

ARTISAN®
TECHNOLOGY GROUP

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

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

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

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

Custom engineering
so your equipment works exactly as you specify.

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

Expert team | Trust guarantee | 100% satisfaction

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

Visit our website - *Click HERE*

RACAL INSTRUMENTS™

1256

SWITCHING SYSTEM

Publication No. 980855 Rev. A

EADS North America Test and Services,
a division of EADS North America, Inc.
4 Goodyear, Irvine, CA 92618
Tel: (800) 722-2528, (949) 859-8999; Fax: (949) 859-7139

info@eads-nadefense.com
sales@eads-nadefense.com
helpdesk@eads-nadefense.com
http://www.eads-nadefense.com



PUBLICATION DATE: August 21, 2008

Copyright 2000 by EADS North America Test and Services, a division of EADS North America, Inc. Printed in the United States of America. All rights reserved. This book or parts thereof may not be reproduced in any form without written permission of the publisher.

THANK YOU FOR PURCHASING THIS EADS NORTH AMERICA TEST AND SERVICES PRODUCT

For this product, or any other EADS North America Test and Services a division of EADS North America, Inc. ("EADS North America Test and Services") product that incorporates software drivers, you may access our web site to verify and/or download the latest driver versions. The web address for driver downloads is:

http://www.eads-nadefense.com/downloads

If you have any questions about software driver downloads or our privacy policy, please contact us at:

info@eads-nadefense.com

WARRANTY STATEMENT

All EADS North America Test and Services products are designed and manufactured to exacting standards and in full conformance to EADS ISO 9001:2000 processes.

This warranty does not apply to defects resulting from any modification(s) of any product or part without EADS North America Test and Services express written consent, or misuse of any product or part. The warranty also does not apply to fuses, software, non-rechargeable batteries, damage from battery leakage, or problems arising from normal wear, such as mechanical relay life, or failure to follow instructions.

This warranty is in lieu of all other warranties, expressed or implied, including any implied warranty of merchantability or fitness for a particular use. The remedies provided herein are buyer's sole and exclusive remedies.

For the specific terms of your standard warranty, or optional extended warranty or service agreement, contact your EADS North America Test and Services customer service advisor. Please have the following information available to facilitate service.

1. Product serial number
2. Product model number
3. Your company and contact information

You may contact your customer service advisor by:

E-Mail: Helpdesk@eads-nadefense.com

Telephone: +1 800 722 3262 (USA)

Fax: +1 949 859 7309 (USA)

RETURN of PRODUCT

Authorization is required from EADS North America Test and Services before you send us your product for service or calibration. Call or contact the Customer Support Department at 1-800-722-3262 or 1-949-859-8999 or via fax at 1-949-859-7139. We can be reached at: helpdesk@eads-nadefense.com.

PROPRIETARY NOTICE

This document and the technical data herein disclosed, are proprietary to EADS North America Test and Services, and shall not, without express written permission of EADS North America Test and Services, be used, in whole or in part to solicit quotations from a competitive source or used for manufacture by anyone other than EADS North America Test and Services. The information herein has been developed at private expense, and may only be used for operation and maintenance reference purposes or for purposes of engineering evaluation and incorporation into technical specifications and other documents which specify procurement of products from EADS North America Test and Services.

DISCLAIMER

Buyer acknowledges and agrees that it is responsible for the operation of the goods purchased and should ensure that they are used properly and in accordance with this handbook and any other instructions provided by Seller. EADS North America Test and Services products are not specifically designed, manufactured or intended to be used as parts, assemblies or components in planning, construction, maintenance or operation of a nuclear facility, or in life support or safety critical applications in which the failure of the EADS North America Test and Services product could create a situation where personal injury or death could occur. Should Buyer purchase EADS North America Test and Services product for such unintended application, Buyer shall indemnify and hold EADS North America Test and Services, its officers, employees, subsidiaries, affiliates and distributors harmless against all claims arising out of a claim for personal injury or death associated with such unintended use.

FOR YOUR SAFETY

Before undertaking any troubleshooting, maintenance or exploratory procedure, read carefully the **WARNINGS** and **CAUTION** notices.



This equipment contains voltage hazardous to human life and safety, and is capable of inflicting personal injury.



If this instrument is to be powered from the AC line (mains) through an autotransformer, ensure the common connector is connected to the neutral (earth pole) of the power supply.



Before operating the unit, ensure the conductor (green wire) is connected to the ground (earth) conductor of the power outlet. Do not use a two-conductor extension cord or a three-prong/two-prong adapter. This will defeat the protective feature of the third conductor in the power cord.



Maintenance and calibration procedures sometimes call for operation of the unit with power applied and protective covers removed. Read the procedures and heed warnings to avoid “live” circuit points.

Before operating this instrument:

1. Ensure the proper fuse is in place for the power source to operate.

2. Ensure all other devices connected to or in proximity to this instrument are properly grounded or connected to the protective third-wire earth ground.

If the instrument:

- fails to operate satisfactorily
- shows visible damage
- has been stored under unfavorable conditions
- has sustained stress

Do not operate until performance is checked by qualified personnel.

Racal Instruments

EC Declaration of Conformity

We

Racal Instruments Inc.
4 Goodyear Street
Irvine, CA 92718

declare under sole responsibility that the

1256 Switching System
P/N 407734

conforms to the following Product Specifications:

Safety: EN 61010-1

EMC: Immunity: EN61326, Class A, Table 1
Emissions: EN61326, Class A, Table 3

Supplementary Information:

The product herewith complies with the requirements of EN61010-1 and EN61326.

Irvine, CA, February 22, 2001

Quality Manager

This page was left intentionally blank.

Table of Contents

Chapter 1 1-1

INTRODUCTION 1-1

Overview 1-1

Features 1-2

Ease Of Use 1-2

Front and Rear Panels 1-2

Powerful Software 1-4

Items Shipped with the 1256 1-4

Chapter 2 2-1

GETTING STARTED 2-1

Mounting Options 2-1

Main AC Power 2-1

Installing Plug-Ins 2-3

Numbering of Plug-In Slots 2-3

Connecting to Plug-Ins 2-4

Activating Main AC Power 2-4

Self-Test 2-5

Verifying the Installed Plug-Ins 2-5

Closing a Relay 2-6

Opening a Relay 2-7

In Case of Trouble 2-7

Switching and Heat Dissipation 2-8

Chapter 3 3-1
USING THE FRONT-PANEL CONTROLS 3-1
Introduction 3-1
Front Panel 3-2
Understanding the Hierarchical Menu 3-2
The Five Menus 3-7
Operations Menu 3-8
Viewing Relay States 3-10
Enabling a Digital Port 3-13
Path Control Menu 3-14
Defining a Path 3-14
Closing a Path 3-15
Opening a Path 3-16
Settings Menu 3-18
Store/Recall Menu 3-25
System Menu 3-29
Complete Menu Listing 3-29

Chapter 4 4-1
GPIB AND RS-232 INTERFACES 4-1
Introduction 4-1
Commands 4-1
Using the IEEE-488 (GPIB) Interface 4-1
Basic GPIB Information 4-1
GPIB Cabling Guidelines 4-2
Connecting the GPIB Cables 4-2

Selecting the GPIB Interface 4-3
Setting the GPIB Address 4-4
Service Request (SRQ) 4-5
Using the RS-232 Interface 4-5
Basic RS-232 Information 4-5
Equipment Required 4-6
DTE Versus DCE 4-6
Connecting to a Computer or Data Terminal 4-6
Connecting to a Modem 4-7
Selecting the RS-232 Interface 4-9
Setting the Flow Control Option 4-10
Description 4-10
Setting the Baud Rate 4-12
Setting the Message Termination Characters 4-13
Setting the Data Bits, Stop Bits, and Parity 4-14
Using Microsoft HyperTerminal 4-16

Chapter 5 5-1
SCPI COMMAND BASICS 5-1
SCPI Command Overview 5-1
Input Format 5-1
Command Keyword Long Form and Short Form 5-3
Case Sensitivity 5-3
Optional Keywords 5-4
Querying Parameter Setting 5-4
SCPI Command Terminator 5-4
IEEE-STD-488.2 Common Commands 5-6

IEEE 488.2 Common Commands 5-6
IEEE-488. 2 Status Description 5-7
Standard EVENT STATUS Register 5-9
STATUS BYTE Register 5-11
SERVICE Register ENABLE Register 5-13
*CLS Command 5-13
*ESE Command 5-14
*ESE? Query 5-15
*ESR? Query 5-15
*SRE Command 5-15
*SRE? Query 5-15
*STB? Query 5-16
*OPC Command 5-16
*OPC? Query 5-16
*IDN? Query 5-16
*RST Command 5-17
*TST? Query 5-18
*RCL Command 5-18
*SAV Command 5-18
*SPE Command 5-19
*TRG Command 5-20
*WAI Command 5-20
SCPI Status Registers 5-20
SCPI Parameter Type 5-23
Numeric Parameters 5-23
Discrete Parameters 5-23
Boolean Parameters 5-24

Command Input Buffer 5-25
Reply Output Buffer 5-26
Specifying Channels in Commands 5-27
Naming Relay Cards 5-28
Defining Module Names 5-28
Removing Module Names 5-29
Reading the Presently Defined Module Names 5-30
Reading the Module Address for a Module Name 5-31
Storing the Module Names in Non-Volatile Memory 5-31
Naming a Path 5-32
Defining Path Names 5-32
Removing Path Names 5-34
Reading the Presently Defined Path Names 5-34
Reading the Channel List for a Path Name 5-35
Storing Path Names in Non-Volatile Memory 5-35
Closing Relays 5-35
Opening Relays 5-37
Checking for Installed Plug-Ins 5-37
System Commands 5-38
Reading Error Messages 5-38
SCPI Compliance Information 5-39
User-Interface Lockout 5-39
Non-Volatile Update 5-40

Chapter 6 6-1
SCPI COMMAND REFERENCE 6-1
General 6-1

Chapter 7 7-1
DIGITAL MODULE OPERATION 7-1
General 7-1
Digital Module Ports 7-1
Asynchronous Digital Operation 7-1
Synchronous Digital Operation 7-2
Mixing Synchronous and Asynchronous Modes of Operation 7-2
Specifying Ports on a Digital Module 7-3
Selecting the Mode of Operation 7-4
Enabling and Disabling the Ports 7-5
Using the Asynchronous Mode of Operation 7-5
Using the Synchronous Mode of Operation 7-6
Setting Up the Synchronous Test 7-8
Arming the Digital Modules 7-12
Checking for Data Transfer Completion 7-13
Reading Data from a Synchronous Input Port 7-14
Clearing Data from Synchronous Input and Output Ports 7-15
Synchronous Control and Status Pins 7-15
Synchronous and Asynchronous Example 7-16

Chapter 8 8-1

ADVANCED SCPI OPERATIONS 8-1

Introduction 8-1

Using ROUTE: CONFIGURE to Control Switching Sequence 8-1

Include Lists 8-2

Using Exclude Lists 8-5

Using External Triggering 8-7

Selecting the Trigger Source 8-7

Selecting the Trigger Count 8-8

Selecting a Trigger Delay 8-9

Arming and Disarming the 1256 8-9

Generating a Single Trigger 8-12

Output Trigger Signals from the 1256 8-12

Selecting an Output Trigger Destination 8-12

Using Scan Lists and External Trigger Input 8-13

Scan Lists 8-14

Defining a Scan List 8-15

Advance Verification Features 8-17

Confidence Mode 8-18

Using Relay Verification 8-18

Saving/Recalling the Verification Masks 8-20

Automatically Recalling Verification Masks at Startup 8-21

Chapter 9 9-1

PRODUCT SUPPORT 9-1

Product Support 9-1

Warranty 9-1

Appendix A A-1

SPECIFICATIONS A-1

General A-1

System A-1

Remote Interface A-2

Environmental A-3

EMC A-3

Safety A-4

Reliability A-4

Mechanical A-4

Appendix B B-1

RACK MOUNTING B-1

Items Included B-1

Installing Rack-Mount Ears B-1

Installing the 1256 Chassis Into a Rack B-2

Appendix C C-1

TROUBLESHOOTING C-1

How to Use this Section C-1

Power-Up C-1

Front-Panel Controls C-2

RS-232 Remote Interface C-3

IEEE-488 (GPIB) Remote Interface C-5

Plug-In Modules C-5

Non-Volatile Memory C-6

Appendix D D-1

PLUG-IN MODULES D-1

How to Use This Section D-1

Index 1

List of Figures

Figure 1-1, 1256 Switching System 1-1
Figure 1-2, 1256 Features, Front Panel 1-2
Figure 1-3, 1256 Features, Rear Panel 1-3
Figure 1-4, Dimensions 1-3

Figure 2-1, Changing Fuse for Main AC Power 2-1
Figure 2-2, Inserting a Plug-In 2-3
Figure 2-3, Numbering of Switch Plug-In Slots 2-4
Figure 2-4, Activating Main AC Power 2-4

Figure 3-1, Front-Panel Controls 3-2

Figure 4-1, Using the GPIB Interface 4-3
Figure 4-2, Using the RS-232 Interface 4-7
Figure 4-3a, Using the 1256 with a Computer 4-8
Figure 4-3b, Using the 1256 with a Modem 4-8
Figure 4-4, HyperTerminal Connection Dialog Box 4-17
Figure 4-5, Phone Number Dialog Box 4-17
Figure 4-6, COM Properties Dialog Box 4-18
Figure 4-7, Main Communications Screen 4-19
Figure 4-8, Properties Dialog Box 4-20
Figure 4-9, Settings Dialog Box 4-20
Figure 4-10, ASCII Setup Dialog Box 4-21

Figure 5-1, IEEE-488.2 Status Reporting Model.....5-8

Figure 7-1 (A), Port 4 Data after DIG:SYNC:DATA (@7(4)),10,20,30,40.....7-10

Figure 7-1 (B), Port 4 Data after DIG:SYNC:DATA (@7(4)),50,60,70,80,90.....7-10

Figure 7-1 (C), Port 4 Data after DIG:SYNC:INDEX (@7(4)),3.....7-11

Figure 7-1 (D), Port 4 Data after DIG:SYNC:DATA (@7(4)),77,78.....7-11

Figure 8-1, State Transition Diagram For Arming and Triggering the 1256.....8-11

Figure 8-2, Using External Trigger In and External Trigger Out.....8-13

Figure B-1, Installing the Rack-Mount Ears.....B-1

Figure B-2, Rack Mounting.....B-2

List of Tables

Table 3-1, Accessibility of Features.....3-1

Table 5-1, Power-On and Reset State.....5-17

Table 6-1, Commands Implemented by the 1256.....6-3

This page was left intentionally blank.

DOCUMENT CHANGE HISTORY

Revision	Date	Description of Change
A	8/21/08	Revised per EO 29290 Revised format to current standards. Company name revised throughout manual. Manual now revision letter controlled. Added Document Change History Page xiii.

This page was left intentionally blank.

Chapter 1
INTRODUCTION

Overview

The 1256 Switching System is a high-performance switching and control system in a compact 2U rack-mountable package. The 1256 can contain up to eight Adapt-a-Switch plug-ins for switching and digital I/O. These plug-ins provide a wide range of switching capability:

- High-current to 13A
- High-voltage to 1kV
- RF and microwave to 18GHz
- Digital I/O with 96 channels per plug-in

The user can easily insert these plug-ins into the 1256 Switching System to meet the needs of nearly any switching application.

Figure 1-1 shows a picture of the 1256.



Figure 1-1, 1256 Switching System

Features

Ease Of Use

The 1256 Switching System has many design features to make it easy and convenient to use:

- **Fast Setup.** There are no configuration jumpers or switches to be set. The 1256 is ready to use as soon as the plug-ins have been inserted.
- **Flexible Controller Interface.** The controller (host computer) may communicate with the 1256 using either IEEE-488 (GPIB) or RS-232 interfaces. This allows an external computer to automate the operation of the 1256.
- **Front-Panel Controls.** The front-panel controls provide an intuitive user interface for manually controlling the plug-ins and setting user preferences. The front-panel controls provide access to all interface parameters, such as GPIB address and RS-232 baud rates. These controls are often helpful during system setup.
- **Command Set Commonality.** The 1256 supports the Standard Commands for Programmable Instruments (SCPI) Language. This command set is used in other switching platforms such as 1260-Series VXI. For an introduction to the SCPI command set and its usage, see **Chapter 5, SCPI COMMAND BASICS**. For a complete description of all SCPI commands for the 1256, see **Chapter 6, SCPI COMMAND REFERENCE**.

Front and Rear Panels

Figures 1-2 and 1-3 show the features of the front and rear panels, respectively. **Figure 1-4** gives the physical dimensions of the 1256 chassis.

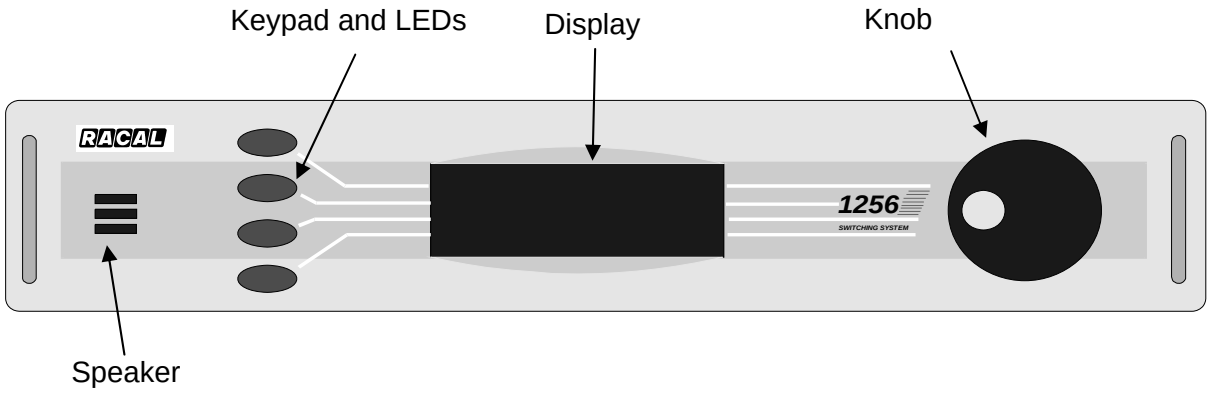


Figure 1-2, 1256 Features, Front Panel

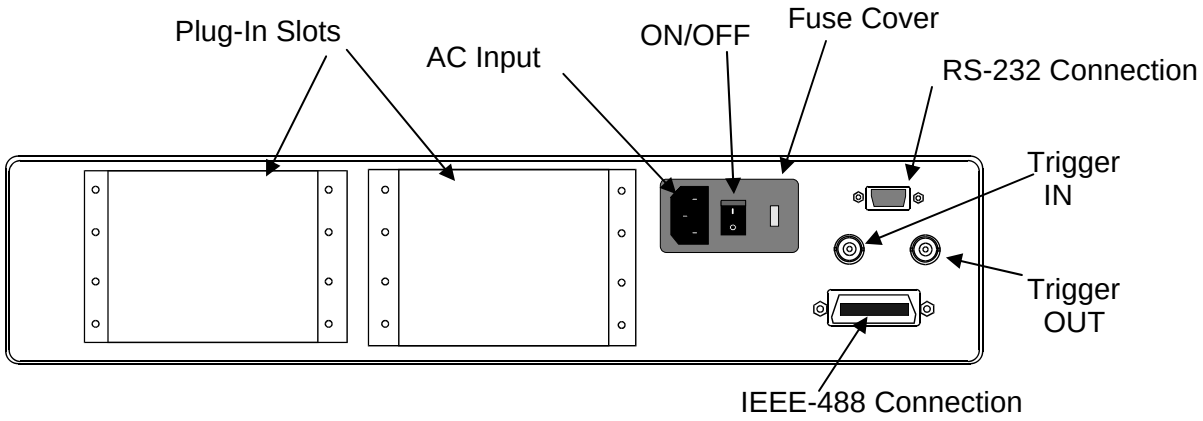


Figure 1-3, 1256 Features, Rear Panel

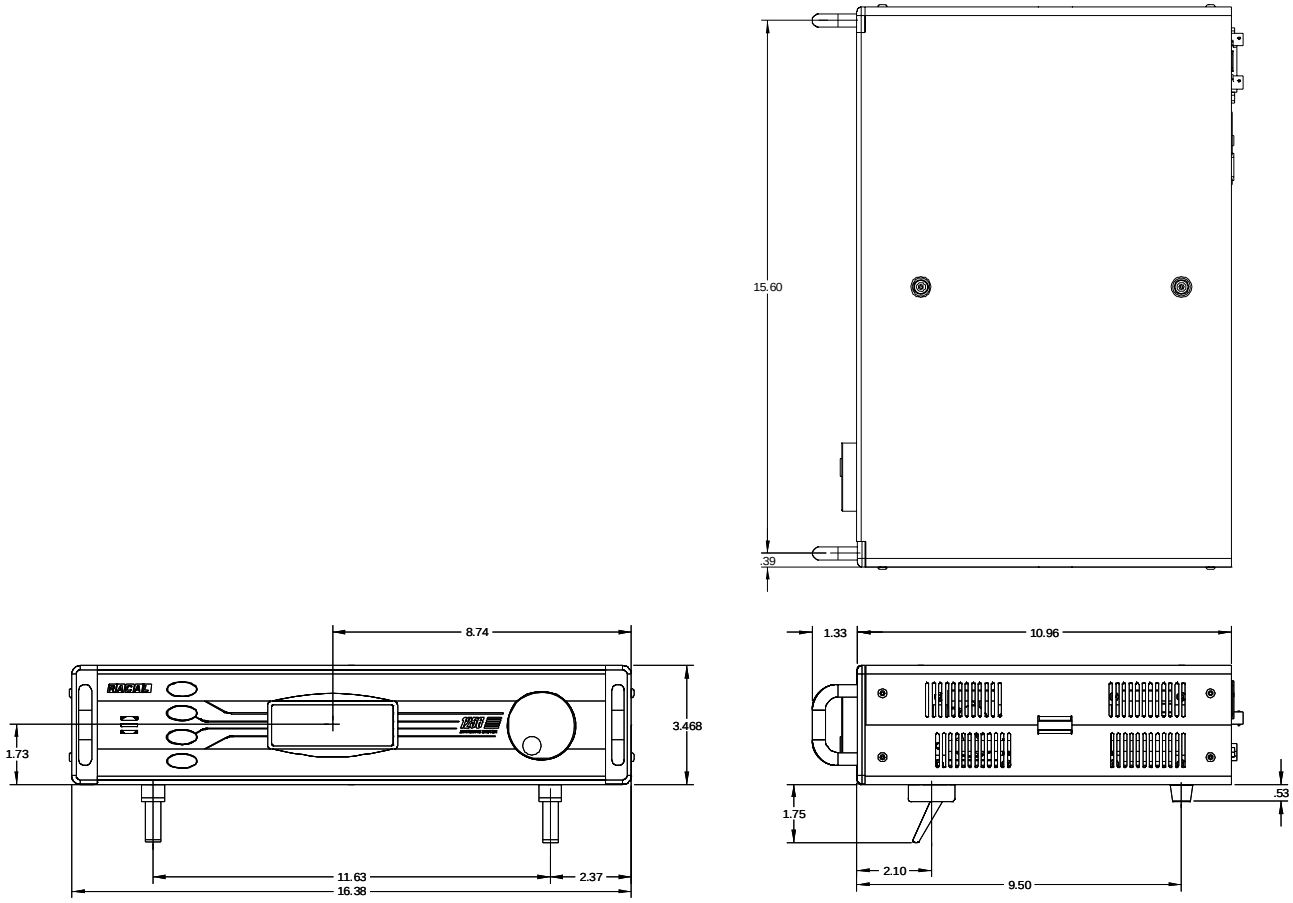


Figure 1-4, Dimensions

Powerful Software

The 1256 provides extensive software capabilities, including:

- **Exclude Lists.** Allows you to prevent simultaneous closure of switches in a specified group. The switches are then mutually exclusive.
- **Equate Lists.** Reduces programming effort. When two or more modules are on an equate list, they respond simultaneously to a command sent to any of them.
- **External Trigger Input.** Allows an external device to trigger switching operations. The external device connects to the *External Trigger In* connector.
- **External Trigger Output.** Provides a pulse on the *External Trigger Out* connector to trigger an external device, such as a DMM, after the relay states have changed.
- **Scan Lists.** Allows you to specify a sequence of relay closures to step through in response to an external trigger source. A scan list may include relays on one module or multiple modules.
- **Programmable Delay.** Allows you to specify the amount of time delay from a relay state change to an external trigger pulse.
- **Verification.** Allows you to verify the status of all switches.
- **Path-Level Switching.** Allows complex paths to be defined with open and close elements, stored in non-volatile memory, and be controlled remotely or through the front-panel display.
- **Confidence Mode.** Automatically verifies correct application of voltage to relay coils.
- **Non-Volatile Memory.** Stores and recalls switch configurations and 1256 user preferences (such as GPIB address and RS-232 baud rate).
- **Power-Up Recall.** Automatically recalls a complete switching system configuration from non-volatile memory at power-up.
- **Self-Test.** Ensures that critical system components are functioning correctly. This test executes automatically at power-up.

Items Shipped

The following items are shipped with each 1256 Switching System:

with the 1256

ITEM	PART NUMBER	QUANTITY
1256 Chassis Assembly	407724	1
User Manual	980855	1
AC Power Mains Cable	600620	1
Blanking Plates (to cover empty plug-in slots)	407667	4
Software Driver	921678	1

The following accessories may be ordered separately:

- Option 60: Rack Mount Kit P/N 407731
- Option 04: European Power Cord (unterminated) P/N 407730
- GPIB Cable, 1-meter P/N 500310-001
- GPIB Cable, 2-meter P/N 500310-002
- RS-232 Cable, 10 ft., 9-pin female D-Sub to 9-pin male D-Sub P/N 602248

This page was left intentionally blank.

Chapter 2
Getting Started

Mounting Options

You may use the 1256 Switching System on a table or mount it in a standard NEMA 19" rack. See **Appendix B, RACK MOUNTING the 1256** for instructions on installing the rack mount kit (Option 60) and placing the 1256 in a rack.

Main AC Power

The 1256 Switching System requires a power input of 90VAC to 250VAC, 43Hz to 63Hz. The 1256 requires no adjustments, settings, or changes in fuse capacity to accommodate variations of voltage and frequency within these ranges. Simply connect the 1256 AC power inlet to the power source.

CAUTION:

Before connecting the unit to a power source, make sure that the installed fuse has the capacity marked on the rear panel of the 1256. For instructions on changing the fuse, refer to Figure 2-1, Changing Fuse for Main AC Power.

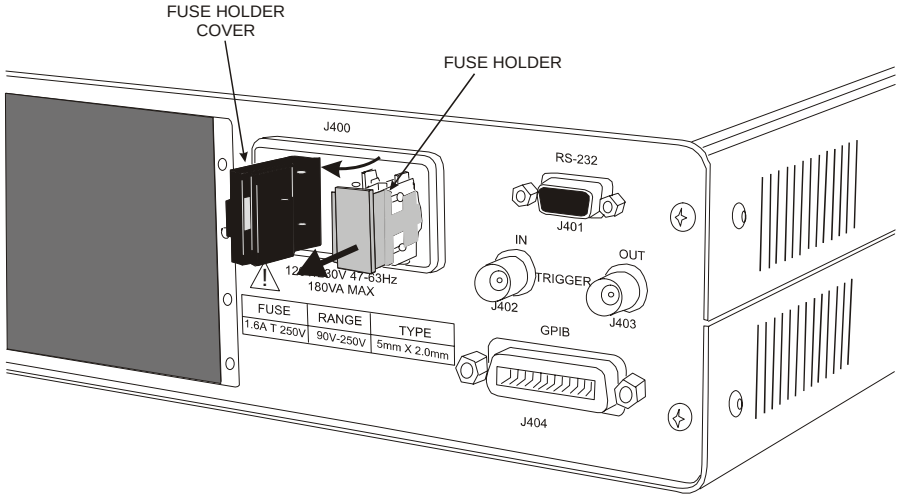


Figure 2-1, Changing Fuse for Main AC Power

The 1256 is grounded in accordance with MIL-T-28800, protecting the user from possible injury due to shorted circuits.



CAUTION:

Use only AC power outlets that have a protective ground. DO NOT USE a two-conductor extension cord or 3-prong to 2-prong adapter that does not provide a protective ground connection.

The rear-panel AC input socket provides a protective ground terminal. The 1256 is supplied with a detachable 3-conductor power cord that connects to this protective terminal. Only this cord, or an equivalent, should be used. Connection of the power cord to the power outlet must be made in accordance with the following standard color code:

Function	Power Outlet Wire Color	
	American	European
Line (Live)	Black	Brown
Neutral	White	Blue
Ground (Earth)	Green	Green/Yellow



CAUTION:

Position the 1256 to allow easy access to the power switch on the rear-panel.



CAUTION:

Using this equipment in a manner not specified in this manual may impair the protection it normally provides.

Installing Plug-Ins

Make sure the power is turned off before installing or removing plug-ins. Inserting or removing plug-ins with power on may cause damage to the 1256 and the plug-ins.

Figure 2-2, Inserting a Plug-In shows the proper way to install an Adapt-a-Switch plug-in into the 1256 chassis. When inserting the plug-in, engage the edges with the card guides of the 1256. Carefully slide the plug-in into the chassis until it stops, then push it firmly to engage its rear connectors with the 1256 backplane. When the plug-in is fully seated, its panel is flush with the rear panel of the 1256 chassis.

Each plug-in has retainer screws at the ends of its front panel. Carefully thread the retainer screws into the 1256 chassis and tighten securely.

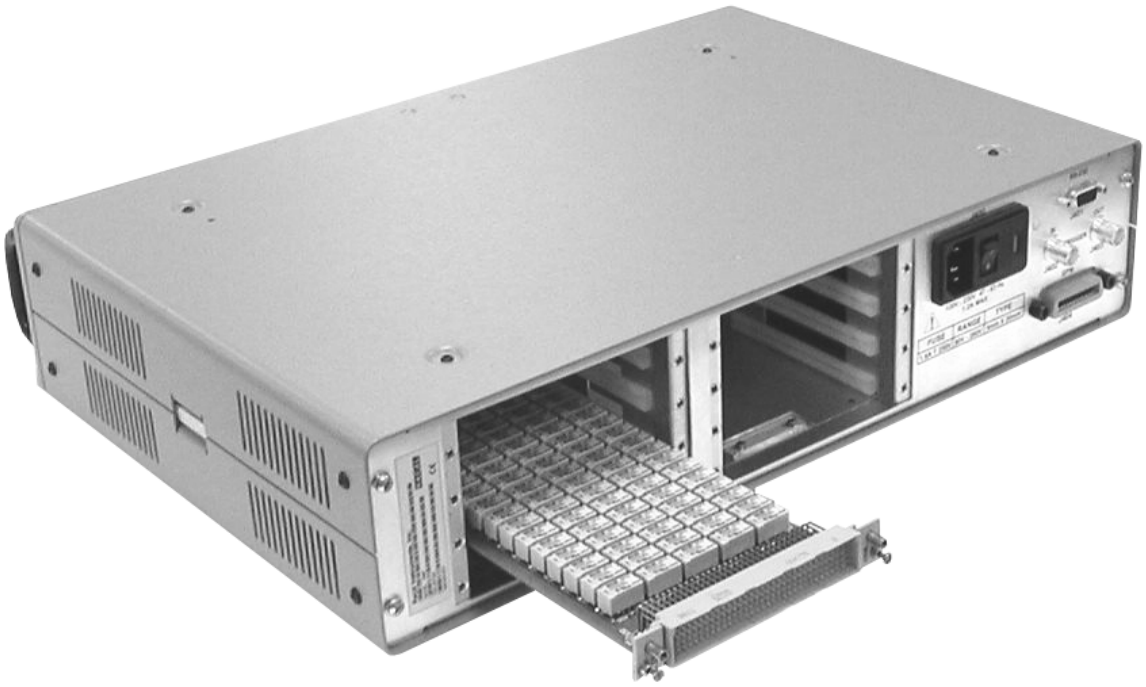


Figure 2-2, Inserting a Plug-In

Numbering of Plug-In Slots

The 1256 references each plug-in by the number of the chassis slot into which it is installed. There are no configuration jumpers or settings. Refer to **Figure 2-3, Numbering of Switch Plug-In Slots**.

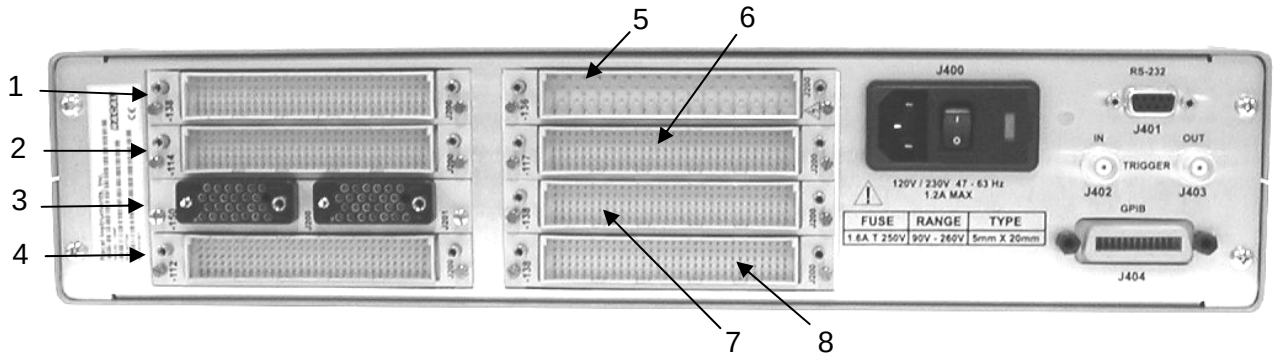


Figure 2-3, Numbering of Switch Plug-In Slots

Connecting to Plug-Ins

For information about connecting the switch plug-ins to your external hardware, refer to the user manuals of the individual plug-ins.

