes (after waiting
155 'for the initial 11,000), and after the capture
160 'sequence has completed, use step mode handshaking
170 PRINT#1,"SET TRIGGER ON SRQ COUNT 1 DELAY 11000 POST
    15000 ATEND STOP"
180 'Arm the Analyzer488 for the events SRQ, TRIGGER,
    CAPTURE
190 PRINT#1,"ARM SRQ TRIGGER CAPTURE"
195 'Make the Wave488 generate an SRQ after 2 seconds
200 PRINT#1,"OUTPUT20;t2000x"
210 PRINT "Wave should generate an SRQ in 2 seconds"
220 'Put the Analyzer in Low Level Controller mode
225 'to allow the use of
230 'different handshake selections
240 PRINT#1,"controller off"
250 PRINT#1,"handshake step"
255 'Assert ATN and REN
258 'to make the Wave488 listen to the following commands
260 PRINT#1,"ASSERT REN ATN"
270 'Make the Analyzer488 a talker
```

```
280 PRINT#1,"talk"
290 'Send the Wave's talk address (&H54 = TAG20)
295 'since the Wave is at address 20
300 PRINT#1,"write &H54"
310 PRINT "Told the Wave488 to talk"
320 'Unassert ATN to allow data transfers to occur
330 PRINT#1,"unassert ATN"
340 'Put Analyzer488 in fast handshaking mode
345 'to allow it to handshake the
350 'data without having to read it in from the serial
    port
360 'data is therefore being accepted but not stored
370 PRINT#1,"handshake fast"
375 'Record the bus transactions
380 PRINT#1,"record on"
390 'Put Analyzer488 in listen mode
395 'to allow it to accept data
400 PRINT#1,"listen"
410 PRINT"Analyzer in Listen mode, sequence in progress"
420 GOTO 480
430 '
440 'Delay Loop
450 FOR I = 1 TO 3000:NEXT I
460 RETURN
470 '
480 PRINT"Waiting for Events"
490 PRINT
500 INITTIME = TIMER
510  IF LOC(1) THEN PRINT INPUT$(LOC(1),1); ELSE PRINT".";
520 T = TIMER
530 IF TIMER - T < .2 GOTO 530
540 IF TIMER - INITTIME > 5 GOTO 560
545 PRINT#1,"status3"
547 INPUT#1,B$:PRINT B$
550 GOTO 510
560 PRINT "Finished With Demo"
570 END
```

IEEE 488 Primer

1. HISTORY

The **IEEE 488** bus is an instrumentation communication bus adopted by the Institute of Electrical and Electronic Engineers in 1975 and revised in 1978. The **Analyzer488** conforms to this most recent revision designated **IEEE 488-1978**.

Prior to the adoption of this standard, most instrumentation manufacturers offered their own versions of computer interfaces. This placed the burden of system hardware design on the end user. If his application required the products of several different manufacturers, then he might need to design several different hardware and software interfaces. The popularity of the **IEEE 488** interface (sometimes called the **G**eneral **P**urpose **I**nterface **B**us or **GPIB**) is due to the total specification of the electrical and mechanical interface as well as the data transfer and control protocols. The use of the **IEEE 488** standard has moved the responsibility of the user from design of the interface to design of the high level software that is specific to the measurement application.

2. GENERAL STRUCTURE

The main purpose of the **GPIB** is to transfer information between two or more devices. A device can either be an instrument or a computer. Before any information transfer can take place, it is first necessary to specify which will do the talking (send data) and which devices will be allowed to listen (receive data). The decision of who will talk and who will listen usually falls on the **System Controller** which is, at power on, the **Active Controller**.

The **System Controller** is similar to a committee chairman. On a well run committee, only one person may speak at a time and the chairman is responsible for recognizing members and allowing them to have their say. On the bus, the device which is recognized to speak is the **Active Talker**. There can only be one Talker at a time if the information transferred is to be clearly understood by all. The act of "giving the floor" to that device is called **Addressing to Talk**. If the committee chairman can not attend the meeting, or if other matters require his attention, he can appoint an acting chairman to take control of the proceedings. For the **GPIB**, this device becomes the **Active Controller**.

