Contents

HP E2291A 16-Channel Isolated Digital Output User's Manual and SCPI Programming Guide

Edition 1

Warranty 3
Safety Symbols 4
WARNINGS 4
Declaration of Conformity 5
Reader Comment Sheet 7

Chapter 1
Configuring the Module 9
What's in This Manual 9
Module Description 9
General Features 9
Module Description 9
Wiring and Configuration 11
Assembling the Field Wiring Connector 12

Chapter 2
SCPI Programming 15
Using This Chapter 15
Reset Condition 15
Example 1: Initial Operation 15
Closing and Opening Channels 17
Channel Lists 17
Example 2: Opening/Closing Multiple Channels 18
Example 3: Setting the Delayed Interrupt Time 19

Chapter 3
SCPI Command Reference 21
DIAGnostic Subsystem 24
DIAGnostic:INTerrupt[:LINe] <intr_line> 24
DIAGnostic:INTerrupt[:LINe]? 25
DIAGnostic:TEST? 25
DIAGnostic:SWITch:TIMe <set_time> 26
DIAGnostic:SWITch:TIMe? 27
DISPlay Subsystem 28
DISPLay:MONitor[:STATe] <boolean> 28
DISPLay:MONitor[:STATe]? 28
ROUTe Subsystem 29
[ROUTe:]CLOSe <channel_list> 29
[ROUTe:]CLOSe? <channel_list> 30
[ROUTe:]OPEN <channel_list> 31
[ROUTe:]OPEN?<channel_list> 32
STATus Subsystem 33
SYSTem Subsystem 34
SYSTem:ERRor? 34
SYSTem:VERSion? 34
SCPI Command Quick Reference 35

IEEE 488.2 Common Command Reference 36

Chapter 4
Register Descriptions 37
About This Chapter 37
Block Diagram Description 37
Module Control 38
ID EEPROM 38
Output Driver and Output Circuit 38
Delay Timer 38
Power Conditioning 38
Register Addressing in the VXI Environment 39
Logical Address 39
A16/A24 Memory Mapping 39
Determining a Module's A16 Base Address 40
Addressing A16 Registers 41
Addressing A24 Registers 41
Program Example 42
Registers in A16 Memory Space 44
VXI ID Register 44
VXI Device Type Register 44
VXI Status/Control Register 45
Interrupt Selection Register 46
Registers in A24 Address Space 48
Status Register 48
Control Register 48
Interrupt Register 49
Delay Register 49
Output Register 49
ID EEPROM Register 50

Appendix A
Specifications 53
M-Module Specification Compliance 53
HP E2291A 16-Channel Isolated Digital Output M-Module 53

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard product is warranted against defects in materials and workmanship for a period of three years from date of shipment. Duration and conditions of warranty for this product may be superseded when the product is integrated into (becomes a part of)other HP products. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard (HP). Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country

HP warrants that its software and firmware designated by HP for use with a product will execute its programming instructions when properly installed on that product. HP does not warrant that the operation of the product, or software, or firmware will be uninterrupted or error free.

Limitation Of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied products or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

The design and implementation of any circuit on this product is the sole responsibility of the Buyer. HP does not warrant the Buyer's circuitry or malfunctions of HP products that result from the Buyer's circuitry. In addition, HP does not warrant any damage that occurs as a result of the Buyer's circuit or any defects that result from Buyer-supplied products.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Notice

The information contained in this document is subject to change without notice. HEWLETT-PACKARD (HP) MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HP shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. HP assumes no responsibility for the use or reliability of its software on equipment that is not furnished by HP.

U.S. Government Restricted Rights

The Software and Documentation have been developed entirely at private expense. They are delivered and licensed as "commercial computer software" as defined in DFARS 252.227- 7013 (Oct 1988), DFARS 252.211-7015 (May 1991) or DFARS 252.227-7014 (Jun 1995), as a "commercial item" as defined in FAR 2.101(a), or as "Restricted computer software" as defined in FAR 52.227-19 (Jun 1987)(or any equivalent agency regulation or contract clause), whichever is applicable. You have only those rights provided for such Software and Documentation by the applicable FAR or DFARS clause or the HP standard software agreement for the product involved

HP E2291A 16-Channel Isolated Digital Output User's Manual and SCPI Programming Guide
Edition 1

Documentation History

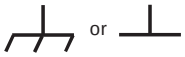
All Editions and Updates of this manual and their creation date are listed below. The first Edition of the manual is Edition 1. The Edition number increments by 1 whenever the manual is revised. Updates, which are issued between Editions, contain replacement pages to correct or add additional information to the current Edition of the manual. Whenever a new Edition is created, it will contain all of the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this documentation history page.

Edition 1 . . . . . .June 1997

Safety Symbols

Instruction manual symbol affixed to product. Indicates that the user must refer to the manual for specific WARNING or CAUTION information to avoid personal injury or damage to the product.

Indicates the field wiring terminal that must be connected to earth ground before operating the equipmentÅprotects against electrical shock in case of fault.



Frame or chassis ground terminal—typically connects to the equipment's metal frame.

Alternating current (AC)

Direct current (DC).

Indicates hazardous voltages.

WARNING Calls attention to a procedure, practice, or condition that could cause bodily injury or death.

CAUTION Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.

WARNINGS

The following general safety precautions must be observed during all phases of operation, service, and repair of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the product. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

Ground the equipment: For Safety Class 1 equipment (equipment having a protective earth terminal), an uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

DO NOT operate the product in an explosive atmosphere or in the presence of flammable gases or fumes.

For continued protection against fire, replace the line fuse(s) only with fuse(s) of the same voltage and current rating and type. DO NOT use repaired fuses or short-circuited fuse holders.

Keep away from live circuits: Operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers or shields are for use by service-trained personnel only. Under certain conditions, dangerous voltages may exist even with the equipment switched off. To avoid dangerous electrical shock, DO NOT perform procedures involving cover or shield removal unless you are qualified to do so.

DO NOT operate damaged equipment: Whenever it is possible that the safety protection features built into this product have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the product until safe operation can be verified by service-trained personnel. If necessary, return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

DO NOT service or adjust alone: Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT substitute parts or modify equipment: Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

Declaration of Conformity

according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name:	Hewlett-Packard Company Loveland Manufacturing Center
Manufacturer's Address:	815 14th Street S.W. Loveland, colorado 80537

declares, that the product:

Product Name:	16-Channel Isolated Digital Output Module
Model Number:	HP E2291A
Product Options:	All

conforms to the following Product Specifications:

Safety:	IEC 1010-1 (1990) Incl. Amend 2 (1996)/EN61010-1 (1993) CSA C22.2 #1010.1 (1992) UL 3111-1 (1994)
EMC:	CISPR 11:1990/EN55011 (1991): Group1 Class A EN61000-3-2:1995 Class A EN50082-1:1992 IEC 801-2:1991: 4kV CD, 8kV AD IEC 801-3:1984: 3 V/m IEC 801-4:1988: 1kV Power Line, 0.5kV Signal Lines ENV50141:1993/prEN50082-1 (1995): 3 Vrms ENV50142:1994/prEN50082-1 (1995): 1 kV CM, 0.5 kV DM IEC1000-4-8:1993/prEN50082-1 (1995): 3 A/m EN61000-4-11:1994/prEN50082-1 (1995): 30%, 10ms 60%, 100ms

Supplementary Information: The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (inclusive 93/68/EEC) and carries the "CE" mark accordingly.

Tested in a typical configuration in an HP C-Size VXI mainframe.

April 7, 1997

Jim White, QA Manager

European contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department HQ-TRE, Herrenberger Straße 130, D-71034 Böblingen, Germany (FAX: +49-7031-14-3143)

Notes:

Please fold and tape for mailing

Reader Comment Sheet

HP E2291A 16-Channel Isolated Digital Output User's Manual & SCPI Programming Guide
Edition 1

You can help us improve our manuals by sharing your comments and suggestions. **In appreciation of your time, we will enter you in a quarterly drawing for a Hewlett-Packard Palmtop Personal Computer** (U.S. government employees are not eligible for the drawing).

Your Name	City, State/Province
Company Name	Country
Job Title	Zip/Postal Code
Address	Telephone Number with Area Code

Please list the system controller, operating system, programming language, and plug-in modules you are using.

fold here

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 37 LOVELAND, CO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD COMPANY
Measurement Systems Division
Learning Products Department
P.O. Box 301
Loveland, CO 80539-9984

cut along this line

fold here

Please pencil-in one circle for each statement below:

	Disagree ◄				► Agree
• The documentation is well organized.	O	O	O	O	O
• Instructions are easy to understand.	O	O	O	O	O
• The documentation is clearly written.	O	O	O	O	O
• Examples are clear and useful.	O	O	O	O	O
• Illustrations are clear and helpful.	O	O	O	O	O
• The documentation meets my overall expectations.	O	O	O	O	O

Please write any comments or suggestions below–be specific.

Chapter 1
Configuring the Module

What's in This Manual

This manual contains a module description, configuration and wiring information, SCPI programming information, register programming information, and specifications for the HP E2291A 16-Channel Isolated Digital Output M-Module.

This chapter contains general information, a simplified schematic, configuring and wiring information for the HP E2291A.

Module Description

General Features

- Four groups of four isolated output channels. Each group requires its own +5VDC power supply (supplied by the user). All channels are isolated from chassis.
- Can be used as simple digital outputs or to drive relays
- Programmable Interrupt Delay Timer provides for delayed acknowledgment of command completion. For example, if you use the module to control relays, you can set the timer to assert interrupt after the relays have settled.
- Each channel can sink up to 200mA and switch voltages up to 36VDC.

Module Description

Refer to Figure 1-3. The HP E2291A provides four groups of four channels (channels 00 - 03, 04 - 07, 08 - 11, and 12 - 15). For group isolation, the user must supply each group with separate +5VDC power supplies:

Table 1-1. HP E2291A External Power Supply Connections

Channels	External +5VDc Connections
00 - 03	$+5V_1$ (Pin 18), $TGND_1$ (pins 2, 3, 32, 33)
04 - 07	$+5V_2$ (pin 22), $TGND_2$ (Pins 6, 7, 36, 37)
08 - 11	$+5V_3$ (Pin 24), $TGND_3$ (Pins 8, 9, 38, 39)
12 - 15	$+5V_4$ (Pin 28), $TGND_4$ (Pins 12, 13, 42, 43)

The power supplies power the optical-isolators in each channel. If isolation is not necessary, one power supply can be used for all four groups.

The output of each channel is an open collector NPN transistor (emitter is connected to the $TGND_n$ of each group's isolated power supply). Each channel is zener diode protected to +36VDC, do not exceed this voltage. Each channel's transistor can sink up to 200mA. Figure 1-1 shows a typical application for the module.