Activating Main AC Power

You are now ready to apply power to the 1256 Switching System. See **Figure 2-4, Activating Main AC Power** for instructions on activating main power.

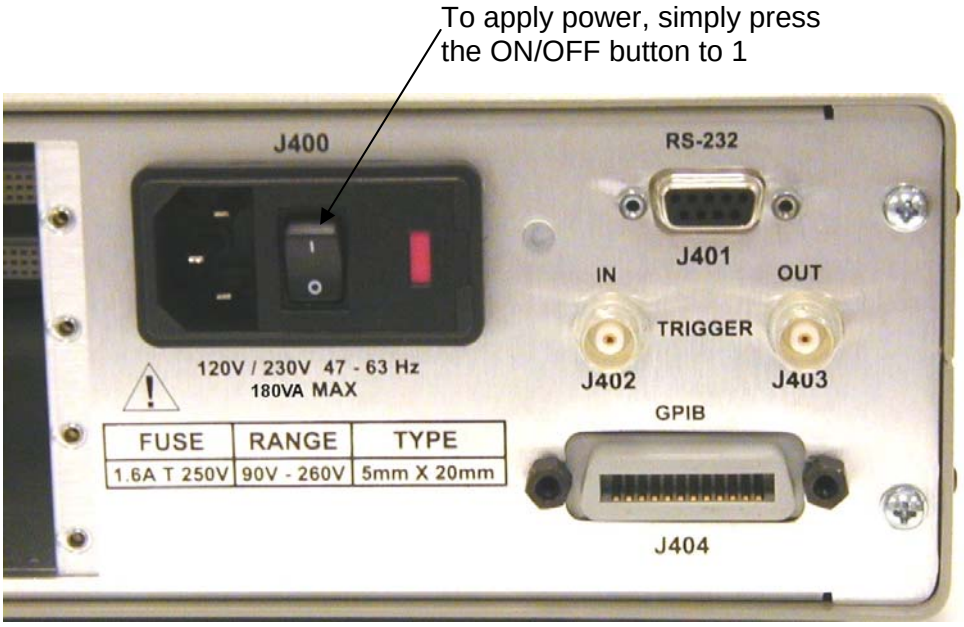


Figure 2-4, Activating Main AC Power

Self-Test

The 1256 automatically performs a self-test immediately after power is applied. The self-test completes within ten seconds after power application. If a failure is detected, a message appears on the front-panel display. If this occurs, refer to **Appendix C, TROUBLESHOOTING** to solve the problem.

If the self-test does not detect a problem, the front-panel display will show the main menu. The top line (line 1) of the display appears as follows:

```
----OPERATIONS----
```

The contents of lines 2 through 4 depend upon which plug-ins are installed.

Verifying the Installed Plug-Ins

The front-panel controls provide a way to verify which plug-ins are installed in the 1256. For example, suppose the following plug-ins are installed:

Slot 1:	1260-121A
Slot 2:	1260-138A
Slot 3:	1260-118
Slot 4:	Empty: no module
Slot 5:	Empty: no module
Slot 6:	Empty: no module
Slot 7:	Empty: no module
Slot 8:	Empty: no module

You may verify the installation of these plug-ins as follows:

1. Press the top key on the keypad next to the display (See **Figure 1-2**). Rotate the knob until the top line reads:

   ```
   >----OPERATIONS----<
   ```

   The markers “>” and “<” indicate that this line has been selected by pressing the corresponding button. When a line is selected, rotating the knob changes the contents of that line.

2. Select line 2 (next to top line) by pressing the key just below top key. The display describes the plug-in that is in slot 1:

   ```
    ----OPERATIONS----
   >Slot 01: 1260-121A<
    Channel: 0
     Status: Open
   ```

   In this example, the plug-in is 1260-121A. Lines 3 and 4 indicate the status of the plug-in (discussed later).

3. Rotate the knob one “click” to the right (clockwise) to check the

contents of the next slot (slot 2):

```
 ----OPERATIONS----
>Slot 02: 1260-138A<
 Channel: 0
  Status: Open
```

The plug-in is 1260-138A.

4. Continue to rotate the knob to view the contents of the remaining slots.

If any of the installed switch modules do not appear on the display as you scroll through the slot numbers, refer to **Appendix C, TROUBLESHOOTING** to correct the problem.

To use the IEEE-488 or RS-232 interfaces to verify plug-ins, see **Chapter 5, SCPI COMMAND BASICS**. Look under the heading **Checking for Installed Plug-Ins**.

Closing a Relay

The front-panel controls may be used to close relays on the plug-ins. For example, suppose a 1260-118 plug-in is installed in slot 3. To close relay channel 4 on this plug-in, proceed as follows:

1. Press the first key next to the display. Rotate the knob until the top line reads:

   ```
   >----OPERATIONS----<
   ```

2. Select line 2 (next to top line) by pressing the second key (just below the first one). This highlights line 2 (indicated by the “>” and “<” markers) so that you may select the slot to work with:

   ```
    ----OPERATIONS----
   >Slot 01: 1260-121A<
    Channel: 0
     Status: Open
   ```

3. Rotate the knob until line 2 shows the contents of slot 3:

   ```
    ----OPERATIONS----
   >Slot 03: 1260-118 <
    Channel: 0
     Status: Open
   ```

4. Press the third key to select line 3 of the display:

   ```
     ----OPERATIONS----
    Slot 03: 1260-118
   >Channel: 0        <
     Status: Open
   ```

5. Rotate the knob until line 3 shows the desired channel number, which is 4 in this example:

```
 ----OPERATIONS----
 Slot 03: 1260-118
>Channel: 4        <
  Status: Open
```

6. Press the fourth key to select line 4 (bottom line). Rotate the knob until the display reads:

```
 ----OPERATIONS----
 Slot 03: 1260-118
 Channel: 0
<Set to?: Closed
```

7. Press the fourth key to confirm the closure of the channel. The relay then closes, and the display shows:

```
 ----OPERATIONS----
 Slot 03: 1260-118
 Channel: 0
> Status: Closed   <
```

Pressing any other key cancels the change.

Opening a Relay

To open a relay, follow the steps given in the above section, **Closing a Relay**, except in step 7 rotate the knob until the display reads:

```
 ----OPERATIONS----
 Slot 03: 1260-118
 Channel: 0
<Set to?: Open
```

Then press the fourth key to confirm the opening of the relay. Pressing any other key cancels the change.

In Case of Trouble

If you have difficulties using the 1256 Switching System, refer to **Appendix C, TROUBLESHOOTING** to resolve the problem. If the problem persists, refer to **Chapter 9, PRODUCT SUPPORT** for instructions on contacting EADS North America Test and Services for assistance.

Switching and Heat Dissipation

The 1256 accommodates all Adapt-a-Switch plug-ins, which gives it exceptionally high-density switching capability. When switching a large quantity of signals near its maximum rated loading current, a plug-in generates heat due to the resistance of its conducting paths and relay contacts. To prevent excessive heat, certain guidelines apply regarding the maximum number of channels that may carry full rated current at one time. In a high-density switching system, it is not reasonable to have all relays simultaneously closed, all carrying full rated current, operating at the maximum rated ambient temperature. Some form of de-rating is necessary.

Whether or not a plug-in requires de-rating, and to what degree, depends on the type of plug-in, its switching density, and the ambient temperature. The following guidelines provide practical de-rating for plug-ins used in the 1256 chassis.

Signal Switch Plug-Ins

Signal switch plug-ins consist of low- to moderate-current modules such as the 1260-117 and 1260-118. For these plug-ins, up to 25% of the channels on each plug-in may be simultaneously closed under full-rated current if the ambient temperature is 50^{O}C or less. If the ambient temperature is greater than 50^{O}C, de-rate the current by 20%.

Power Switch Plug-Ins

Power switch plug-ins are high-current modules such as the 1260-120 and 1260-121. For these, their full rated current is allowed on all channels simultaneously at ambient temperatures up to 50^{o}C. For ambient temperatures above 50^{o}C, de-rate the current by 20%.

Digital I/O, RF/Microwave, Matrix, and Multiplexer Plug-Ins

Digital I/O (e.g., 1260-114), RF/microwave (e.g. 1260-150), matrix (e.g., 1260-145), and multiplexer (e.g. 1260-138A) plug-ins require no de-rating. As always, the current through any connector pin must not exceed the rated channel current. For example, if current passes simultaneously through two inputs of the same multiplexer, the common side carries the current from both. The common current must not exceed the rated channel current.

Chapter 3
Using the Front-Panel Controls

Introduction

The front-panel controls allow you to use the main features of the 1256 without the use of a remote interface such as IEEE-488 (GPIB) or RS-232. This may be useful during initial setup or when troubleshooting a system.

Table 3-1, Accessibility of Features lists the features of the 1256 and indicates their accessibility via remote interface and front-panel controls.

Table 3-1, Accessibility of Features

Feature	Accessibility	
	Remote Interface	Front-Panel Controls
Open/Close Relays	•	•
Write/View Digital Channels	•	•
Select Remote Interface		•
Set RS-232/GPIB Parameters (e.g. address)		•
Set Display Parameters (e.g. brightness, format, screen saver)		•
View Firmware Rev. Letter	•	•
Work With Scan Lists	•	
Work With Include Lists	•	
Work With Exclude Lists	•	
Generate *Trigger Out* Pulse	•	
Set Up Response To *Trigger In* Pulse	•	
Define/Save/Recall Pathnames	•	
Open/Close Pathnames	•	•
Save/Recall Switch States (flash memory)	•	•
Save/Recall Menu Preferences (flash memory)		•

Front Panel

The front-panel controls consist of four keypad switches (pushbuttons), a four-line alphanumeric display, and a rotary knob (See **Figure 3-1, Front-Panel Controls**).

The display contains four alphanumeric lines of text. To the left of the display are the four keys, each corresponding to one line of the display. The keys select the display line to work with. After selecting a display line, the user may rotate the knob to change the parameter indicated on the selected line.

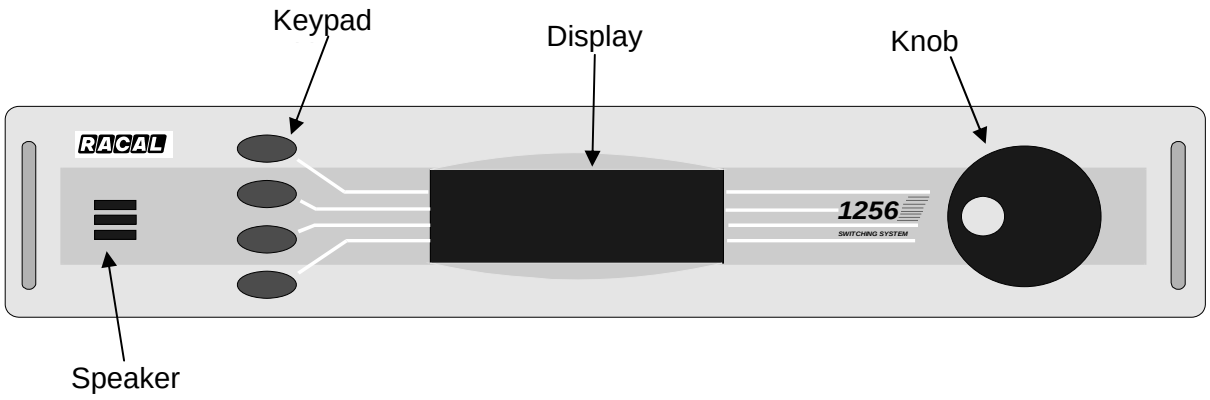


Figure 3-1, Front-Panel Controls

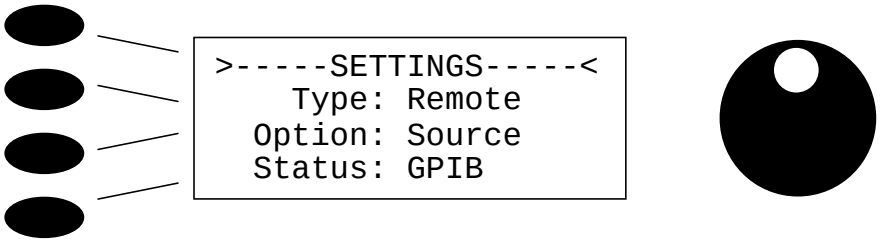
Understanding the Hierarchical Menu

The 1256 uses a multiple-level, hierarchical menu tree. To help the user visualize the menu tree, the menu is limited to four levels. Each hierarchical level is shown on its own line on the front-panel display. The top line shows the top level.

The user begins by selecting a general type of function on line 1 (top line). Then, on line 2, the 1256 displays choices that are more specific to the general function that was selected on line 1. After the user selects from the choices on line 2, the 1256 shows related parameters on line 3. Once the user chooses a parameter from line 3, the 1256 displays the current status, or value, of this parameter on line 4. On line 4, the user may choose from among different values for the parameter.

For example, suppose slot 3 of the 1256 contains a 1260-120 plug-in. To close relay channel 2 on this plug-in, the user proceeds as follows:

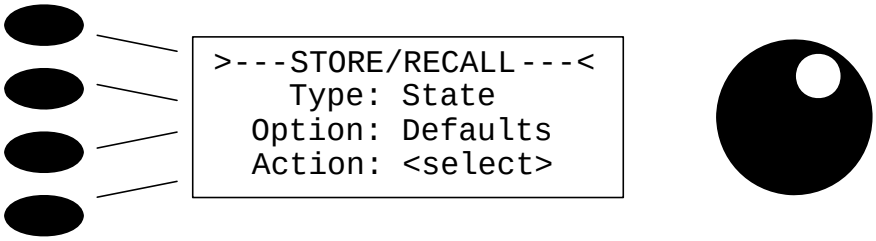
1. Press the top keypad switch. This selects line 1 (top line) of the display, as follows:



NOTE:

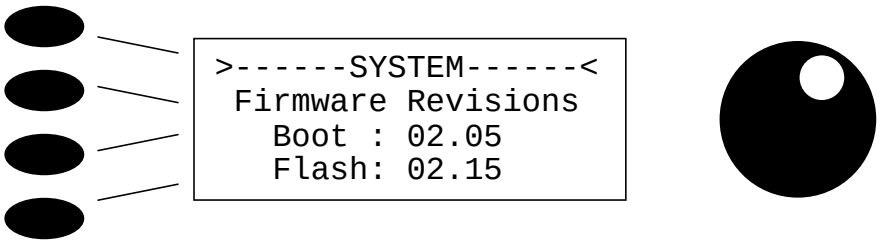
The selection markers “>” and “<” indicate the display line that is currently selected.

2. Rotate the knob to the right (clockwise). The front-panel speaker makes a clicking noise as the knob rotates. At each click, the selected display line shows another choice. After the first click, the display shows the following choice on line 1:

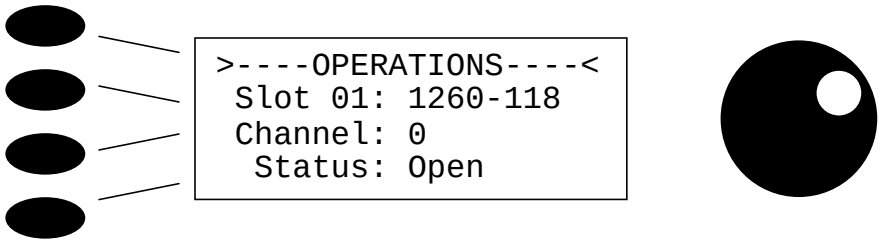


Note that the top line has changed, along with the related information on lines 2, 3, and 4.

3. Rotate the knob another click to the right. The display now shows:

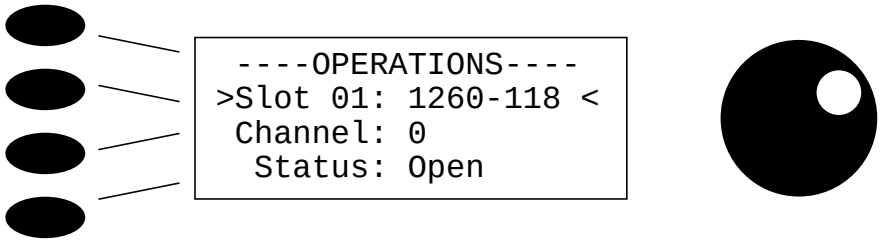


4. Rotate the knob one more click to the right. The display shows:



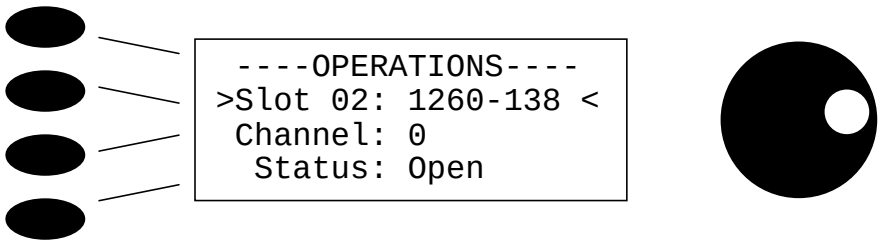
Now the OPERATIONS menu is active. This menu contains the functions for operating the relays.

5. Press the second keypad switch (corresponding to the second display line). This selects the second line, indicated by the selection symbols “>” and “<”:

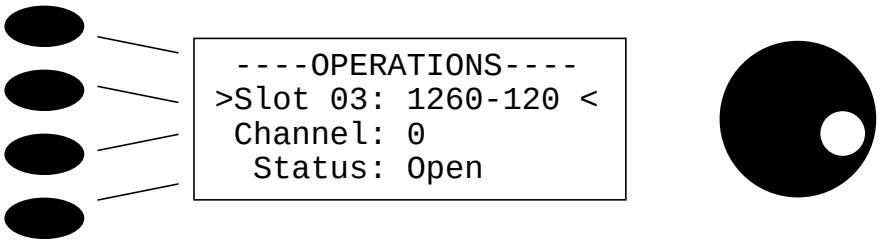


Line 2 also indicates that slot 1 contains a 1260-118 plug-in.

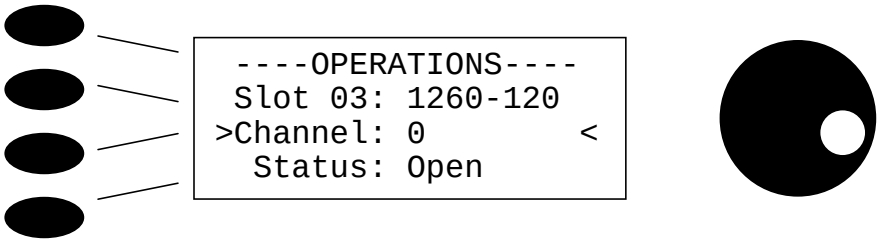
6. Rotate the knob one click to the right. Line 2 of the display now shows the contents of slot 2 (1260-138):



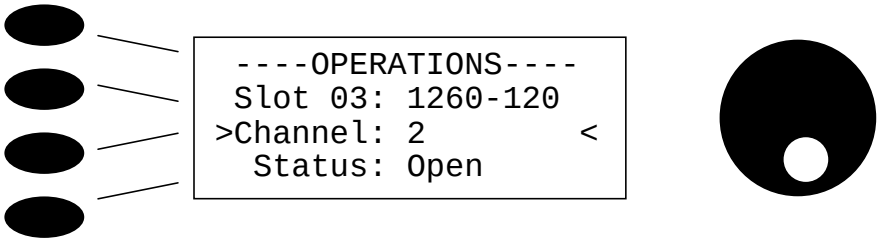
7. Rotate the knob one more click to the right. Line 2 now shows the contents of slot 3 (1260-120 plug-in):



8. Press the third keypad switch (corresponding to line 3 of the display). This selects line 3 so that the relay channel can be selected:

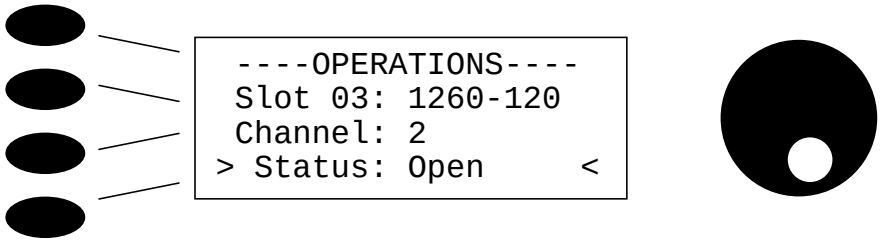


9. Rotate the knob to the right. As the knob rotates, the channel number indicated by line 3 increments upward. Stop when the channel number reaches 2:

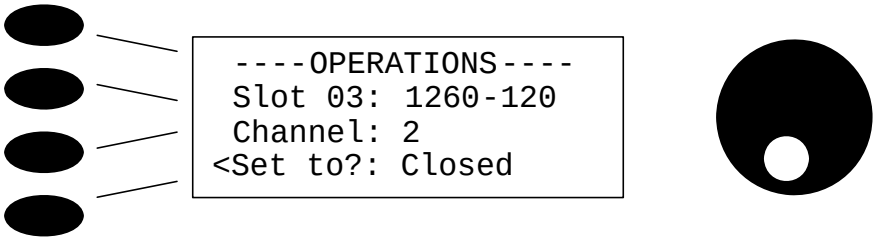


Line 4 shows the status of the channel 2 relay on the 1260-120 plug-in.

10. Press the fourth keypad switch (corresponding to line 4 of the display).

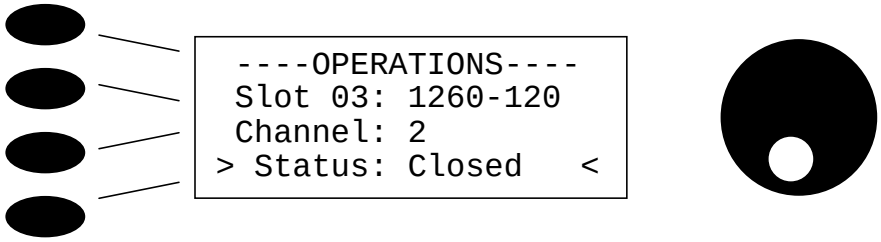


11. Rotate the knob one click to the right, to view the next choice for the relay status:



On the fourth line, the 1256 asks whether to close the relay channel:

12. Press the fourth keypad switch to confirm that the relay channel should be closed (press any other key to cancel). The display then shows the updated relay status:



In the above example, the user starts at the top of the menu hierarchy and proceeds downward through the following levels:

Level 1: Menu type: OPERATIONS, PATH CONTROL, SETTINGS, STORE / RECALL, SYSTEM (This example used OPERATIONS).

Level 2: Slot number: Selects the switch plug-in to control.

Level 3: Channel number: Selects the relay channel to control.

Level 4: Status: Shows whether the relay is open or closed, and lets the user change the status.

The above example illustrates that the user interface has a hierarchical menu structure. The top line shows the top level of the menu. The second, third, and fourth lines show successively lower levels of the hierarchy.

Some menus do not use all four lines of the display. When a line is unused, or contains nothing that the user can change, the LED in the corresponding key is not illuminated. Pressing such a key has no effect. However, when the LED in a key is illuminated, that key represents a line that can be edited.

This menu system is easy to use because the display shows all levels simultaneously. In addition, the user may instantly go to a higher level by simply pressing the corresponding key.

The Five Menus

At the top of the menu hierarchy, on the first display line, are five menus, or general types of functions. These menus, and the functions they include, are:

- **Operations:** Open and close relays on switching plug-ins, and set logic levels high or low on digital I/O plug-ins.
- **Path Control:** Open and close user-selected pathnames stored in non-volatile memory (paths must be defined and stored through the remote user-interface prior to accessing them through the display).
- **Settings:** Set menu preferences such as IEEE-488 address, RS-232 baud rate, and display brightness.

- **Store/Recall:** Store switch states and menu preferences to non-volatile (flash) RAM, and recall these states and preferences.

- **System:** Show firmware revisions for boot memory and flash memory. This facilitates efficient response when contacting Customer Service.

To work with one of these menus, first push the top key, corresponding to line 1 (top line) of the display. This selects the top line:

```
>----OPERATIONS----<
 Slot 01: 1260-121A
 Channel: 0
  Status: Open
```

Next, rotate the knob until the desired menu title is displayed on line 1.

The following sections discuss in detail how to use these menus.

Operations Menu

Description

The Operations Menu provides direct control of switching and digital I/O plug-ins. It allows the following operations:

- View the states of the relays (closed or open) on a switching plug-in.

- Close or open a relay on a switching plug-in.

- View the logic state of a digital input or output port on a digital I/O plug-in.

- Change the logic state of a digital output port on a digital I/O plug-in.

Closing a Relay

The 1256 front-panel enables the user to manually close and open relays on switch plug-ins. (For connector pin assignments of a plug-in, refer to its user manual.)

For example, suppose a 1260-118 plug-in is installed in slot 3. To close relay channel 4 on this plug-in, proceed as follows:

1. Press the top key next to the display. This selects the top display line so that you can change it. The selection markers ">" and "<" appear on the line to indicate that it is selected. Rotate the knob until the top line reads:

```
>----OPERATIONS----<
```

2. Select line 2 (next to the top line) by pressing the second key. This selects line 2 so that you may select the slot to work with:

```
 ----OPERATIONS----
>Slot 01: 1260-121A    <
 Channel: 0
  Status: Open
```

3. Rotate the knob until line 2 indicates the contents of slot 3:

```
 ----OPERATIONS----
>Slot 03: 1260-118     <
 Channel: 0
  Status: Open
```

4. Press the third key to select line 3 of the display (channel number):

```
 ----OPERATIONS----
 Slot 03: 1260-118
>Channel: 0        <
  Status: Open
```

5. Rotate the knob until line 3 shows the desired relay channel number, which in this example is 4:

```
 ----OPERATIONS----
 Slot 03: 1260-118
>Channel: 4        <
  Status: Open
```

6. Press the fourth key to select line 4. Rotate the knob until the display reads:

```
 ----OPERATIONS----
 Slot 03: 1260-118
 Channel: 4
<Set to?: Closed
```

7. Press the fourth key to confirm the closure of the relay channel. The relay then closes, and the display indicates the updated status on line 4:

```
 ----OPERATIONS----
 Slot 03: 1260-118
 Channel: 4
> Status: Closed   <
```

Opening a Relay

To open a relay, follow the steps given in the above section, **Closing a Relay**, except in step 6 rotate the knob until line 4 (bottom line) of the display reads as shown below:

```
  ----OPERATIONS----
  Slot 03: 1260-118
  Channel: 4
 <Set to?: Open
```

Then press the fourth key to open the relay.

Viewing Relay States

It is easy to view the states of the relay channels on a switch plug-in by using the OPERATIONS menu:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >----OPERATIONS----<
   ```

2. Select line 2 (next to the top line) by pressing the second key. This selects line 2:

   ```
    ----OPERATIONS----
   >Slot 01: 1260-121A<
    Channel: 0
     Status: Open
   ```

3. Rotate the knob until line 2 shows the slot number of the plug-in you wish to view. This example uses slot 3:

   ```
    ----OPERATIONS----
   >Slot 03: 1260-118 <
    Channel: 0
     Status: Open
   ```

4. Press the third key to select line 3 of the display (channel number). Line 4 of the display shows the state of the channel 0 relay.

   ```
    ----OPERATIONS----
    Slot 01: 1260-118
   >Channel: 0        <
     Status: Open
   ```

5. Rotate the knob one “click” to the right to view the state of the next relay channel (channel 1):

   ```
    ----OPERATIONS----
    Slot 01: 1260-118
   >Channel: 1        <
     Status: Closed
   ```

6. Rotate the knob right or left to scan through the channels as desired.

Controlling a Digital Output Port

The 1256 front-panel provides control of the output ports of a digital I/O plug-in. (For connector pin assignments of a plug-in, refer to its user manual.)

For example, suppose a 1260-114TTL plug-in is installed in slot 3. Also suppose that port 7 of this plug-in is configured as an output (see **Chapter 7, DIGITAL MODULE OPERATION** for configuration instructions). To change the output of port 4, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >----OPERATIONS----<
   ```

2. Select line 2 (next to the top line) by pressing second key. The display shows line 2 as being selected:

   ```
    ----OPERATIONS----
   >Slot 01: 1260-121A<
    Channel: 0
     Status: Open
   ```

3. Rotate the knob until line 2 shows the contents of slot 3:

   ```
    ----OPERATIONS----
   >Slot 03: -114TTL  <
    Port 00: Write
      Value: 255
   ```

4. Press the third key to select line 3 of the display (port number):

   ```
    ----OPERATIONS----
    Slot 03: -114TTL
   >Port 00: Write    <
      Value: 255
   ```

5. Rotate the knob until line 3 shows the desired port number, which in this example is 7, and the desired mode, which is Write:

   ```
    ----OPERATIONS----
    Slot 03: -114TTL
   >Port 07: Write    <
      Value: 255
   ```

   Line 4 now shows the current states of the eight bits of port 7, expressed as an eight-bit decimal number. By default, the display uses decimal notation. To change the notation to hexadecimal or binary, refer to the **Display Format** section in this chapter, under the **Settings** heading.

6. Press the fourth key to select line 4. Rotate the knob until line 4 shows the desired data to write to the port. This example uses 170, which equals AA_{16}, or an alternating 1/0

pattern:

```
 ----OPERATIONS----
 Slot 03: -114TTL
 Port 07: Write
<Set to : 170?
```

7. Press the fourth key to confirm the change. The port output then changes, and the display shows:

```
 ----OPERATIONS----
 Slot 03: -114TTL
 Port 07: Write
>  Value: 170      <
```

Viewing a Digital Input Port

The 1256 front-panel controls allow the user to view the data from input ports of digital I/O plug-ins. For example, suppose a 1260-114TTL plug-in is installed in slot 3. Also suppose that port 5 of this plug-in is configured as an input (see **Chapter 7, DIGITAL MODULE OPERATION** for configuration instructions). To view the status of port 5, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

```
>----OPERATIONS----<
```

2. Select line 2 (next to the top line) by pressing the second key. The display shows line 2 as being selected:

```
 ----OPERATIONS----
>Slot 01: 1260-121A<
 Channel: 00
  Status: Open
```

3. Rotate the knob until line 2 shows the contents of slot 3:

```
 ----OPERATIONS----
>Slot 03: -114TTL  <
 Port 00: Write
   Value: 255
```

4. Press the third key to select line 3 of the display (port number):

```
 ----OPERATIONS----
 Slot 03: -114TTL
>Port 00: Write    <
   Value: 255
```

5. Rotate the knob until line 3 shows the desired port number, which in this example is 5, and the desired mode, which is Read:

```
 ----OPERATIONS----
```

```
 Slot 03: -114TTL
>Port 05: Read     <
   Value: 43
```

Now line 4 shows the states of the eight input signals of port 5, expressed as an eight-bit decimal number. By default, the display uses decimal notation. To change the notation to hexadecimal or binary, refer to the **Display Format** section of this chapter, under the **Settings** heading.

6. If desired, rotate the knob in either direction to view the states of other digital ports. Ports programmed as inputs reflect the states of their input signals. Ports configured as outputs show the most recent data written to them.