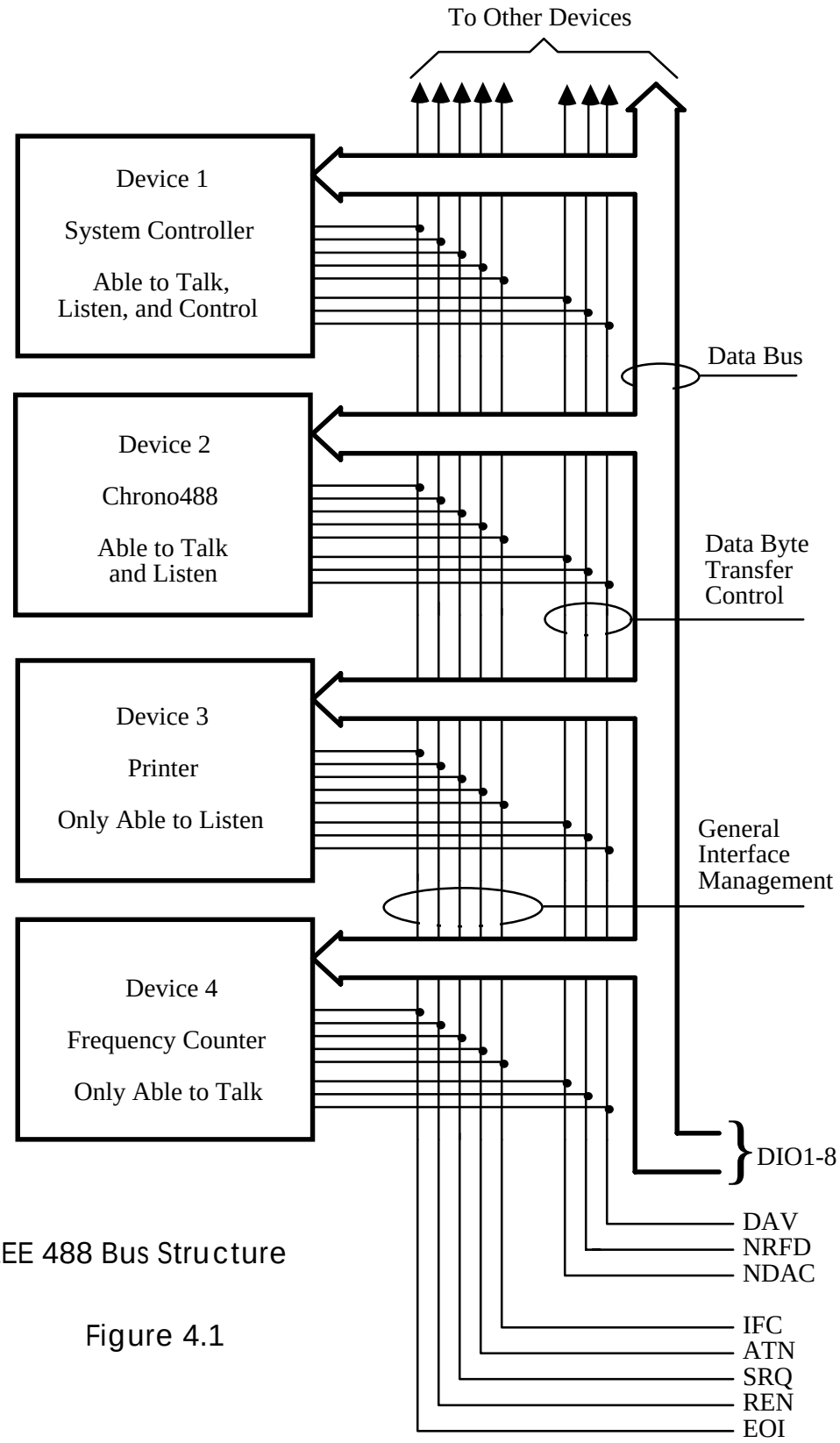
At a committee meeting, everyone present usually listens. This is not the case with the **GPIB**. The **Active Controller** selects which devices will listen and commands all other devices to ignore what is being transmitted. A device is instructed to listen by being **Addressed to Listen**. This device is then referred to as an **Active Listener**. Devices which are to ignore the data message are instructed to **Unlisten**.