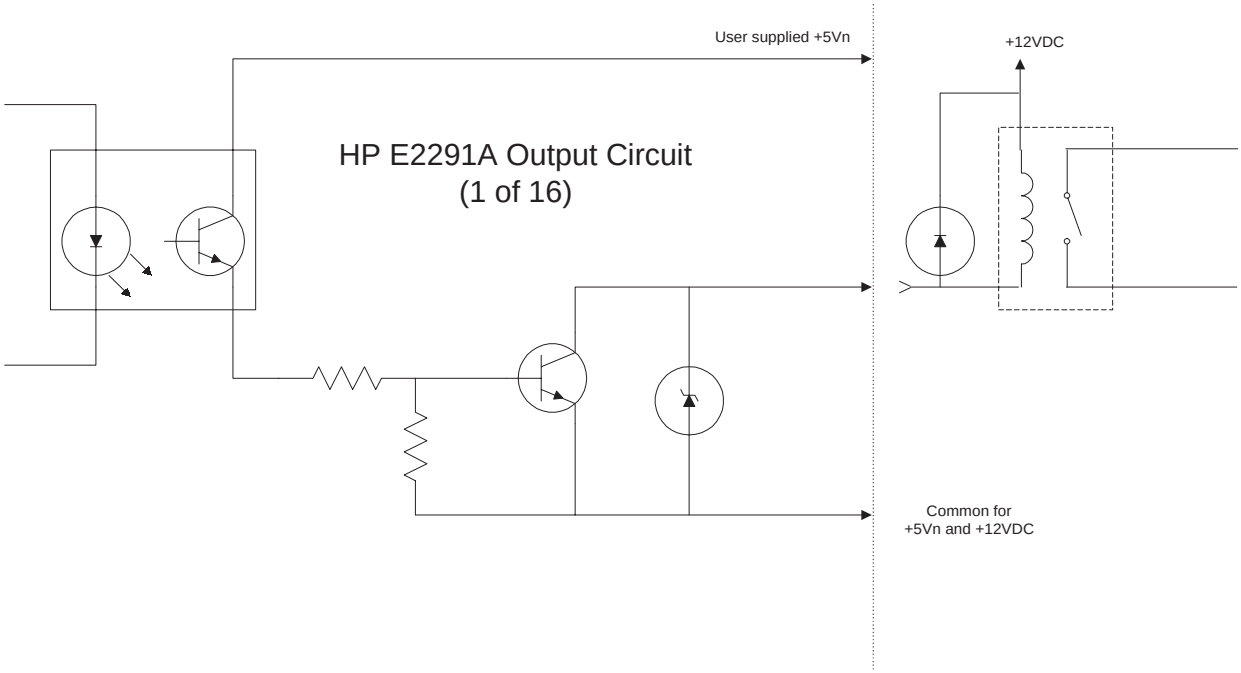


Figure 1-1. Sample Application

Programmable Interrupt Delay Timer

When a command to close a channel or open a channel (CLOSE means the output transistor is biased ON, a low collector-emitter impedance; OPEN means the transistor is biased OFF, a high collector-emitter impedance) completes execution, the module asserts an interrupt. For some external circuits, you may not want the interrupt until some period of time after the module finishes execution. For example, if you are using the module to control relays with a 13 - 15 mS settling time, you can program a delay so that the module does not assert interrupt until after the relays have settled. The delay time is valid for all 16 channels. Programmable range is 0.031875 mS to 2089 mS in 1 mS increments. Default is 13 mS.

Wiring and Configuration

This section describes how to connect user wiring to the HP E2291A.

Note The procedures in this section assume the M-Module has already been installed into an M-Module carrier. Since installation is dependent on the carrier used, instructions for installing M-Modules into the carrier are not included here. Refer to you M-Module carrier documentation for installation procedures. Each M-Module is shipped with identifying labels that should be installed on the carrier.

WARNING **SHOCK HAZARD. Only service-trained personnel aware of the hazards involved should install, configure, or remove the M-Module. Before installing or removing any module or carrier, disconnect power from the mainframe and user wiring.**

Caution **VOLTAGE/CURRENT. Pay more attention to the limitation of maximum voltage/current and maximum power listed in Appendix A. Exceeding any limit or use outside the parameters specified may damage the modules and impair the protection provided by the modules.**

Caution **STATIC ELECTRICITY. Static electricity is a major cause of component failure. To prevent damage to the electrical components on an M-Module or the carrier, observe anti-static techniques whenever installing, removing, or working on a carrier or M-Module.**

Assembling the Field Wiring Connector

Each HP M-Module includes a 44-pin connector and hood (HP kit part number E2273-01203). You must supply your own cable. The drawing below shows how to connect wiring and assemble the connector and hood.

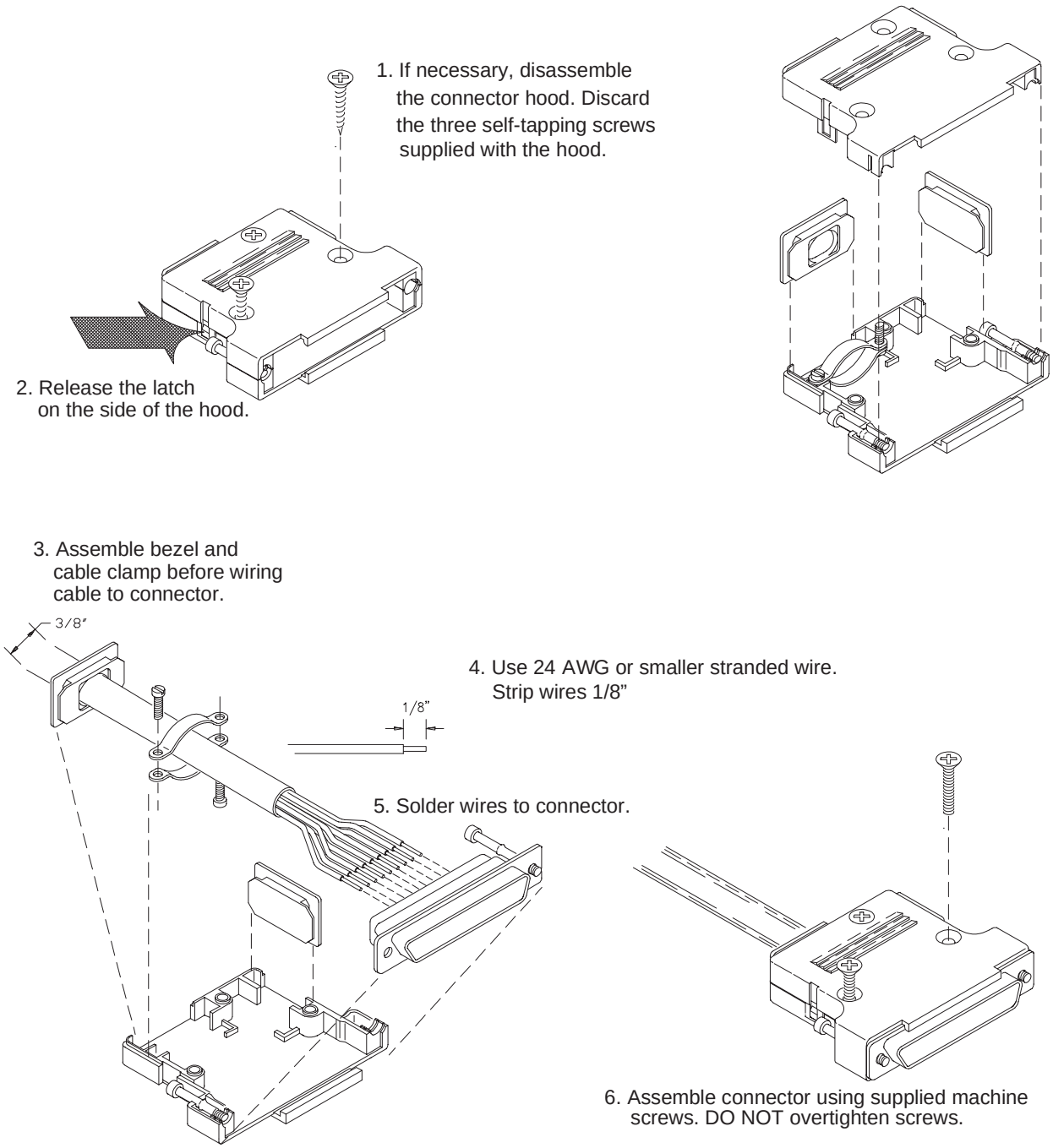
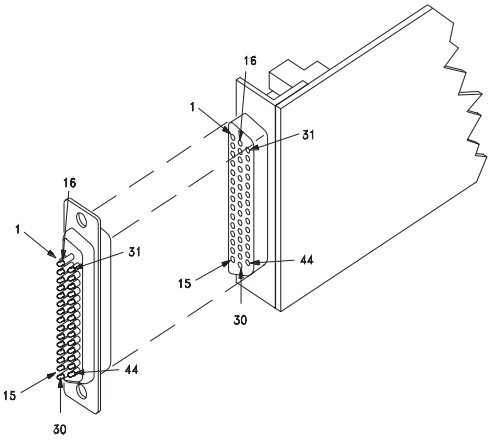


Figure 1-2. Wiring the 44-Pin Field Wiring Connector



Note: User must supply +5Vdc to the module. You can supply up to four isolated supplies (+5V$_1$/TGND$_1$, +5V$_2$/TGND$_2$, +5V$_3$/TGND$_3$, +5V$_4$/TGND$_4$) or they can be combined and one supply used. Refer to the drawing below.

CAUTION: The maximum voltage is 36 VDC or 36 VAC-Peak per channel. Maximum current is 200 mA DC or 200 mA AC-peak per channel.