Enabling a Digital Port

Before using a digital port as an output, its output buffers must be enabled. For example, suppose a 1260-114TTL plug-in is installed in slot 3. To enable output port 4, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >----OPERATIONS----<
   ```

2. Select line 2 (next to the top line) by pressing the second key. The display shows line 2 as being selected:

   ```
    ----OPERATIONS----
   >Slot 01: 1260-121A<
    Channel: 00
     Status: Open
   ```

3. Rotate the knob until line 2 shows the contents of slot 3:

   ```
    ----OPERATIONS----
   >Slot 03: -114TTL  <
    Port 00: Write
      Value: 255
   ```

4. Press the third key to select line 3 of the display (port number):

   ```
    ----OPERATIONS----
    Slot 03: -114TTL
   >Port 00: Write    <
      Value: 255
   ```

5. Rotate the knob until line 3 shows the desired port number, which in this example is 4, and the desired mode, which is Enable:

   ```
    ----OPERATIONS----
    Slot 03: -114TTL
   >Port 04: Enable   <
   ```

```
  Status: Disabled
```

Now line 4 shows the enable status of the eight-bit buffer for port 5 (Enabled or Disabled). In this case, it is disabled.

6. Press the bottom key to select the bottom line of the display. Then rotate the knob until the display reads as follows:

```
 ----OPERATIONS----
 Slot 03: -114TTL
>Port 04: Enable   <
<Set to?: Enabled
```

7. Press the bottom key to complete the operation.

Path Control Menu

Description

The Path Control Menu automatically retrieves pathnames stored in non-volatile memory at startup and provides the user with direct control of pathnames from the 1256 front-panel display. It allows the following operations:

- View the states of pathnames (closed or open) in the system.
- Set pathname filter masks to wildcards or to specific characters to limit the display of pathnames to those meeting particular user-criteria.

Defining a Path

The 1256 front-panel enables the user to manually close or open pathnames after they have been defined remotely using the SCPI path definition command through one of the remote interfaces. Refer to "**Naming a Path**" section in Chapter 5 for more details on defining path names.

Once a path is defined through a remote interface, it is immediately available for display and use on the front-panel and will remain visible until either 1) the power is removed from the system or 2) the path is removed remotely using the SCPI path delete command. If the intent is to define the pathnames and make them accessible to the front panel after power is cycled to the 1256, the defined pathnames must also be stored in non-volatile memory. Refer to "**Storing Path Names in Non-Volatile Memory**" section in Chapter 5 for more details.

It is important to realize that the SCPI command language allows pathname lengths that exceed the display capability of the 1256 (18 characters). If pathnames are defined greater than 18 characters, only the first 18 characters will be visible. It is also only possible to selectively filter the first 18 characters using the

display filter masks. Characters extending beyond this limit are treated with wildcards when the filter mask is applied.

Closing a Path

The 1256 front-panel enables the user to manually close and open pathnames.

For example, suppose four pathnames have been defined using the remote interface for a combination of different relays and plug-in cards – “PATH1”, “PATH2”, “PATH3”, and “PATH4”:

1. Press the top key next to the display. This selects the top display line so that you can change it. The selection markers “>” and “<” appear on the line to indicate that it is selected. Rotate the knob until the top line reads:

   ```
   >---PATH CONTROL---<
   ```

2. Select line 3 (next to the top line) by pressing the third key. This allows the pathname to be selected from a list of those that are presently defined in the 1256 switching system:

   ```
    ---PATH CONTROL---
    Select Filter Mask
   >PATH1_____________<
     Status: Open
   ```

3. Rotate the knob until line 3 indicates the desired pathname (“PATH2” for this example):

   ```
   ---PATH CONTROL---
    Select Filter Mask
   >PATH2_____________<
     Status: Open
   ```

4. Press the fourth key to select line 4. Rotate the knob until the display reads:

   ```
   ---PATH CONTROL---
    Select Filter Mask
    PATH2_____________
   <Set to?: Closed
   ```

5. Press the fourth key to confirm the closure of the desired pathname. The entire path then closes, and the display indicates the updated status on line 4:

   ```
   ---PATH CONTROL---
    Select Filter Mask
    PATH2_____________
   ```

```
> Status: Closed   <
```

Opening a Path

To open a relay, follow the steps given in the above section, **Closing a Relay**, except in step 4 rotate the knob until line 4 (bottom line) of the display reads as shown below:

```
---PATH CONTROL---
Select Filter Mask
PATH2_____________
<Set to?: Open
```

Then press the fourth key to open the relay.

Viewing Path States

It is easy to view the states of pathnames on a 1256 system from the PATH CONTROL menu:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >---PATH CONTROL---<
   ```

2. Select line 3 by pressing the third key:

   ```
   ---PATH CONTROL---
    Select Filter Mask
   >PATH1_____________<
     Status: Open
   ```

3. Rotate the knob until line 3 indicates the desired pathname (“PATH2” for this example):

   ```
   ---PATH CONTROL---
    Select Filter Mask
   >PATH2_____________<
     Status: Open
   ```

4. Rotate the knob right or left to scan through the pathnames as desired. Each pathname when displayed will indicate whether it is open or closed on the fourth line of the display.

Setting Filter Masks

Since a large number of path permutations may exist for a given selection of cards, it may take time to scroll through all the possibilities if a large number are defined in the 1256 system. To minimize scrolling time on the display, filter masks can be easily set from the front panel to limit the pathnames displayed to only those that meet user-specified criteria.

Setting filter selection masks is performed easily as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

```
>---PATH CONTROL---<
```

2. Select line 2 (next to the top line) by pressing second key. The display shows line 2 as being selected and will show a flashing cursor on the first position of the mask (wildcards are the default and appear as an asterisk “*”):

```
 ---PATH CONTROL---
>******************<
   Select Paths(s)
```

3. Move the flashing cursor to the desired character position in the filter mask by pressing the line 2 key as many times as required to advance the cursor to the desired mask position.

4. Once in the desired position, rotate the knob left or right to select the appropriate filter character or wildcard character.

```
---PATH CONTROL---
>P***3*************<
   Select Paths(s)
<Reset Mask
```

5. When the mask is fully defined, pressing the line 3 key “Select Path(s)” option will apply the mask condition to defined paths. If the system has paths meeting the mask criteria, they can be displayed, selected, and controlled in the manner described earlier in this section:

```
---PATH CONTROL---
 Select Filter Mask
>PATH3_____________<
  Status: Open
```

If no paths exist that meet the filter criteria, the 1256 will display:

```
>---PATH CONTROL---<
 Select Filter Mask
  {no paths found}
```

While selecting a mask, pressing the line 1 key or line 4 key will institute other changes. If a mask is selected that is not all wildcards, the display will give the option to reset the mask:

```
---PATH CONTROL---
>P***0*************<
   Select Paths(s)
<Reset Mask
```

By pressing the line 4 key, all mask characters are reset to

wildcards "*" and the cursor position is returned to the first character.

Pressing the line 1 key will effectively abandon any changes made to the mask and revert back to the prior mask.

Settings Menu

Description

The Settings Menu provides control of the following options:

- Remote interface source: GPIB or RS-232.
- Local lockout during remote operations: enabled or disabled.
- RS-232 parameters such as baud rate and parity.
- GPIB address and SRQ settings.
- Display format: hexadecimal, decimal, or binary.
- Display intensity (brightness).
- Display screen saver: enabled or disabled.
- Screen saver timeout, in minutes.

Selecting the Remote Interface Source

The user may control the 1256 remotely by either of two remote interfaces: GPIB or RS-232. To select one of these as the active interface, follow these steps:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until line 2 of the display reads "`Type: Remote`", as follows:

   ```
    -----SETTINGS-----
   >   Type: Remote   <
     Option: Lockout
     Status: Disabled
   ```

3. Select line 3 by pressing the third key. Rotate the knob until line 3 reads as shown below:

```
 -----SETTINGS-----
    Type: Remote
> Option: Source   <
  Status: RS-232
```

4. Line 4 now indicates the remote interface source that is active. The two choices are GPIB and RS-232. If the choice shown is not the desired one, then rotate the knob until the correct choice is displayed:

```
 -----SETTINGS-----
    Type: Remote
  Option: Source
<Update?: GPIB
```

Then update to this choice by pressing the fourth key. The display now reads:

```
 -----SETTINGS-----
    Type: Remote
  Option: Source
> Status: GPIB     <
```

To set the GPIB parameters, such as the address, refer to **Chapter 4, GPIB AND RS-232 INTERFACES**.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Local Lockout

In some applications, it may be desirable to lock out (disable) the front-panel controls while controlling the 1256 via a remote interface. The 1256 provides two types of local lockout:

- **Automatic lockout:** When the 1256 receives any command via the RS-232 or GPIB interface, it shows a message on the front-panel display, stating that the unit is in remote operation, and which interface (GPIB or RS-232) is active. For example, after receiving a command via the RS-232 interface, the display reads:

```
 *REMOTE OPERATION*

 (RS-232 is active)
<Local Mode
```

At this time, the knob and all keys are disabled, and the 1256 receives commands from the remote interface

only. The user may restore local control only by issuing a KLOCK OFF command. The display then returns to the state that it was in when the 1256 entered lockout mode. Manual control is also restored if the user cycles the power to the 1256.

- **Manual lockout:** When the 1256 receives the command SYST:KLOCK ON (local lockout) via the RS-232 or GPIB interface, it displays the following message:

```
*REMOTE OPERATION*

(RS-232 is active)
Local Mode Lockout
```

In this case, all front-panel controls are disabled. The only way to restore local (front-panel) control is to send the command SYST:KLOCK OFF via the remote (RS-232 or GPIB) interface. For details on sending the commands SYST:KLOCK ON and SYST:KLOCK OFF, refer to **Chapter 4: GPIB AND RS-232 INTERFACES**.

The automatic lockout may be enabled or disabled via the front-panel controls (if local mode is currently active). When lockout is enabled, the automatic lockout mode is activated when the 1256 receives any command via a remote interface. When lockout is disabled, the front-panel controls remain active even after the 1256 receives a command (other than SYST:KLOCK ON) via the remote interface. However, the secure lockout mode becomes active if the 1256 receives a SYST:KLOCK ON command, regardless of whether the automatic lockout is enabled or disabled.

To enable or disable the automatic local lockout, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until the display reads:

   ```
    -----SETTINGS-----
   >   Type: Remote   <
     Option: Source
     Status: GPIB
   ```

3. Select line 3 by pressing the third key. Rotate the knob until line 3 reads as shown below:

   ```
    -----SETTINGS-----
       Type: Remote
   ```

```
> Option: Lockout  <
  Status: Disabled
```

4. If line 4 already shows the desired status, then no further action is necessary. Otherwise, select line 4. Then rotate the knob until the desired state (“Enabled” or “Disabled”) shows on line 4. This example uses “Enabled”:

```
 -----SETTINGS-----
    Type: Remote
  Option: Lockout
<Update?: Enabled
```

   Then press the fourth key (bottom) to update the status. Pressing any other key cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting RS-232 Parameters

For detailed instructions on setting RS-232 parameters, such as baud rate and parity, refer to **Chapter 4, GPIB AND RS-232 INTERFACES**.

Setting GPIB Address and SRQ Settings

For detailed instructions on setting the GPIB address and SRQ settings, refer to **Chapter 4, GPIB AND RS-232 INTERFACES**.

Setting the Display Format

The display may be set to display digital port data in hexadecimal, decimal, or binary format. To set the display format, proceed as follows:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to top line) by pressing the second key. Rotate the knob until line 2 of the display reads

"Type: Display:"

```
 -----SETTINGS-----
>   Type: Display  <
  Option: Intensity
  Status: 75%
```

3. Press the third key to select line 3 of the display (option): Rotate the knob until line 3 of the display reads as follows:

```
 -----SETTINGS-----
    Type: Display
> Option: Format   <
  Status: Hex
```

4. If line 4 already shows the desired format, then no further action is necessary. Otherwise, select line 4. Then rotate the knob until line 4 shows the desired format. This example uses "Binary":

```
 -----SETTINGS-----
    Type: Display
  Option: Format
<Update?: Binary
```

Then press the fourth key (bottom) to update the status. Pressing any other key cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting the Display Intensity

The display intensity (brightness) is adjustable to accommodate different ambient lighting conditions. It may be set to 25%, 50%, 75%, or 100% of its maximum brightness. To set the display intensity, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

```
>-----SETTINGS-----<
```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until line 2 of the display reads "Type: Display":

```
 -----SETTINGS-----
>   Type: Display  <
```

```
  Option: Format
  Status: Hex
```

3. Press the third key to select line 3 of the display (option). Then rotate the knob until line 3 of the display reads as follows:

```
 -----SETTINGS-----
    Type: Display
> Option: Intensity<
  Status: 75%
```

4. If line 4 already shows the desired intensity, then no further action is necessary. Otherwise, select line 4. Then rotate the knob until line 4 shows the desired intensity. This example uses “50%”:

```
 -----SETTINGS-----
    Type: Display
  Option: Intensity
<Update?: 50%
```

   Then press the fourth key (bottom) to update the status. Pressing any other key cancels the change.

Enabling or Disabling the Screen Saver

The purpose of the screen saver is to maximize the life of the fluorescent display. The screen saver automatically turns the display off after no controls have been operated for a certain time interval. To set this time interval, refer to **Setting the Screen Saver Timeout**, later in this section.

To enable or disable the screen saver, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

```
>-----SETTINGS-----<
```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until line 2 of the display reads “`Type: Display`“:

```
 -----SETTINGS-----
>   Type: Display  <
  Option: Format
  Status: Hex
```

3. Press the third key to select line 3 of the display (option): Rotate the knob until line 3 of the display reads as follows:

```
 -----SETTINGS-----
    Type: Display
> Option: Saver    <
  Status: Enabled
```

4. If line 4 already shows the desired status, then no further action is necessary. Otherwise, select line 4. Then rotate the knob until line 4 shows the desired status (enabled or disabled). This example uses “Disabled”:

```
 -----SETTINGS-----
    Type: Display
  Option: Saver
<Update?: Disabled
```

   Then press the fourth key (bottom) to update. Pressing any other key cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting the Screen Saver Timeout

When the screen saver is enabled, it blanks the screen after no controls have been operated for a certain period of time. The purpose is to maximize display life. The time interval is adjustable from one minute to one hour. To set the time interval, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

```
>-----SETTINGS-----<
```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until line 2 of the display reads “`Type: Display`“:

```
 -----SETTINGS-----
>   Type: Display  <
  Option: Format
  Status: Hex
```

3. Press the third key to select line 3 of the display (option): Rotate the knob until line 3 of the display reads:

```
 -----SETTINGS-----
    Type: Display
> Option: Timeout  <
  Status: 1 min
```

4. If line 4 already shows the desired timeout value, then no further action is necessary. Otherwise, select line 4. Then rotate the knob until line 4 shows the desired value. This example uses “15 min”:

```
 -----SETTINGS-----
    Type: Display
  Option: Timeout
<Update?: 15 Min
```

   Press the fourth key (bottom) to update the timeout value. Pressing any other key cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Store/Recall Menu

Description

After the user has set up the 1256 Switching System, the setup may be stored in non-volatile (flash) RAM. Later, the user may recall this setup to instantly restore the 1256 to the same state it was in earlier. The STORE/RECALL menu contains the functions used for doing this.

The non-volatile RAM can store two types of setup information: switching states and menu preferences.

Storing the Switching State

The switching state includes the positions (closed or open) of all relays.

The 1256 stores the states of all plug-ins together as a single state. When a switching state is recalled, it automatically restores the states of all relays on all installed plug-ins.

The 1256 can store up to 100 switching states, or presets, plus an additional state that is restored automatically when the power is turned on.

NOTE:

After storing the switch state, the non-volatile image must be written to non-volatile RAM. Refer to the section Updating Non-Volatile RAM for instructions.

Storing the Menu Preferences

Menu preferences include all preferences set by the user, such as remote interface settings, display intensity, and display format. The 1256 stores all menu preferences together as a single state.

The 1256 can store one menu preference state, which is automatically restored when the power is turned on.

NOTE:

After storing the switch state, the non-volatile image must be written to non-volatile RAM. Refer to the section Updating Non-Volatile RAM for instructions.

Storing the Power-Up Switching State

To store the current switching state as the power-up state, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >---STORE/RECALL---<
   ```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until line 2 of the display reads as follows:

   ```
    ---STORE/RECALL---
   >   Type: State    <
     Config: Defaults
     Action: <select>
   ```

3. Press the third key to select line 3 of the display (option). Rotate the knob until line 3 reads:

   ```
    ---STORE/RECALL---
       Type: State
   > Config: Powerup  <
     Action: <select>
   ```

4. Press the fourth key to select line 4 of the display. Rotate the knob until line 4 reads:

   ```
    ---STORE/RECALL---
       Type: State
   > Config: Powerup  <
   <    Do?: Store
   ```

Press the fourth key (bottom) to store the power-up state. Pressing any other key cancels the change.

NOTE:

After storing the switch state, the non-volatile image must be written to non-volatile RAM. Refer to the section Updating Non-Volatile RAM for instructions.

Updating Non-Volatile RAM

Non-volatile (flash) RAM is inherently slow. If presets were stored directly in non-volatile RAM, the user would have to wait several seconds for each store operation. This may not always be an inconvenience when working with the front-panel controls, but would certainly slow down software that automatically sets up and stores many presets.

To avoid the delay after storing each preset, the presets are written to a temporary area of the static random-access memory (SRAM). This memory is volatile (will erase upon power-down), but is very fast. This allows presets to be stored rapidly. Once the last desired preset has been stored, the presets stored in SRAM (known as the SRAM image) may be written to non-volatile RAM all at once. Since the non-volatile RAM is block-oriented, it takes the same amount of time to write many presets as to write just one. Thus, by writing all of the desired presets to non-volatile RAM just once, much time is saved.

To update the non-volatile RAM with the SRAM image, follow these steps:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >---STORE/RECALL---<
   ```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until line 2 of the display reads as follows:

   ```
    ---STORE/RECALL---
   >   Type: NVRAM    <
     Config: Image
     Status: Erase
   ```

3. Press the fourth key to select line 4 of the display. Rotate the knob until line 4 reads:

   ```
    ---STORE/RECALL---
       Type: NVRAM
     Option: Image
   <    Do?: Update
   ```

4. Press the fourth key (bottom) to store the power-up state. Pressing any other key cancels the change.

Erasing Non-Volatile RAM

The entire non-volatile RAM may be erased if desired. This erases all presets, including switching states, the power-up switching state, and the menu preferences. The menu preferences will be set to the factory defaults.

To erase the entire non-volatile RAM, proceed as follows:

1. Press the top key next to the display. Rotate the knob until the top line reads:

   ```
   >---STORE/RECALL---<
   ```

2. Select line 2 (next to the top line) by pressing the second key. Rotate the knob until line 2 of the display reads as follows:

   ```
    ---STORE/RECALL---
   >   Type: NVRAM    <
     Config: Image
     Action: <select>
   ```

3. Press the fourth key to select line 4 of the display. If line 4 does not read `Erase All`, then rotate the knob until it does:

   ```
    ---STORE/RECALL---
      Type: NVRAM
     Config: Image
   <    Do?: Erase All
   ```

4. Press the fourth key (bottom) to erase the non-volatile RAM. Pressing any other key cancels the operation and leaves the memory intact.

System Menu

Description

The SYSTEM menu allows viewing of system information. This consists of the firmware revision numbers of the boot ROM and flash program. This information may be valuable when contacting customer service to resolve a problem.

Viewing Firmware Revision Numbers

To view the revision letters of the firmware, follow these steps:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >------SYSTEM------<
   ```

2. The display will appear similar to the following:

   ```
   >------SYSTEM------<
    Firmware Revisions
      Boot : 1.1
     Flash : 1.1
   ```

The third and fourth lines indicate the revision numbers of the boot ROM and the operating program that is currently stored in flash memory.

Complete Menu Listing

For quick reference, this section lists the complete menu hierarchy for the 1256 Switching System. Menu items are indented according to their levels of hierarchy. For example:

LEVEL 1 (top line of display)
    LEVEL 2 (second line of display)
        LEVEL 3 (third line of display)
            LEVEL 4 (fourth line of display)

The complete menu listing is as follows:

OPERATIONS
    Plug-In (slot number and model number) to control
        State of channel (relay or digital I/O)
            Close or Open (relay)
            or...
            High or Low (digital I/O)

PATH CONTROL
    Filter Mask
        Pathname
            Close or Open (relay)
            or...
            High or Low (digital I/O)

- SETTINGS
  - Remote
    - Source
      - GPIB
      - RS-232
    - Lockout
      - Disabled
      - Enabled
  - RS-232
    - Baud Rate
      - 1,200
      - 2,400
      - 4,800
      - 9,600
      - 19,200
      - 38,400
      - 57,600
      - 115,200
    - Data Bits
      - 8
      - 7
    - Parity
      - None
      - Even
      - Odd
    - Stop Bits
      - 1
      - 2
    - Flow Control
      - Disabled
      - XON/XOFF
      - RTS/CTS
    - EOL Mode
      - LF Only
      - CR/LF
      - CR Only
  - GPIB
    - Address
      - 0
      - 1
      - 2
      - ...
      - 30
    - PwrOnSRQ
      - Enabled
      - Disabled
  - Display
    - Format
      - Hex
      - Binary
      - Decimal

- Intensity
  - 25%
  - 50%
  - 75%
  - 100%
- Screen Saver
  - Enabled
  - Disabled
- Timeout
  - 1 min
  - 5 mins
  - 10 mins
  - 15 mins
  - 30 mins
  - 45 mins
  - 1 hr

STORE/RECALL

- State
  - Defaults
    - Recall
  - Power Up
    - Store
    - Recall
  - Preset #1 to 100
    - Store
    - Recall
- Non-Volatile RAM
  - Image
    - Erase All
    - Update
- Menu
  - Defaults
    - Recall
  - Power Up
    - Store
    - Recall

SYSTEM

- Firmware Revisions
  - Boot
  - Flash

This page was left intentionally blank.

Chapter 4

GPIB and RS-232 INTERFACES

Introduction

The 1256 Switching System provides two remote interfaces that may be used for controlling the 1256:

- **IEEE-488.** Also known as the General-Purpose Interface Bus (GPIB), commonly used in test and measurement applications.
- **RS-232.** A serial bus used by modems and data terminals.

The remote interface allows access to the operating features of the 1256 by sending commands manually or automatically via a computer program.

Commands

The 1256 remote interfaces comply with the Standard Commands for Programmable Instruments (SCPI), a universal programming language for test and measurement instruments.

Chapter 5, SCPI COMMAND BASICS introduces SCPI and discusses the commands most frequently used with the 1256. For a discussion of all commands that may be used via the 1256 remote interfaces, see **Chapter 6, SCPI COMMAND REFERENCE.**

Using the IEEE-488 (GPIB) Interface

Basic GPIB Information

The GPIB standard permits up to 15 devices (including the controller) to interconnect and reside on a single bus. GPIB cables have identical stackable connectors at each end, allowing several cables to connect to a single source. System devices and instruments may be interconnected in almost any desired configuration. There must be a path from the controller to every device residing on the bus.

GPIB cables are available in two lengths:

- GPIB Cable, 1 meter (3.28 ft.) P/N 500310-001
- GPIB Cable, 2 meters (6.56 ft.) P/N 500310-002

To use the IEEE-488 (GPIB) interface with the 1256 Switching System, you also need a computer equipped with a GPIB controller.

GPIB Cabling Guidelines

Adhere to the following guidelines to maximize the data transfer rate of the GPIB system:

- Total cable length of no more than 20 meters (65.6 feet).
- Maximum separation of 4m (13.2 feet) between devices. Average separation of no more than 2 meters (6.6 feet).
- Maximum of 15 devices (including controller) connected to the bus.

Connecting the GPIB Cables

Use a GPIB cable to connect the rear-panel GPIB connector on the 1256 to the GPIB port of the host computer. Refer to **Figure 4-1, Using the GPIB Interface**.

CAUTION:

Avoid stacking more than three cables on any single connector. If the stack is too large, side forces may damage the connector mounting.

CAUTION:

Tighten the locking screws finger-tight on each connector. Do not over-tighten.

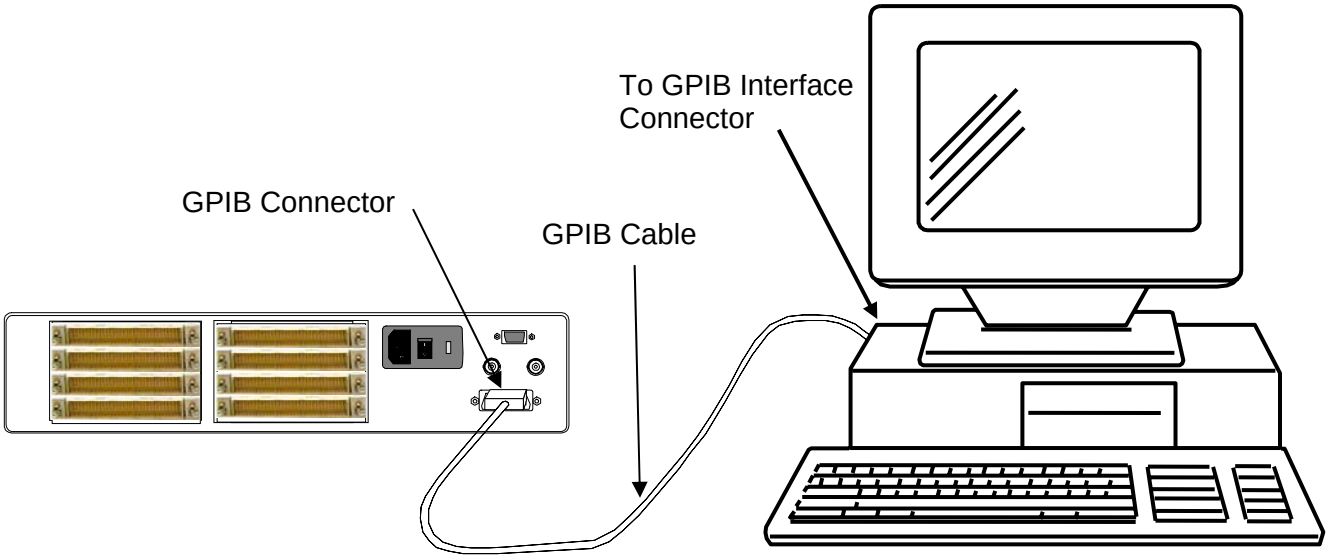


Figure 4-1, Using the GPIB Interface

Selecting the GPIB Interface

Before using the GPIB interface with the 1256, you must select the GPIB interface via the front-panel controls, as follows:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to top line) by pressing the second key. This selects line 2 so that you may select the type of setting to work with:

   ```
    -----SETTINGS-----
   >   Type: Display  <
     Option: Format
     Status: Decimal
   ```

3. Select the remote interface settings mode by rotating the knob until line 2 reads as shown below (lines 3 and 4 may differ):

   ```
    -----SETTINGS-----
   >   Type: Remote   <
     Option: Lockout
     Status: Disabled
   ```

4. Press the third key to select line 3 of the display. Rotate the knob until line 3 indicates the “Source” option as shown below:

   ```
    -----SETTINGS-----
       Type: Remote
   > Option: Source   <
     Status: GPIB
   ```

5. If line 4 of the display already indicates the status as “GPIB”, then the GPIB interface is already active; no further action is necessary. Otherwise, press the fourth key. Rotate the knob until line 4 indicates “GPIB”:

```
 -----SETTINGS-----
    Type: Remote
  Option: Source
<Update?: GPIB
```

   Then press the fourth key to confirm the update. Pressing any other button cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting the GPIB Address

The 1256, as a GPIB bus member, must be assigned a unique address. This address can be set only by means of the front-panel controls. Set the address to a number from 0 to 31, as follows:

1. Press the top key. Rotate the knob until the top line reads:

```
>-----SETTINGS-----<
```

2. Select line 2 (next to top line) by pressing the second key. This selects line 2:

```
 -----SETTINGS-----
>   Type: Display  <
  Option: Format
  Status: Decimal
```

3. Select the GPIB settings mode by rotating the knob until line 2 reads as shown below (lines 3 and 4 may differ):

```
 -----SETTINGS-----
>   Type: GPIB     <
  Option: Lockout
  Status: Disabled
```

4. Press the third key to select line 3 of the display. Rotate the knob until line 3 reads as shown below:

```
 -----SETTINGS-----
    Type: GPIB
> Option: Address  <
  Status: 14
```

5. If line 4 of the display already indicates the desired address

number, then no further action is necessary. Otherwise, press the fourth key. Rotate the knob until line 4 indicates the desired address. In this example, 15 is the desired address:

```
 -----SETTINGS-----
    Type: GPIB
  Option: Address
<Update?: 15
```

Then press the fourth key to confirm the update. Pressing any other button cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Service Request (SRQ)

The 1256 may generate a service request (SRQ) in response to a variety of events. These events include detecting a programming error or confidence mode failure, generating a response to a query, and many others.

Consult the **IEEE-488.2 Status Description** section in **Chapter 5, SCPI Command BASICS** for a description of how to program the 1256 to generate a service request.

Using the RS-232 Interface

Basic RS-232 Information

The RS-232 interface provides a simple way to control the 1256 from virtually any computer. Nearly every computer has a serial (Com) port. The only extra hardware required is a serial cable.

This section gives information about connecting the 1256 to the computer via serial cable, setting up the 1256 for serial port operation, and communicating with the 1256 by using a terminal emulation program.

Equipment Required

To use the RS-232 interface, you need an RS-232 data terminal or a computer equipped with an RS-232 serial port, such as the COM port of a personal computer. In addition, a shielded cable is required for compliance with the radiated emissions standards specified for the 1256. A shielded RS-232 cable, suitable for connecting the 1256 to a computer, is available from EADS North America Test and Services:

RS-232 Cable, 10 feet (3 meters) P/N 602248

DTE Versus DCE

All equipment with an RS-232 interface must be designed as either Data Communications Equipment (DCE) or Data Terminal Equipment (DTE), depending upon the end of the communications link from which it will operate. Computers are usually DTE, and modems are DCE. This allows a “straight-through” connection, where connector pins 1, 2, 3, ... at one end connect to pins 1, 2, 3, ... , respectively, at the other end.

The 1256 is designed as DCE to facilitate direct connection to a computer or data terminal (DTE). Connecting to a modem requires a “null modem”, described in the section **Connecting to a Modem**.

Connecting to a Computer or Data Terminal

Electrical Connection

Using a shielded RS-232 cable, connect the nine-pin female RS-232 connector on the 1256 rear panel to the RS-232 serial port on the computer or data terminal. Refer to **Figure 4-2, Using the RS-232 Interface**. A suitable cable is available from EADS North America Test and Services:

- RS-232 cable, 10 ft. (3.05 meter) P/N 602248

This cable has a DB-9 connector at each end. The male connector connects to the 1256 and the female connector connects to the data terminal or computer.

If you wish to make your own cable, **Figure 4-3a** provides the details.

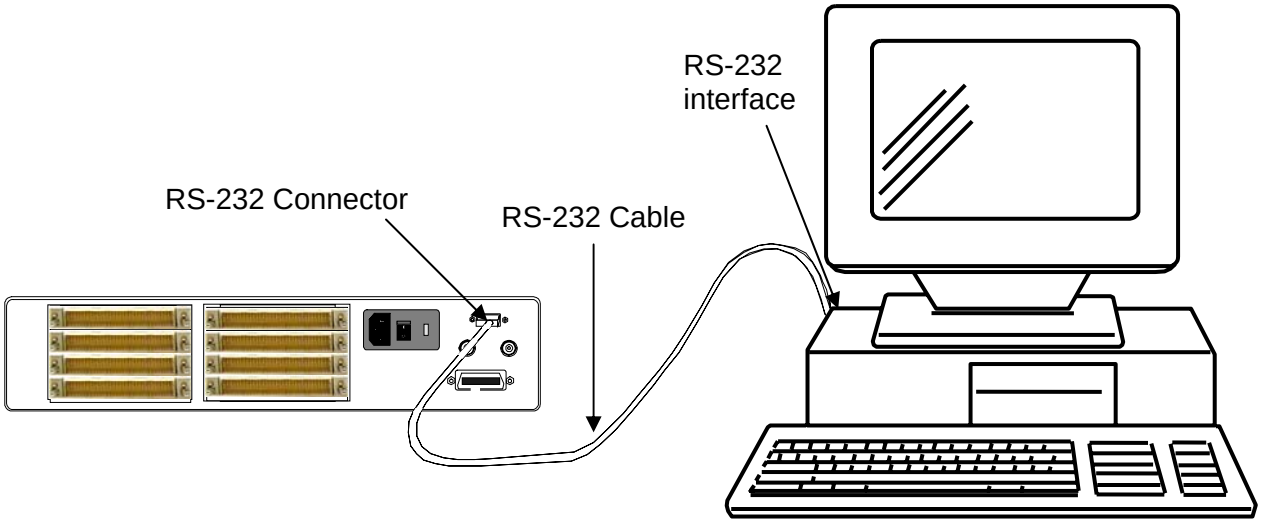


Figure 4-2, Using the RS-232 Interface

Connecting to a Modem

If you wish to connect the 1256 directly to a modem instead of a computer or data terminal, the connection will be DCE-to-DCE. A DCE-to-DCE connection requires a special serial cable, known as a null-modem. In a null-modem cable, the Received Data (RxD) and Transmitted Data (TxD) signals are reversed at one connector, and the Clear-to-Send (CTS) and Request-to-Send (RTS) signals are also reversed at one connector. This can be accomplished by constructing a special nine-pin serial cable with the signals wired as shown in **Figure 4-3b, Using the 1256 with a Modem**. Note that the RS-232 connector at the 1256 is a 9-pin female D-Subminiature (DB-9) type.

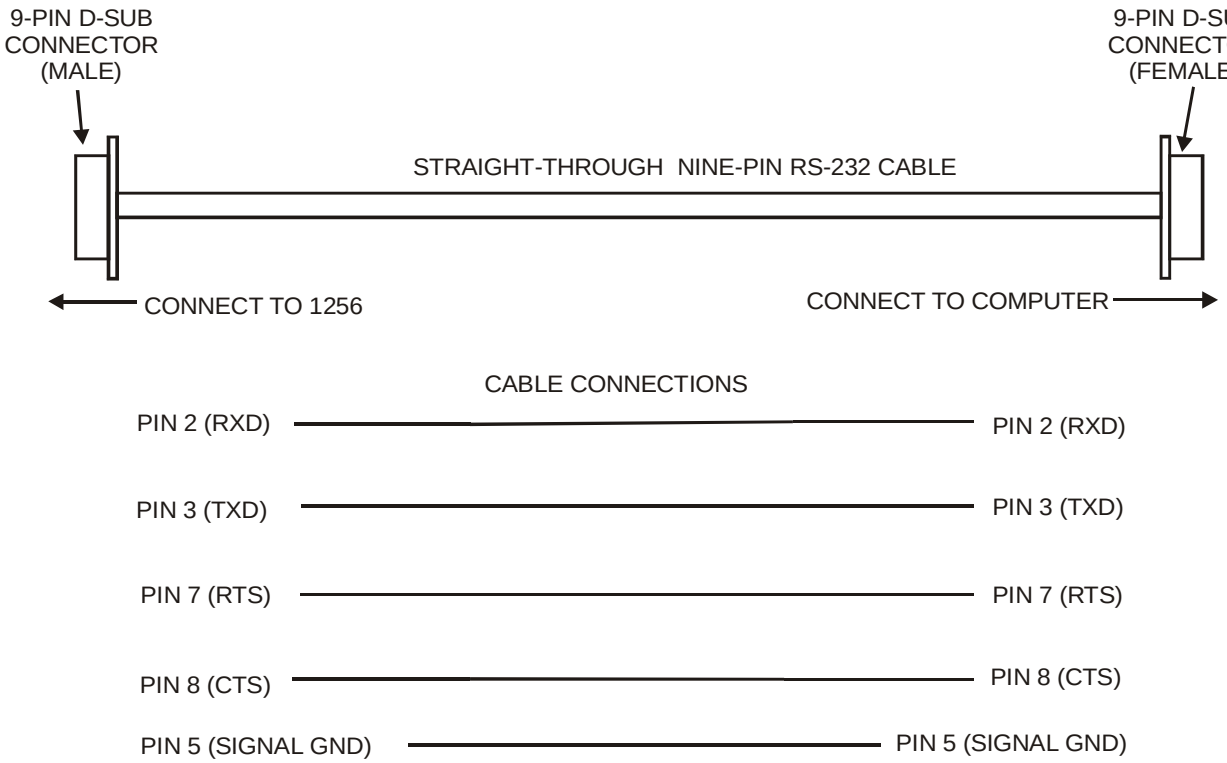


Figure 4-3a, Using the 1256 with a Computer

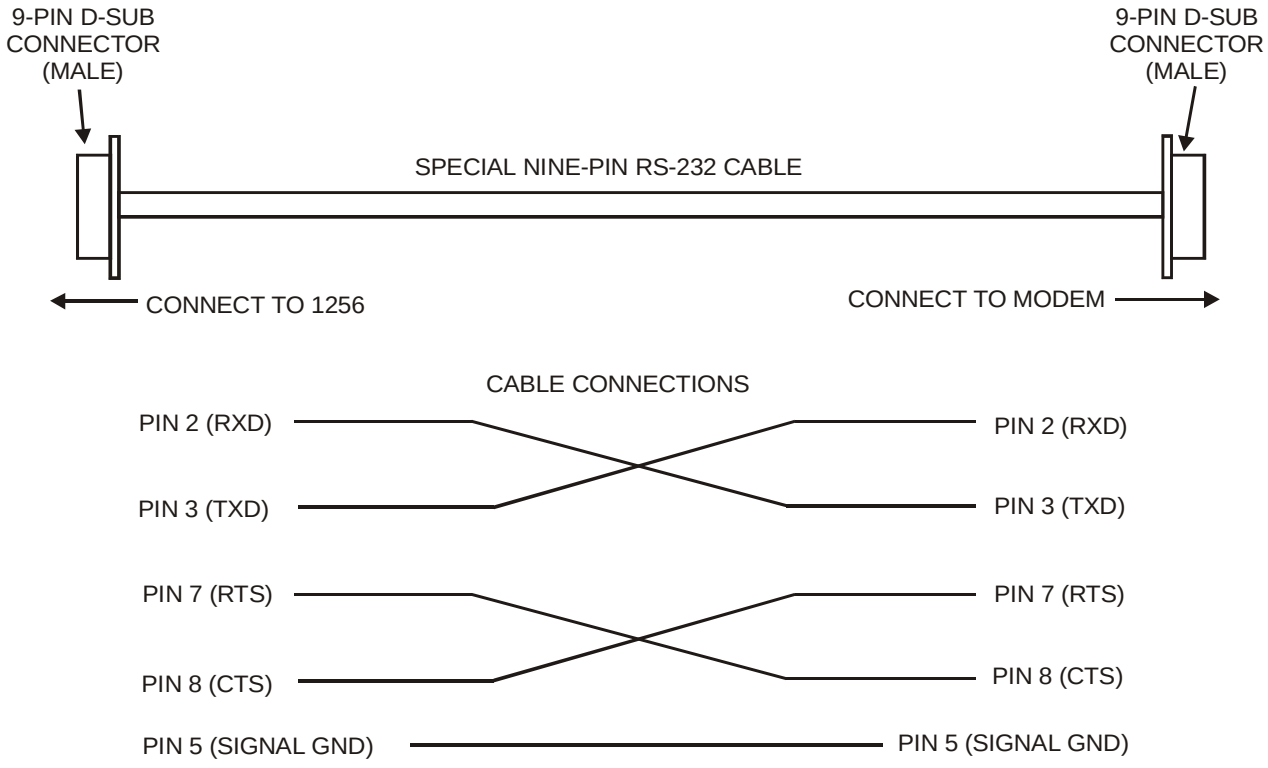


Figure 4-3b, Using the 1256 with a Modem

Selecting the RS-232 Interface

To use the RS-232 interface with the 1256, the RS-232 interface must be selected via the front-panel controls, as follows:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to top line) by pressing the second key. This selects line 2 so that you may choose the type of setting to work with:

   ```
    -----SETTINGS-----
   >   Type: Display  <
     Option: Format
     Status: Decimal
   ```

3. Rotate the knob until line 2 reads as shown below (lines 3 and 4 may differ):

   ```
    -----SETTINGS-----
   >   Type: Remote   <
     Option: Lockout
     Status: Disabled
   ```

4. Press the third key to select line 3 of the display. Rotate the knob until line 3 indicates the “Source” option as shown below:

   ```
    -----SETTINGS-----
       Type: Remote
   > Option: Source   <
     Status: GPIB
   ```

5. If line 4 of the display already indicates the status as “RS-232”, then the RS-232 interface is already active; no further action is necessary. Otherwise, press the fourth key. Rotate the knob until line 4 indicates the RS-232 interface:

   ```
    -----SETTINGS-----
       Type: Remote
     Option: Source
   <Update?: RS-232
   ```

   Then press the fourth key to confirm the update. Pressing any other button cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting the Flow Control Option

Description

Flow control is a means by which the data terminal (or host computer acting as a terminal) may pace the flow of data coming from the 1256 so that the terminal has time to properly receive the data. It also allows the 1256 to pace commands coming from the terminal so that the 1256 has sufficient time to execute the commands as they arrive.

The 1256 provides three flow control options:

- **XON/XOFF: Software flow control.** The RS-232 device that is receiving data may send an XOFF character (ASCII 19_{10}) to the sending device, commanding the sending device to suspend data output. Later, when the receiving device is ready for more data, it may send an XON character (ASCII 17_{10}) to the sending device, indicating that data transmission may be resumed.
- **CTS/RTS: Hardware flow control.** The 1256 normally holds the RS-232 Clear-To-Send (CTS) line in the TRUE state (-12 volts). When the 1256 is receiving data, and needs to have data transmission suspended, it takes the CTS line to the FALSE state (+12 volts). When it is ready for more data, it returns the CTS line to the TRUE state. Similarly, the data terminal uses the RS-232 Request-To-Send (RTS) line to pace data coming from the 1256. The data terminal normally holds the RTS line in the TRUE (-12 volts) state. To suspend data from the 1256, it takes the RTS signal to the FALSE state (+12 volts). To resume data transfer, the terminal takes the RTS signal TRUE again.
- **Disabled: No flow control.** If your terminal does not support flow control, set the 1256 to disable flow control. Note that sending large blocks of data without flow control may necessitate a slower baud rate to avoid data overrun problems.

Setting the Flow Control Option

To select the flow control option, or to disable flow control, proceed as follows:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to top line) by pressing the second key. This selects line 2:

   ```
    -----SETTINGS-----
   >   Type: Display  <
     Option: Format
     Status: Decimal
   ```

3. Rotate the knob until line 2 reads as shown below (lines 3 and 4 may differ):