The reason some devices are instructed to **Unlisten** is quite simple. Suppose a college instructor is presenting the day's lesson. Each student is told to raise their hand if the instructor has exceeded their ability to keep up while taking notes. If a hand is raised, the instructor stops his discussion to allow the slower students the time to catch up. In this way, the instructor is certain that each and every student receives all the information he is trying to present. Since there are a lot of students in the classroom, this exchange of information can be very slow. In fact, the rate of information transfer is no faster than the rate at which the slowest note-taker can keep up. The instructor, though, may have a message for one particular student. The instructor tells the rest of the class to ignore this message (**Unlisten**) and tells it to that one student at a rate which he can understand. This information transfer can then happen much quicker, because it need not wait for the slowest student.

The **GPIB** transfers information in a similar way. This method of data transfer is called **handshaking**. More on this later.

For data transfer on the **IEEE 488**, the **Active Controller** must…

a) **Unlisten** all devices to protect against eavesdroppers.
b) Designate who will **talk** by **addressing** a device to **talk**.
c) Designate all the devices who are to **listen** by **addressing** those devices to **listen**.
d) Indicate to all devices that the data transfer can take place.



IEEE 488 Bus Structure

Figure 4.1

3. SEND IT TO MY ADDRESS

In the previous discussion, the terms **Addressed to Talk** and **Addressed to Listen** were used. These terms require some clarification.

The **IEEE 488** standard permits up to 15 devices to be configured within one system. Each of these devices must have a unique address to avoid confusion. In a similar fashion, every building in town has a unique address to prevent one home from receiving another home's mail. Exactly how each device's address is set is specific to the product's manufacturer. Some are set by DIP switches in hardware, others by software. Consult the manufacturer's instructions to determine how to set the address.

Addresses are sent with **universal** (**multiline**) commands from the **Active Controller**. These commands include **My Listen Address** (MLA), **My Talk Address** (MTA), **Talk Address Group** (TAG), and **Listen Address Group** (LAG).

4. BUS MANAGEMENT LINES

Five hardware lines on the **GPIB** are used for bus management. Signals on these lines are often referred to as **uniline** (single line) commands. The signals are active low, i.e. a low voltage represents a logic "1" (asserted), and a high voltage represents a logic "0" (unasserted).

4.1 Attention (ATN)

ATN is one of the most important lines for bus management. If Attention is asserted, then the information contained on the data lines is to be interpreted as a multiline command. If it is not, then that information is to be interpreted as data for the **Active Listener**s. The **Active Controller** is the only bus device that has control of this line.

4.2 Interface Clear (IFC)

The **IFC** line is used only by the **System Controller**. It is used to place all bus devices in a known state. Although device configurations vary, the **IFC** command usually places the devices in the Talk and Listen Idle states (neither **Active Talker** nor **Active Listener**).

4.3 Remote Enable (REN)

When the **System Controller** sends the **REN** command, bus devices will respond to remote operation. Generally, the **REN** command should be issued before any bus programming is attempted. Only the **System Controller** has control of the **Remote Enable** line.

4.4 End or Identify (EOI)

The **EOI** line is used to signal the last byte of a multibyte data transfer. The device that is sending the data asserts **EOI** during the transfer of the last data byte. The **EOI** signal is not always necessary as the end of the data may be indicated by some special character such as carriage return.

The **Active Controller** also uses **EOI** to perform a **Parallel Poll** by simultaneously asserting **EOI** and **ATN**.