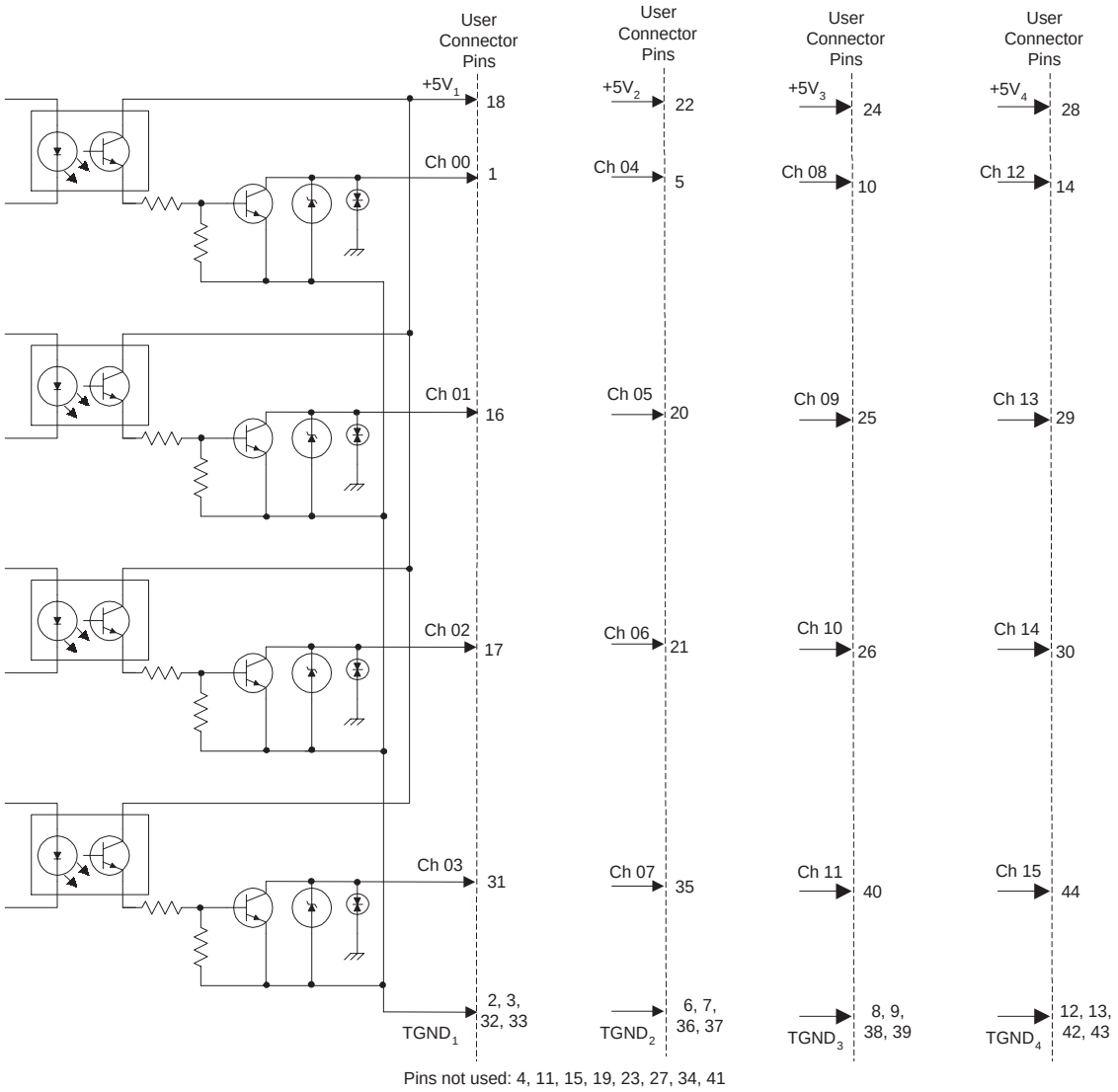


Figure 1-3. Simplified E2291A User Connections and Simplified Schematic

Chapter 2
SCPI Programming

Using This Chapter

This chapter contains SCPI program examples that demonstrate how to read a module ID, perform self test, set the programmable busy time, and close channels. The program examples were written in C language. To run any of these programs you must have the HP SICL library, the HP VISA library, an HP-IB interface module installed in an external PC, an HP E1406 Command Module, and the HP E2291A M-Module installed on an HP Carrier such as the HP E2251A.

Reset Condition

At power-on or following the reset of the modules (*RST command), all the channels (channels 00 through 15) are open (the output transistors are high collector-emitter impedance).

Example 1: Initial Operation

The following example reads the module ID string, performs module self-test, and displays the results.

```
#include <visa.h>
#include <stdio.h>
#include <stdlib.h>

    /* Interface address is 9, M-Module secondary address is 3*/
#define INSTR_ADDR "GPIB0::9::3::INSTR"

int main()
{
   ViStatus errStatus;                  /*Status from each VISA call*/
   ViSession viRM;                      /*Resource mgr. session */
   ViSession m_mod;                     /* M-module session  */
   char id_string[256];                 /*ID string*/
   char selftst_string[256];            /*self-test string*/

    /* Open the default resource manager  */
   errStatus = viOpenDefaultRM (&viRM);
   if (VI_SUCCESS > errStatus){
      printf ("ERROR: viOpenDefaultRM() returned 0x%x\n",errStatus);
      return errStatus;}
```

```
    /* Open the M-Module instrument session  */
    errStatus = viOpen(viRM,INSTR_ADDR, VI_NULL,VI_NULL,&m_mod);
    if (VI_SUCCESS > errStatus){
        printf ("ERROR: viOpen() returned 0x%x\n",errStatus);
        return errStatus;}

    /* Reset the M-Module */
    errStatus = viPrintf(m_mod, "*RST\n");
    if (VI_SUCCESS > errStatus){
        printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
        return errStatus;}

    /* Perform M-Module Self-Test */
    errStatus = viQueryf(m_mod,"*TST?\n","%t",selftst_string);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
        return errStatus;}
    printf ("Self Test Result is %s\n",selftst_string);

    /* Query the M-Module ID string */
    errStatus = viQueryf(m_mod,"*IDN?\n","%t",id_string);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
        return errStatus;}
    printf("ID is %s\n",id_string);

    /* Close the M-Module Instrument Session */
    errStatus = viClose (m_mod);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viClose() returned 0x%x\n",errStatus);
        return 0;}

    /* Close the Resource Manager Session */
    errStatus = viClose (viRM);
    if (VI_SUCCESS > errStatus) {
        printf("ERROR: viClose() returned 0x%x\n",errStatus);
        return 0;}

    return VI_SUCCESS;
}
```

Closing and Opening Channels

The [ROUTe]:CLOSe<*channel_list*> closes one or more channels on the module. “Closed” means that the output transistor is biased ON; a low impedance between collector and emitter. The [ROUTe]:OPEN<*channel_list*> command opens one or more channels. “Open” means that the output transistor is biased OFF; a high impedance between collector and emitter.

Note The [ROUTe]: portion of the command is optional syntax and can be omitted. For example, the command [ROUTe]:CLOSe<*channel_list*> can be shortened to CLOSe<*channel_list*>.

Channel Lists

The <*channel_list*> parameter in the CLOSe or OPEN command has the form (@*ccnn*), where *cc* = card number and *nn* = channel numbers (00-15).

Note The SCPI driver supports single modules only. Therefore, *cc* is always 1. To simplify programming, the card number (*cc*) can be eliminated. The remainder of this manual will use the shortened (no card number) channel list format (@*nn*).

You can specify single channels (@nn); use commas to specify multiple channels (@nn,nn,...); or use a colon to specify a range of sequential channels (@nn:nn). You can also specify any combination of single channels, multiple channels, and channel ranges. Some examples:

CLOS (@00,03)
Close channels 00 and 03.

OPEN (@01,02,03,10)
Open channels 01, 02, 03 and 10.

OPEN (@00:15)
Open all channels (0 through 15)

CLOS (@02:05,07,09:11)
Close channels 02 through 05, 07, and 09 through 11

Note A range of channels (@nn:nn) must be specified in ascending order, that is lower channel number on the left, high channel number on the right.

Example 2: Opening/Closing Multiple Channels

The following example closes channels 01 and 10 through 13 on the M-Module. The program then opens channels 01 and 11. The program assumes an M-Module secondary address of 3 and an interface address of 9.

```
#include <visa.h>
#include <stdio.h>
#include <stdlib.h>

    /* Interface address is 9, M-Module secondary address is 3*/
#define INSTR_ADDR "GPIB0::9::3::INSTR"

int main()
{
   ViStatus errStatus;                  /*Status from each VISA call*/
   ViSession viRM;                      /*Resource mgr. session */
   ViSession m_mod;                     /* M-module session  */

    /* Open the default resource manager  */
   errStatus = viOpenDefaultRM (&viRM);
   if(VI_SUCCESS > errStatus){
      printf ("ERROR: viOpenDefaultRM() returned 0x%x\n",errStatus);
      return errStatus;}

    /* Open the M-Module instrument session  */
   errStatus = viOpen(viRM,INSTR_ADDR, VI_NULL,VI_NULL,&m_mod);
   if (VI_SUCCESS > errStatus){
      printf ("ERROR: viOpen() returned 0x%x\n",errStatus);
      return errStatus;}

    /* Reset the M-Module */
   errStatus = viPrintf (m_mod, "*RST\n");
   if (VI_SUCCESS > errStatus){
      printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
      return errStatus;}

    /* Close channels 1 and 10 through 13 on the M-Module */
   errStatus = viPrintf (m_mod,"ROUT:CLOS (@01,10:13)\n");
   if (VI_SUCCESS > errStatus) {
      printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
      return errStatus;}

    /* Open channels 1 and 11 on the M-Module */
   errStatus = viPrintf (m_mod,"ROUT:OPEN (@01,11)\n");
   if (VI_SUCCESS > errStatus) {
      printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
      return errStatus;}
```

```
    /* Close the M-Module Instrument Session */
    errStatus = viClose (m_mod);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viClose() returned 0x%x\n",errStatus);
        return 0;}

    /* Close the Resource Manager Session */
    errStatus = viClose (viRM);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viClose() returned 0x%x\n",errStatus);
        return 0;}

    return VI_SUCCESS;
}
```

Example 3: Setting the Delayed Interrupt Time

The HP E2291A provides the convenience of setting set a wide range of switching times (the switch time can be from 0.031875 ms to 2,089.00 ms). The settling time is valid for all 16 channels. This program queries the current delay time and then sets the delay time to 15mS.

```
#include <visa.h>
#include <stdio.h>
#include <stdlib.h>

    /* Interface address is 9, M-Module secondary address is 3*/
#define INSTR_ADDR "GPIB0::9::3::INSTR"

int main()
{
    ViStatus errStatus;                 /*Status from each VISA call*/
    ViSession viRM;                     /*Resource mgr. session */
    ViSession m_mod;                    /* M-module session  */
    char ch_time[25]                    /* Channel settling time */

    /* Open the default resource manager  */
    errStatus = viOpenDefaultRM (&viRM);
    if (VI_SUCCESS > errStatus){
        printf ("ERROR: viOpenDefaultRM() returned 0x%x\n",errStatus);
        return errStatus;}

    /* Open the M-Module instrument session  */
    errStatus = viOpen(viRM,INSTR_ADDR, VI_NULL,VI_NULL,&m_mod);
    if (VI_SUCCESS > errStatus){
        printf ("ERROR: viOpen() returned 0x%x\n",errStatus);
        return errStatus;}
```

```
    /* Query the current Switch Settling Time */
    errStatus = viQueryf(m_mod,"DIAG:SWIT:TIME?\n","%s",ch_time);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
        return errStatus;}
    printf ("Channel Settling Time is %s\n",ch_time);

    /* Set channel settling time to 15mS */
    errStatus = viPrintf(m_mod,"DIAG:SWIT:TIME 15\n");
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viPrintf() returned 0x%x\n",errStatus);
        return errStatus;}

    /* Close the M-Module Instrument Session */
    errStatus = viClose (m_mod);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viClose() returned 0x%x\n",errStatus);
        return 0;}

    /* Close the Resource Manager Session */
    errStatus = viClose (viRM);
    if (VI_SUCCESS > errStatus) {
        printf ("ERROR: viClose() returned 0x%x\n",errStatus);
        return 0;}

    return VI_SUCCESS;
}
```

Chapter 3
SCPI Command Reference

Using This Chapter

This chapter describes the **Standard Commands for Programmable Instruments** (SCPI) command set and the **IEEE-488.2 Common Commands** for the HP E2291A 16-Channel Isolated Digital Output M-Module. This chapter contains the following sections:

- Command Fundamentals . . . . . Page 21
- SCPI Command Reference . . . . . Page 29
- Common Command Reference . . . . . Page 35
- Command Quick Reference . . . . . Page 36

Command Fundamentals

Commands are separated into two types: IEEE 488.2 Common commands and SCPI commands.