```
 -----SETTINGS-----
>   Type: RS-232   <
  Option: Baud Rate
  Status: 9600
```

4. Press the third key to select line 3 of the display. Rotate the knob until line 3 indicates the “Flow Control” option as shown below:

```
 -----SETTINGS-----
    Type: Remote
> Option: FlowCtrl <
  Status: Disabled
```

5. The three available flow control options on line 4 are:

   - Disabled
   - XON/XOFF
   - RTS/CTS

   If line 4 of the display already indicates the desired choice, then no action is necessary. Otherwise, press the fourth key. Rotate the knob until the desired selection shows on line 4. In this example, XON/XOFF is desired:

```
 -----SETTINGS-----
    Type: RS-232
  Option: FlowCtrl
<Update?: XON/XOFF
```

   Then press the fourth key to confirm the update. Pressing any other button cancels the change.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting the Baud Rate

To set the baud rate (bit rate, in bits per second), proceed as follows:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to top line) by pressing the second key. This selects line 2 so that you may select the type of setting to work with:

   ```
    -----SETTINGS-----
   >   Type: Display  <
     Option: Format
     Status: Decimal
   ```

3. Rotate the knob until line 2 reads as shown below (lines 3 and 4 may differ):

   ```
    -----SETTINGS-----
   >   Type: RS-232   <
     Option: Data Bits
     Status: 8
   ```

4. Press the third key to select line 3 of the display. Rotate the knob until line 3 appears as shown below:

   ```
    -----SETTINGS-----
       Type: RS-232
   > Option: Baud Rate<
     Status: 2400
   ```

5. If line 4 of the display already indicates the desired baud rate, then no action is necessary. Otherwise, press the fourth key. Rotate the knob until the desired baud rate shows on line 4. In this example, 9600 is desired:

   ```
    -----SETTINGS-----
       Type: RS-232
     Option: Baud Rate
   <Update?: 9600
   ```

   Then press the fourth key to confirm the update. Pressing any other button cancels the change.

Set the RS-232 port of your data terminal or computer to operate at the same baud rate as the 1256. Refer to the user's manual for your terminal for instructions on making these settings. If you are using a computer with terminal emulation, refer to the user's manual for the terminal emulation software.

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting the Message Termination Characters

Each message to or from the 1256 must end with an end-of-line (EOL) character known as a terminator. You may select the EOL terminator to be any one of the following:

- Carriage return (CR) only
- Line feed (LF) only (for non-terminal-based applications, such as controlling via LabVIEW programs)
- Carriage return and line feed (CR-LF)

The factory default is line feed only.

To select the EOL terminator, proceed as follows:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to top line) by pressing the second key. This selects line 2 so that you may choose the type of setting to work with:

   ```
    -----SETTINGS-----
   >   Type: Display  <
     Option: Format
     Status: Decimal
   ```

3. Rotate the knob until line 2 reads as shown below (lines 3 and 4 may differ):

   ```
    -----SETTINGS-----
   >   Type: RS-232   <
     Option: Data Bits
     Status: 8
   ```

4. Press the third key to select line 3 of the display. Rotate the knob until line 3 appears as shown below:

   ```
    -----SETTINGS-----
       Type: RS-232
   > Option: EOL Mode <
     Status: CR Only
   ```

5. If line 4 of the display already indicates the desired EOL terminator, then no action is necessary. Otherwise, press the fourth key. Rotate the knob until the desired EOL terminator shows on line 4. In this example, “CR/LF” is desired:

```
-----SETTINGS-----
   Type: RS-232
 Option: Baud Rate
<Update?: CR/LF
```

   Then press the fourth key to confirm the update. Pressing any other button cancels the change.

When using a data terminal, or a computer with terminal-emulation software, it is recommended that you set the 1256 to use the CR-only terminator. In this case, also set the data terminal or computer to half-duplex operation (local echo), with line feed appended to incoming messages (see your data terminal manual for information on how to do this).

NOTE:

Menu preferences are not normally retained after the power is removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Setting the Data Bits, Stop Bits, and Parity

The 1256 allows the user to set the following additional RS-232 parameters:

- **Data bits:** Data bits per frame. The RS-232 interface allows either 7 or 8 bits per data frame. For most applications, set this to 8.

- **Stop bits:** Stop bits at end of each frame. The RS-232 interface can insert either one or two stop bits at the end of each data frame. Slow data terminals may require the extra delay resulting from two stop bits. However, for most applications, one stop bit is sufficient.

- **Parity:** Type of parity checking. Parity checking provides a level of confidence that data has been received properly. When parity checking is used, an extra bit (the parity bit), is inserted at the end of each data frame. The RS-232 interface provides three parity options: “none” (no checking performed), “even” (sum of all bits including parity bit is an

even number or zero), and “odd” (sum of all bits including parity

is an odd number). For most applications, this is set to “none”.

To set these parameters, proceed as follows:

1. Press the top key. Rotate the knob until the top line reads:

   ```
   >-----SETTINGS-----<
   ```

2. Select line 2 (next to top line) by pressing the second key. This selects line 2 so that you may select the type of setting to work with:

   ```
    -----SETTINGS-----
   >   Type: Display  <
     Option: Format
     Status: Decimal
   ```

3. Rotate the knob until line 2 reads as shown below (lines 3 and 4 may differ):

   ```
    -----SETTINGS-----
   >   Type: RS-232   <
     Option: Data Bits
     Status: 8
   ```

4. Press the third key to select line 3 of the display. Rotate the knob until line 3 indicates the desired parameter as shown below (in this example, the parity option is chosen):

   ```
    -----SETTINGS-----
       Type: RS-232
   > Option: Parity   <
     Status: Even
   ```

5. If line 4 of the display already indicates the desired choice for the parameter, then no action is necessary. Otherwise, press the fourth key. Rotate the knob until line 4 displays the desired choice (in this example, “None” is the desired choice):

   ```
    -----SETTINGS-----
       Type: RS-232
     Option: Parity
   <Update?: None
   ```

   Then press the fourth key to confirm the update. Pressing any other button cancels the change.

Set the data terminal or computer to match the settings you select for the 1256.

NOTE:

Menu preferences are not normally retained after the power is

removed from the instrument. To make (a) menu preference(s) permanent, make the desired change(s) to settings first and then proceed to the section Storing the Menu Preferences to make the changes permanent. Following this procedure will put the settings into non-volatile memory and allow them to be restored during instrument power-up.

Using Microsoft HyperTerminal

One of the easiest ways to manually control the 1256 through the RS-232 interface is to use HyperTerminal. HyperTerminal is a terminal emulation program that is included with Windows 95 and 98.

Before using HyperTerminal, or any other terminal emulation program, set up the 1256 RS-232 parameters as follows (refer to the above sections for details about accessing these settings via the front-panel controls):

Baud rate:	9600
Data Bits:	8
Parity:	None
Stop Bits:	1
Flow Control:	Disabled
EOL Mode:	CR/LF

To run HyperTerminal, click on Start, click on Run, then type:

hypertrm

(use the exact spelling above). Then press Enter.

The HyperTerminal program then starts, and displays the Connection Description dialog box as shown in **Figure 4-4**.

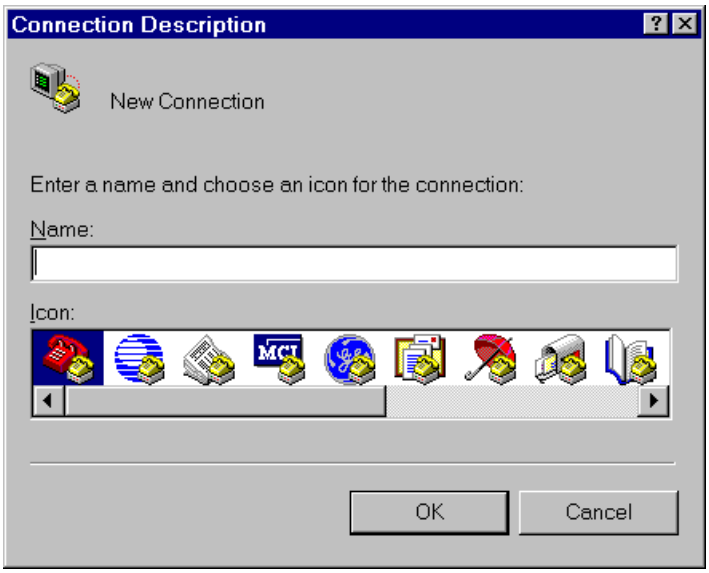


Figure 4-4, HyperTerminal Connection Dialog Box

Enter a name, such as “1256”, for the new connection. Then click on OK.

HyperTerminal then displays the Phone Number dialog box as shown in **Figure 4-5** (your screen may differ slightly).

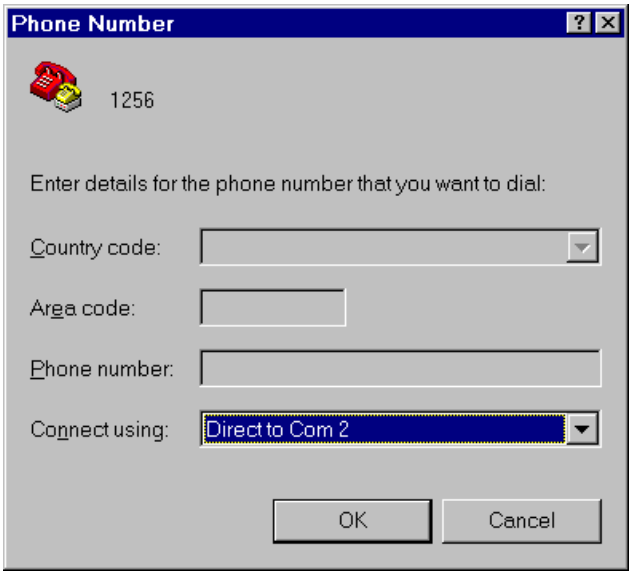


Figure 4-5, Phone Number Dialog Box

On the Connect Using line, select the serial port you wish to use. This example uses Com 2. Click on OK.

HyperTerminal then displays the COM Properties dialog box shown in **Figure 4-6**.

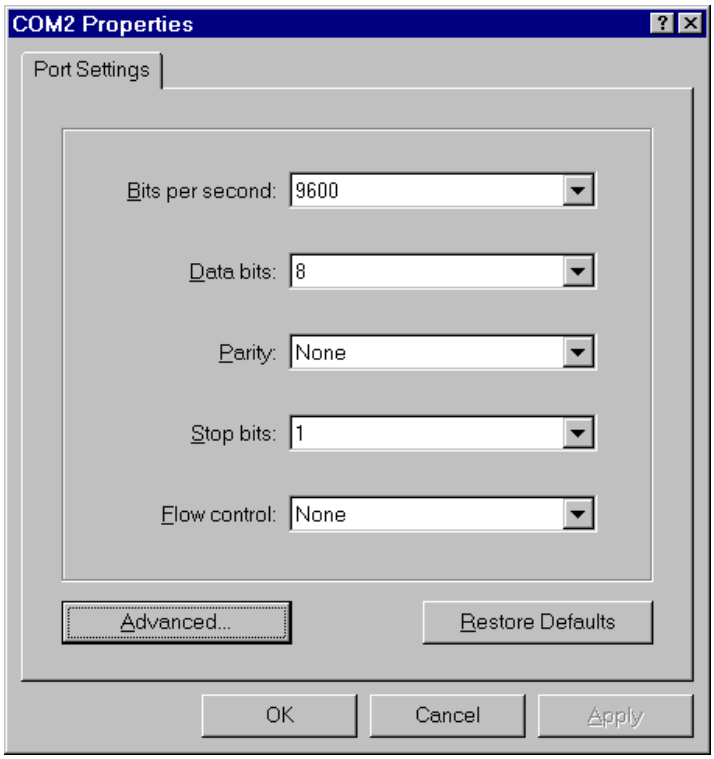


Figure 4-6, COM Properties Dialog Box

Set the baud rate to match the 1256 baud rate. Set the port for 8 data bits, no parity, one stop bit, and no flow control, as shown in **Figure 4-6**.

Click on OK. HyperTerm then displays the main terminal screen as shown in **Figure 4-7**.

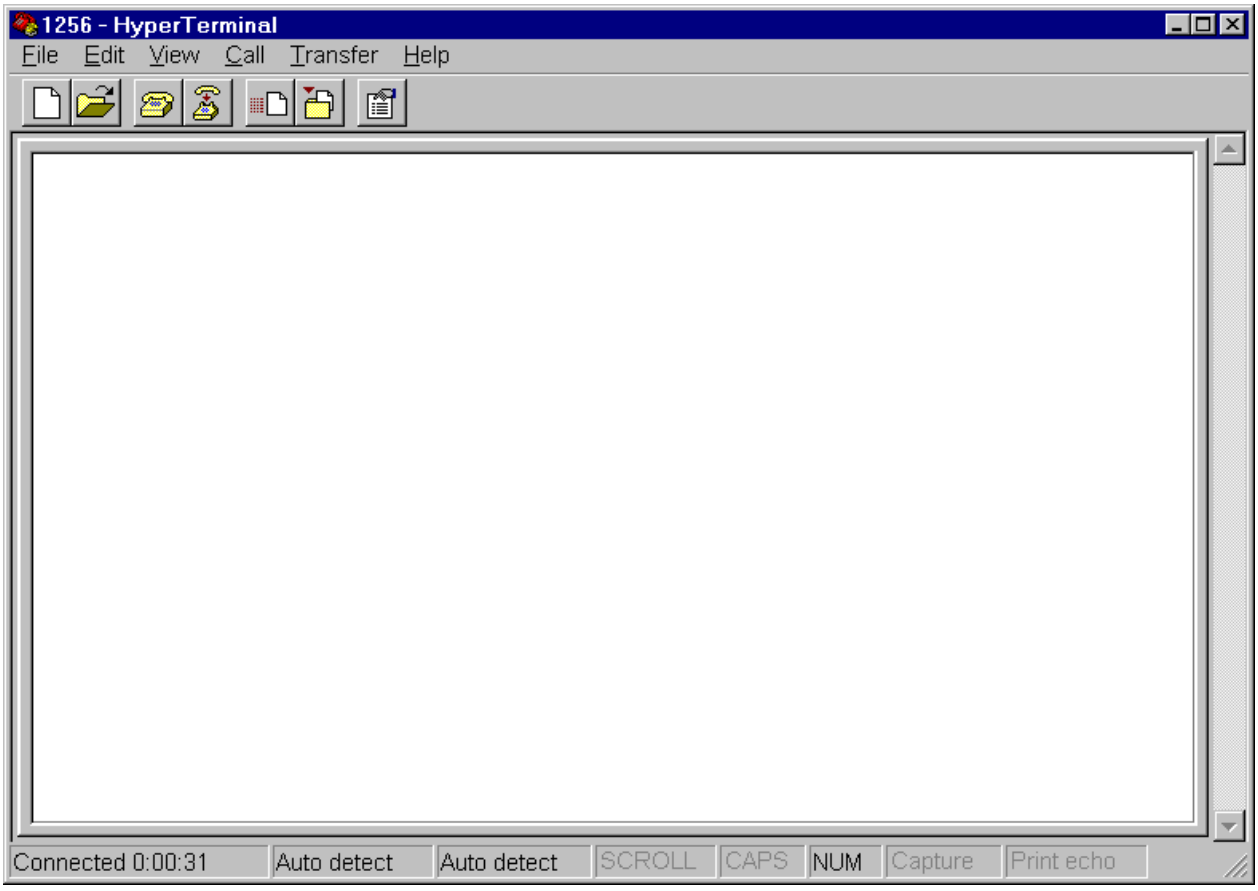


Figure 4-7, Main Communications Screen

Click on the File menu, then on Properties, to display the Properties dialog box shown in **Figure 4-8** (your screen may differ slightly).

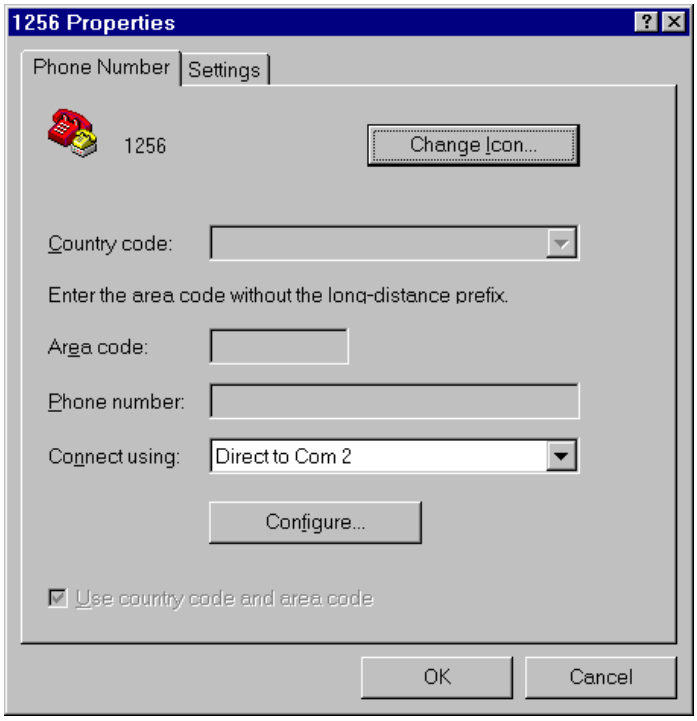


Figure 4-8, Properties Dialog Box

Click on the Settings tab. The Settings dialog box then appears as shown in **Figure 4-9** (your screen may differ slightly).

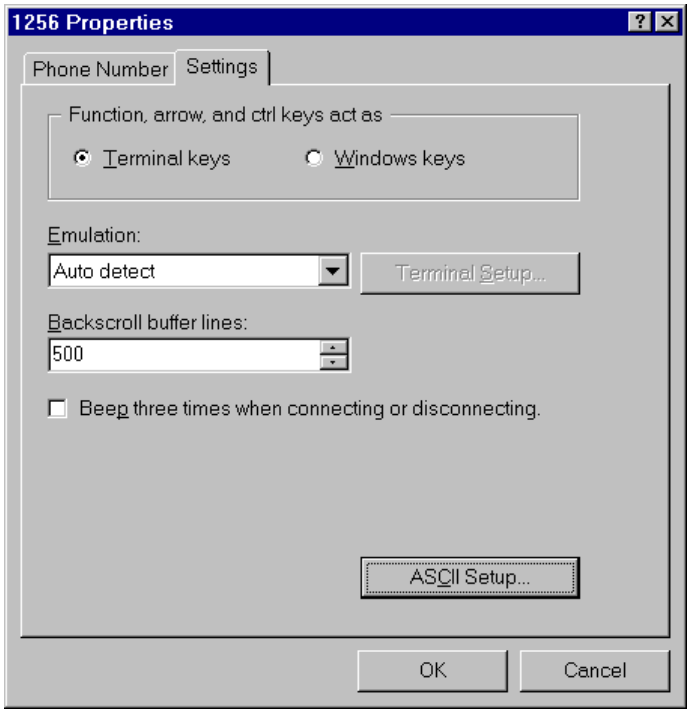


Figure 4-9, Settings Dialog Box

Click on ASCII Setup. HyperTerminal then displays the ASCII Setup dialog box shown in **Figure 4-10**.

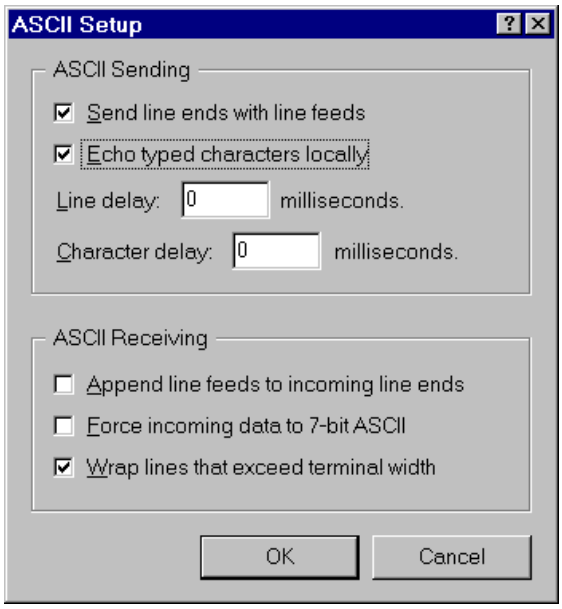


Figure 4-10, ASCII Setup Dialog Box

Using the mouse, check the boxes as shown in the figure. Make sure all other boxes are unchecked. Then click on OK, taking you back to Setting Dialog Box. Click OK again. The main communications screen then appears as shown in **Figure 4-7**.

To finalize the connection, click on the Call menu, then click on Connect.

To test the communications between the computer and the 1256, type the following command:

```
*IDN?
```

Then press Enter. The 1256 should respond with:

```
Racal Instruments 1256 Switch System, 03.10
```

The number at the end of the response is the firmware revision number, which may differ from what is shown above.

NOTE:

The version of HyperTerminal that is shipped with Microsoft Windows may not echo characters to the screen properly. An updated version is available from Hilgraeve, Inc. at:

http://www.hilgraeve.com

The new version is called HyperTerminal Private Edition.

Chapter 5
SCPI COMMAND BASICS

SCPI Command Overview

The 1256 accepts commands over either the GPIB or serial interfaces. These commands follow the rules defined by the **SCPI** standard. SCPI is an acronym for "Standard Commands For Programmable Instruments", and defines standard command names and syntax rules for commands to the instrument and replies from the instrument.

Input Format

SCPI is an ASCII-based instrument command language designed for test and measurement instruments. SCPI commands are based on a hierarchical structure known as a tree system. In this system, associated commands are grouped together under a common root, thus forming command subsystems. Throughout this chapter, the following conventions are used for SCPI command syntax.

Square Brackets ([ ])	Enclose optional keywords or parameters.
Braces ({ })	Enclose possible parameters within a command.
Triangle Brackets (< >)	Substitute a value for the enclosed parameter.
Vertical Bar (|)	Separate multiple parameter choices.
Bold Typeface Letters	Designate factory default values.
(Command Only)	This indicates the command cannot be used in query form.
(Query Only)	This indicates the command can only be used as a query. The command form, without the question mark is not allowed.

To illustrate the SCPI notation, a part of the ROUTE command

subsystem is shown below:

```
[:ROUTe]
      :CLOSe <channel list>
      :OPEN <channel list>
            :ALL
```

ROUTE is the root keyword of the command. This keyword is optional, since it is shown enclosed in square brackets. CLOSE and OPEN are the next level keywords. The “ALL” keyword is below the OPEN keyword in this command tree.

The colon (:) is used to separate keywords from different levels on the command tree. Each keyword is separated from the next by a single colon.

The SCPI commands which may be formed by this tree are shown below (a <channel list> of (@5(0)) is used in these examples):

```
ROUTE:CLOSE (@5(0))
CLOSE (@5(0))
ROUTE:CLOSE? (@5(0))
CLOSE? (@5(0))
ROUTE:OPEN (@5(0))
OPEN (@5(0))
ROUTE:OPEN? (@5(0))
OPEN? (@5(0))
ROUTE:OPEN:ALL
OPEN:ALL
```

Note that the optional ROUTE keyword is omitted in many of the examples.

Command Keyword Long Form and Short Form

Each keyword defined by SCPI has both a **long form** and a **short form**. The long form is formed by using all letters shown in the keyword. The short form is formed by using only those letters shown in upper-case in the command tree.

The short form is normally three or four letters in length, and ends with a consonant where possible.

For example, the ROUTE keyword may be specified by either of the following:

```
ROUT
ROUTE
```

since the final “e” is shown in lower-case in the command tree.

Only the long form or short form may be used. For example, the keyword “DEFINE” is shown as:

```
DEFine
```

Therefore, the following two command keywords are valid:

```
DEFINE
DEF
```

But the keywords

```
DE
DEFI
DEFIN
```

are NOT valid.

Case Sensitivity

Command keywords are **NOT** case sensitive. Command parameters are not case sensitive, unless the parameter is string data enclosed in quotes.

For example, the following commands are equivalent:

```
CLOSE
Close
```

Optional Keywords

Command keywords enclosed in square brackets are optional. For example, the following commands are valid and equivalent:

```
ROUTE:CLOSE (@5(0))
CLOSE (@5(0))
```

Querying Parameter Setting

Most of the commands in SCPI have an equivalent query form. The query is used to read the present state of the item that is set with the command. The query is formed by adding a question mark (?) to the end of the command keyword.

For example, the command:

```
ROUTE:CLOSE (@5(0))
```

Has an equivalent query:

```
ROUTE:CLOSE? (@5(0))
```

The command instructs the 1256 to close channel 0 on relay module 5. The query inquires about the present open or close state of channel 0 on relay module 5.

SCPI Command Terminator

SCPI commands can be terminated several ways depending on whether the GPIB or RS-232 serial interface is used.

For commands sent through the GPIB interface, a command string sent to the 1256 must be terminated with one of the following:

1. An ASCII linefeed character (decimal 10, hex 0A, or a character '\n').
2. The last character with the EOI signal asserted.
3. An ASCII linefeed character with the EOI signal asserted.

For commands sent through the RS-232 interface, several different rules apply. Generally speaking, the RS-232 interface can be accessed a number of different ways that directly impact the termination method:

1. An ASCII linefeed character (decimal 10, hex 0A, or a character '\n').
2. An ASCII carriage return (decimal 13, hex 0D, or a character '\r').

3. An ASCII carriage return character followed by a linefeed character.

Because of the wide variability in termination schemes in serial mode, the 1256 must be configured through the front-panel user interface to indicate whether it should expect carriage returns only, linefeeds only, or carriage returns followed immediately by linefeeds.

To make a termination election, select `-----SETTINGS-----` on the top most line, select `Type: RS-232` on the second line, and `Option: EOL Mode` from the third line, and then choose the appropriate setting. The default power-up state of the instrument is linefeed only. To elect a different power-up state, this setting like all other menu options can be saved to non-volatile memory. For further details on front-panel usage refer to **Chapter 3: USING THE FRONT PANEL CONTROLS**.

The 1256 has the ability to interface with terminals and terminal emulation software. Unfortunately, a wide variation of capabilities exists in terminal/terminal emulators. Most terminal/terminal emulators generally treat the “ENTER” key as a carriage return, will operate in half-duplex mode, and have the ability to add linefeeds to incoming carriage returns.

To minimize frustration and obtain the most legible communications, set the terminal/terminal emulator to operate in half-duplex mode and add linefeeds automatically to incoming carriage returns. In this configuration, the 1256 should be configured to expect a carriage return only. Using this method, the terminal display will not have any overwritten lines and be much easier to read and use.

Another option for terminal/terminal emulators that support character translation is to have carriage returns translated into linefeeds. In this scenario, the terminal/terminal emulator still needs to operate in half-duplex mode, however, the 1256 if configured to accept linefeeds only, will now send a linefeed with replies. This eliminates the need to add linefeeds to carriage returns since no carriage returns come into play on either side.

The two methods mentioned here are only suggestions and do not cover all possibilities. The 1256 is a flexible instrument and will work in nearly all configurations.

IEEE-STD-488.2 Common Commands

The IEEE-STD-488.2 standard defines a set of common commands that perform functions like reset, trigger and status operations. Common commands begin with an asterisk (*), are four to five characters in length, and may include parameters. The command keyword is separated from the first parameter by a blank space. A semicolon (;) may be used to separate multiple commands as shown below:

```
*RST; *STB?; *IDN?
```

The IEEE-488. 2 common commands implemented by the 1256 are described later in this chapter. The 1256 complies with this standard for both GPIB and RS-232 operational modes.

IEEE 488.2 Common Commands

The 1256 supports all required IEEE-488.2 commands. In addition, the optional *OPT?, *SAV, and *RCL commands are supported. The following paragraphs describe the IEEE-488.2 commands supported by the 1256. In addition, the status reporting model implemented by the 1256 is described. The 1256 complies with this standard for both GPIB and RS-232 operational modes.

The following commands are implemented:

*IDN?	Identification query
*RST	Instrument reset
*TST?	Commanded self-test
*CLS	Clear status
*ESE	Set the Standard Event Status Enable register
*ESE?	Read the Standard Event Status Enable register
*ESR?	Read the Standard Event Status register
*SRE	Set the Service Request Enable register
*SRE?	Read the Service Request Enable register
*STB?	Read the status byte
*OPC	Set the OPC bit of the Standard Event Status register
*OPC?	Reply with “1" when executed (used for synchronizing)

*TRG	Send a trigger to the instrument over the bus
*SAV	Store relay states in non-volatile memory
*RCL	Recall relay states from non-volatile memory
*OPT?	Read if any options are installed

These commands are described in greater detail in this section of the manual.

IEEE-488. 2 Status Description

The IEEE-488.2 Status Reporting Model is shown in **Figure 5-1**. This figure shows how the status reporting data structures are implemented and the commands used to set and read each of the registers. While the GPIB operational mode of the 1256 is fully compliant with this standard, it is impossible for the 1256, or any instrument operating in the RS-232 operational mode for that matter, to satisfy the IEEE status requirements in all respects. A good example of non-compliance relates to the SRQ. SRQ is a feature that does not exist in RS-232 communications.

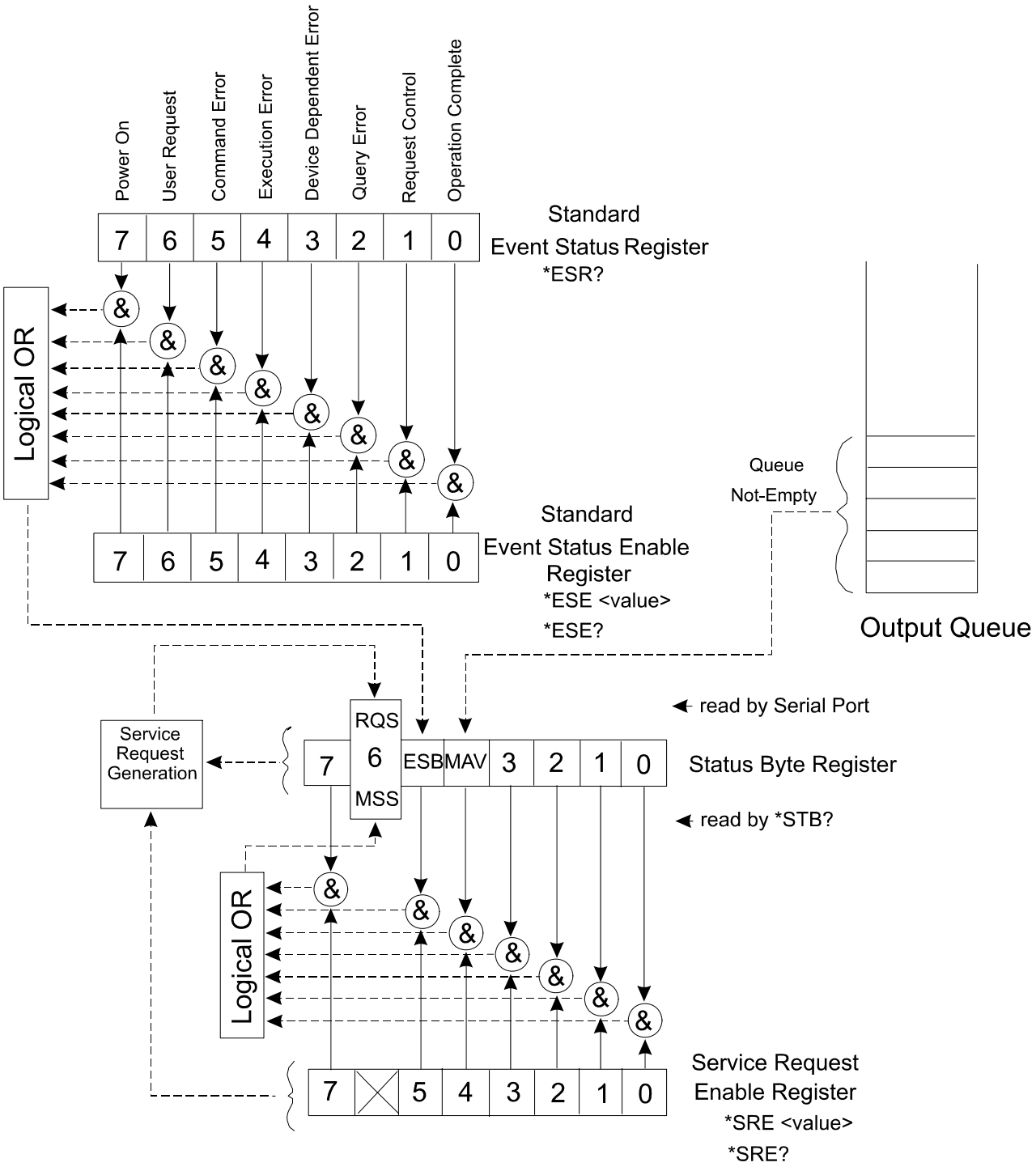


Figure 5-1, IEEE-488.2 Status Reporting Model

Standard EVENT STATUS Register

Figure 5-1 shows four related registers. The *Standard Event Status Register*, the *Standard Event Status Enable Register*, the *Status Byte Register*, and the *Service Request Enable Register*.

The *Standard Event Status Register* reflects the present status of the instrument. This register consists of eight 1-bit flags. Each flag represents a true or false indication of the corresponding condition. The bits of this register are:

PON Power-On

Bit 7 (MSB), Bit weight = 128 decimal = 80 hexadecimal

This flag is set when the instrument is powered on.

URQ User Request

Bit 6, Bit weight = 64 decimal = 40 hexadecimal

This bit is never set by the 1256 and will always read 0.

CME Command Error

Bit 5, Bit weight = 32 decimal = 20 hexadecimal

This bit is set when a command error is detected by the 1256.

Errors of this type will also result with an error added to the error queue. The error may be read using the SYST:ERR? query to determine the cause of the error

EXE Execution Error

Bit 4, Bit weight = 16 decimal = 10 hexadecimal

This bit is set when a valid command is received but cannot be executed for some reason. For example, an error occurs while executing the *SAV command. When this error occurs, an error message is added to the message queue and may be read using the SYST:ERR? query.

DDE Device Dependent Error

Bit 3, Bit weight = 8 decimal = 8 hexadecimal

This bit is set when a device-dependent error is detected. For example, a 1260-series relay module is read during power-up but contains an unknown identification byte value. When this error occurs, an error is added to the error message queue.

QYE Query Error

Bit 2, Bit weight = 4 decimal = 4 hexadecimal

This bit is set when a query error is detected. For example, a query is sent, but the reply is not read, and a second query or command is sent. When this error occurs, an error is added to the error message queue.

RQC Request Control

Bit 1, Bit weight = 2 decimal = 2 hexadecimal

This bit is never set by the 1256 and will always read 0.

OPC Operation Complete

Bit 0, Bit weight = 1 decimal = 1 hexadecimal

This bit is set when the *OPC command is executed. This may be used to synchronize the 1256 with the commands (to ensure that the 1256 command buffer is empty).

A bit is set in this register when the corresponding condition becomes true. It remains set until the *ESR? query is executed. When the query is executed, the reply contains the present value of the register, and the register is then cleared to 0.

The value returned by the *ESR? query represents a sum of the bit-weight values for all conditions that are true. For example, if the PON bit is set and the QYE bit is set, and the rest of the bits are cleared, then the value returned for the *ESR? query is:

PON + QYE = 128 + 4 = 132

The *Standard Event Status Enable Register* provides a mask register. The value of this register is logically ANDed with the *Standard Event Status Register*. If the value of this ANDing is nonzero, then bit 5 of the *Status Byte Register* is set. This bit is known as the “Event Summary Bit”, or ESB.

For example, if the PON and QYE bits of the *Standard Event Status Register* are set, but the *Standard Event Status Enable Register* value is 0, then the ESB of the *Status Byte Register* will not be set. If either, or both, bits 7 and bit 2 of the *Standard Event Status Enable Register* are set, then the ESB bit of the *Status Byte Register* will be set.

STATUS BYTE Register

Another way of viewing the *Standard Event Status Enable Register* is that it selects which conditions reflected in the *Standard Event Status Register* are enabled to set the ESB bit of the status byte.

The *Status Byte Register* is similar to the *Standard Event Status Register*. Each bit of this register reflects the true or false condition of the corresponding bit. These bits reflect the PRESENT value of the condition, whereas *the Standard Event Status Register* bits are latched. That is, once a bit in the *Standard Event Status Register* is set, it remains set until a *CLS command is executed or an *ESR? Query is executed. However, the bits of the *Status Byte Register* change states as the corresponding condition becomes true or false. These bits are NOT latched.

Also, the Status Byte is not cleared by reading the register. Each bit of the Status Byte remains set until the condition indicated by the bit is no longer present.

The following bits are assigned in the *Status Byte Register*. All other bits are not used and will return “0" when read with the *STB? query:

OSE Operation Status Event

Bit 7, bit weight = 128 decimal = 80 hexadecimal

This bit is set when any of the bits of the Operation *Status Event Register* are set. (This bit is NOT shown on the diagram. For a description of the *Operation Status Event Register*, consult the “**SCPI Status Registers**” section of this chapter.

MSS Master Summary Status.

Bit 6, bit weight = 64 decimal = 40 hexadecimal

This bit is set when one or more of the “enabled” bits of the Status Byte are set. In other words:

(Status Byte bit 0 AND SRE bit 0)
OR

(Status Byte bit 1 AND SRE bit 1)
OR

(Status Byte bit 2 AND SRE bit 2)
OR

(Status Byte bit 3 AND SRE bit 3)
OR

(Status Byte bit 4 AND SRE bit 4)
OR

(Status Byte bit 5 AND SRE bit 5)
OR

(Status Byte bit 7 AND SRE bit 7)

where the SRE is the *Service Request Enable Register*.

ESB Event Summary Bit

Bit 5, bit weight = 32 decimal = 20 hexadecimal

This bit is set when one of the *enabled Standard Event Status Enable Register* bits is set. The previous paragraphs describe the formation of the ESB bit.

MAV Message Available

Bit 4, bit weight = 16 decimal = 10 hexadecimal

This bit is set when there is a message in the output buffer of the 1256

All other bits (3, 2, 1, and 0) of the Status Byte are not assigned and will always return 0.

When the MSS transitions from a 0 to a 1, the GPIB SRQ interrupt is generated. The MSS will remain 1 until all enabled bits of the Status Byte have returned to 0.

SERVICE Register ENABLE Register

The *Service Request Enable Register* is used to individually enable bits in the Status Byte to set the MSS bit of the Status Byte, thereby generating an interrupt. When the bit of the Service Request Enable bit is set, and the corresponding bit of the *Status Byte Register* is set, then the MSS bit will be a 1.

The following IEEE-488.2 Common Commands relate to the Status Reporting Model:

*CLS	Clear status
*ESE	Set the Standard Event Status Enable register
*ESE?	Read the Standard Event Status Enable register
*ESR?	Read the Standard Event Status register
*SRE	Set the Service Request Enable register
*SRE?	Read the Service Request Enable register
*STB?	Read the status byte
*SPE	Serial Poll Enable

These common commands are described below.

*CLS Command

The *CLS command clears the SCPI and IEEE-488.2 defined status event registers. These include:

- The *Standard Event Status Register* is cleared to 0. This register is read using the *ESR? query.
- The *Standard Event Status Enable Register* is cleared to 0. This register is set using the *ESE command, and read using the *ESE? query.
- The *Service Request Enable Register* is cleared to 0. This register is set using the *SRE command, and read using the *SRE? query.
- The *Operation Status Enable Register* is cleared to 0. This register is set using the STATUS:OPERATION:ENABLE command. This register can also be read using the STATUS:OPERATION:ENABLE? query.
- The *Operation Status Event Register* is cleared to 0 This register is read using the STATUS:OPERATION:EVENT? query.

- The *Questionable Status Enable Register* is cleared to 0. To set this register, the STATUS:QUESTIONABLE:ENABLE command is issued. To read this register, the query STATUS:QUESTIONABLE:ENABLE? indicates whether this feature is enabled.
- The *Questionable Status Event Register* is cleared to 0. It is read using the STATUS:QUESTIONABLE:EVENT? Query.

***ESE Command**

The *ESE command sets the value of the *Standard Event Status Enable Register*. The value of this register is logically ANDed with the contents of the *Standard Event Status Register* (see the *ESR? query description). If any bits of this AND operation are set, then bit 5 of the Status Byte is set. Bit 5 is known as the Event Summary Bit, or ESB, and is shown in **Figure 2-3**.

ESB of Status Byte =

(bit 0 of ESE AND bit 0 of ESR)
OR

(bit 1 of ESE AND bit 1 of ESR)
OR

(bit 2 of ESE AND bit 2 of ESR)
OR

(bit 3 of ESE AND bit 3 of ESR)
OR

(bit 4 of ESE AND bit 4 of ESR)
OR

(bit 5 of ESE AND bit 5 of ESR)
OR

(bit 7 of ESE AND bit 7 of ESR)

where:

ESE is the value as set by the *ESE command

ESR is the value which may be read with *ESR? query

The *ESE command has the format:

```
*ESE <ESE value>
```

where the “<ESE value>” is an integer numeric value in the range from 0 through 255.

***ESE? Query**

This query reads the value presently programmed for the *Standard Event Status Enable Register*. This reads the value as programmed by the "*ESE" command.

***ESR? Query**

This query reads the value of *the Standard Event Status Register*. Each bit of this register indicates a true/false status condition. When the bit is set, the condition is TRUE; when the bit is cleared, the condition is FALSE. The bit assignments are defined by the IEEE-488.2 specification. The bit assignments are described in the text following **Figure 5-1**.

The bits of the *Standard Event Status Register* are cleared at power-on, except for bit 7, which is set. As conditions become true, the corresponding bit in the register are set. These bits remain set until:

- The *ESR? query is executed
- The *CLS command is executed

The reply to the *ESR? query is a numeric integer value in the range "0" to "255".

***SRE Command**

The *SRE command sets the value of *Service Request Enable Register*. The value of this register is logically ANDed with the contents of the Status Byte. If any bits of this AND operation are set, bit 6 of the Status Byte is set. Bit 6 is also known as the Master Status Summary bit. Consult the text immediately following **Figure 5-1** for a description of the MSS bit and the Service Request Enable register.

The command has the format:

```
*SRE <SRE value>
```

where the "<SRE value>" is an integer numeric value in the range 0 to 255. The value of bit 6 of this register is ignored, since it does not make sense to enable an interrupt when an interrupt is generated.

***SRE? Query**

This query reads the value presently programmed for *the Service Request Enable Register*. This reads the value as programmed by the *SRE command. The reply to this command is a numeric value in the range 0 to 255.

***STB? Query**

This query reads the value of the *Status Byte Register*. Each bit of this register indicates a true/false status condition. When the bit is set, the corresponding condition is TRUE; when the bit is cleared, the condition is FALSE. The bit assignments are defined by the IEEE-488.2 specification. The bit assignments are described in the Status Byte Register section of this chapter.

The value returned by the *STB? Query may also be read using the GPIB serial poll command.

Note that bit 6 (MSS) of the *Status Byte Register* remains set until all enabled conditions are cleared. This is in contrast to the GPIB serial poll operation, where the SRQ bit is set until it is read once, and cleared after.

***OPC Command**

The *OPC command will cause the Operation Complete bit of the Standard Event Status Register to be set when the command is executed. This is bit 0 of the register.

This command could be used to cause an interrupt (if bit 0 of the *Standard Event Status Enable Register* is set, and bit 5 of the *Service Request Enable Register* is set). This provides a means of synchronizing the application program with the 1256 and ensuring that all commands have been parsed and executed before continuing execution of the application program.

The *OPC command has no parameters. The only valid syntax for this command is:

```
*OPC
```

***OPC? Query**

The *OPC? query causes the 1256 to reply with the value of “1" when the query is executed. This query may be used to ensure that all previous commands have been executed so the application program may be sure that relays have been programmed to their desired states before continuing execution of the application program.

***IDN? Query**

This query requests the instrument to identify itself. The EMS responds to this query with the following reply:

```
Racal Instruments 1256 Switch System,<revision>
```

This reply indicates the manufacturer of the instrument is EADS North America Test and Services, that it is a 1256 Switch System, and the current firmware revision. The firmware revision is a numeric, floating point value. An example firmware revision is “3.10". A sample is shown below:

```
Racal Instruments 1256 Switch System,3.10
```

***RST Command**

The *RST command resets the instrument to its power-on default state. These settings are shown in **Table 5-1** of this manual.

This command does **NOT** change the value of SCPI Operation or Questionable status registers or IEEE-488.2 status registers, condition registers, or enable registers. This command does **NOT** clear the error message queue, the input command buffer, or the output reply buffer.

Table 5-1, Power-On and Reset State

Attribute	Related Command(s)	Reset State

Relay States	`*RCL 0`
	`OPEN`
	`CLOSE`
		The states are recalled from non-volatile memory location 0 (power-up state). As shipped from the factory, these are all in the OPEN position.

Trigger Input Source	`TRIGGER:SOURCE`	IMMediate
Trigger Count	`TRIGGER:COUNT`	1
Trigger Input Delay	`TRIGGER:DELAY`	0.0 seconds
Trigger Output	`OUTPUT:TRIG`	Off
Trigger Output Delay	`OUTPUT:DELAY`	0.0 seconds
Confidence Mode	`MONITOR:STATE`	Off
Pathname List	`ROUTE:PATH`	Pathnames stored in non-volatile memory are automatically recalled at power-up.
Scan List	`ROUTE:SCAN`	No Scan List Defined
Include List	`ROUTE:INCLUDE`	No Include Lists Defined
Exclude List	`ROUTE:EXCLUDE`	No Exclude Lists Defined

***TST? Query**

The *TST? query is supported only to be compliant with the IEEE-488.2 standard. The 1256 performs an extensive self-test at every power-up. If the system detects any failures, the condition is indicated directly on the front-panel display. Normally self-test errors are considered to be fatal conditions and result in a system halt. In this condition, communication activity ceases. For this reason, communications only function if self-test passes and therefore a *TST? will always report a value of “0” indicating a successful self-test.

***RCL Command**

The *RCL command will recall the relay states from non-volatile memory. The *RCL command may specify a non-volatile memory location from which to recall the instrument state. That is, both of the following formats are accepted:

```
*RCL

*RCL <location>
```

If <location> is specified, it must be in the range 0 to 100. If <location> is not specified, it will default to 100.

Location “0” has a special purpose. This location is used by the 1256 at power-up to set the initial relay states.

The *RCL command recalls the states of all relays in the system. The relay states are stored using the *SAV command.

Note that path names, module names, status registers, include lists, and so on are NOT affected by the *RCL command.

***SAV Command**

The *SAV command will prepare to store a given state into non-volatile memory. The actual process of updating non-volatile memory for this command requires a separate SYSTEM:NVUPD command. This allows multiple relay states and module names to be updated in rapid succession, followed by a single, slower update of the actual non-volatile memory.

The *SAV command accepts one of two formats:

```
*SAV

*SAV <location>
```

If <location> is specified, it must be in the range from 0 through 100. If <location> is not specified, the instrument state will be saved into non-volatile memory location 100.

State 0 is recalled at power-up. **The 1256 is shipped without any**

data in state 0. This effectively tells the 1256 to open all relays at power-up. This default may be overwritten by placing all relays in the desired power-up state, and then executing the command.

```
*SAV 0
```

In the manner indicated earlier, a separate SYSTEM:NVUPD command must follow to actually update the non-volatile memory.

If new relay modules are added to the system after the *SAV 0 command has been executed, the new relay modules will not be programmed at power-up. Also, if module addresses are changed after the execution of the *SAV 0 command, the modules whose addresses have changed will not be programmed.

In general, whenever new modules are added, or module addresses are changed, then the *SAV 0 command should be used to place the relays of the new modules into the desired power-up state.

***SPE Command**

The status byte is returned to the controller in response to a Serial Poll Enable (SPE) message.

The status byte carries the following information:

Bit Number	Bit Value	Description
1 (LSB)	1	Unused
2	2	Unused
3	4	Unused
4	8	Scan list at break point
5	16	Ready
6	32	Programming error
7	64	Service requested
8 (MSB)	128	Unused

NOTE:

When bit 7 of the status byte (Service Requested) is set to show that SRQ is asserted, the 1256 will not respond to any GPIB commands until the interrupt has been serviced. Servicing may be done with a serial poll. After the interrupt has been serviced, the error code generated must be obtained via GPIB.

Bits 6 and 7 are cleared after each Serial Poll Enable (SPE) command. Bit 5 is cleared by sending instructions to the 1256, and is set when the 1256 finishes executing a command. These transitions coincide with the rising and falling edges, respectively, of the *External Trigger Out* signal. All status bits are active-high.

***TRG Command**

The *TRG command is required by the IEEE-488.2 specification. If the 1256 is armed (see the INIT:IMMEDIATE and INIT:CONTINUOUS commands), and the trigger source is “BUS” (see the TRIGGER:SOURCE command), then this will cause the next scan list action to occur.

This is equivalent to sending a GPIB bus trigger.

***WAI Command**

The *WAI command is required by the IEEE-488.2 specification. This command is accepted but has no effect on the 1256.

SCPI Status Registers

SCPI defines two additional registers beyond those shown in **Figure 5-1**. These are the *Operation Status Register* and the *Questionable Status Register*.

The *Operation Status Register* consists of three logical registers: a condition register, an enable register, and an event register. If any bit is set in the operation event register, bit 7 in the status byte register will also be set.

The *Operation Status Condition Register* holds the present condition of various instrument attributes. This register is a set of 1-bit flags. The conditions assigned to the bits of the register are shown below:

Digital Trigger Event — Bit 7, Bit weight = 128 decimal = 80 hexadecimal

This bit is set when a trigger event occurs on a 1260-114 digital I/O card.

Waiting For Arm — Bit 6, Bit weight = 64 decimal = 40 hexadecimal

This bit is set when a Scan List has been defined, but the 1256 is not armed. Use the INIT:IMMEDIATE or INIT:CONTINUOUS command to arm the 1256

Waiting for Trigger — Bit 5, Bit weight = 32 decimal = 20 hexadecimal

This bit is set when a Scan List has been defined, and the 1256 has been armed, but is waiting for a trigger. The TRIGGER:SOURCE command may be used to select a trigger source. If the trigger source is “BUS”, then the *TRG command will satisfy a trigger.

Settling — Bit 1, Bit weight = 2 decimal = 2 hexadecimal

This bit is set when the relays are settling due to a CLOSE, OPEN, or SCAN operation. This bit is cleared after the settling time has elapsed.

All of the other bits of this register are not used by the 1256. These bits will return a value of 0 when read.

The *Operation Status Enable Register* enables individual bits to pass through to the Operation Status Event Register. The bits of the *Operation Status Enable Register* are ANDed with the bits of the *Operation Status Condition Register*. If both bits are set, then the corresponding bit in the *Operation Status Event Register* is set. For example, if bits 1, 5, and 6 of *the Operation Status Condition Register* are set, and bits 5 and 6 of *the Operation Status Enable Register* are set, then bits 5 and 6 of the *Operation Status Event Register* will be set.

The *Operation Status Event Register* latches the status information. Once a bit is set in the *Operation Status Event Register*, it remains set until the bit is cleared by reading the register with the STATUS:OPERATION:EVENT? query, or by sending the *CLS command.

When any of the bits of the *Operation Status Event Register* are set, bit 7 of the *Status Byte Register* will be set.

The *Questionable Status Register* is not used by the 1256. When the *Questionable Status Condition Register* or *Questionable Status Event Register* are read, they will return a value of 0. The *Questionable Status Event Register* may be programmed and queried, but will have no effect on the operation of the 1256.

The following SCPI command tree shows the syntax of the SCPI STATUS commands:

```
:STATus
      :OPERation
            [:EVENt]?
            :CONDition?
            :ENABle
      :QUEStionable
            [:EVENt]?
            :CONDition?
            :ENABle
```

Examples of the commands are shown below. To enable the “Wait for Trigger” and “Wait for Arm” bits the following command achieves this:

```
STAT:OPER:ENABLE 96
```

To read the value of the enable register:

```
STAT:OPER:ENABLE?
```

To read the value of the event register (and clear the event register):

```
STAT:OPER?
```

To read the value of the condition register:

```
STAT:OPER:COND?
```

These commands augment the IEEE-488.2 Common Commands to provide additional status information.

SCPI Parameter Type

The SCPI language defines several different data formats to be used in program messages and response messages.

Numeric Parameters

Commands that require numeric parameters will accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation.

```
TRIG:DELAY 0.035
```

When a real, non-integer value is returned in a reply from the 1256, the floating point notation will be used. The only non-integer values returned from the 1256 correspond to the TRIGGER:DELAY and the OUTPUT:DELAY. The values will be a number between 0.0 and 10.0. At most, 6 digits follow the decimal point.

Integer values may be sent in the command using decimal, octal, hexadecimal, or binary values. The default base for values is decimal.

To specify a hexadecimal value, use the prefix #H. To specify an octal value, use the prefix “#Q”. To specify a binary value, use the prefix “#B”. The following values are all equivalent.

`123`	123 decimal
`#B1111011`	1111011 binary = 123 decimal
`#H7B`	7B hex = 123 decimal
`#Q173`	173 octal = 123 decimal

When an integer value is returned In a reply from the 1256, the value will be a decimal number.

Discrete Parameters

Discrete parameters are used to program settings that have a limited number of values. Parameters are NOT case sensitive. As an example of the discrete parameter, the TRIGGER:SOURCE command is specified as:

:TRIGger

:SOURce { BUS | HOLD | IMMediate | EXT }

Meaning the parameter must be one of the following:

```
BUS
HOLD
IMM
IMMEDIATE
EXT
```

Note that, just like command keywords, discrete parameters may

be specified using either the long form or the short form.

Whenever a discrete parameter is used, the query form of the command returns the SHORT form of the parameter value, in upper-case characters. That is, the command may be specified using either “IMMEDIATE” or “IMM”, but the query:

```
TRIGGER:SOURCE?
```

will return the reply

```
IMM
```

NOT

```
IMMEDIATE
```

Boolean Parameters

Boolean parameters represent a single binary condition that is either true or false. The 1256 accepts "OFF" or "0" for a false condition. The 1256 accepts "ON" or "1" for a true condition. The following command uses a boolean parameter:

```
[:ROUTe]
      :MONitor
            [:STATe]  { OFF | ON | 0 | 1 }
```

The following commands turn the monitor OFF:

```
ROUTE:MONITOR:STATE OFF
ROUTE:MONITOR:STATE 0
MON OFF
MON 0
```

The following commands turn the monitor ON

```
ROUTE:MONITOR:STATE ON
ROUT:MON:STAT ON
MON 1
```

When Boolean parameters are queried, the 1256 always replies with a “1", if the state is on, or “0", if the state is off. The keywords “ON” and “OFF” are NOT returned in the reply to ROUTE: MONITOR:STATE?, or any other query.

Command Input Buffer

The 1256 uses a command buffer to store commands sent either through the RS-232 or GPIB interfaces. The input buffer is 1024 characters in length.

If the input buffer fills to capacity and the GPIB interface is the active interface, the NRFD signal is asserted on the GPIB bus. This indicates to the system controller that the 1256 is not able to accept more data until it parses existing commands and clears space in the input buffer.

If the input buffer fills to capacity and RS-232 is the active interface, several possibilities exist depending on flow control arrangements. The 1256 instrument supports either no flow control, software flow control (XON/XOFF), or hardware flow control (CTS/RTS). Each scenario is described in more detail below.

If no flow control is selected, the input buffer is full, and additional commands are sent, the buffer will accept characters as it makes space. It is entirely possible therefore, to have only parts of a message queuing to the input buffer. This leaves unknown commands on the buffer that when parsed will generate errors. For this reason, if RS-232 communications are used without any flow control, the user has the responsibility to insure that buffer overrun conditions do not occur.

If software flow control is selected, and the input buffer reaches 80% capacity, the 1256 will send an XOFF character to the host. The 80% threshold provides a reasonable safety margin to allow for host software overhead. When the input buffer reaches the 60% capacity threshold, the 1256 will send an XON character to the host to resume data transfer.

If hardware flow control is selected, and the input buffer is full, the RTS signal from the 1256 will assert indicating to the remote host than no further characters will be received. As soon as room exists in the input buffer, the RTS signal will de-assert indicating that room exists in the input buffer.

To make a flow control selection, select:

```
-----SETTINGS-----
```

on the top line, then select:

```
Type: RS-232
```

on the second line, and then:

```
Option: FlowCtrl
```

on the third line. Then choose the appropriate setting on the

bottom line. The default power-up state of the instrument is no flow control. To select a different power-up state, this setting like all other menu options can be saved to non-volatile memory. For further details on front-panel usage refer to **Chapter 3: USING THE FRONT PANEL CONTROLS**.

The 1256 waits for a linefeed (carriage return or carriage return/line feed will also work in RS-232 mode if the 1256 is configured properly) or GPIB EOI terminator before any command is parsed. If 1024 characters are sent to the 1256 without any terminators, the 1256 will flush both input and output buffers and then queue and error message.

Because the 1256 uses an input buffer to store commands, it is possible that multiple CLOSE or OPEN commands may be stored in the input buffer before the first command is executed. To synchronize the application program with the 1256, and to ensure that relays are in the programmed state, a query may be sent to the 1256. Once the reply to the query is read, you can be sure that the relays are in the programmed state.

For example, the following command sequence may be used:

```
CLOSE (@5(0))
OPEN (@5(12))
CLOSE (@5(17))
OPEN (@5(16,18))
*OPC?
<read the reply>
```

Once the reply from the *OPC? query is read, all of the previous relay operations have been completed.

Reply Output Buffer

The 1256 maintains an output buffer for sending replies to commands. This buffer is 1024 characters in length.

Although unlikely, the output buffer could become filled with replies to commands. If the output buffer is filled, the SCPI-defined “QUERY Deadlock” condition will be detected by the 1256. In this case, the output buffer will be cleared and an error will be added to the error queue. In addition, the query error bit (QYE) of the IEEE-488.2 Standard Event Status Register will be set. The error queue may be read using the “SYST:ERR?” query, while the Standard Event Status Register may be read using the *ESR? query.

When a reply is in the output queue, the message available, or MAV, bit of the status byte is set. The status byte of the 1256 may be read using the GPIB serial poll feature.

Specifying Channels in Commands

To select a single channel in a command, both the address of the relay plug-in which contains the relay, and the channel number for the relay must be specified. The syntax to describe a single channel is:

```
(@<address> ( <channel> ) )
```

where

<address>	This is a number in the range “1” to “8” and corresponds to the slot number (1 through 8) of the relay plug-in.
<channel>	This is a number which identifies a single relay to operate. The range of valid values for <channel> depends on the particular Adapt-a-Switch plug-in being controlled.

So, to close the relay channel 17 on the plug-in with address 3 (slot 3), use the command:

```
CLOSE (@3(17))
```

Multiple channels for a single plug-in may be specified using the syntax:

```
(@<address>(<channel>,<channel>...))
```

So the command:

```
CLOSE (@3(1,5,9,11))
```

may be used to close channels 1, 5, 9, and 11 on the plug-in with address 3.

A range of channels for a single plug-in may be specified by using the syntax:

```
(@<address>(<channel1>:<channel2>))
```

This format indicates that all relays between <channel1> and <channel2> are to be operated. The command:

```
CLOSE (@3(1:10))
```

closes channels 1 through 10, inclusive, on the relay plug-in with address 3.

A range of relays and a list of single relays may be mixed in a command. For example, the command:

```
CLOSE (@3(1:10,12,15,17:19))
```

Closes channels 1 through 10, 12, 15, 17, 18, and 19 on the plug-

in with address 3.

Multiple relays on multiple plug-ins may also be specified. In general, the syntax:

```
(@<address>(<channel list>),<address>(<channel list>),...)
```

is used. For example:

```
CLOSE (@3(1:10, 17), 7(15),8(8:10))
```

This closes the following relays:

Slot 3	channels 1 through 10 and 17
Slot 7	channel 15
Slot 8	channels 8 through 10

Naming Relay Cards

Each plug-in module controlled by a 1256 may be given a name. This name may be used in place of the slot number in any command used to control a relay.

Defining Module Names

The MODULE:DEFINE command (MOD:DEF) may be used to define a name of a plug-in module. The syntax for this command is:

```
MOD:DEF <module name> , <module address>
```

The <module name> is a string of up to 44 characters. It must begin with a letter between “A” and “Z”. After the first letter, all other letters may be “A” - “Z”, “0" through “9", and the underscore “_”.

NOTE:

The module name may be 44 characters. However, to maintain true SCPI compatibility, it must not exceed 12 characters.

Examples of valid commands are as follows.

Assigns name to plug-in module 1 (slot 1):

```
MOD:DEF A,1
```

Assigns name to plug-in module 7:

```
MOD:DEF ABCDEFGHIJKL,7
```

Assigns name to plug-in module 8:

```
MOD:DEF A12345678901,8
```

Assigns name to plug-in module 2:

```
MOD:DEF ZZZZZZ2,2
```

Examples of invalid commands are as follows.

Incorrect; the module name must appear first:

```
MOD:DEF 5,ABCD
```

Incorrect; the module name must begin with a letter:

```
MOD:DEF 4ASDF,8
```

Incorrect (for SCPI compatibility only); the module name is 13 characters long and it must be ≤ 12:

```
MOD:DEF A123456789012,5
```

The following examples show how to use module names in place of plug-in addresses.

Assigns the name *matrix* to plug-in 5:

```
MOD:DEF matrix,5
```

Closes row 0, column 23 of *matrix*:

```
CLOSE (@matrix(23))
```

Assigns the name *Power* to module 6:

```
MOD:DEF Power,6
```

Close *Power* relay channels 7 through 12:

```
CLOSE (@Power(7:12))
```

Closes relays on two modules:

```
CLOSE (@Power(8),matrix(102:104))
```

Removing Module Names

The MODULE:DELETE command may be used to delete a single plug-in module name from the list of known module names. The MODULE:DELETE:ALL command may be used to remove all presently defined module names.

The syntax for these commands is:

```
[:ROUTe]
```

```
:MODule
      :DELete <module name>
            :ALL
```

The following examples illustrate the use of this command.

Removes the name “scanner”:

```
ROUTE:MODULE:DELETE scanner
```

Removes the name “matrix”:

```
MOD:DEL matrix
```

Remove all names:

```
MOD:DEL:ALL
```

Reading the Presently Defined Module Names

The MODULE:CATALOG? command may be used to read back all of the presently defined module names. The format for this command is:

```
[:ROUTe]
      :MODule
            :CATalog?
```

The reply to this command consists of the presently defined module names, separated by a comma. Module names are stored internally in upper-case characters by the 1256. These upper-case names are returned in the reply.

For example, suppose the following commands have been executed:

```
MODULE:DEFINE scanner,1
MODULE:DEFINE matrix,2
MODULE:DEFINE power,5
MODULE:DEFINE rf_mux,4
```

Then the query:

```
MOD:CAT?
```

returns the reply:

```
SCANNER,MATRIX,RF_MUX,POWER
```

Reading the Module Address for a Module Name

The module address associated with a module name may be read using the MODULE:DEFINE? query. The syntax for this command is:

```
[:ROUTe]
      :MODule
            :DEFine? <module name>
```

Using the example in the previous section of this manual, the query:

```
MODULE:DEFINE? matrix
```

returns the reply:

```
2
```

Storing the Module Names in Non-Volatile Memory

All module names presently defined may be prepared for storage in non-volatile memory using the MODULE:SAVE command. The actual process of updating non-volatile memory for this command requires a separate SYSTEM:NVUPD command. This allows multiple relay states, exclude lists, include lists, and module names to be updated in rapid succession, followed by a single, slower update of the actual non-volatile memory.

All module names stored in non-volatile memory may be recalled from non-volatile memory using the MODULE:RECALL command**.** The syntax for these commands is shown below:

```
[:ROUTe]
      :MODule
            :SAVe
            :RECall
```

These commands are NOT defined in the SCPI standard. These commands follow the syntax rules of SCPI to implement this functionality.

Naming a Path

A group of channels may also be assigned a name. When a group of channels is named, it is called a “path”. A path may consist not only of elements that must be closed (the closed channel list) but also of elements that must be opened (the open channel list) to complete the path. This may at first seem contradictory to close a path with switching elements that must be opened, but it nevertheless occurs.

A good example which illustrates the concept of an open channel list is a typical 1P2T failsafe microwave switch. Because this type of switch is failsafe, it only requires a single control line. By default, with no power applied, the first throw is always connected to the common. Energizing the control line opens the first throw and closes the connection between the second throw and the common. Both are obviously mutually exclusive.

Defining a path through the second throw of the example switch can be done strictly with a close channel list. The problem occurs when a path must go through the failsafe throw (first throw). To make this path the switch must be in an open or off state. The open channel list readily accomplishes this task.

Defining Path Names

The PATH:DEFINE command may be used to associate a name with one or more relays. The format for this command is:

```
[:ROUTe]
      :PATH
            :DEFine <path name> ,
                  <close channel list> ,
                  [,<open channel list>]
```

The <path name> follows the same name requirements as a <module name>, as described in the previous paragraphs.

The <close channel list> and the optional <open channel list> follows the syntax rules described in the “**Specifying Relays in Commands**” section of this chapter.

The following examples illustrate the use of a path name.

This associates the name path1 with the close channel list encompassing relays 6 through 9 on plug-in module 8 and relay 77 on plug-in module 4:

```
PATH:DEF path1,(@8(6:9),4(77))
```

This associates the name dmm_to_P177 with the close channel list, channels 205 and 305, on the module whose name is defined as “matrix”:

```
PATH:DEF dmm_to_P177,(@matrix(305,205))
```

This associates the name oscope1 with a close channel list for channels 0 and 3 on plug-in module 3 and an open channel list for plug-in module 5, channel 15.

```
PATH:DEF oscope,(@3(0,3)),(@5(15))
```

Once a path name is defined, it may be used in a CLOSE or OPEN command:

```
OPEN (@path1)
CLOSE (@dmm_to_P177)
CLOSE (@path1,dmm_to_P177,7(0:10))
```

A CLOSE command, for a given pathname, will effectively close all channels listed on the close channel list and open those listed on the open channel list.

An OPEN command, for a given pathname, will open all channels listed on the close channel list. Channels on the open channel list are ignored when using this command.

Path names may also be used in defining an “Include List”, an “Exclude List”, or a “Scan List”. The following sections describe each of these concepts.

When using path names with the “Include List”, “Exclude List” and “Scan List”, the presently defined path is used. That is, if the following sequence of commands is received:

```
PATH:DEF PATH1,(@5(0),7(0))
INCLUDE (@PATH1,1(0))
PATH:DEF PATH1,(@6(17),8(23))
CLOSE (@1(0))
```

then the path definition in effect when the INCLUDE command was defined is used. For the example shown then, channel 0 on module 5 and channel 0 on module 7 are affected; channel 17 on module 6 and channel 23 on module 8 are NOT affected.

Removing Path Names

The PATH:DELETE command may be used to delete a single path name from the list of known path names. The PATH:DELETE:ALL command may be used to remove all presently defined path names.

The syntax for these commands is:

```
[:ROUTe]
      :PATH
            :DELete <path name>
                        :ALL
```

The following examples illustrate the use of this command.

Removes the name “path1”:

```
ROUTE:PATH path1
```

Removes the name “testit”:

```
PATH:DEL testit
```

Removes all path names:

```
PATH:DEL:ALL
```

Reading the Presently Defined Path Names

The PATH:CATALOG? command may be used to read back all of the presently defined path names. The format for this command is:

```
[:ROUTe]
      :PATH
            :CATalog?
```

The reply to this command consists of the presently defined path names, each of which is separated by a comma. Path names are stored internally in upper-case characters by the 1256. These upper-case names are returned in the reply.

For example, suppose the following commands have been executed:

```
PATH:DEFINE dmm_2_pin1,(@1(117),2(17))
PATH:DEFINE dmm_2_pin2,(@1(116),2(14),7(23))
PATH:DEFINE cntr_2_pin1,(@1(217),2(24))
PATH:DEFINE cntr_2_pin2,(@1(216),2(37),7(3))
```

Then the query:

```
PATH:CAT?
```

returns the reply:

```
DMM_2_PIN1,DMM_2_PIN2,CNTR_2_PIN1,
CNTR_2_PIN2
```

Reading the Channel List for a Path Name

The module address associated with a module name may be read using the PATH:DEFINE? query. The syntax for this command is:

```
[:ROUTe]
      :PATH
           :DEFine? <path name>
```

Using the example in the previous section of this manual, the query:

```
PATH:DEFINE? dmm_to_pin1
```

returns the reply:

```
(@1(117),2(17)
```

Storing Path Names in Non-Volatile Memory

All paths presently defined may be prepared for storage in non-volatile memory using the PATH:SAVE command.

NOTE:

The actual process of updating non-volatile memory requires a separate SYSTEM:NVUPD command. This allows multiple relay states, exclude lists, include lists, and module names to be updated in rapid succession, followed by a single, slower update of the actual non-volatile memory.

Closing Relays

The CLOSE command may be used to close channels. To say a channel is closed, it means either:

a. The input of the channel is connected to the output (Single-Pole Single Throw).

b. The input of the channel is disconnected from the “normally closed” output and connected to the “normally open” output (Single-Pole Double-Throw).

c. The common input of a mux is connected to the output denoted by the channel number.

The syntax for the CLOSE command is:

```
[:ROUTe]
      :CLOSe <channel list>
```

The format for a “<channel list>” is described in the previous paragraphs of this section of the manual.

The CLOSE? command may be used to query the present state of the relays in the system. This command returns a reply of a sequence of “0" and “1", each of which are separated by a single ASCII space character. The value of the reply is “0" if the corresponding relay is opened, or “1" if the corresponding relay is closed.

The reply is one-for-one with the <channel-list>. For example, assume channels for a particular relay module at module address 7 are numbered as follows:

```
0, 1, 2, 3, 4, 10, 11, 12, 13, 14, 20, 21,
22, 23, 24, 30, 31, 32, 33, 34
```

This example module consists of 20 channels. Suppose that only the following channels are closed, while the remainder are open:

```
3, 20, 31
```

The following examples show the replies to the “CLOSE?” queries.

Command:

```
CLOSE? (@7(0:34))
```

Reply:

```
0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0
```

Command:

```
CLOSE? (@7(0))
```

Reply:

```
0
```

Command:

```
CLOSE? (@7(3, 20, 31))
```

Reply:

```
1 1 1
```

Opening Relays

The OPEN command may be used to open channels. To say a channel is open means:

a. The input of the channel is disconnected to the output (Single-Pole Single Throw)

b. The input of the channel is disconnected from the “normally open” output and connected to the “normally closed” output (Single-Pole Double-Throw).

The syntax for the “OPEN” command is:

```
[:ROUTe]
      :OPEN <channel list>
```

The format for a “<channel list>” is described in the previous paragraphs of this section of the manual.

The OPEN? command may be used to query the present state of the relays in the system. This command returns a reply of a sequence of “0" and “1", each of which are separated by a single ASCII space character. The value of the reply is “1" if the corresponding relay is opened, or “0" if the corresponding relay is closed. **Note that this is the opposite state from the** CLOSE? **query.**

Checking for Installed Plug-Ins

The MOD:LIST? command may be used to query the 1256 for the types and slot numbers of the cards present in the system.

```
[ROUTe]
      :MODule
           :LIST? [module address]
```

As an example, if a 1256 contains a 1260-136A, 1260-167B, and a 1260-138 in slots 4, 7, and 8 respectively, supposing that the following command was executed:

```
MOD:LIST?
```

the reply would be:

```
4 : 1260-136A HIGH CURRENT 1X42 (2X21)
MUX,7 : 1260-167B DUAL 1x6 RF SWITCHING
MODULE,8 : 1260-138 64-CHANNEL DPST 2A MUX
```

Individual slots can also be queried:

```
MOD:LIST? (@7)
```

Where the reply would be:

```
7 : 1260-167B DUAL 1x6 RF SWITCHING MODULE
```

This command is NOT defined in the SCPI standard. This command follows the syntax rules of SCPI to implement this functionality.

System Commands

The 1256 instrument supports several system commands. These commands generally access errors, SCPI standard information, as well as user-interface issues. Additionally, the 1256 uses a special command to update non-volatile memory.

Reading Error Messages

Whenever an error is encountered by the 1256, it will perform two actions:

1. One of the bits of the *Standard Event Status Register* will be set. This register may be read using the *ESR? Query.

2. An error message will be added to the error message queue. The error message queue may be read using the SYSTEM:ERROR? query (or SYST:ERR?).

Consult Appendix B for a list of error messages and more detailed explanations.

The error queue holds up to 15 error messages. Each time an error is detected by the 1256, it adds a new error to the error queue. Each time the SYSTEM:ERROR? query is received, the oldest (least recent) error message is returned.

The syntax for this command is shown below:

```
:SYSTem
      :ERRor?
```

The reply to the SYSTEM:ERROR? query uses the format:

<error code> , “<error message>”

where:

<error code>	This parameter is a numeric value. This value is “0” if there are no errors remaining in the error queue. This value is negative when an error exists on the error queue.

<error message>	This parameter is a string enclosed in double quotes. The error message provides some additional information about the error.

For example, the reply:

```
0,"No error"
```

indicates that no errors remain on the error queue, while the reply:

```
-102,"Syntax error ; missing @ sign"
```

indicates that a syntax error was detected in a previous command to the 1256. The reply:

```
-350,"Queue overflow"
```

is returned to indicate that all 15 places in the queue have been occupied and that the error queue is full.

SCPI Compliance Information

The SYSTEM:VERSION? query provides information about which SCPI standard the 1256 references and is required for compliance with SCPI.

The syntax for this command is shown below:

```
:SYSTem
      :VERSion?
```

For example, the query:

```
SYST:VERS?
```

always returns the following:

```
1994.0
```

User-Interface Lockout

The syntax for this command is shown below:

```
:SYSTem
      :KLOCK ON | OFF
      :KLOCK?
```

The SYSTEM:KLOCK command and SYSTEM:KLOCK? query are used to both control and determine the present state of the user-interface. In certain applications or environments where the 1256 is under automatic control, it may be desired or required that access to the 1256 be prohibited through the front panel user-interface.

For example, the command:

```
SYST:KLOCK ON
```

will cause the 1256 to display a remote operation screen on the display with a message indicating that local mode is in a lockout condition.

The command:

```
SYST:KLOCK OFF
```

will also display a remote operation screen, however, the fourth line will indicate a local option the user can activate by pressing the fourth line soft key.

To query the present lockout state:

```
SYST:KLOCK?
```

will reply with either ON or OFF depending on the present state of the 1256.

The 1256 operates in either automatic or manual user-interface lockout mode. In either case, the 1256 will always power-up in local mode. In manual lockout mode, the SYSTEM:KLOCK ON command must be explicitly issued to prevent local operations. In automatic lockout mode, any remote command detected by the 1256 puts the instrument in remote mode with lockout mode automatically enabled. To allow local access, the 1256 must receive a SYSTEM:KLOCK OFF command or experience a power cycle event.

To make a lockout election from the user interface, select `-----SETTINGS-----` on the top most line, select `Type: Display` on the second line, and `Option: Lockout` from the third line and then choose the appropriate setting. The default power-up state of the instrument is automatic mode disabled. To elect a different power-up state, this setting like all other menu options can be saved to non-volatile memory. For further details on front-panel usage refer to **Chapter 3: USING THE FRONT PANEL CONTROLS**.

Non-Volatile Update

The syntax for this command is shown below:

```
:SYSTem
      :NVUPD
      :NVUPD?
```

The SYSTEM:NVUPD and SYSTEM:NVUPD? query are used to both initiate and determine whether a non-volatile update cycle is in progress.

To improve system reliability and eliminate the need for batteries, the 1256 stores configuration information into flash memory. The downside of flash memory is that it updates more slowly than battery-backed memories.

To minimize the effects of the flash memory update cycle, the 1256 allows individual selections (i.e. include list, exclude lists, path names, menu selections, etc) to be instantaneously saved into intermediate volatile memory. When all of these intermediate operations are completed, the SYSTEM:NVUPD command starts the actual process of transferring the contents of intermediate volatile memory into non-volatile flash memory with a single, slower update.

For example, the command:

```
SYST:NVUPD
```

will cause the 1256 to start the transfer process.

The command:

```
SYST:NVUPD?
```

will return:

```
ACTIVE
```

during the update process and:

```
IDLE
```

at any other time or after the update process has completed.

Attempting to change intermediate memory during a flash update cycle is prohibited. To illustrate this point, sending a MOD:SAV while SYST:NVUPD? returns ACTIVE, will cause the 1256 to ignore the request and push an error onto the system error queue.

These commands are NOT defined in the SCPI standard. These commands follow the syntax rules of SCPI to implement this functionality.

This page was left intentionally blank.

Chapter 6

SCPI Command REFERENCE

General

This section contains a detailed description of each command that is either specific to the 1256 or common to multiple switch modules. The commands are presented in **Table 6-1**.

The commands accepted by the 1256 are shown using the SCPI syntax notation. The command syntax in **Table 6-1** is displayed using the following conventions:

Square Brackets ([ ])	Enclose optional keywords or parameters.
Braces ({ })	Enclose possible parameters within a command.
Triangle Brackets (< >)	Substitute a value for the enclosed parameter.
Vertical Bar (|)	Separate multiple parameter choices.
Bold Typeface Letters	Designate factory default values.
(Command Only)	This indicates the command cannot be used in query form.
(Query Only)	This indicates the command can only be used as a query. The command form, without the question mark is not allowed.

To illustrate the SCPI notation, a part of the ROUTE command subsystem is shown below:

```
[:ROUTe]
      :CLOSe <channel list>
      :OPEN <channel list>
            :ALL     (Command Only)
```

This example shows the following:

- The ROUTE command keyword is optional, since it is enclosed in square braces
- The ROUTE command keyword may be specified as ROUTE (long form) or as ROUT (short form), since the last character is displayed in lower-case
- The CLOSE keyword must be followed by a <channel list>. A <channel list> represents any of a number of possible channels in a variety of formats. The format for the <channel list> and other command parameters are described in **Chapter 5: SCPI COMMAND BASICS**.
- Since the ROUTE keyword is optional, and the CLOSE keyword is shown indented from the ROUTE keyword, valid commands consist of:

```
ROUTE:CLOSE

CLOSE
```

- The command ROUTE:OPEN:ALL is a command only. Therefore, the command ROUTE:OPEN:ALL? is not allowed. Since they are NOT shown as command-only, the commands ROUTE:CLOSE? and ROUTE:OPEN? are valid commands.

Table 6-1 contains a synopsis of the commands implemented by the 1256. The maximum and minimum values and resolution are shown for numeric parameters. The commands marked with an asterisk (*) are NOT defined in the SCPI language. These commands follow the SCPI syntax rules for implementing the operation of the command.

Table 6-1, Commands Implemented by the 1256

Command	Max	Min	Default	See page	Notes
ROUTE Subsystem:				**6-1, 8-1**	
[:ROUTe]				**5-2, 5-25**	
:CLOSe <channel list>				**5-2, 5-28**	
:CLOSe?				**5-38, 6-1**	
:OPEN <channel list>				**5-2, 5-27**	
:OPEN?				**5-37, 6-1**	
:ALL (Command Only)				**5-2, 6-1**	
:SCAN <scan list>				**8-15**	
:PATH				**5-33**	
:CATalog? (Query Only)				**5-35**	
:DEFine <path name> , <close channel list> [, <open channel list>]				**5-36**	
:DEFine? <path name>				**5-36**	no chan list for query
:DELete				**5-35**	
[:NAME] <path name> (Command Only)				**5-35**	
:ALL (Command Only)				**5-35**	
:SAVe* (Command Only)				**5-33**	
:RECall* (Command Only)				**5-33**	
:MODule				**5-29, 8-3**	
:CATalog? (Query Only)				**5-31**	
:LIST?* [<module list>] (Query Only)				**5-38**	
:DEFine <module name> , <module number>				**5-29**	
:DEFine? <module name>				**5-29**	no module number for query
:DELete				**5-31**	
[:NAME] <module name> (Command Only)					
[:SAVe]* (Command Only)				**5-33**	
[:RECall]* (Command Only)				**5-33**	
:CONFigure <module list> , { BBM | MBB | IMMediate }			BBM	**8-1**	no relay mode for query
:CONFigure? <module list>				**8-1**	
:EXCLude* <channel list>				**8-5**	no exclude list
:EXCLude?				**8-5**	
:DELete <channel list>				**8-6**	
:ALL				**8-6**	
:INCLude* <channel list>				**8-2**	
:INCLude?				**8-2**	
:DELete <channel list>				**8-2**	
:ALL				**8-2**	
:MONitor*				**8-17**	
[:STATe ON | OFF | 1 | 0]				**5-25**	
:VERify*				**8-18**	
:MASK <channel list>, 0 | 1 | X				**8-18**	

Table 6-1, Continued

Command	Max	Min	Default	See page	Notes
:MASK? <channel list>				**8-18**	
:SAVE				**8-18**	
:RECALL				**8-18**	
[:STATE ON | OFF| 1 | 0]				**8-20**	
:VERify? <channel list or path name>				**8-18**	
:VERify:ALL? (Query Only)				**8-18**	
SYSTEM Subsystem:				**5-36**	
:SYSTem					
:ERRor? (Query Only)				**5-40**	
:VERSion? (Query Only)				**5-41**	1994.0
:NVUPD*				**5-43**	
:NVUPD?				**5-43**	
:KLOCK { ON | OFF }				**3-20, 5-41**	
:KLOCK?				**5-41**	
STATUS Subsystem:				**5-21**	
:STATus				**5-23**	
:PRESet (Command Only)					
:OPERation				**5-23**	
[:EVENt]? (Query Only)				**5-23**	
:CONDition? (Query Only)	255	0	0	**5-23**	
:ENABle <enable mask>				**5-23**	
:ENABle?				**5-23**	
:QUEStionable					When the register or status event are read, they will return a value of zero (0). These commands are not used with the 1256
[:EVENt]? (Query Only)					
:CONDition? (Query Only)					
:ENABle <enable mask>					
:ENABle?					
TRIGGER (and Related Commands) Subsystem:				**8-7**	
:TRIGger				**8-7**	
[:SEQuence]				**8-8**	
:COUNt <trigger count>	2^{31}-1 (~ 2 billion)	1	1	**8-8**	
:DELay <trigger delay>	10.0	0.0	0.0	**8-9**	resolution = 0.000001 = 1 microsecond
:SOURce { BUS | HOLD | IMMediate | EXT }			IMMediate	**8-8**	
:IMMediate				**8-11**	
:OUTPut					
:DELay <output trigger delay time>	10.0	0.0	0.0	**8-12**	resolution = 0.000001 = 1 microsecond
:TRIGger				**8-12**	
[:STATe] { ON | OFF | 1 | 0 }				**8-12**	
:INITiate				**8-9**	

Table 6-1, Continued

Command	Max	Min	Default	See page	Notes

:IMMediate ON | OFF | 1 | 0 (Command Only)					
:CONTinuous ON | OFF | 1 | 0 (Command Only)					
:ABORt					
DIGITAL Subsystem:				**7-4**	
:DIGital*				**7-4**	
:STATe <port list> , { ON | OFF | 1 | 0 }				**7-4**	
:CONFigure <port>				**7-4**	
:OUTPut				**7-5**	
[:DATA] <port list> , <data list>	255	0	n/a	**7-5**	maximum / minimum for each data byte
:INPut? <port list>				**7-6**	
:SYNChronous				**7-7**	maximum / minimum for each data byte
:DATA <port> , <data list>	255	0	n/a	**7-9**	
:DATA? <port>				**7-15**	
:INDex <port list> , <index>				**7-10**	
:INDex? <port list>				**7-15**	
:POINts <port> , <number of points>	256	0	0	**7-17**	
:POINts? <port>				**7-17**	
:CLEar <port list> (Command Only)				**7-16**	
:CLOCk				**7-16**	
[:POLarity] <module list> , { NORMal | INVerted }					
:BUSY				**7-16**	
[:POLarity] <module list> , { NORMal | INVerted }					
IEEE-488.2 Common Commands:				**5-6**	
*IDN? (Query Only)				**5-17**	
*RST (Command Only)				**5-17**	
*TST? (Query Only)				**5-19**	
*CLS (Command Only)					
*ESE <register value>	255	0	0	**5-14**	
*ESE?				**5-15**	
*ESR? (Query Only)				**5-15**	
*SPE				**5-20**	
*SRE <register value>	255	0	0	**5-15**	
*SRE?				**5-16**	
*STB? (Query Only)				**5-16**	
*OPC				**5-16**	
*OPC?				**5-16**	
*TRG (Command Only)				**5-21**	
*SAV [<state>] (Command Only)	0	100	100	**5-19**	
*RCL [<state>] (Command Only)	0	100	100	**5-19**	
*OPT? (Query Only)					
*WAI				**5-21**	no function in the 1256

This page was left intentionally blank.

Chapter 7
Digital Module Operation

General

The 1256 may be used to control the 1260-114TTL, 1260-114CMOS, 1260-114OC, and 1260-114HVOC digital modules. These modules may be programmed to work in either the **synchronous** or **asynchronous** modes of operation. The modules may be programmed to operate in a combination of the two modes, with some ports operating in the asynchronous mode and some ports operating in the synchronous mode.

Digital Module Ports

The 1260-114 modules are programmed at the “port” level. A port is a set of 8 consecutive input/output pins. Data is written to the ports in 8-bit bytes. The 1260-114TTL, 1260-114CMOS, and 1260-114OC modules consist of twelve 8-bit ports, for a total of 96 input/output pins. In the case of the 1260-114HVOC card, it consists of six 8-bit ports.

For the TTL/CMOS versions of the 1260-114, each port may be defined as an input or an output. All 8 pins of an output port will be enabled to output data when the port is enabled. Data may be read from an output port, but data written to an input port will not appear on the signal pins.

The open-collector versions of the 1260-114 are generally thought of as output only modules. If an output bit is turned off (high-impedance state) such that the 1260-114 is not actively pulling a signal low, reading the port will return the present signal level. It is possible, therefore, to determine whether something else is driving the system and whether the signal is either high or low when referenced to CMOS levels.

Asynchronous Digital Operation

When the 1260-114 modules are programmed to operate in the asynchronous mode, data is written to the output pins immediately upon receipt of the DIGITAL:OUTPUT command. Likewise, the data is read from the input pins immediately upon receipt of the DIGITAL:INPUT? command.

Synchronous Digital Operation

When the 1260-114 modules are programmed to operate in the synchronous mode, data will be clocked in and out of the module by using the CLKIN line on the module's edge connector. To output data in the synchronous mode, one must:

A) Enable the drivers on the ports.

B) Tell the 1256 which ports of the module are to be used as synchronous ports.

C) Load data into the memory of the 1256.

D) Arm the 1260-114 module.

E) Generate TTL-level pulses on the CLKIN pin of the module. One data pattern is written per clock pulse.

To input data in the synchronous mode, one must:

A) Disable the drivers on the ports.

B) Tell the 1256 which ports of the module are to be used as synchronous ports.

C) Tell the 1256 how many data bytes will be read from the ports.

D) Arm the 1260-114 module.

E) Generate TTL-level pulses on the CLKIN pin of the module. One data pattern is read per clock pulse.

F) When the number of bytes specified in step (C) has been captured, the 1260-114 automatically disarms itself.

G) Read the number of data bytes captured from the 1256.

Mixing Synchronous and Asynchronous Modes of Operation

Both synchronous and asynchronous modes of operation may be used on a 1260-114 module simultaneously. The following restrictions apply:

A) You cannot use the asynchronous read command DIGITAL:INPUT? to read data from a synchronous port.

B) All synchronous ports on a module must be contiguous. The synchronous ports are always the lower numbered ports on the module. The synchronous ports begin with port 0 and go through port N. The asynchronous ports begin with port (N+1), and go through port 11. For example, if the last

synchronous port is 7, then ports 0 through 7 will use the synchronous mode, and ports 8 through 11 will use the asynchronous mode.

The following paragraphs describe the commands used to operate the 1260-114 modules.

Specifying Ports on a Digital Module

When a <port> is shown on the command syntax, it represents the combination of the module address and the port number. The syntax for a <port> is:

(@ <module address> (<port number>))

where <module address> is a number in the range 1 through 12, or a module name defined using the "MODULE:DEFINE" command. This selects the 1260-114 or 1260-114C module.

The <port number> is a value in the range 0 through 11. This selects which of the 12 ports on the module will be used.

Some commands accept a more general form of specifying multiple ports. This is shown in the command syntax as a <port list>. A <port list> uses the format:

(@ <module address> (<port range>)
[, <module address> (<port range>)])

where the <module address> is a number in the range 1 through 12, or a module name. The <port range> is a sequence of port specifies, separated by commas.

A <port range> may be a:

<port number>

where a single port number is specified. The port number is in the range 0 to 11. Or:

<port1> , <port2>

where two individual ports are specified. Each of these is a number in the range 0 to 11. Or:

<port1> : <port2>

which specifies a range of inclusive ports. All ports between <port1> and <port2> are included. Each of <port1> and <port2> is a number in the range 0 to 11.

Examples of <port list> are shown below.

Port 0 on module 7:

```
(@7(0))
```

Ports 0 and 11 on module 7:

```
(@7(0,11))
```

Ports 0 through 11 on module 7:

```
(@7(0:11))
```

Port 0 on module 7; port 4 on module 8:

```
(@7(0),8(4))
```

Ports 0, 2 and 3 on module 7; ports 7 through 11 on module 8:

```
(@7(0,2,3),8(7:11))
```

Selecting the Mode of Operation

The mode of operation may be selected using the DIGITAL:CONFIGURE command. This command has the following syntax:

```
:DIGital
    :CONFigure <port>
```

Examples of this command are shown below:

To set all ports of the 1260-114 module with address 7 to the asynchronous mode of operation:

```
DIG:CONF (@7(0))
```

To set ports 0 through 3 to the synchronous mode, and ports 4 through 11 to the asynchronous mode:

```
DIG:CONF (@7(12)): DIG:CONF (@7(4))
```

To read the presently selected mode of operation for port 10 on module 7, issue the following command. The reply returns “SYNC” if it is synchronous, or “ASYN” if it is asynchronous:

```
DIG:CONF? (@7(10))
```

Enabling and Disabling the Ports

The DIGITAL:OUTPUT:STATE command is used to enable or disable the output drivers on one or more ports of the 1260-114 module. **The 1260-114OC and 1260-114HVOC modules have open collector outputs, and so this command has no effect**. This command affects the ports used in either the synchronous or the asynchronous mode of operation.

The command uses the syntax:

```
:DIGital
   :OUTPut
      :STATe <port list> , { ON | OFF | 1 | 0 }
```

Examples of the command are shown below.

Enable port 11 of module 7 as an output:

```
DIG:OUTP:STATE (@7(11)),ON
```

Disable ports 3 and 6 of module 7:

```
DIG:OUTP:STATE (@7(3,6)),OFF
```

Enable port 0 of module 7 and port 3 of module 8:

```
DIG:OUTP:STATE (@7(0),8(3)),ON
```

Enable ports 1 through 4, and port 6 of module 7, and ports 2 through 10 of module 8:

```
DIG:OUTP:STATE (@7(1:4,6),8(2:10)),1
```

Disable port 3 of the module whose name is “digio”:

```
DIG:OUTP:STATE (@digio(3)),off
```

Using the Asynchronous Mode of Operation

The syntax for the asynchronous mode commands is shown below:

```
:DIGital
    :OUTPut
        [:DATA] <port list> , <data list>
            :INPut? <port list>
```

The DIGITAL:OUTPUT command is used to output data to one or more ports on the 1260-114 modules.

The <port list> defines which port or ports to be written.

The <data list> is a list of comma-separated numeric values, each

of which is in the range 0 to 255. The number of data bytes in the <data list> must agree with the number of ports specified in the <port list>.

Examples of this command are shown below.

Output the data value 38 decimal to port 0 of module 7:

```
:DIG:OUTP (@7(0)),38
```

Output data value 55 hex (= 85 decimal) to port 0 and AA hex (= 170 decimal) to port 3 of module 7:

```
:DIG:OUTP (@7(0,3)),#H55,#HAA
```

Output the value 0 to port 0, 1 to port 1, 2 to port 2, and so on. Data is output to each of the 12 ports on the module whose name has been defined as “digio”:

```
:DIG:OUTPUT(@digio(0:11)),0,1,2,3,4,5,6,
      7,8,9,10,11
```

Output the value 1 to port 5 of module 7; output 2 to port 11 of module 7, and output 3 to port 4 of module 8:

```
DIGITAL:OUTPUT (@7(5,11),8(4)),1,2,3
```

The DIGITAL:INPUT? query is used to read data from the port. The <port list> specifies which ports will be read. The reply to this query is a comma-separated list of values, one per port. Data are returned in decimal values.

Examples of this query are shown below.

Read data from port 4 of module 7, and return the data in the reply. An example of the reply is “37":

```
DIG:INPUT? (@7(4))
```

Read data from ports 1 through 6 (in reverse order) from the 1260-114 or 1260-114C with module address 11:

```
DIGITAL:INPUT?  (@11(6:1))
```

An example reply is:

```
38,44,255,0,94,77
```

which indicates that port 6 read a value of 38, port 5 read a value of 44, port 4 read a value of 255, and so on.

When data is read from an enabled output port, the data should equal the last value written to that port using the DIGITAL:OUTPUT command.

Using the

The syntax for the synchronous mode commands is shown below:

Synchronous Mode of Operation