4.5 Service Request (SRQ)

When a device desires the immediate attention of the **Active Controller** it asserts **SRQ**. It is then the Controller's responsibility to determine which device requested service. This is accomplished with a **Serial Poll** or a **Parallel Poll**.

5. HANDSHAKE LINES

The **GPIB** uses three handshake lines in an "I'm ready - Here's the data - I've got it" sequence. This handshake protocol assures reliable data transfer, at the rate determined by the slowest Listener. One line is controlled by the Talker, while the other two are shared by all Active Listeners. The handshake lines, like the other **IEEE 488** lines, are active low.

5.1 Data Valid (DAV)

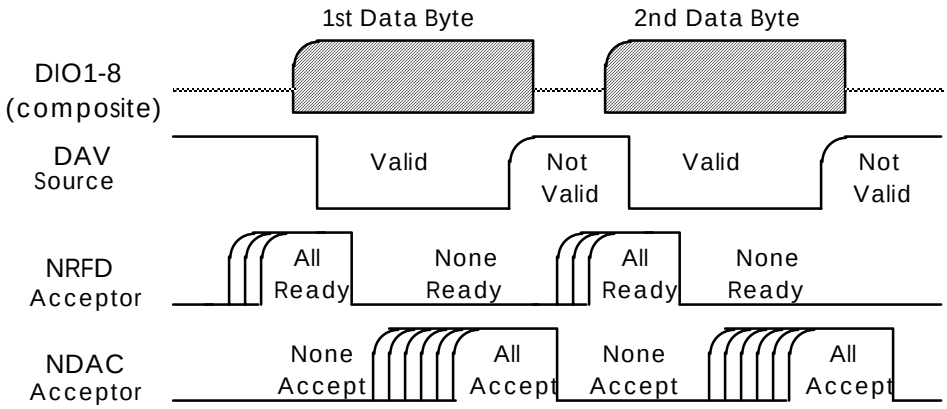
The **DAV** line is controlled by the **Talker**. The **Talker** verifies that **NDAC** is asserted (active low) which indicates that all Listeners have accepted the previous data byte transferred. The **Talker** then outputs data on the bus and waits until **NRFD** is unasserted (high) which indicates that all Addressed Listeners are ready to accept the information. When **NRFD** and **NDAC** are in the proper state, the **Talker** asserts **DAV** (active low) to indicate that the data on the bus is valid.

5.2 Not Ready for Data (NRFD)

This line is used by the **Listeners** to inform the **Talker** when they are ready to accept new data. The **Talker** must wait for each **Listener** to unassert this line (high) which they will do at their own rate when they are ready for more data. This assures that all devices that are to accept the information are ready to receive it.

5.3 Not Data Accepted (NDAC)

The **NDAC** line is also controlled by the **Listeners**. This line indicates to the **Talker** that each device addressed to listen has accepted the information. Each device releases **NDAC** (high) at its own rate, but the **NDAC** will not go high until the slowest Listener has accepted the data byte.



IEEE Bus Handshaking

6. DATA LINES

The **GPIB** provides eight data lines for a bit parallel/byte serial data transfer. These eight data lines use the convention of **DIO1** through **DIO8** instead of the binary designation of **D0** to **D7**. The data lines are bidirectional and are active low.

7. MULTILINE COMMANDS

Multiline (bus) commands are sent by the **Active Controller** over the data bus with **ATN** asserted. These commands include addressing commands for talk, listen, Untalk and Unlisten.

7.1 Go To Local (GTL)

This command allows the selected devices to be manually controlled. ($01)

7.2 Listen Address Group (LAG)

There are 31 (0 to 30) listen addresses associated with this group. The 3 most significant bits of the data bus are set to 011 while the 5 least significant bits are the address of the device being told to listen.

7.3 Unlisten (UNL)

This command tells all bus devices to Unlisten. The same as Unaddressed to Listen. ($3F)

7.4 Talk Address Group (TAG)

There are 31 (0 to 30) talk addresses associated with this group. The 3 most significant bits of the data bus are set to 101 while the 5 least significant bits are the address of the device being told to talk.