Common Command Format

The IEEE 488.2 standard defines the Common commands that perform functions like reset, self-test, status byte query, etc. Common commands are four or five characters in length, always begin with the asterisk character (*****), and may include one or more parameters. The command keyword is separated from the first parameter by a space character. Some examples of Common commands are shown below:

```
*RST     *ESR 32     *STB?
```

SCPI Command Format

The SCPI commands perform functions like closing switches, making measurements, and querying instrument states or retrieving data. A subsystem command structure is a hierarchical structure that usually consists of a top level (or root) command, one or more lower-level commands, and their parameters. The following example shows part of a typical subsystem:

```
[ROUTe:]
    CLOSe <channel_list>
```

[ROUTe:] is the root command, CLOSe is second-level commands with parameter.

Command Separator

A colon (**:**) always separates one command from the next lower-level command as shown below:

```
[ROUTe:]OPEN?
```

Abbreviated Commands

The command syntax shows most commands as a mixture of upper- and lowercase letters. The uppercase letters indicate the abbreviated spelling for the command. For shorter program lines, send the abbreviated form. For better program readability, you

may send the entire command. The instrument will accept either the abbreviated form or the entire command.

For example, if the command syntax shows MEASure, then MEAS and MEASURE are both acceptable forms. Other forms of MEASure, such as MEASU or MEASUR will generate an error. You may use upper- or lowercase letters. Therefore, MEASURE, measure, and MeAsUrE are all acceptable.

Implied Commands

Implied commands are those which appear in square brackets ([ ]) in the command syntax. ***(Note that the brackets are not part of the command and are not sent to the instrument.)*** Suppose you send a second-level command but do not send the preceding implied command. In this case, the instrument assumes you intend to use the implied command and it responds as if you had sent it.

Examine the [ROUTe:] subsystem shown below:

```
[ROUTe:]
    CLOSe <channel_list>
    CLOSe? <channel_list>
    OPEN <channel_list>
    OPEN? <channel_list>
```

The root command [ROUTe:] is an implied command (indicated by square brackets [ ]). To close channels in a channel list, you can send either of the following command statements:

[ROUTe:]CLOSe (@100:103) *or* CLOSe (@100:103)

These commands function the same - closing channels 00 through 03 on the M-Module.

Parameters

Parameter Types. The following table contains explanations and examples of parameter types you might see later in this chapter.

Parameter Type	Explanations and Examples
Numeric	Accepts all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. 123, 123E2, -123, -1.23E2, .123, 1.23E-2, 1.23000E-01. Special cases include MIN, MAX, and INF.
Boolean	Represents a single binary condition that is either true or false. ON, OFF, 1, 0.

Linking Commands

Linking IEEE 488.2 Common Commands with SCPI Commands. Use a semicolon (**;**)between the commands. For example:

```
*RST;CLOS (@103)
```

Linking Multiple SCPI Commands. Use both a semicolon (**;**)and a colon (**:**) between the commands. For example:

```
OPEN (@102);:CLOS (@103)
```

Channel Designations

The [ROUTe]:CLOSe<*channel_list*> closes one or more channels on the module. “Closed” means that the output transistor is biased ON; a low impedance between collector and emitter. The [ROUTe]:OPEN<*channel_list*> command opens one or more channels. “Open” means that the output transistor is biased OFF; a high impedance between collector and emitter.

Note The [ROUTe]: portion of the command is optional syntax and can be omitted. For example, the command [ROUTe]:CLOSe<*channel_list*> can be shortened to CLOSe<*channel_list*>.

Channel Lists

The <*channel_list*> parameter in the CLOSe or OPEN command has the form (@*ccnn*), where *cc* = card number and *nn* = channel numbers (00-15).

Note The SCPI driver supports single modules only. Therefore, *cc* is always 1. To simplify programming, the card number (*cc*) can be eliminated. The remainder of this manual will use the shortened (no card number) channel list format (@*nn*).

You can specify single channels (@nn); use commas to specify multiple channels (@nn,nn,...); or use a colon to specify a range of sequential channels (@nn:nn). You can also specify any combination of single channels, multiple channels, and channel ranges. Some examples:

CLOS (@00,03)
Close channels 00 and 03.

OPEN (@01,02,03,10)
Open channels 01, 02, 03 and 10.

OPEN (@00:15)
Open all channels (0 through 15)

CLOS (@02:05,07,09:11)
Close channels 02 through 05, 07, and 09 through 11

Note A range of channels (@nn:nn) must be specified in ascending order, that is lower channel number on the left, high channel number on the right.

DIAGnostic subsystem controls setting and querying M-Module's interrupt level, setting and querying the switch delay time, and self test.

Subsystem Syntax

```
DIAGnostic
    :INTerrupt[:LINe] <intr_line>
    :INTerrupt[:LINe]?
    :TEST?
    :SWITch:TIMe <number>
    :SWITch:TIMe?
```

DIAGnostic:INTerrupt[:LINe] *<intr_line>*

DIAGnostic:INTerrupt[:LINe] <*intr_line*> sets the VXIbus interrupt line the M-Module will use. The M-Module generates an interrupt after a channel closing or opening completes. Refer to your command module documentation for more information.

Note

The VXIbus Interrupt Line is controlled by the VXIbus M-Module Carrier NOT by the M-Module. DIAGnostic:INTerrupt:LINE reprograms the HP E2251 M-Module Carrier. It will work properly only if the M-Module is installed in an HP E2251 M-Module Carrier.

Parameter

Parameter Name	Parameter Type	Range of Values	Default Value
<intr_line>	int16	0 through 7	1

Comments

- Normally, the VXIbus interrupt line does not need to be modified. Only one value (1 through 7) can be set at one time.
- The default value of <*number*> is 1 (lowest interrupt level).
- Setting the interrupt line to 0 disables the interrupt.
- If you are using the HP E2291A to drive relays, you can set a delay between when the [ROUTe:]CLOSe or [ROUTe:]OPEN command executes and when the interrupt occurs with the DIAGnostic:SWITch:TIMe command.

Reset Condition

*RST does not affect the interrupt line.

Related Commands

DIAGnostic:INTerrupt:LINE?, DIAGnostic:SWITch:TIMe

DIAGnostic:INTerrupt[:LINe]?

DIAGnostic:INTerrupt[:LINe]? returns the VXIbus interrupt line being used be the module.

Returned Data

Type	Range	Default
int16	0 through 7	1

Note The VXIbus Interrupt Line is controlled by the VXIbus M-Module Carrier NOT by the M-Module. DIAGnostic:INTerrupt:LINE reprograms the HP E2251 M-Module Carrier. It will work properly only if the M-Module is installed in an HP E2251 M-Module Carrier.

Reset Condition

*RST does not affect the interrupt line.

Related Commands

DIAGnostic:INTerrupt:LINE

DIAGnostic:TEST?

DIAGnostic:TEST? returns the result of the module's self test. This command performs an extended self test that opens all channels on the module and then closes each channel one at a time. For each channel closure, the test waits for an interrupt and then reads the M-Module register to verify that the channel did close. If an interrupt does not occur, the command returns a 3 indicating a missing interrupt. If the value read back from the module does not match the channel closed, a 2 is returned indicating a problem in the driver circuitry. A problem with the switch time circuitry returns a 1. A 0 is returned if all tests pass.

Caution **The extended self test opens and closes all channels on the module. Before performing the test, make certain that external devices connected to the module will not be affected by these actions. It is recommended that external devices be disconnected from the M-Module while executing DIAGnostic:TEST?.**

Returned Data

Type	Description of Numerical Response	Possible Strings Returned
int16, string	0 = self-test passsed	“Self Test passed”
	1 = ERROR: status register	“Busy, full bit failed. Expect 4 got X”
	2 = ERROR: register readback	“Readback reg X failed, expect 0 got X”
	3: ERROR: interrupt	“Interrupt failed VISA error X”

Comments

- A query response of 0 means that the module is operating properly, a non-zero result means an error occurred.
- The extended self test does not measure the actual state of the output transistor.The self test queries the state of the Control Register circuitry. A peek (read) of the channel’s VXI register returns the state of the Channel Control Register and not the actual state of the output circuit. It is possible to pass DIAGnostic:TEST? (return a 0) and still have output circuitry failures.
- After the test, the module is left in its power-on/reset (*RST) state.

Reset Condition

*RST does not affect this query.

Related Commands

*TST?, DIAGnostic:INTeruppt:LINE

DIAGnostic:SWITch:TIMe *<set_time>*

DIAGnostic:SWITch:TIMe <***set_time***> sets the time period between completion of the [ROUTe:]CLOSe or [ROUTe:]OPEN command and the return of the channel closed/open interrupt. The M-Module generates an interrupt indicating it has completed a channel close/open. If you are using the HP E2291A to control relays, for example, with a settling time of 13mS, you can set the DIAGnostic:SWITch:TIMe to 13 (or greater) so that the interrupt is not asserted until the relays have settled.

Parameters

Parameter Name	Parameter Type	Range of Values (ms)	Default Time
<set_time>	numeric	0.31875mS to 2089mS	13

Comments

- The delay time set by **DIAGnostic:SWITch:TIMe** <***number***> should be equal to or longer than the longest close/open time of the devices connected to the module. All 16 channels use the same delay time.

Reset Condition

*RST resets the delay time to default 13mS.

Related Commands

[ROUTe:]CLOSE, [ROUTe:]OPEN

DIAGnostic:SWITch:TIMe?