```
:DIGital
     :SYNChronous
          :STATe <module list> ,
               { ON | OFF |1| 0 }
          :DATA <port> , <data list>
          :DATA? <port>
          :INDex <port list> , <index>
          :INDex? <port list>
          :POINts <port> ,
               <number of points>
          :POINts? <port>
          :CLEar <port list>
          :CLOCk
               [:POLarity] <module list> ,
                    { NORMal | INVerted }
          :CLOCk
               [:POLarity]? <module list>
          :BUSY
               [:POLarity] <module list> ,
                    {NORMal | INVerted}
          :BUSY?
               [:POLarity]? <module list>
```

The sequence for using the 1260-114 modules is as follows:

A) Use the DIGITAL:CONFIGURE command to select which port(s) will be used in the synchronous mode of operation, and which will be used in the asynchronous mode of operation.

B) For each port to be used as a synchronous port, determine if the port will be used as a synchronous input port or a synchronous output port.

C) For each Synchronous Output Port:

   1) Use the DIGITAL:STATE command to enable the port.

   2) Use the DIGITAL:SYNCHRONOUS:DATA command to load the data bytes which will be clocked out.

D) For each Synchronous Input Port:

   1) Use the DIGITAL:STATE command to disable the port.

   2) Use the DIGITAL:SYNCHRONOUS:POINTS command to define the number of data points that will be acquired and stored for the port.

E) Arm the 1260-114 module by using the DIGITAL:SYNCHRONOUS:STATE command.

F) Generate pulses on the CLKIN line of the 1260-114 module. The 1260-114 module will disarm itself when the number of pulses is equal to, or greater than, the maximum of:

1) The greatest number of data points downloaded to any synchronous output port on the module (equals the value returned by DIGITAL:SYNC:INDEX?).

2) The greatest number of data points defined for any synchronous input port on the module (equals the value returned by DIGITAL:SYNC:POINTS?).

All data operations are complete when the module disarms itself. The DIGITAL:SYNCHRONOUS:STATE? query may be used to determine if the specified module(s) are armed or disarmed.

Setting Up the Synchronous Test

The DIGITAL:SYNCHRONOUS:DATA command is used to load data into the memory of the 1260-114. This defines the data that will be clocked out, one byte per CLKIN pulse.

Each synchronous output port may hold up to 256 data bytes. Each byte may have a value between 0 and 255. Each time the DIGITAL:SYNCHRONOUS:DATA command is executed, new data bytes are added to any data that was previously downloaded.

For example, the commands:

```
DIG:SYNC:DATA (@7(4)),10,11,12

DIG:SYNC:DATA (@7(4)),50,60,70,80,90

DIG:SYNC:DATA (@7(4),255,254
```

are equivalent to the command:

```
DIG:SYNC:DATA(@7(4)),10,11,12,50,60,70,80,
      90,255,254
```

The DIGITAL:SYNCHRONOUS:INDEX command provides flexibility for loading new data to output. This command sets the location at which the next DIG:SYNC:DATA command will download.

The value passed as the <index> for this command must satisfy the following conditions:

Minimum value: 0

Maximum value: 255, but less than, or equal to, the number of data points loaded

Figures 7-1 (A-D) show the interaction between the DIG:SYNC:DATA command and the DIG:SYNC:INDEX command.

After power-on, or a *RST command, no data is loaded into any output port buffer of a 1260-114. The index for each port points to the first location in the buffer.

After executing the command:

```
DIG:SYNC:DATA (@7(4)),10,20,30,40
```

the memory buffer for port 4 of module 7 is filled with data as depicted in **Figure 7-1 (A)**. There are four bytes loaded. The index for this port points to the next available position in the port buffer, shown as location 4 in the figure.

After executing the command:

```
DIG:SYNC:DATA (@7(4)),50,60,70,80,90
```

the memory buffer for port 4 of module 7 tracks that depicted in **Figure 7-1 (B)**. There are now a total of nine bytes loaded. The index for this port now points to the next available position in the buffer.

After executing the command:

```
DIG:SYNC:INDEX (@7(4)),3
```

the index for the port buffer now points to the fourth entry in the buffer. This is where the next data will be loaded. This is depicted in **Figure 7-1 (C)**.

After executing the command:

```
DIG:SYNC:DATA (@7(4)),77,78
```

the fourth and fifth data bytes are overwritten. The previous data (40 and 50) is replaced by the new data (77 and 78). The index now points to the sixth entry, which has a value of 60. The next DIG:SYNC:DATA command would begin by overwriting this data. This final configuration is depicted in **Figure 7-1 (D)**.

Figure 7-1 (A), Port 4 Data after DIG:SYNC:DATA (@7(4)),10,20,30,40

Location	Data
9	
8	
7	
6	
5	
Index -> 4	
3	40
2	30
1	20
0	10

Figure 7-1 (B), Port 4 Data after DIG:SYNC:DATA (@7(4)),50,60,70,80,90

Location	Data
Index -> 9	
8	90
7	80
6	70
5	60
4	50
3	40
2	30
1	20
0	10

Figure 7-1 (C), Port 4 Data after DIG:SYNC:INDEX (@7(4)),3

Location	Data
9	
8	90
7	80
6	70
5	60
4	50
Index -> 3	40
2	30
1	20
0	10

Figure 7-1 (D), Port 4 Data after DIG:SYNC:DATA (@7(4)),77,78

Location	Data
9	
8	90
7	80
6	70
Index -> 5	60
4	78
3	77
2	30
1	20
0	10

The query form of the command may be used to read the present index location for the specified port. For example, the command:

```
DIGITAL:SYNC:INDEX? (@7(4))
```

would return the reply:

```
5
```

The DIGITAL:SYNCHRONOUS:DATA? query may be used to read the data that has been previously loaded to a synchronous output port. This command will return all of the data bytes (up to 256) downloaded to the specified port.

The DIGITAL:SYNCHRONOUS:POINTS command is used to define the maximum number of data bytes that will be acquired for the specified synchronous input port. This tells the 1256 how many data bytes to collect for the specified port(s). The value must be between 0 and 256.

The DIGITAL:SYNCHRONOUS:POINTS? query may be used to read the number of points defined for the synchronous input port(s) specified.

After power-up, and after executing a *RST command, the number of points to acquire for all ports is set to 0.

The DIGITAL:SYNCHRONOUS:INDEX? query may be used to determine the number of data bytes stored in a synchronous input port's buffer. This number will always be less than or equal to the maximum number of points for the port as specified with the DIGITAL:SYNCHRONOUS:POINTS? command.

Arming the Digital Modules

The DIGITAL:SYNCHRONOUS:STATE command is used to "arm" one or more 1260-114 digital modules. These modules must be "armed" before they will respond to CLKIN pulses and perform the synchronous operations.

The command accepts one or more module numbers or module names. Note that entire modules, not individual ports, are armed using this command. The following examples illustrate the use of the command.

Arm the digital module at module address 7:

```
DIGITAL:SYNC:STATE (@7),ON
```

Arm the digital modules at module addresses 3, 5, and 9:

```
DIG:SYNC:STATE (@3,5,9),ON
```

Arm the digital modules at module addresses 3, 4, 5, 6, 9, and 11:

```
DIG:SYNC:STATE (@3:6,11,9),ON
```

After the module has been armed, it will remain armed until:

A) The DIGITAL:SYNCHRONOUS:STATE command is used to disarm the module; OR:

B) The number of data bytes collected for all input ports matches the number of points defined for the port (DIGITAL:SYNC:POINTS command), and the number of data bytes output for each output port matches the number of data bytes loaded (DIGITAL:SYNC:DATA command).

One byte will be clocked out for each output port on the module for each CLKIN pulse received. One byte will be read for each input port on the module for each CLKIN pulse received.

Checking for Data Transfer Completion

The DIGITAL:SYNCHRONOUS:STATE? query is used to read whether the digital module is armed. When all of the data transfer has been completed for a given module, the module will automatically be disarmed.

The following examples illustrate the use of this query.

This queries the present arm state of module 7. The reply will be “1" if the module is armed, or “0" if the module is disarmed:

```
DIG:SYNC:STATE? (@7)
```

This queries the present arm state of each of the following modules: 7, 8, 9, and 10:

```
DIGITAL:SYNCHRONOUS:STATE? (@7:10)
```

The reply consists of a single “1" or “0" for each port. Each state indication is separated by a comma. For example, the reply:

```
0,1,0,0
```

indicates that module 8 is armed, while modules 7, 9, and 10 are not.

Reading Data from a Synchronous Input Port

Once the data transfer is complete, each of the synchronous input ports may be read. The DIGITAL:SYNCHRONOUS:DATA? query is used to read data from the synchronous input ports.

The following examples illustrate the use of this query.

Reads all of the data from port 4 of module 7:

```
DIG:SYNC:DATA? (@7(4))
```

The reply to this query consists of a series of numeric values, each separated by a comma. For example, the following reply could be returned. This indicates that seven data bytes were stored by the module from port 4:

```
255,0,128,93,66,17,23
```

Reads all of the data from port 11 of the module whose name is “digio”. The name must have been defined using the MODULE:DEFINE command:

```
DIG:SYNC:DATA? (@digio(11))
```

The number of points stored in the port memory for a synchronous input port may be read using the DIGITAL:SYNCHRONOUS:INDEX? query. This query will return the number of data points stored for the specified port(s). One numeric reply is returned for each port specified in the query.

If multiple ports are specified, then multiple numeric values are returned. In this case, each number is separated from the next by a comma. Examples are shown below.

Reads the index for port 4 of module 7:

```
DIG:SYNC:INDEX? (@7(4))
```

Reads the index for each of the ports 3, 4, and 5 on module 7.

```
DIGITAL:SYNC:INDEX? (@7(3:5))
```

The reply will be similar to:

```
123,14,79
```

This reply indicates that there are 123 bytes stored in port 3, 14 bytes stored in port 4, and 79 bytes stored in port 5.

Clearing Data from Synchronous Input and Output Ports

Data may be cleared from all synchronous input and output ports with the DIGITAL:SYNCHRONOUS:CLEAR command. This command does the following:

A) Clears any data loaded into a synchronous output port.

B) Clears any data read into a synchronous input port.

C) Sets the Index for the output and input ports to 0.

Examples of the command are shown below.

Clears module 7 synchronous input and output ports:

```
DIG:SYNC:CLEAR (@7)
```

Clears modules 7, 11, and the module assigned the name “DIGIO”.

```
DIG:SYNC:CLEAR (@7,DIGIO,11)
```

Synchronous Control and Status Pins

The CLKIN input pin of the 1260-114 is used to clock synchronous data into and out of the module. Each time a clock edge is presented to this input, the module generates an interrupt to the 1256.

The 1256 responds to this interrupt by writing the next data byte from each synchronous output port, and reading a new data byte from each synchronous input port.

The BUSY status line on the 1260-114 module is asserted until the 1256 is ready to accept a new CLKIN line. Any new CLKIN pulses received while the BUSY line is asserted will be ignored.

The CLKIN signal is edge sensitive. The active edge is positive at power-up and after a *RST command. This may be set to the negative (falling) edge by using the command:

```
DIG:SYNC:CLOCK INVERTED
```

command. It may be set back to the positive (rising) edge by using the command:

```
DIG:SYNC:CLOCK NORMAL
```

The BUSY signal is active high after power-on and after a *RST command. This may be set to be active low by sending the command:

```
DIG:SYNC:BUSY INVERTED
```

and may be set back to active high using the command:

```
DIG:SYNC:BUSY NORMAL
```

Synchronous and Asynchronous Example

To illustrate a command sequence, assume module 7 is a 1260-114 module. The following command sequence may be used to:

A) Select ports 0 and 1 as synchronous output ports.

B) Select ports 2 and 3 as synchronous input ports.

C) Select ports 4, 5, 8, and 9 as asynchronous output ports.

D) Select ports 6, 7, 10, and 11 as asynchronous input ports.

E) Output values 1, 2, 3, 4 to port 0 in synchronous mode.

F) Output values 10, 20, 30 to port 1 in synchronous mode.

G) Read 5 data bytes from port 2 in synchronous mode.

H) Read 256 data bytes from port 3 in synchronous mode.

I) Output the value 44 to port 4, 55 to port 5, 88 to port 8, and 99 to port 9 in asynchronous mode.

J) Input a byte from ports 6, 7, 10, and 11 in asynchronous mode.

Resets the 1260-114 modules to power-up defaults:

```
*RST
```

Enable ports 0, 1, 4, 5, 8, and 9 as output ports. Others are defaulted (by *RST) as input ports:

```
DIG:STATE (@7(0,1,4,5,8,9)),ON
```

Sets ports 0 to 3 as synchronous, ports 4 to 11 as asynchronous:

```
DIG:CONF (@7(4))
```

Loads the data bytes 1, 2, 3, and 4 into port 0 buffer:

```
DIG:SYNC:DATA (@7(0)),1,2,3,4
```

Loads the data bytes 10, 20, and 30 into port 1 buffer:

```
DIG:SYNC:DATA (@7(1)),10,20,30
```

Set the number of points to 5 for port 2:

```
DIG:SYNC:POINTS (@7(2)),5
```

Set the number of points to 256 for port 3:

```
DIG:SYNC:POINTS (@7(3)),256
```

Outputs 44 to port 4, 55 to port 5, 88 to port 8, and 99 to port 9:

```
DIG:OUTPUT (@7(4,5,8,9)),44,55,88,99
```

Commands the 1260-114 to read the data. The reply will be a set of 4 data byte values, in decimal format, separated by commas:

```
DIG:INPUT? (@7(6,7,10,11))
```

Enables the synchronous mode for module #7:

```
DIG:SYNC:STATE (@7),ON
```

After this operation is complete, generate 256 clock pulses on the CLKIN input line of module #7.

Reads whether module #7 is still armed. If the reply is “1", then the module is still armed, expecting more CLKIN pulses. If the reply is “0", then the module is disarmed:

```
DIG:SYNC:STATE? (@7)
```

Read the data bytes captured by port 2 during the synchronous test. The reply will be a set of 5 data bytes, in decimal format, separated by commas:

```
DIG:SYNC:DATA? (@7(2))
```

Read the data bytes captured by port 3 during the synchronous test. The reply will be a set of 256 data bytes, in decimal format, separated by commas:

```
DIG:SYNC:DATA? (@7(3))
```

This page was left intentionally blank.

Chapter 8
Advanced SCPI Operations

Introduction

The 1256 Switching System offers a number of advanced functions to give you precise control over the instrument. These functions can be accessed only via the IEEE-488 (GPIB) and RS-232 interfaces.

Using ROUTE:CONFIGURE to Control Switching Sequence

The ROUTE:CONFIGURE command controls the sequence in which relays are opened and closed when more than one relay has been commanded to change. The command has the form:

```
ROUTE:CONFIGURE (@<module address>),<seqmode>
```

The <seqmode> parameter may be one of three modes:

- **IMM (Immediate):** Causes channels (relays) to be opened and closed as quickly as possible. The sequence of opening and closing is indeterminate. In this mode, care must be taken to avoid damaging equipment while the relays are in transition, i.e. by short circuiting power supplies, connecting high-power outputs together, or applying excessive power to low-impedance inputs.
- **BBM (Break Before Make):** Causes existing channel closures to be opened before new channel closures are made. BBM is the default mode on power-up, or after the 1255A has been reset.
- **MBB (Make Before Break):** Causes new channel closures to be made before existing channel closures are opened.

For example, the following command:

```
ROUTE:CONFIGURE (@4),MBB
```

causes switch module 4 (the switch plug-in installed in slot 4) to operate in Make-Before-Break mode.

Include Lists

The 1256 provides the capability to define sets of relays which operate together. This feature is called an “include list”. When a relay on an include list is closed, all relays on that include list are closed. Likewise, when a relay on an include list is opened, all relays on that include list are opened.

A channel may reside on at most one include list. An attempt to place a relay on a second include list results in an error. The error is placed on the error queue and may be read using the SYSTEM:ERROR? query.

The syntax for defining an include list is shown below:

```
[:ROUTe]
      :INCLude <channel list>
      :INCLude? [ <channel list> ]
            :DELete <channel list>
                  :ALL
```

The following examples illustrate the use of this command to define include lists:

```
INCLUDE (@3(5,15))

CLOSE (@3(5))
```

This first command places two relays on the include list. Channels 5 and 15 of the module with address 3 are included together. The second command closes channels 5 AND 15 on the module with address 3.

```
INCLUDE (@3(12),8(0))

OPEN (@8(0))
```

The first command places channel 12 from module 3 and channel 0 from module 8 on an include list. The second command opens both channel 6 from module 3 and channel 0 from module 8.

```
INCLUDE (@3(5,16:19),7(0:5),6(9:12,17))
```

This command groups the following relays on a single include list:

Module 3, channels 5, 16, 17, 18, and 19

Module 7, channels 0, 1, 2, 3, 4, and 5

Module 6, channels 9, 10, 11, 12, and 17

Any command which opens any of these relays will open all of them; and any command which closes any of these relays will close all of them.

```
MOD:DEF power,3
MOD:DEF matrix,5
PATH:DEF thispath,(@8(0,4,12),power(14))
INCL (@power(15),matrix(323),thispath)
```

The first three commands define module names and a path. The last command shows that the INCLUDE command can use these definitions in an include list.

The include list relationship for a particular relay, or multiple relays, may be removed by using the INCLUDE:DELETE command. The following examples illustrate the use of the INCLUDE:DELETE command:

```
INCLUDE (@3(0:4))

INCLUDE:DELETE (@3(2))
```

The first command defines an include list consisting of channels 0 through 4 on module 3. The second command removes channel 2 from any include list definitions. After the second command is executed, the include list now consists of channels 0, 1, 3 and 4. These two commands are thus equivalent to a single command INCLUDE (@3(0,1,3,4)).

```
INCLUDE (@1(0:19),2(0:19))

INCL:DEL (@1(5:8),2(11,15,17))
```

The first command defines an include list of channels 0 through 19 on both modules 1 and 2. The second command removes include list association for channels 5 through 8 on module 1 and channels 11, 15, and 17 on module 2. These two commands are equivalent to the command:

```
INCLUDE (@1(0:4,9:19),2(0:10,12:14,16,18,19))

INCL:DELETE:ALL
```

All include list definitions are deleted.

The include list association for a single channel, for multiple channels, or for all channels in the system may be checked using the INCLUDE? query.

For the remainder of this discussion of the INCL? query, assume the following commands have been executed:

```
INCL (@1(0),2(0),4(0))
INCL (@2(7:10))
INCL (@3(16,19))
INCL (@1(3,5))
INCL (@4(1:4,14,23))
```

These commands define five distinct include groups.

The reply to the query be one of the following:

A) If no include groups are defined for any of the channels passed as a parameter, the reply will be a single ASCII linefeed character. For example, if the query:

```
INCL? (@1(15))
```

is specified, the reply will be a single linefeed character, since channel 15 of module 1 has not been placed on an include list.

B) If all channels specified in the query reside on a single include group, the reply will be a single channel list, and terminated with an ASCII linefeed character. For example, the command:

```
INCL? (@2(0))
```

returns the reply:

```
(@1(0),2(0),4(0))
```

Note that the reply maintains the order in which the include group was defined. That is, the reply begins with channel “1(0)”, even though the query was sent for channel “2(0)”.

C) If multiple channels are specified in the query, and the channels reside on different include lists, then multiple lists will be returned in the reply. The lists will be separated from each other by a comma. For example, the command:

```
INCL? (@1(0:10),2(0:10),3(0:10))
```

returns the reply:

```
(@1(0),2(0),4(0)),(@1(3,5))
```

and the reply:

```
INCL?
```

returns the reply:

```
(@1(0),2(0),4(0),(@1(3,5)),(@2(7:10)),
(@3(16,19)),(@4(1:4,14,23))
```

The INCLUDE? query with no parameters passed returns all presently defined include groups. Each group is separated from each other by a comma. Note that if there are many include lists defined, the query may result in a reply which is longer than the length of the output reply buffer (1024 characters). In this case, the "Query Interrupted" error will be placed on the error queue and the output reply buffer will be cleared. To avoid this, always specify a channel list with this query.

The reply to the INCLUDE? query does NOT contain module names, nor does it contain path names. In addition, when 3 or more consecutive channels are defined in the include list, the reply will represent the channels as a range of channels, with a colon (:) between the first and last channels in the range.

For example, if the following commands are executed:

```
MODULE:DEFINE matrix,1
MODULE:DEFINE power,2
INCLUDE (@matrix(14,103,104,105,106),power
      (3:7,12,16,17,18))
```

then the query:

```
INCLUDE? (@matrix(105))
```

returns the reply:

```
(@1(14,103:106),2(3:7,12,16:18))
```

After power-up, and after a "*RST" command has been executed, all INCLUDE lists are deleted.

Using Exclude Lists

The 1256 provides the capability to define sets of relays which are "mutually exclusive". This feature is called an "exclude list". When one relay on an exclude list is closed, all others in the exclude list are opened. This prevents two relays in an exclude list from being closed simultaneously.

This feature can be used to ensure two or more relays are not shorting system resources, such as power supplies, together.

A channel may reside on at most one exclude list. An attempt to place a channel on a second exclude list results in an error. The error is placed on the error queue and may be read using the SYSTEM:ERROR? query.

A channel cannot be on an include list with another channel if it is also on an exclude list with that second channel. That is, the following definitions result in an error:

```
INCLUDE:DEF (@1(0:10))
EXCLUDE:DEF (@1(0,11:15,6))
```

The error occurs because channels 1(0) and 1(6) are on an include list together and they are on an exclude list together. This creates a conflict since an attempt to close channel 1(0) would attempt to close channel 1(6) due to the include list association, but would be prevented from doing so by the exclude list.

The syntax for defining an exclude list is shown below:

```
[:ROUTe]
      :EXCLude <channel list>
      :EXCLude? [ <channel list> ]
            :DELete <channel list>
                  :ALL
```

The following examples illustrate the use of this command to define exclude lists:

```
EXCLUDE (@1(0:19),2(0:19))
CLOSE (@1(0))
CLOSE (@2(11))
CLOSE (@1(15,17))
```

The first command establishes an exclude list of channels 0 through 19 on module 1 and channels 0 through 19 on module 2.

The second command closes channel 0 on module 1.

The third command closes channel 11 on module 2. This command will cause channel 0 on module 1 to open, since both channels are on the exclude group together.

The fourth command closes channel 17 on module 2. Channel 15 is never closed, since it is excluded by channel 17. Channel 11 on module 2 is opened prior to closing channel 17, since channels 11 and 17 are on the exclude group together.

The INCLUDE and EXCLUDE relationships of channels can cause multiple closures and openings when commanding a single channel to close. For example, assume the following commands have been executed:

```
INCLUDE (@1(0:5,10,12))
INCLUDE (@1(13:19))
EXCLUDE (@1(0,13)
EXCLUDE (@1(1,14))
EXCLUDE (@1(2,15))
CLOSE (@1(0))
```

After these commands have been executed, channel 0 of module 1 is closed. Also, channels 1 through 4, 10 and 12 are closed, since these are on the same include list as channel 0.

Now, if the command:

```
CLOSE (@1(13))
```

is executed, the following actions occur:

A) Channel 0 is opened since it is on an exclude list with channel 13

B) Channels 1, 2, 3, 4, 10, and 12 are opened since they are on an include list with channel 0

C) Channel 13 is then closed.

After power-up, and after a "*RST" command has been executed, all EXCLUDE lists are deleted.

Using External Triggering

You may provide an *External Trigger In* signal to control when the 1256 closes or opens relays. Also, the 1256 can provide an *External Trigger Out* signal to tell your equipment when the relays have settled after opening or closing.

These signals connect to the *External Trigger In* and *External Trigger Out* connectors on the 1256 rear panel. Refer to **Figure 7-1, Using *External Trigger In* and *External Trigger Out*** for connector location and signal information.

Selecting the Trigger Source

The TRIGGER:SOURCE command selects which source is used to trigger the scan list. Each time a trigger is received from the selected trigger source, the 1256 advances to the next element on the scan list.

The syntax for the TRIGGER:SOURCE command is:

```
:TRIGger
     [ :SEQuence ]
          :SOURce  { BUS | HOLD |
               IMMediate | EXT }
```

The possible trigger sources are:

BUS	The GPIB bus trigger, or the "*TRG" command, advances to the next scan list element.
HOLD	Triggers are ignored, and the scan list will not advance.
IMMediate	The scan list, once initiated, advances as fast as possible after considering the relay settling time, trigger input delay, and trigger output delay. The

Scan List operation is initiated by sending an INIT:IMMEDIATE command (or INIT:CONTINUOUS command).

EXT — This selects the 1256 external trigger input line. Each time a low-going pulse is received on the line, the scan list advances to the next element.

After power-up, and after a *RST command, the trigger source is set to IMMEDIATE.

The present trigger source may be queried using the TRIGGER:SOURCE? query. The reply to this query will be one of the following:

BUS

HOLD

IMM

EXT

Selecting the Trigger Count

The trigger count determines how many elements will be scanned in the scan list. That is, if the trigger count is 10, then the scan list will advance 10 times before halting. Any triggers received from the selected trigger source will be ignored after the trigger count has been satisfied.

The syntax for the TRIGGER:COUNT command is:

```
:TRIGger
     [:SEQuence]
          :COUNt <trigger count>
```

The <trigger count> is a numeric value between 1 and 21747483647 (2^{31}-1).

The present trigger count may be queried by using the TRIGGER:COUNT? query.

Selecting a Trigger Delay

The 1256 may be programmed to delay before acting on a trigger. This provides the ability to slow down the scan list operation if so desired. The TRIGGER:DELAY command is used to program the trigger delay. The syntax for this command is:

```
:TRIGger
      [:SEQuence]
            :DELay <trigger delay>
```

The <trigger delay> is a real number between 0.0 and 10.0. The trigger delay is programmed in seconds, with a resolution of 1 microsecond. This is the **resolution** and not the **accuracy**. That means that you will receive 0.00000 (5 decimal places) when you read it. Any delay over 10 milliseconds rounds the delay to the nearest 10 milliseconds.

When a non-zero trigger delay is programmed, the 1256 performs an “idle wait” for the trigger delay period before acting on the trigger.

The present trigger delay may be queried by using the TRIGGER:DELAY? command.

Arming and Disarming the 1256

Before scanning of channels can be performed, the 1256 must be armed. Arming is accomplished with the INITIATE command. The syntax for the INITIATE command is:

```
:INITiate
      :IMMediate {ON|1|OFF|0}
      :CONTinuous {ON|1|OFF|0}
```

The INITIATE:IMMEDIATE command arms the 1256. This enables the 1256 to accept new triggers and continue scanning. Each time the INIT:IMMEDIATE command is received by the 1256, it enables triggering for the scan list. Each time this command is received, the number of elements which may be scanned is set to the trigger count.

Each time the INIT:IMMEDIATE command is received, scanning resumes from the point at which it was stopped, NOT from the beginning of the scan list. That is, if the commands:

```
SCAN (@1(0:19))
TRIG:COUNT 3
TRIG:SOUR BUS
INIT:IMMEDIATE
```

are executed, the 1256 will accept up to 3 triggers. After the third trigger, scanning will be disabled since the trigger count value has been reached. At this point, channel 2 from module 1 will be closed, because it is the third element in the scan list.

Now if a second INIT:IMMEDIATE command is received, the 1256 will accept up to 3 more triggers. The first trigger received will open channel 2, and close channel 3.

The INIT:CONTINUOUS command also arms the 1256. This command allows the 1256 to continuously scan through the channels in the scan list. The 1256 will NOT stop scanning after the number of triggers equal to the trigger count have been received. This command effectively bypasses the trigger count limit.

The ABORT command disarms the 1256. Once the 1256 receives the ABOR or ABORT command, it will not perform any scan list actions until it is rearmed by using the INIT:IMMEDIATE or INIT:CONTINUOUS commands.

At power-up, and after executing a *RST command, the 1256 is disarmed.

The trigger and arm state diagram is shown in **Figure 8-1**. This diagram shows that the 1256 must be armed before it responds to triggers. Once it is armed, it remains armed and sequences through the scan list until the number of triggers equal to the trigger count is received. After that, it is disarmed and must be rearmed again. The 1256 may also be manually disarmed with the ABORT command.

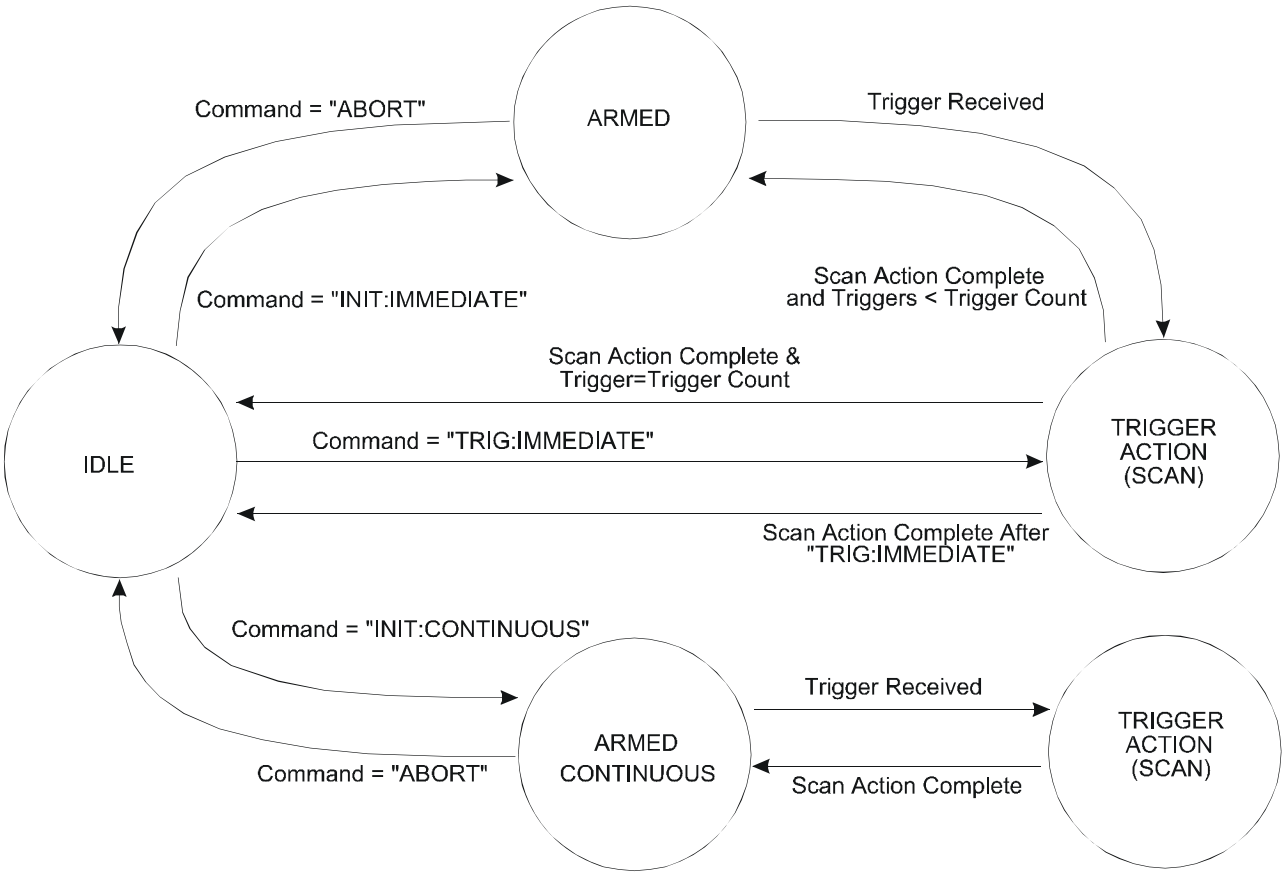


Figure 8-1, State Transition Diagram For Arming and Triggering the 1256

Generating a Single Trigger

The 1256 may be armed, and a single trigger may be sent, using the TRIGGER:IMMEDIATE command. The syntax for this command is:

```
:TRIGger
      [:SEQuence]
            :IMMediate
```

This command is equivalent to the following:

A) Programming the trigger source

B) Arming the 1256 with an INIT:IMMEDIATE command

C) Sending a single trigger on the selected trigger source

This command may be used to execute the next step of a scan list.

Output Trigger Signals from the 1256

The 1256 may be programmed to generate an output trigger pulse each time a relay operation has been completed. The output trigger pulse may be placed on the external trigger output found on the rear panel of the 1256. Refer to **Figure 7-1: Using *External Trigger In* and *External Trigger Out*** for connector location and signal information. This trigger pulse may be used to inform other instruments that the relay(s) have settled, and a new measurement may take place.

Selecting an Output Trigger Destination

The OUTPUT:TRIGGER command is used to enable or disable the external trigger line. The syntax for this command is:

```
:OUTput
      :TRIGger
            [:STATe] { ON | 1 | OFF | 0 }
```

Once the trigger output is enabled, it may be disabled by:

A) Deselecting the line, using the “OFF” parameter.

B) Sending the *RST command.

For example, the following commands may be sent.

To enable the external trigger output:

```
OUTPUT:TRIGGER ON
```

To disable the external trigger output:

```
OUTP:TRIG 0
```

The present destination for the trigger output pulse may be read by

sending each of the following queries:

```
OUTPUT:TRIG?
```

If the trigger output is disabled, the reply to this query will be a "0". If an output trigger is enabled, the reply will be "1".

A delay may also be used to prevent the trigger form getting to the UUT until a certain time has passed.