7.5 Untalk (UNT)

This command tells bus devices to Untalk. The same as Unaddressed to Talk. ($5F)

7.6 Local Lockout (LLO)

Issuing the **LLO** command prevents manual control of the instrument's functions. ($11)

7.7 Device Clear (DCL)

This command causes all bus devices to be initialized to a predefined or power up state. ($14)

7.8 Selected Device Clear (SDC)

This causes a single device to be initialized to a predefined or power up state. ($04)

7.9 Serial Poll Disable (SPD)

The **SPD** command disables all devices from sending their Serial Poll status byte. ($19)

7.10 Serial Poll Enable (SPE)

A device which is Addressed to Talk will output its Serial Poll status byte after **SPE** is sent and **ATN** is unasserted. ($18)

7.11 Group Execute Trigger (GET)

This command usually signals a group of devices to begin executing a triggered action. This allows actions of different devices to begin simultaneously. ($08)

7.12 Take Control (TCT)

This command passes bus control responsibilities from the current **Controller** to another device which has the ability to control. ($09)

7.13 Secondary Command Group (SCG)

These are any one of the 32 possible commands (0 to 31) in this group. They must immediately follow a talk or listen address. ($60 to $7F)

7.14 Parallel Poll Configure (PPC)

This configures devices capable of performing a **Parallel Poll** as to which data bit they are to assert in response to a **Parallel Poll**. ($05)

7.15 Parallel Poll Unconfigure (PPU)

This disables all devices from responding to a **Parallel Poll**. ($15)

8. MORE ON SERVICE REQUESTS

Most of the commands covered, both uniline and multiline, are the responsibility of the **Active Controller** to send and the bus devices to recognize. Most of these happen routinely by the interface and are totally transparent to the system programmer. Other commands are used directly by the user to provide optimum system control. Of the uniline commands, **SRQ** is very important to the test system and the software designer has easy access to this line by most devices. Service Request is the method by which a bus device can signal to the **Controller** that an event has occurred. It is similar to an interrupt in a microprocessor based system.

Most intelligent bus peripherals have the ability to assert **SRQ**. A DMM might assert it when its measurement is complete, if its input is overloaded or for any of an assortment of reasons. A power supply might **SRQ** if its output has current limited. This is a powerful bus feature that removes the burden from the **System Controller** to periodically inquire, "Are you done yet?". Instead, the **Controller** says, "Do what I told you to do and let me know when you're done" or "Tell me when something is wrong."

Since **SRQ** is a single line command, there is no way for the **Controller** to determine which device requested the service without additional information. This information is provided by the multiline commands for **Serial Poll** and **Parallel Poll**.

8.1 Serial Poll

Suppose the **Controller** receives a service request. For this example, let's assume there are several devices which could assert **SRQ**. The **Controller** issues an **SPE** (Serial Poll enable) command to each device sequentially. If any device responds with DIO7 asserted it indicates to the **Controller** that it was the device that asserted **SRQ**. Often times the other bits will indicate why the device wanted service. This **Serial Polling** sequence, and any resulting action, is under control of the software designer.

8.2 Parallel Poll

The **Parallel Poll** is another way the **Controller** can determine which device requested service. It provides the who but not necessarily the why. When bus devices are configured for Parallel Poll, they are assigned one bit on the data bus for their response. By using the Status bit, the logic level of the response can be programmed to allow logical OR/AND conditions on one data line by more than one device. When **SRQ** is asserted, the **Controller** (under user's software) conducts a **Parallel Poll**. The **Controller** must then analyze the eight bits of data received to determine the source of the request. Once the source is determined, a **Serial Poll** might be used to determine the why.

Of the two polling types, the **Serial Poll** is the most popular due to its ability to determine the who and why. In addition, most devices support **Serial Poll** only.

$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 Address Group
TAG = Talk Address Group
SCG = Secondary Command Group