DIAGnostic:SWITch:TIMe? returns the current delay time value.

Returned Data

Type	Range	Default	Description
string	0.31875mS to 2089mS	+13	Current Delay Time

The DISPlay subsystem monitors the channel state of a selected module. The DISPlay command subsystem only operates with an RS-232 terminal connected to the HP E1406 command module's RS-232 port. These commands control the display on the terminal, and are generally executed directly from the terminal keyboard.

Subsystem Syntax

```
DISPlay
    :MONitor
        [:STATe]  ON | OFF | 1 | 0)
        [:STATe]?
```

DISPLay:MONitor[:STATe] <boolean>

DISPlay:MONitor[:STATe] turns the monitor mode ON or OFF. When monitor mode is on, the RS-232 terminal display presents an array of values indicating the open/close state of each channel on the module. This display is dynamically updated each time a switch is opened or closed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
<boolean>	boolean	ON | OFF | 1 | 0	OFF | 0

Comments

- **Monitoring Instrument State:** DISPlay:MONitor:STATe ON or DISPlay:MONitor:STATe 1 turns the monitor mode ON.
- Executing another command on the terminal will cause the DISPlay:MONitor[:STATe] to automatically be set to OFF. NOTE: Use of the OFF parameter is useful only if the command is issued across the HP-IB interface.
- **Monitor Mode Display:** The HP E2291A monitor display shows which channels are closed by displaying the channel number. Open channels do not display a channel number. For example:

  ```
  Chan:  ,1,2, , , , ,7, , , , ,11,12, , ,15
  ```

  shows channels 1, 2, 7, 11, 12, and 15 as closed; and channels 0, 3 - 6, 8 - 10, 13, and 14 as open.
- ***RST Condition:** Turns the monitor mode off. Opens all channels on the M-Module.

DISPLay:MONitor[:STATe]?

DISPlay:MONitor[:STATe]? returns the monitor mode state. The command returns a “1” if monitor mode is on or a “0” if monitor mode is off.

The [ROUTe] subsystem opens and closes channels on the HP E2291A. OPEN means the channel's output transistor is biased off, a high impedance between collector - emitter. CLOSED means the output transistor is biased on, a low collector - emitter impedance.

Subsystem Syntax

```
[ROUTe:]
    CLOSe <channel_list>
    CLOSe? <channel_list>
    OPEN <channel_list>
    OPEN? <channel_list>
```

[ROUTe:]CLOSe <*channel_list*>

[ROUTe:]CLOSe <*channel_list*> closes channels specified in the <*channel_list*>.

Parameters

Parameter Name	Parameter Type	Range of Values
<channel_list>	numeric	00-15

Comments

- The <*channel_list*> parameter in the CLOSe or OPEN command has the form (*@ccnn*) where *cc* is the card number and *nn* is the channel number. The SCPI Driver supports single modules only, therefore *cc* is always 1. To simplify programming, the card number (*cc*) can be eliminated.
- You can specify a single channel (*@nn*), use commas to specify multiple channels (*@nn,nn,...*), or use a colon to specify a range of channels (*@nn:nn*). You can also specify any combination of single channels, multiple channels, and channel ranges.
- A range of channels (*@nn:nn*) must be specified in ascending order, that is lower channel number on the left, higher channel number on the right.
- If any of the channels in the list cannot be closed, an execution error is reported.
- **Closure Order:** The order channels close when specified from a single command is not guaranteed. Use sequential CLOSe commands if needed.

Reset Condition

All 16 channels on the module are open.

Related Commands

[ROUTe:]OPEN, [ROUTe:]CLOSe?

Error Conditions

Error Number	Message	Probable Cause
-222	“Data out of range”	An invalid channel range was specified. Valid channel numbers are 00 to 15 or 100 to 115. Both channel elements in the range must be of the same type (either (@*nn*) or (@*ccnn*)) where *cc* is either 0 or 1.
2000	“Invalid Card Number”	A channel element was specified using the (@*ccnn*) format but *cc* was not 0 or 1.
2001	“Invalid Channel Number”	A channel element is improperly specified. Valid channel numbers are 0 - 15 or 100 - 115.
2009	“Channel number to big”	A channel element is improperly specified. Valid channel numbers are 0 - 15 or 100 - 115.
2011	“Empty channel list”	No channel element specified in the channel list.
2012	“Bad Range”	An invalid channel range was specified.

[ROUTe:]CLOSe? <*channel_list*>

[ROUTe:]CLOSe? <***channel_list***> returns the current state of the specified channel(s). The *channel_list* is in the form (@ccnn).

Comments

- **Query is Software Readback:** The [ROUTe:]CLOSe? command returns the current state of the hardware controlling the specified channel. It does not account for a failed switch element or a channel closed by direct register access.
- The command returns 1 for each channel in the list that is in the CLOSEd state or returns a 0 for each channel in the OPEN state. If a list of channels is queried, a comma delineated list of 0 or 1 values is returned in the same order of the channel list. *Note: The response of the [ROUTe:]CLOSe? is the opposite of the [ROUTe:]OPEN?*.
- The <*channel_list*> parameter has the form (@*ccnn*) where *cc* is the card number and *nn* is the channel number. The SCPI Driver supports single modules only, therefore *cc* is always 1. To simplify programming, the card number (*cc*) can be eliminated.
- You can specify a single channel (@*nn*), use commas to specify multiple channels (@*nn,nn,...*), or use a colon to specify a range of channels (@*nn:nn*). You can also specify any combination of single channels, multiple channels, and channel ranges.
- A range of channels (@*nn:nn*) must be specified in ascending order, that is lower channel number on the left, higher channel number on the right.

[ROUTe:]OPEN <*channel_list*>

[ROUTe:]OPEN <*channel_list*> opens channels specified in the <*channel_list*>.

Parameters

Parameter Name	Parameter Type	Range of Values
<channel_list>	numeric	00 to 15

Comments

- The <*channel_list*> parameter in the CLOSe or OPEN command has the form (*@ccnn*) where *cc* is the card number and *nn* is the channel number. The SCPI Driver supports single modules only, therefore *cc* is always 1. To simplify programming, the card number (*cc*) can be eliminated.
- You can specify a single channel (*@nn*), use commas to specify multiple channels (*@nn,nn,...*), or use a colon to specify a range of channels (*@nn:nn*). You can also specify any combination of single channels, multiple channels, and channel ranges.
- A range of channels (*@nn:nn*) must be specified in ascending order, that is lower channel number on the left, higher channel number on the right.
- If any of the channels in the list cannot be opened, an execution error is reported.
- **Opening Order:** The order channels open when specified from a single command is not guaranteed. Use sequential OPEN commands if needed.

Related Commands

[ROUTe:]CLOSe, [ROUTe:]OPEN?

Reset Condition

All module channels are open.

Error Conditions

Error Number	Message	Probable Cause
-222	“Data out of range”	An invalid channel range was specified. Valid channel numbers are 00 to 15 or 100 to 115. Both channel elements in the range must be of the same type (either (*@nn*) or (*@ccnn*)) where *cc* is either 0 or 1.
2000	“Invalid Card Number”	A channel element was specified using the (*@ccnn*) format but *cc* was not 0 or 1.
2001	“Invalid Channel Number”	A channel element is improperly specified. Valid channel numbers are 0 - 15 or 100 - 115.
2009	“Channel number to big”	A channel element is improperly specified. Valid channel numbers are 0 - 15 or 100 - 115.
2011	“Empty channel list”	No channel element specified in the channel list.
2012	“Bad Range”	An invalid channel range was specified.

[ROUTe:]OPEN?<*channel_list*>

[ROUTe:]OPEN? <*channel_list*> returns the current state of the specified channel(s). The *channel_list* is in the form (@ccnn).

Comments

- **Query is Software Readback:** The [ROUTe:]OPEN? command returns the current state of the hardware controlling the specified channel. It does not account for a failed switch element.
- The command returns 1 for each channel in the list that is in the OPEN state or returns a 0 for each channel in the CLOSEd state. If a list of channels is queried, a comma delineated list of 0 or 1 values is returned in the same order of the channel list. *Note: The response of the [ROUTe:]CLOSe? is the opposite of the [ROUTe:]OPEN?.*
- The <*channel_list*> parameter command has the form (@*ccnn*) where *cc* is the card number and *nn* is the channel number. The SCPI Driver supports single modules only, therefore *cc* is always 1. To simplify programming, the card number (*cc*) can be eliminated.
- You can specify a single channel (@*nn*), use commas to specify multiple channels (@*nn,nn,...*), or use a colon to specify a range of channels (@*nn:nn*). You can also specify any combination of single channels, multiple channels, and channel ranges.
- A range of channels (@*nn:nn*) must be specified in ascending order, that is lower channel number on the left, higher channel number on the right.

SCPI uses four status groups - the Status Byte, the Standard Event status group, the Operation status group, and the Questionable Data status group.

Note This subsystem is included in the SCPI driver for compatibility reasons only. None of the events in the STATus:OPERation or STATus:QUEStionable registers are used by the HP E2291A.

Each status group consists of a condition register, transition filters, event register, and enable register. The STATus subsystem consists of commands set and query the registers.

Subsystem Syntax

```
STATus
    :OPERation
        :CONDition?              Returns Condition Register
        :ENABle  <unmask>        Sets Enable Register
        :ENABle?                 Returns Enable Register value
        :EVENt?                  Returns Event Register value
    :PRESet                      Clears operation/questionable enable registers
    :QUEStionable
        :CONDition?              Sets Condition Register
        :ENABLE <mask>           Sets Enable Register
        :ENABle?                 Returns Enable Register value
        :EVENt?                  Returns Event Register value
```

The SYSTem subsystem queries system information.

Subsystem Syntax

```
SYSTem
        :ERRor?
        :VERSion
```

SYSTem:ERRor?

SYSTem:ERRor? returns an error number and corresponding error message in the error queue.

Returned Data

Type	Range	Default	Description
int16	-32768 to +32767	none	Error Number
string			Error Message

Comments

- The response format is: **`error_number,"error description string"`**
- **Error Numbers/Messages in the Error Queue:** Each error generated has both an error number and corresponding error message. The error message can be up to 255 characters long.
- **Clearing the Error Queue:** An error number/message is removed from the queue each time the SYSTem:ERRor? command is sent. The errors are cleared first-in, first-out. When the queue is empty, each following SYSTem:ERRor? query returns **`+0, "No error"`**. To clear all error numbers/messages in the queue, execute the *CLS command.
- **Maximum Error Numbers/Messages in the Error Queue:** The queue holds a maximum of 30 error numbers/messages. If the queue overflows, the last error number/message in the queue is replaced by -350, "Too many errors". The least recent error numbers/messages remain in the queue and the most recent are discarded.