```
OUTP:DELay <output trigger delay time>
```

This produces a delay from the time the relay closes till the trigger is felt. The trigger is programmed in seconds, with a resolution of 1 microsecond. This is the **resolution** and not the **accuracy**. That means that you will receive 0.00000 (5 decimal places) when you read it. Any delay over 10 milliseconds rounds the delay to the nearest 10 milliseconds.

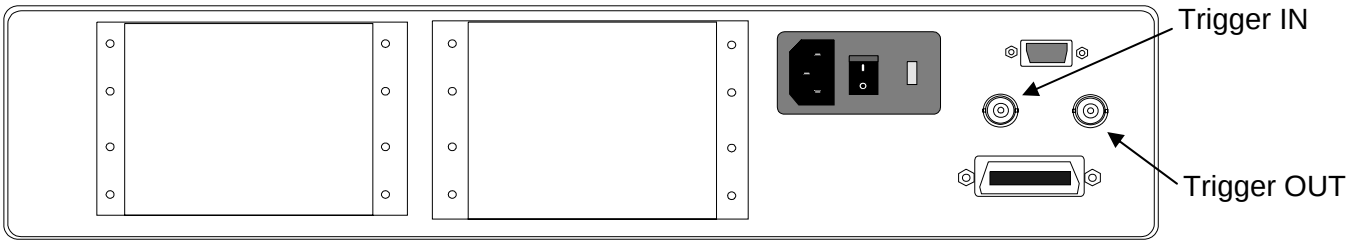


Figure 8-2, Using External Trigger In and External Trigger Out

Using Scan Lists and External Trigger Input

A scan list allows you to specify a list of switching commands to be performed. Once you have defined this list, the 1256 will automatically sequence through the scan list, executing one command each time it detects a trigger pulse at the external trigger input of the SYNC/TRIG connector.

Three types of commands may be entered into a scan list:

- Close a specific relay
- Recall a state stored in a non-volatile memory location
- Break (stop scan list processing)

Scan Lists

The 1256 may be programmed to sequence through a list of channels. This feature is known as a “Scan List”.

When a Scan List is used, the user defines a list of channels to operate. Each time a trigger is received by the 1256, the presently closed channel is opened, and the next channel on the list is closed. This capability may be used in conjunction with a Digital Multimeter (DMM) or other measurement device to allow a group of measurements to be made without having to program the 1256 for each relay operation.

For example, without the Scan List feature, for **each** DMM reading, you must:

A) Program the 1256 to close a channel
B) Wait/Query the 1256 to ensure the channel is closed
C) Program the DMM to take a reading
D) Program the 1256 to open a channel

With the Scan List feature, the DMM and 1256 are programmed to trigger each other for each DMM reading. After this, sending a trigger to the 1256 intiates the cycle of:

Close Relay, Take DMM Reading, Open Relay

automatically. When all of the desired readings have been made, the DMM may be read to obtain the group of measurements. This method usually results in increased measurement rate by minimizing the command processing time of the instruments involved.

To set up the Scan List, the programmer must:

A) Program the 1256 to define the list of channels to scan through, using the SCAN command.
B) Program the 1256 to select an input trigger source. This input trigger source should be the external trigger input. The TRIGGER:SOURCE command is used to select the input trigger.
C) Program the 1256 to generate a trigger each time a relay is closed. The output trigger must be enabled, through the OUTPUT:TRIGGER command.
D) Program the DMM (or other instrument) to select an output trigger source. The output trigger source should be the input trigger source of the 1256, as indicated in step (B) above.
E) Program the DMM to select an input trigger source. The input trigger source should be the output trigger line of the 1256, as indicated in step (C) above.
F) Program the DMM to select a trigger count, which sets the

number of channels to scan through.

G) Program the DMM to select the number of readings to store.

H) Send the command TRIGGER:IMMEDIATE to the 1256 to initiate the Scan operation.

I) Repeatedly query the DMM to wait until the measurements have completed.

J) Read the measurements from the DMM.

The commands used to program the DMM (or other instrument) depend on the instrument being used. Consult the user documentation of the instrument for a description of how to perform steps (D), (E), (G), (J), and (K) above.

The remainder of this section describes how to define a scan list, select input and output trigger sources, and so on.

Defining a Scan List

A Scan List may be defined using the SCAN command. The syntax for the SCAN command is shown below:

```
[:ROUTe]
      :SCAN <scan list>
          :DELete
              [:ALL]
:SCAN?
```

The <scan list> is the same as a <channel list> used with the CLOSE, OPEN, and PATH commands, but with one addition: special keywords may be used to indicate that all channels controlled by the 1256 are placed in the state as recalled from nonvolatile memory.

In short, the <scan list> is comprised of:

A) Individual channel designators. Examples:

```
3(0)
6(37)
1(323)
```

B) A list of single channels, separated by commas:

```
3(0,2,4,6)
7(9,2,1,10)
```

C) A range of relays, separated by a colon:

```
3(1:8)
7(10:2)
```

D) A path name:

```
Path1
Thispath
```

E) A state name. State names begin with the letters “STATE”,

and end with a number between 0 and 100. The following are valid state names:

```
STATE0
STATE7
State53
State100
```

For example, the commands:

```
PATH:DEF example,(@7(0,5,10,13))

     SCAN (@1(323),4(0:2),5(8:5),example,
     1(0),state14,1(224))
```

Define a path name (“example”) and then define a SCAN list. Each time a trigger is received, the next channel, path, or state in the scan list is closed after the previous channel or path is opened.

Using the SCAN command example above, the following sequence of actions occurs:

A) After a trigger is received, channel 323 of module 1 is closed

B) After the next trigger, channel 1(323) is opened, channel 4(0) is closed

C) After the next trigger, channel 4(0) is opened, channel 4(1) is closed

D) After the next trigger, channel 4(1) is opened, channel 4(2) is closed

E) After the next trigger, channel 4(2) is opened, channel 5(8) is closed

F) After the next trigger, channel 5(8) is opened, channel 5(7) is closed

G) After the next trigger, channel 5(7) is opened, channel 5(6) is closed

H) After the next trigger, channel 5(6) is opened, channel 5(5) is closed

I) After the next trigger, channel 5(5) is opened, and all channels defined for the path “example” are closed. (Module 7, channels 0, 5, 10, and 13)

J) After the next trigger, all the channels defined for the path “example” are opened, and channel 1(0) is closed

K) After the next trigger, channel 1(0) is opened, and the state of all channels controlled by the 1256 is recalled from nonvolatile memory. The state of the relays is recalled from state location 14.

L) After the next trigger, channel 1(224) is closed. Note that the channels recalled from nonvolatile memory are **NOT** opened.

M) After the next trigger, channel 1(224) is opened, and channel 1(323) is closed. The scan list has "wrapped around" to the beginning of the list.

Each time the SCAN command is executed, the previously defined scan list is deleted.

The present scan list may be deleted using the command SCAN:DELETE command.

The presently defined scan list may be read using the SCAN? command. For example, if the scan list shown for the example above is presently defined, the query:

```
SCAN?
```

will generate the reply:

```
(@1(323),9(0:2),10(8:5),example,1(0),
state14,1(224))
```

The reply will NOT include module names, even if the SCAN command used to define the list included module names. Module numbers are always used when replying to the SCAN? query.

Advance Verification Features

In some applications, it may be desirable for the reason of reliability to confirm that switches are actually closed or open based on the commanded state of the system. The 1256 provides the user with capabilities to detect certain types of switching failures.

From the standpoint of the 1256, switching failures fall into one of two classes – detectable and undetectable. An example of an undetectable failure is where a relay contact has welded closed through misuse. In this example, the relay coil voltages will verify properly even though the contact has suffered permanent damage. There is no easy way to detect this condition without invasively monitoring the switched-side of the relay. This obviously would defeat the isolation characteristics of a relay.

The other class of failures – those that are detectable – include coil

and driver failures. The 1256 switching system offers two methods for identifying and locating these problems.

Confidence Mode

The 1256 supports an operational mode that performs a high-level system-wide verification. When Confidence Mode is enabled, the 1256 checks the status of all switches after it closes or opens any of them. The syntax for the command is shown below:

```
[:ROUTe]
      :MONitor
          [:STATE] { ON | 1 | OFF | 0 }
```

If any relay is not in the correct state and Confidence Mode is enabled, the 1256 pushes an error message onto the system error queue and asserts the “Execution Error” bit in the *Event Status Register*.

If the 1256 is accessed through the GPIB interface, this condition can further be made to generate an SRQ interrupt. This is accomplished by writing a “1” into the “Execution Error” mask bit of the *Standard Event Status Register* and a “1” into the “ESB” mask bit of the *Service Request Enable Register*.

When Confidence Mode is enabled, commands are executed more slowly because of the time consumed in reading the status of all relays. The amount of slowdown depends on the quantity and types of switch modules that are installed.

Using Relay Verification

The 1256 system also supports the ability to verify that a switch or a range of switches is closed or open. The syntax for these commands is shown below:

```
[:ROUTe]
      :VERify
            :ALL?
            :MASK <channel list>, 0 | 1 | X
            :MASK? <channel list>
```

The 1260 line of switching cards used in the 1256 typically utilizes open-drain or open-collector drivers. A separate signal between the coil of each relay and associated driver feeds back into the 1256 for verification and confidence mode checks.

Verification is somewhat different than confidence mode which is enabled with the MONITOR:STATE command. With confidence mode enabled, the 1256 verifies the state of all relays in a system

against expectations after performing any relay state changes (i.e. open or close commands). The result of this test is a high-level "pass" or "fail" indication for the entire system. Confidence mode therefore has the ability to quickly test a system, but it is unable to resolve or identify the exact cause of the failure if one is detected.

Unlike confidence mode, relay verification can be initiated by the user on a channel by channel basis and can be performed at anytime (not just at relay state changes). To be able to perform relay verification properly, several steps need to be performed:

1. Set the proper verification mask indicating whether the readback channel is normal, inverted, or don't care. A channel's mask defaults to don't care. For relay cards utilizing open-drain drivers (virtually all 1260-series relay cards), the readback signal will be pulled low by the driver when the driver is enabled (high or on state). For relay cards, the verification mask should be manually set to the inverted mode of operation.

2. Issue a verification command for the relays or range of relays of interest.

This process is illustrated with several examples. If the following command was sent:

```
VERIFY:MASK (@2(0:10)),1
```

the 1256 would set the verification masks for plug-in module 2, channels 0 through 10 to "inverted" feedback. The following command would do the same thing except set the same channels to normal (non-inverted) feedback:

```
VERIFY:MASK (@2(0:10)),0
```

If a channel needs to be set to don't care, this is easily accomplished by sending a value of "X" for the mask:

```
VERIFY:MASK (@6(5),7(10:15)),X
```

Assuming in the first example that the plug-in card located in slot 2 had channels 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12, issuing the following query:

```
VERIFY:MASK? (@2(0:12))
```

would return:

```
1 1 1 1 1 1 1 1 1 1 1 X X
```

which would indicate the masks for the first 11 channels were inverting and the last two were don't cares. If this hypothetical card had a bad relay on channel 11, issuing the command:

```
VERIFY? (@2(0:12))
```

would simply return:

```
OK
```

If on the other hand, channels 3, 5 and 11 were bad, sending the same verification command would return:

```
2 : 3,2 : 5
```

As was the case earlier, channel 11 is excluded by the mask and will not show up in the error report. In addition, the 1256 will show only the first ten failures it finds when a large number of channels are specified in a query.

These commands are NOT defined in the SCPI standard. These commands follow the syntax rules of SCPI to implement this functionality.

Saving/Recalling the Verification Masks

All verification masks presently defined may be prepared for storage in non-volatile memory using the VERIFY:SAVE command.

All module names stored in non-volatile memory may be recalled from non-volatile memory using the VERIFY:RECALL command**.** The syntax for these commands is shown below:

```
[ROUTe]
      :VERify
            :SAVe
            :RECall
```

These commands are NOT defined in the SCPI standard. These commands follow the syntax rules of SCPI to implement this functionality.

NOTE:

The actual process of updating non-volatile memory requires a separate SYSTEM:NVUPD command. This allows multiple relay states, exclude lists, include lists, and module names to be updated in rapid succession, followed by a single, slower update of the actual non-volatile memory.

Automatically Recalling Verification Masks at Startup

In addition to defining verification masks, the user has the ability to determine whether the masks stored in non-volatile memory are recalled on application of power to the system. The command syntax is as follows:

```
[ROUTe]
      :VERify
            :RECall
                  :STATE {OFF | ON | 0 | 1}
                  :STATE?
```

When a state of `ON` or `1` is specified, a special flag is stored inside the 1256 non-volatile memory structure that instructs it to automatically load any stored verification masks each time power is applied to the 1256.

These commands are NOT defined in the SCPI standard. These commands follow the syntax rules of SCPI to implement this functionality.

NOTE:

The actual process of updating non-volatile memory requires a separate SYSTEM:NVUPD command. This allows multiple relay states, exclude lists, include lists, and module names to be updated in rapid succession, followed by a single, slower update of the actual non-volatile memory.

This page was left intentionally blank.

Chapter 9
PRODUCT SUPPORT

Product Support

EADS North America Test and Services, has a complete Service and Parts Department. If you need technical assistance or should it be necessary to return your product for repair or calibration, call 1-800-722-3262. If parts are required to repair the product at your facility, call 1-949-859-8999 and ask for the Parts Department.

For worldwide support and the office closest to your facility, refer to the website for the most complete information http://www.eads-nadefense.com.

Warranty

Use the original packing material when returning the 1256 to EADS North America Test and Services, Inc. for calibration or servicing. The original shipping container and associated packaging material will provide the necessary protection for safe reshipment.

If the original packing material is unavailable, contact EADS North America Test and Services, Customer Service at 1-800-722-3262 for information.

This page was left intentionally blank.

Appendix A
Specifications

General

Switch Plug-In Interface	Compliant with Adapt-a-Switch™ platform Compatible with all Adapt-a-Switch modules Eight plug-in slots
Front Panel	Vacuum fluorescent display Menu soft keys Optically-encoded knob
Rear-Panel Connectors	IEEE-488 (standard GPIB) RS-232 (9-pin D-SUB) Trigger in (BNC) Trigger out (BNC) AC power input (IEC)

System

Analog Bus	Four two-wire buses on backplane for interconnecting the plug-ins
Non-Volatile Memory	101 non-volatile memory locations for storage of switch states and user preferences Additional power-on state
Switch Settling Time	Automatically selected by 1256 according to each plug-in

Remote Interface

IEEE-488 (GPIB)

Compliance: IEEE-488.2

Address Selection: Manual, via front-panel controls

RS-232

Hardware Protocol: Data Communications Equipment (DCE). Connects directly to computer COM port with straight-through cable; no null modem.

Baud Rates:

1,200
2,400
4,800
9,600
19,200
38,400
57,600
115,200

Start bits: 1

Stop bits: 1 or 2

Number of bits: 7 or 8

Parity: Even, odd, none

Flow Control: Software (XON/XOFF)
Hardware (RTS/CTS)

Command Set

SCPI. Identical to current 1260-Series switch family except for modifications to triggering functions to accommodate non-VXIbus operation.

Software Drivers

Drivers and soft front panels provided for all Adapt-a-Switch plug-ins.

Electrical

Power Supply Type

Switched-mode PWM

AC Input	90VAC to 250VAC, auto-ranging, 47-63Hz, 180VA max.
DC Supply Output	+5VDC @ 20A

Environmental

General	MIL-T-28800E Type III, Class 5, Style F	
Temperature	**Operating:**	-20^{o} to +60^{o}C
	Storage:	-40^{o}C to +70^{o}C
Humidity	95% non-condensing	
Altitude	**Operating:**	10,000 feet
	Storage:	15,000 feet
Shock	30g, 11ms, ½ sine wave	
Vibration	0.013 in. (P-P), 5-55Hz	
Bench Handling	4-inch drop at 45^{o}	

EMC

Immunity	EN61326, Class A, Table 1
Emissions	EN61326, Class A, Table 3

Safety

Safety	CE, EN61010-1

Reliability

MTBF	>42,390 hours (with fan)

Mechanical

Dimensions	3.5”	high
	16.6”	wide
	11.3”	deep
Chassis Weight	7 lbs., 11 oz. (3.5 kg)	
Installation	**Rack mount:**	Use optional rack-mount ears (Option 65) for forward or reverse mounting
	Bench-top:	Tilt legs for tabletop use

Appendix B
RACK MOUNTING

Items Included

The rack-mount kit, Option 60, includes the following items.

Item	Description	Qty.	Part Number
1	Rack-mount Ear	2	456884
2	6X32X¾ Pan-Head Screw	4	616257
3	Instruction Sheet, Option 60 Rack Mount	1	980858

Installing Rack-Mount Ears

NOTE:

There is no need to open the unit.

1. Remove the feet by removing the screws that secure them.
2. Remove two pan-head allen screws from each side of the chassis. They are the top and the bottom screws. Refer to the diagram in **Figure B-1**.
3. Secure rack mount ear (Item 1) to the unit, using 6X32 panhead screws (Item 2) as shown.

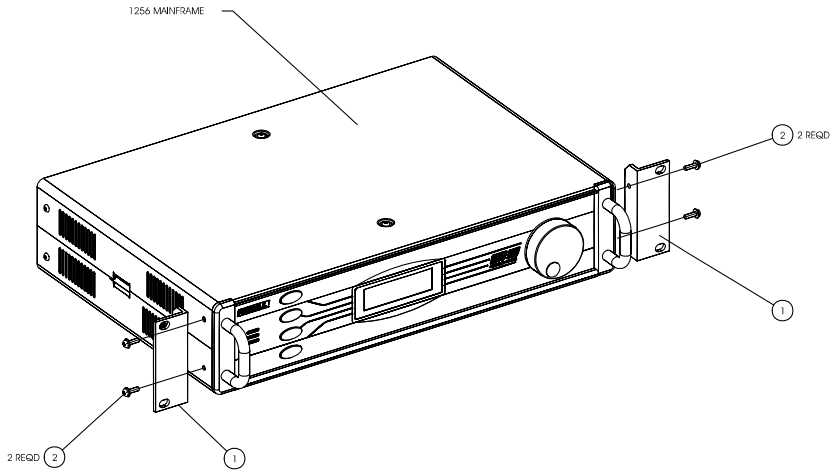


Figure B-1, Installing the Rack-Mount Ears

CAUTION:

DO NOT attempt to support the mainframe in the rack with the rack mount ears only. Right angle runners or a base plate MUST be used to support the mainframe.

Installing the 1256 Chassis Into a Rack

Slide the instrument fully into the rack on previously installed right angle runners (**Figure B-2**) or a base plate, until the rack mounting ears strike the front rail. Secure the instrument in place using two (each side) Phillips Pan-Head #6-32 x 3/4 screws (Item 2)

Fixed Rack Mount installation is now complete.

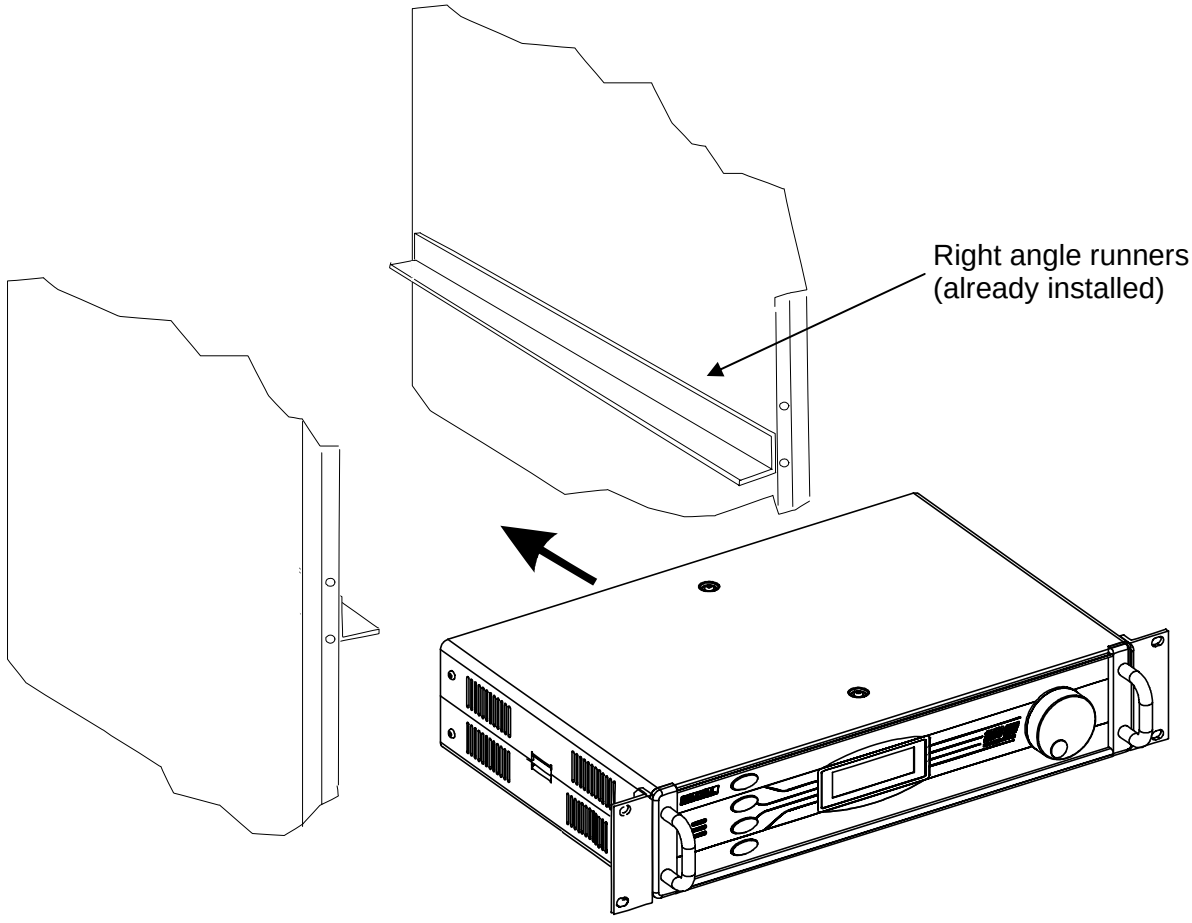


Figure B-2, Rack Mounting

Appendix C
Troubleshooting

How to Use this Section

If you have difficulty using the 1256 Switching System, refer to the following troubleshooting tables. If you are still unable to resolve the problem, contact the EADS North America Test and Services Customer Support Department (refer to **Chapter 9, PRODUCT SUPPORT**).

Power-Up

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
Unit does not turn on (fan not running, blank display, and no indicator lights).	• Not connected to proper power source.	• Make sure that the 1256 is connected to a live power source, using a functional power cord. For input power requirements, see **Chapter 2, Getting Started**, under the heading **Main AC Power**.
	• Fuse missing, blown, or incorrectly installed.	• Check fuse, verifying that it has the required capacity as marked on the 1256 rear panel. Make sure the fuse is properly installed (see **Chapter 2, Getting Started**, in **Figure 2-1, Changing Fuse for Main AC Power**).

Front-Panel Controls

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
Display goes blank (was working).	• Power has been interrupted.	• Verify that the 1256 AC input is connected to a live mains outlet. • Verify that the 1256 power switch is still in the ON position.
	• 1256 screen saver has been activated.	• Screen saver activation is normal after a period of no front-panel control activity. To restore the display, press any key on the front panel. • To increase the delay time, or disable the screen saver, refer to Chapter 3: USING THE FRONT-PANEL CONTROLS, under the Settings heading. Note that leaving the screen saver enabled maximizes the life of the fluorescent display.
No response to front-panel controls.	• Key is not currently enabled. Enabled keys have their LEDs illuminated.	• Read display options carefully. Not all keys are valid for all menus.

	• 1256 is in Automatic Local lockout mode.	• If display reads:
		`*REMOTE OPERATION*`
		`(RS-232 is active)`
		`<Local Mode`
		then press key 4 (bottom) to restore display and controls.

	• 1256 is in Manual Lockout mode.	• If display reads:
		`*REMOTE OPERATION*`
		`(RS-232 is active)`
		`Local Mode Lockout`
		then the 1256 is in Manual Lockout mode. To restore controls, send a `SYST:KLOCK OFF` command to the 1256 via remote interface.

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
Display is too dim or too bright.	• Display intensity needs to be set to personal preference.	• Set display intensity (see **Chapter 3, Using the Front-Panel Controls**, under the heading **Enabling or Setting the Display Intensity**.

RS-232 Remote Interface

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
Remote terminal receives unintelligible text from 1256, but only immediately after power turn-on. Text is properly received later on.	• Upon turn-on, the 1256 sends synchronization data at 115,200 baud to the RS-232 port in case a flash programmer is connected. If a programmer is present, this indicates to the programmer that the 1256 is ready for programming operations. Unintelligible characters on a normal terminal at power up are expected if terminal baud rate is not set to 115,200.	• Ignore unintelligible text that is received immediately after power turn-on.
Remote terminal receives unintelligible text from 1256. No intelligible text is received at all.	• Baud rate of 1256 not matched to baud rate of terminal.	• Set baud rate of 1256 to match that of the data terminal (see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the Baud Rate**).
	• Parity setting of 1256 not matched to parity of terminal.	• Set parity of 1256 to match that of the data terminal (see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the Data Bits, Stop Bits, and Parity**).
	• Number of data bits setting of 1256 not matched to number of bits for terminal.	• Set number of data bits of 1256 to match that of the data terminal (see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the Data Bits, Stop Bits, and Parity**).
	• Slow data terminal response (usually when using an older terminal).	• Set number of stop bits of 1256 to 2 (see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the Data Bits, Stop Bits, and Parity**).

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
	• Baud rate too high for cable and terminal being used.	• Change to a lower baud rate. Make sure that the new 1256 baud rate matches the new terminal baud rate (to set the 1256 baud rate, see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the Baud Rate**).
	• Cable is excessively long, or not of high quality.	• Use a high-quality, shielded cable.
Text from 1256 to remote terminal overwrites itself, staying on one line.	• Termination characters of 1256 not compatible with terminal setup.	• Set 1256 to use carriage return and line feed (CR-LF) as termination characters (see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the Message Termination Characters**).
Text from 1256 to remote terminal is double-spaced (lines of text are separated by blank lines).	• Termination characters of 1256 not compatible with terminal setup.	• Set 1256 to use only carriage return (CR) as termination character (see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the Message Termination Characters**).
Remote terminal does not display characters that are typed.	• Echo mode of remote terminal is not set to half-duplex (local echo).	• Set terminal for half-duplex (local echo) mode. Consult the terminal user's manual for instructions.
	• HyperTerminal (included with Microsoft Windows) can fail to echo characters properly.	• Download newer version, known as "HyperTerminal Private Edition" from: http://www.hilgraeve.com

IEEE-488 (GPIB) Remote Interface

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
1256 does not respond to GPIB commands.	• 1256 is not set for remote GPIB operation.	• Use front-panel controls to set the 1256 for GPIB operation. See **Chapter 4, GPIB AND RS-232 INTERFACES**, under the heading **Selecting the GPIB Interface**.
	• Commands sent to wrong GPIB address.	• Make sure that the commands are sent to the GPIB address currently assigned to the 1256 (to set the 1256 GPIB address, see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the GPIB Address**).
	• The 1256 is set to the same GPIB address as another device in the system.	• Make sure that the 1256 is set to a unique GPIB address(to set the 1256 GPIB address, see **Chapter 4, GPIB and RS-232 Interfaces**, under the heading **Setting the GPIB Address**).

Plug-In Modules

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
Slot containing a plug-in is indicated as an “empty slot” when using front-panel controls to scroll through slots.	• Plug-in not properly seated in slot.	• Loosen the retaining screws of the plug-in. Pull the plug-in part-way out, then gently push it back in until it stops. Then press it firmly to seat its connector with the backplane connector. When properly installed, the panel of the plug-in is flush with the rear panel of the 1256.

Non-Volatile Memory

PROBLEM	POSSIBLE CAUSES	WHAT TO DO
Switch states or menu settings that have been saved to non-volatile memory are lost the next time power is turned on.	• Switch states and menu settings must be saved in a two-step process: 1. Save to the non-volatile memory image, located in SRAM. 2. Save the image to non-volatile memory. The reason for this implementation is that non-volatile memory is inherently much slower than SRAM. By saving the image to non-volatile memory just one time instead of performing a separate block write for each piece of data, the slow non-volatile write cycle is encountered just once.	• After saving switch states or menu preferences to the image, save the image to non-volatile memory before turning power off (see **Chapter 3, Using the Front-Panel Controls**, under the heading **Store/Recall Menu**).

Appendix D
PLUG-IN Modules

How to Use This Section

Every Adapt-a-Switch switch plug-in comes with an information packet providing technical information that is specific to that module. Insert these packets after this page so that they become part of this manual.

This page was left intentionally blank.

INDEX

AC power, 2-1, 2-2

automatic lockout, 5-42

baud rate, 4-10, 4-11, 4-12

bus trigger, 5-21

CLOSE, 5-2, 5-3, 5-4, 5-18, 5-22, 5-27, 5-28, 5-29, 5-30, 5-31, 5-33, 5-34, 5-36, 5-37, 5-38,5-39

CME, 5-9

Command Error, 5-9

Command Input Buffer, 5-25

command set, 5-14, 5-15

confidence mode, 4-12

controls, 2-5, 2-6, 4-4, 4-10, 4-15, 4-16

CTS, 4-14, 4-10, 4-18, 5-26

data bits, 4-24

DCE, 4-12

DDE, 5-9

DEFINE, 5-3, 5-29, 5-32,5-33, 5-35, 5-36

delay, 4-13

Device Dependent Error, 5-9

discrete parameters, 5-24

DTE, 4-12

echo, 4-10

EOL, 4-12, 4-15, 4-20, 5-5

error code, 5-21, 5-40

error message, 5-9, 5-10, 5-17, 5-27, 5-40

Exclude Lists, 5-18

EXE, 5-9

Execution Error, 5-9

firmware revision, 4-4, 5-16

flow control, 4-10, 4-18, 5-25, 5-26

front panel, 2-3, 5-41

fuse, 2-1

GPIB, 4-1, 4-2, 4-4, 4-10, 4-16, 5-4, 5-6, 5-7, 5-12, 5-14, 5-16, 5-21, 5-25, 5-26, 5-27

GPIB interface, 4-10, 5-4, 5-25

half-duplex, 4-20, 5-5

IEEE-488, 2-6, 4-1, 4-12, 5-6, 5-7, 5-8, 5-12, 5-13, 5-15, 5-17, 5-19, 5-21, 5-23, 5-27

IMM, 5-24, 5-25

linefeed, 5-4, 5-5, 5-26

local echo, 4-20

lockout, 5-41, 5-42

manual lockout, 5-42

Master Summary Status, 5-11

MAV, 5-12, 5-27

Message Available, 5-12

modem, 4-12, 4-14

module name, 5-19, 5-29, 5-30, 5-31, 5-32, 5-33, 5-36, 5-37

MSS, 5-11, 5-12, 5-15, 5-16

non-volatile memory, 5-5, 5-7, 5-18, 5-19, 5-20, 5-26, 5-32, 5-36, 5-37, 5-40, 5-42

non-volatile update, 5-42

null modem, 4-12

numeric parameters, 5-23

OPC, 5-6, 5-10, 5-16, 5-27

OPEN, 5-2, 5-18, 5-22, 5-27, 5-33, 5-34, 5-37, 5-38, 5-39

Operation Complete, 5-10, 5-16

Operation Status Enable Register, 5-13, 5-22

Operation Status Event, 5-11, 5-13, 5-22

Operation Status Event Register, 5-11, 5-13, 5-22

OSE, 5-11

parity, 4-13, 4-14

path name, 5-19, 5-33, 5-34, 5-35, 5-36, 5-43

PON, 5-9, 5-10, 5-11

Power-On, 5-9, 5-18

Query Error, 5-10

Questionable Status Enable Register, 5-13

Questionable Status Event Register, 5-13, 5-22

QYE, 5-10, 5-11, 5-27

recall, 5-19

remote interface, 4-1, 4-10

Reply Output Buffer, 5-27

Request Control, 5-10

ROUTE, 5-2, 5-3, 5-4, 5-14, 5-18, 5-25, 5-31, 5-35

RQC, 5-10

RS-232, 2-6, 4-1, 4-6, 4-10, 4-12, 4-13, 4-14, 4-15, 4-16, 4-18, 4-20, 5-4, 5-5, 5-6, 5-7, 5-25, 5-26

RS-232 interface, 2-6, 4-12, 4-13, 4-14, 4-15, 4-16, 5-4

RS-232 parameters, 4-13, 4-15

RTS, 4-10, 4-14, 4-18, 5-26

safety, 5-26

SCPI, 2-6, 4-1, 4-12, 5-2, 5-3, 5-4, 5-11, 5-13, 5-14, 5-17, 5-21, 5-22, 5-23, 5-30, 5-33, 5-34, 5-38, 5-39, 5-40, 5-41, 5-42, 5-43

self-test, 2-5, 5-6, 5-19

service, 4-12, 9-1

service request, 4-12

Service Request Enable Register, 5-9, 5-12, 5-13, 5-15, 5-16

SRQ, 4-12, 5-7, 5-12, 5-16, 5-21

Standard Event Status Enable Register, 5-9, 5-10, 5-11, 5-12, 5-13, 5-14, 5-16

Standard Event Status Register, 5-9, 5-10, 5-11, 5-13, 5-14, 5-15, 5-16, 5-27, 5-40

status byte, 5-6, 5-11, 5-13, 5-20, 5-21, 5-27

stop bits, 4-13

system commands, 5-40

Trigger Input Delay, 5-18

Trigger Input Source, 5-18

Trigger Output Delay, 5-18

trigger source, 5-21, 5-22

URQ, 5-9

User Request, 5-9

Waiting For Arm, 5-21

warranty, 9-1

XON/XOFF, 4-10, 4-18, 5-26

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

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

ARTISAN® TECHNOLOGY GROUP