Reset Condition

*RST does not affect the status system. Use *CLS to clear the error queue.

SYSTem:VERSion?

SYSTem:VERSion? returns the version of SCPI this instrument complies with.

Returned Data

Type	Range	Default	Description
string	"1990.0"	none	SCPI Version

The following table summarizes the SCPI commands for the HP E2291A 16-Channel Isolated Digital Output Module.

DIAGnostic	:INTerrupt[:LINe] *<number>* :INTerrupt[:LINe]? :TEST? :SWITch:TIMe *<number>* :SWITch:TIMe?	Set M-Module interrupt level. Query interrupt level. Do diagnostic test to fix specific error. Set delay time between command execution and interrupt. Query the delay time.
DISPlay	:MONitor[:STATe] ON|OFF|1|0 :MONitor[:STATe]?	Turns monitor mode on or off. Queries the monitor mode.
[ROUTe:]	CLOSe <*channel_list*> CLOSe? <*channel_list*> OPEN <*channel_list*> OPEN? <*channel_list*>	Closes channel(s). Queries channel(s) closed. Opens channel(s). Queries channel(s) opened.
STATus	:OPERation:CONDition? :OPERation:ENABle :OPERation:ENABle? :OPERation[:EVENt]? :PRESet	Returns contents of the Operation Condition Register. Enables events in the Operation Event Register to be reported. Returns the mask value set by the :ENABle command. Returns the contents of the Operation Event Register. Sets Enable Register bits to 0.
SYSTem	:ERRor? :VERSion	Returns error number/message in Error Queue. Returns the version number of SCPI driver.

IEEE 488.2 Common Command Reference

The following table lists the IEEE 488.2 Common (*) commands accepted by the HP E2291A 16-Channel Isolated Digital Output M-Module. The operation of some of these commands is described in Chapter 2 of this manual. For more information on Common commands, refer to the user's manual for your HP E1400/E1401 Mainframe, or the ANSI/IEEE Standard 488.2-1987.

Command	Command Description
*CLS	Clears all status registers (see STATus:OPERation[:EVENt]?) and clears the error queue.
*ESE<*unmask*>	Enable Standard Event.
*ESE?	Enable Standard Event Query.
*ESR?	Standard Event Register Query.
*IDN?	Instrument ID Query; returns identification string of the module.
*OPC	Operation Complete.
*OPC?	Operation Complete Query.
*RCL<*n*>	Recalls the instrument state saved by *SAV.
*RST	Resets the module. Opens all channels.
*SAV<*n*>	Stores the instrument state **but does not** save the channel list.
*SRE<*unmask*>	Service request enable, enables status register bits.
*SRE?	Service request enable query.
*STB?	Read status byte query.
*TST?	Self-test. Executes an internal self-test and returns only the first error encountered. Does not return multiple errors. The following is a list of responses you can obtain. +0 if self test passes. +01 for firmware error.
*WAI	Wait to Complete.

Chapter 4
Register Descriptions

About This Chapter

This chapter describes how to program the HP E2291A at the register level in an HP E2251 M-Module Carrier installed in a VXIbus mainframe. Register programming is recommended only if you are unable to use the module's higher-level VXI*plug&play* or SCPI driver. For information on using the VXI*plug&play* driver, refer to the on-line help on the CD ROM.

Block Diagram Description

In order to register program the HP E2291A, it is important to understand its operation at the block diagram level. Figure 4-1 shows the block diagram of the HP E2291A. The following paragraphs describe the major sections of the module.

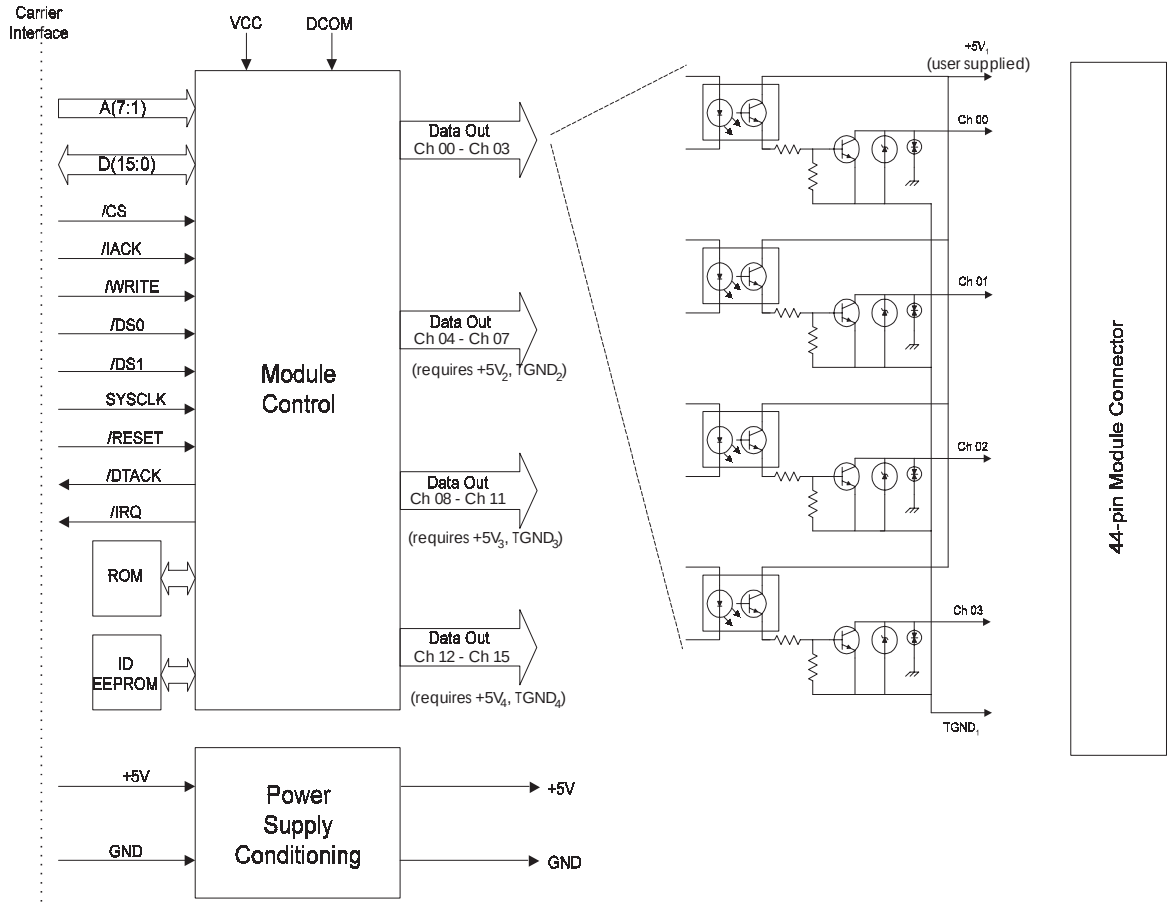


Figure 4-1. HP E2291A Block Diagram

Module Control

This block contains all of the logic for the module including all registers, Carrier interface and driver control.

ID EEPROM

The EEPROM holds sixty-four 16-bit words of M-Module ID information and VXI M-Module data. Refer to “ID EEPROM Register” on page 50 for EEPROM contents.

Output Driver and Output Circuit

The Output Register directly drives the channel output optical isolators. Refer to “Simplified E2291A User Connections and Simplified Schematic” on page 13. The channel optical isolators are powered by external power supplies providing channel isolation. Each optical isolator drives one NPN transistor configured as an open collector driver.

Delay Timer

The module asserts an interrupt when a channel OPEN or CLOSE command completes. If you are using this M-Module to control relays for example, you may not want the interrupt to be asserted until after the relays have settled. By setting a delay time equal to or greater than the relay settling time, the M-Module will not assert the interrupt until after the relays have settled. The default value of the timer is 13 mS.

Power Conditioning

This block filters +5VDC power to produce VCC power (+5VDC) for logic and isolates the various grounds used by the module.

Register Addressing in the VXI Environment

Logical Address

Each module in a VXIbus system, whether standard VXI modules or M-Modules, must have a unique logical address. The HP E2251A Carrier provides a logical address for each installed M-Module. Refer to the *HP E2251A Installation and Wiring Manual* for details (if you are using a different carrier, refer to that carrier's documentation for register-based addressing information).

Register Addresses for register-based devices are located in the upper 25% of VXI A16 address space. Every VXI device (up to 256 devices) is allocated a 32-word (64-byte) block of addresses.

The HP E2291A is a register based device with two memory windows. One is the same as any other VXIbus register based device (in the A16 address space), all the configuring registers are stored in this area. The second is the real I/O registers which located in the A24 address space. Below paragraphs describe how to determine the registers addresses.

A16/A24 Memory Mapping

The VXI Specification allows for only 64 bytes of address space in A16 memory. However, the M-Module specification defines 256 bytes of address space. To resolve this conflict, the HP E2251 Carrier provides two memory segments for each installed M-Module. The first is in VXI A16 memory space and contains the standard VXI registers. The second memory segment is in the A24 memory space and contains all other M-Module registers (these are described starting on Page 48. Figure 4-2 shows the A16/A24 mapping for a typical M-Module.

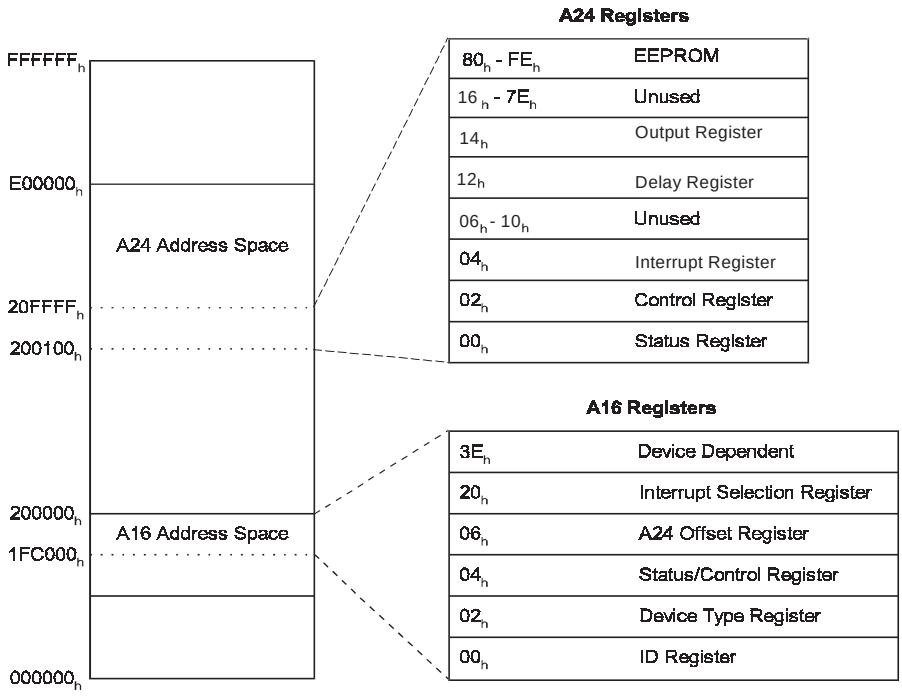


Figure 4-2. A16/A24 Register Mapping

Determining a Module's A16 Base Address

To access a register in A16 memory, you must specify a hexadecimal or decimal register address. This address consists of a base address plus a register offset. The A16 base address depends on whether or not you are using an HP E1406 Command Module.

A16 Address Space Outside the Command Module

When the HP E1406 Command Module is <u>not</u> part of the VXIbus system, the module's base address is computed as:

$C000_h + (LADDR_h * 40_h)$ *or decimal* 49,152 + (LADDR * 64),

Where:

$C000_h$ (49,152) is the starting location of the register addresses,
LADDR is the M-Module's logical address
40_h (64) is the number of address bytes allocated per VXI device.

For example, if one M-Module's logical address is 120 (78_h), the module's register will have a base address of:

$C000_h + (78_h * 40_h) =$ **$DE00_h$**
or
49,152 + (120 * 64) = 49,152 + 7,680 = **56,832**

A16 Address Space Inside the Command Module or Mainframe

When the A16 address space is inside the HP E1406 Command Module, the M-Module's base address is computed as:

$1FC000_h + (LADDR_h * 40_h)$ *or decimal* $2,080,768_{10}$ + (LADDR * 64)

Where:

$1FC000_h$ or 2,080,768 is the VXI A16 starting address,
LADDR is the M-Module's logical address,
40_h (64) is the number of address bytes allocated per VXI device.

For example, if the M-Module's logical address is 78_h (120), then the M-Module will have a base register address of:

$1FC000_h + (78_h * 40_h) =$ **$1FDE00_h$**

or (decimal)

$2,080,768 + (120 * 64_0 = 2,080,768 + 7,680 =$ **2,080,768**

Addressing A16 Registers

Figure 4-2 shows that VXI registers for an M-Module are mapped into A16 address space. To access one of these registers, add the A16 base address to the register offset. For example, an M-Module's VXI Status/Control Register has an offset of 04_h. To access this register (assuming the system does not have an HP E1406 Command Module), use the register address:

$1FDE00_h + 04_h = 1FDE04_h$
or decimal
2,088,488 + 4 = 2,088,452

For the HP E2291A, there are only five registers in the A16 address space:

- VXI ID Register at offset 00_h
- VXI Device Type Register at offset 02_h
- VXI Status/Control Register at offset 04_h
- VXI A24 Offset Register at offset 06_h
- M-Module's Interrupt Control Register at offset 08_h

Addressing A24 Registers

Figure 4-2 shows that most of the HP E2291A registers are mapped into A24 memory space. To access these registers:

1. Obtain the A24 base address by reading the VXI A24 Offset Register (06_h) in A16 memory.

2. Add the A24 base address to the register offset (see "Registers in A24 Address Space" on page 48).

For example, if the A24 base address is 200100_h, then to access the I/O Register (10_h offset):

$200100_h + 10_h = 200110_h$
or decimal
2,097,408 + 16 = 2,097,424

Program Example

The following C language program demonstrates how to program at the register level. The program reads the ID, Device Type, Status, and A24 Offset Registers then sets channels 00 and 02 to HI state. This program was written and tested in Microsoft Visual C++ but should compile under any standard ANSI C compiler.

To run this program you must have the HP SICL library, the HP VISA library, an HP-IB interface module installed in your PC, and an HP E2406 Command Module.

```
#include <visa.h>
#include <stdio.h>
#include <stdlib.h>

ViSession viRM,m_mod;
int main()
{

unsigned short id_reg,dt_reg ;                  /* ID & Device Type Registers */
unsigned short stat_reg, a24_offset ;           /* Status Register & A24 offset
                                                   register */
short value;

ViStatus errStatus;                             /*Status from each VISA call*/

    /* Open the default resource manager  */
errStatus = viOpenDefaultRM ( &viRM);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viOpenDefaultRM() returned 0x%x\n",errStatus);
   return errStatus;}

    /* Open the M-Module instrument session ; Logical Address = 8 */
errStatus = viOpen(viRM,"GPIB-VXI0::8",VI_NULL,VI_NULL,&m_mod);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viOpen() returned 0x%x\n",errStatus);
   return errStatus;}

    /* read  and print the module's ID Register  */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x00,&id_reg);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viIn16() returned 0x%x\n",errStatus);
   return errStatus;}
printf("ID register = 0x%4X\n", id_reg);

    /* read  and print the module's Device Type Register   */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x02,&dt_reg);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viIn16() returned 0x%x\n",errStatus);
   return errStatus;}
printf("Device Type register = 0x%4X\n", dt_reg);
```

```
    /* read and print the module's Status Register  */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x04,&stat_reg);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viIn16() returned 0x%x\n",errStatus);
   return errStatus;}
printf("Status register = 0x%hx\n", stat_reg);

    /* read and print the module's A24 Offset Register  */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x06,&a24_offset);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viOpen() returned 0x%x\n",errStatus);
   return errStatus;}
printf("A24 Offset register value = 0x%hx\n", a24_offset);

    /* Drive Bits 00 and 02 to HI State */
errStatus = viOut16(m_mod,VI_A24_SPACE,0x14,0x05);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viOut16() returned 0x%x\n",errStatus);
   return errStatus;}

    /* Close the M-Module Instrument Session */
errStatus = viClose (m_mod);
if (VI_SUCCESS > errStatus) {
   printf("ERROR: viClose() returned 0x%x\n",errStatus);
   return 0;}

    /* Close the Resource Manager Session */
errStatus = viClose (viRM);
   if (VI_SUCCESS > errStatus) {
   printf("ERROR: viClose() returned 0x%x\n",errStatus);
return 0;}

return VI_SUCCESS;
}
```

Registers in A16 Memory Space

Table 4-1 lists the five registers in the A16 memory space. The following paragraphs describe each register.

Table 4-1. VXIbus A16 Memory Instrument Registers

Address Mapping	Registers
00_{16}	VXI ID Register
02_{16}	VXI Device Type Register
04_{16}	VXI Status/Control Register
06_{16}	VXI Offset Register
20_{16}	M-Module Interrupt Control Register

VXI ID Register

The ID Register is a read only register at address 00_h (MSB) and 01_h (LSB).

b+00$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read	Device Class		Address Space		Manufacturer ID											

- **Device Class:** this field should always be 11 indicating a register-based device.
- **Address Space:** 00 indicating A16/A24 device
- **Manufacturer ID**: 4095 (decimal) for Hewlett-Packard M-Modules

VXI Device Type Register

The Device Type Register is a read only register at address 02_h (MSB) and 03_h (LSB). Reading this register returns a unique identifier for each M-Module.

b+02$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read	Required Memory				M-Module Model Code											

- **Required Memory**: F_h indicating 256 byte block required.
- **M-Module Model Code**: $F261_h$ for the HP E2291A.

VXI Status/Control Register

The Status/Control Register is a read/write register (address 04_h and 05_h) that controls the module and indicates its status.

b+04$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write (Control)	A24 Enable	Reserved													Sysfail Inhibit	Reset
Read (Status)	A24 Active	MODID*	M-Module Device Dependent										Ready	Passed	Device Dependent	

- **A24 Enable.** A 1 in this field means access to the devices A24 registers is enabled.
- **Sysfail Inhibit.** Writing a 1 disables the M-Module from driving the SYSFAIL* line.
- **Reset.** Writing a 1 to this field forces the M-Module to reset.
- **A24 Active.** A 1 in this field indicates the M-Module's registers in A24 memory space can be accessed. Default = 1.
- **MODID*.** A 1 in this field indicates that the M-Module is not selected via the P2 MODID line. A 0 indicates the M-Modules is selected by a high state on the P2 MODID line.
- **Ready.** A 1 in this field indicates that the M-Module is ready to accept commands. A 0 indicates the M-Module is busy and not ready to accept commands.
- **Passed.** A 1 in this field indicates the M-Module passed its self test successfully. A 0 indicates the M-Module is either executing or has failed its self test.

VXI Offset Register

The Offset Register (address 06_h and 07_h) contains the value of the base address for accessing registers in the A24 address space.

b+06$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	A24 Space Base address for those M-Modules needing A24 memory															
Read	A24 Space Base address for those M-Modules needing A24 memory															

Interrupt Selection Register

The Interrupt Selection Register (base + 20_h) specifies which VXI interrupt line the M-Module will use. M-Modules may generate interrupts to indicate that a SCPI command has completed. These interrupts are sent to and acknowledged by the HP Command Module or other system controller via one of seven VXI backplane interrupt lines. Different controllers treat the interrupt lines differently, and you should refer to your controller's documentation to determine how to set the interrupt level. HP Command Modules configured as VXI Resource Managers treat all interrupt lines as having equal priority. For interrupters using the same line, priority is determined by which slot they are installed in; lower-numbered slots have higher priority than higher-numbered slots. HP Command Modules service line 1 by default, so it is normally correct to leave the interrupt level set to the factory default of IRQ1.

b+20$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Reserved												INT	VXI Interrupt Line		
Read (default value)	Reserved												INT 1	VXI Interrupt Line 0	0	1

If your controller's documentation instructs you to change the interrupt level, you need to specify the level in the VXI Interrupt Selection Register. To cause the M-Module to interrupt on one of the VXI interrupt lines, write to the appropriate bits (refer to table below). To disable the module's interrupt, set the bits to 000. Selecting other than the default interrupt line 1 is not recommended. Reading the default value of this register returns the value XXX9_h.

Bits 2 - 0	Selected Interrupt Line
000	NONE (Interrupt Disabled)
001	IRQ1 (default)
010	IRQ2
011	IRQ3
100	IRQ4
101	IRQ5
110	IRQ6
111	IRQ7

M-Module specifications define three types of interrupts. The INT bit (bit 3) determines which M-Module interrupt style is supported. If INT is set to a 0, the M-Module supports interrupt types A and B. If INT is set to a 1, the M-Module supports interrupt type C (this is the default).

Type A Interrupts The interrupting M-Module removes the interrupt request upon a register access (software method) to the interrupting M-Module (such as reading the Status Register). DTACK* is not asserted during interrupt acknowledge.

Type B Interrupts The interrupting M-Module removes the interrupt request via a hardware method (on IACK* going low) but provides no vector information for the interrupt. This is the same as Type C interrupts except that no vector is supplied and DTACK* is not asserted.

Type C Interrupts The interrupting M-Module removes the interrupt request via a hardware method and provides an interrupt vector on the data bus and DTACK* is asserted during the interrupt acknowledge cycle. The M-Module removes the interrupt request by IACK* going low.

In VXI specifications however, only two types of interrupts are defined; RORA (Release on Register Access) and ROAK (Release on Acknowledge). The HP E2251A Carrier converts M-Module Type A interrupts to RORA and Types B and C interrupts to ROAK (default).

RORA Interrupts The interrupting device provides its logical address on the data bus (D0 - D7) during the interrupt acknowledge cycle that was initiated in response to its interrupt request. It does not remove the interrupt request until its Status/Control register is accessed.

ROAK Interrupts The interrupting device removes the interrupt request upon the presence of a properly addressed interrupt acknowledge cycle and provides its logical address on the data bus (D0 - D7). A cause/status byte is also placed on the data bus (D15 - D8)

Registers in A24 Address Space

Table 4-2 lists the six registers in A24 memory. The following paragraphs describe each register.

Table 4-2. HP E2291A A24 Memory Registers

Address Mapping	Registers
FE_h	EEPROM
(16_h - FD_h)	(Reserved)
14_h	Output Register
12_h	Delay Register
10_h	Not Used
(06_h - $0F_h$)	(Reserved)
04_h	Interrupt Register
02_h	Control Register
00_h	Status Register

Status Register

The offset of Status Register is 00_{16}. It is a Read only register.

b+00_h	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read									BUSY*							RIRQX

- BUSY*: 0-Relay is busy (not stable yet).
- RIRQX: 1-Relay interrupt.

Control Register

The offset of Control Register is 02_{16}. It is a Read/Write register.

b+02_h	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read															Re-enable	Soft Reset

- RENABLE: 1 - Enable relay interrupt (After BUSY timer).
- Soft Reset: 1 - Soft Reset the M-Module.

When power-on or reset, all bits of Control Register are set to zero.

Interrupt Register

The offset of Interrupt Register is 04_{16}. It is a Read only register.

b+04$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read																RIRQX

- RIRQX: Relay interrupt

Delay Register

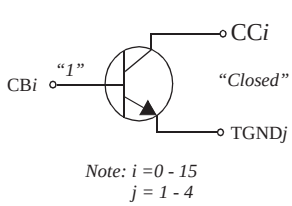
The offset of HP E2291A Delay Register is 12_h. It is a write only register. The value of this register indicates the delay time between command execution and asserting interrupt. The register has a default delay time of **13ms**. It's value can be determined by the formula:

Delay Time = (Register Value + 1) * 0.031875ms

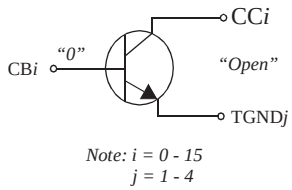
where: Register Value = 0000_h - $FFFF_h$

Output Register

The offset of the Output Register is 14_{16}, it's a read/write register. Writing a "1" to a specific bit closes that channel; writing a 0 opens that channel. Closed means the output transistor is biased on - a low collector-emitter impedance. Open means the output transistor is biased off - a high collector-emitter impedance.



Write a "1" to Close a Channel



Write a "0" to Open a Channel

Figure 4-3. Write a "1" or "0" to Close/Open One Channel

b+14$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Ch 15	Ch 14	Ch 13	Ch 12	Ch 11	Ch 10	Ch 9	Ch 8	Ch 7	Ch6	Ch 5	Ch 4	Ch 3	Ch 2	Ch 1	Ch 0
Read	Ch 15	Ch 14	Ch 13	Ch 12	Ch 11	Ch 10	Ch 9	Ch 8	Ch 7	Ch6	Ch 5	Ch 4	Ch 3	Ch 2	Ch 1	Ch 0

ID EEPROM Register

The ID EEPROM Register allows you to access the contents of the ID EEPROM. The ID EEPROM contains sixty-four 16-bit words of M-Module ID data and VXI M-Module data.

Note

This register is intended to be used by the higher-level software driver. The software driver must perform a series of many reads and writes to this register to perform the required functions within the EEPROM. **When register programming, it is much easier to read the module ID data from the VXI registers (A16 memory area) instead of reading the ID EEPROM Register.** A16 addressing is discussed earlier in this chapter. Do NOT attempt to read the ID EEPROM. Do not attempt to read the EEPROM Registers.

b+80$_h$ - b+FE$_h$	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Write	Unused															
Read (default value)														Chip Select	Clock	Data In/Out

Reset Condition -- Bits 15 - 08 = logic “1”, Bits 07 - Bit 00 = logic “0”.

Caution

Do not attempt to write to Bit 00 of the ID EEPROM register. You could possibly write-over the contents of the ID EEPROM.

Bit Definitions

Data In/Out -- Reading this bit returns the value returned from the Data Out pin of the ID EEPROM.

Clock -- Writing a logic “1” to this bit forces the SK pin of the ID EEPROM high and writing a logic “0” drives it low. This bit is used as a clock to the ID EEPROM for reading data out. Reading this bit always returns logic “0”.

Chip Select -- Writing a logic “1” to this bit selects the ID EEPROM. Writing a logic “0” to this bit deselects the EEPROM. Reading this bit always returns logic “0”.

Table 4-3. ID EEPROM Contents

Word Number	Description	ID EEPROM Contents
0	Sync Code	5346_h
1	M-Module Number (binary code)	$069B_h$ (binary-coded 1690)
2	Revision Number (binary code)	0001_h
3	Module Characteristics	0868_h
4 - 7	Reserved	0000_h
8 - 15	User Defined	0000_h
16	VXI Sync Code	$ACBA_h$
17	VXI ID	$CFFF_h$ (Hewlett Packard)
18	VXI Device Type	$F261_h$
19 - 31	Reserved	0000_h
32 - 63	User Defined	0000_h

Appendix A
Specifications

M-Module Specification Compliance

HP E2291A, the 16-Channel Open Collector M-module complies with the Mezzanine Concept M-Module Specification.

HP E2291A 16-Channel Isolated Digital Output M-Module

Number of Channels:

- 16 Open Collector (arranged as four groups of four channels)

Open Collector-Emitter Voltage:

- 36V max.

Saturation Collector-emitter Voltage (DC):

- <0.5V

Sink Current (per Channel):

- 200mA

Isolation voltage:

- 42VDC

Programmable Busy Time:

- 0.031875 mS to 2089 mS

General Characteristics:

- Connector Type: 44-pin D-Sub (female)

Power Requirements:

	I_{PM} (A)	I_{DM} (A)
+ 5VDC	0.200	0.190
+12VDC	0	0
-12VDC	0	0

- External +5VDC Power Supplies; up to four required. Used for powering isolation optical isolators.

Symbols

*CLS, 36
*ESE, 36
*ESE?, 36
*ESR?, 36
*IDN?, 36
*OPC, 36
*OPC?, 36
*RCL, 36
*RST, 36
*SAV, 36
*SRE, 36
*SRE?, 36
*STB?, 36
*TST?, 36
*WAI, 36

A

A16 Address Space, 40
A24 address space, 39
A24 Offset Register, 45
Abbreviated Commands, 21

C

CLOSe, 29
CLOSe?, 30
Command Module, 40
Command Quick Reference, 36
Command Separator, 21
Command Types, 21
Commands
  abbreviated, 21
  IEEE 488.2, 21, 36
  implied, 22, 22
  linking, 22
  parameters, 22
  Quick Reference, 35, 36
  types, 21
Common (*) Commands, 21, 36
  *CLS, 36
  *ESE, 36
  *ESE?, 36
  *ESR?, 36
  *IDN?, 36
  *OPC, 36
  *OPC?, 36
  *RCL, 36
  *RST, 36
  *SAV, 36
  *SRE, 36
  *SRE?, 36
  *STB?, 36
  *TST?, 36
  *WAI, 36
Compliance, M-Module specification, 53

D

Description, module, 9
Device Type Register, 41, 44
DIAGnostic Subsystem, 24
DIAGnostic:INTerrupt:LINe, 24
DIAGnostic:INTerrupt:LINe?, 25
DIAGnostic:SWITch:TIMe, 26
DIAGnostic:SWITch:TIMe?, 27
DIAGnostic:TEST?, 25
DISPlay Subsystem, 28
DISPLay:MONitor:STATe, 28
DISPLay:MONitor:STATe?, 28

E

ERRor?, 34

I

ID Register, 44
IEEE 488.2 Common Commands, 21, 36
Implied Commands, 22
Interrupt Selection Register, 46

Interrupts
ROAK, 47
RORA, 47
Type A, 46
Type B, 47
Type C, 47

L

Linking Commands, 22

M

Mainframe, 40
M-Module specification compliance, 53
Module description, 9
Module Specifications, 53

O

Offset Register, 45
OPEN, 31
OPEN?, 32

P

Parameters, 22
types, 22

R

Register
Device Type, 44
Device Type Register, 41
ID, 44
ID Register, 41
Interrupt Selection, 46
Offset, 45
Status/Control, 45
Register Address, 39
register-based device, 39
ROAK Interrupts, 47
RORA Interrupts, 47
ROUTe Subsystem, 29
ROUTe:CLOSe, 29, 29
ROUTe:CLOSE?, 30
ROUTe:OPEN, 31
ROUTe:OPEN?, 32

S

SCPI Command Format, 21
SCPI Command Reference
DIAGnostic:INTerrupt:LINe, 24
DIAGnostic:INTerrupt:LINe?, 25
DIAGnostic:SWITch:TIMe, 26
DIAGnostic:SWITch:TIMe?, 27
DIAGnostic:TEST?, 25
DISPLay:MONitor:STATe, 28
DISPLay:MONitor:STATe?, 28
ROUTe:CLOSe, 29
ROUTe:CLOSE?, 30
ROUTe:OPEN, 31
ROUTe:OPEN?, 32
SYSTem:ERRor?, 34
SYSTem:VERSion?, 34
SCPI Commands, 21
abbreviated, 21
format, 21
implied, 22
linking multiple commands, 22
linking with Common commands, 22
parameters, 22
quick reference, 35
Specification compliance, M-Module, 53
Specifications, 53
STATus Subsystem, 33
Status/Control Register, 45
Subsystem
DIAGnostic, 24
DISPlay, 28
ROUTe, 29
STATus, 33
System, 34
System Subsystem, 34
SYSTem:ERRor?, 34
SYSTem:VERSion?, 34

T

Type A Interrupts, 46
Type B Interrupts, 47
Type C Interrupts, 47

V

VERSion?, 34
VXI Device Type Register, 44
VXI ID Register, 44
VXI Offset Register, 45
VXI Status/Control Register, 45