

ARTISAN®
TECHNOLOGY GROUP

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

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

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

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

Custom engineering
so your equipment works exactly as you specify.

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

Expert team | Trust guarantee | 100% satisfaction

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

Find the *Keysight / Agilent Z2417A* at our website: *Click HERE*

HP 75000 Series C

32-Channel Mercury-Wetted Form C Switch Module HP Z2417A

User's Manual

Copyright © Hewlett-Packard Company, 1991

Manual Part Number: Z2417-90001

Z2417-90001

Printed January 1992 Edition 1
Printed in U.S.A

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Printing History

The Printing History shown below lists all Editions and Updates of this manual and the printing date(s). The first printing 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 the current Edition of the manual. Updates are numbered sequentially starting with Update 1. When a new Edition is created, it contains all the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this printing history page. Many product updates or revisions do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one–to–one correspondence between product updates and manual updates.

Edition 1 (Part Number Z2417–90001) January 1992

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 equipment protects 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 fuseholders.

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 one year from date of shipment. Duration and conditions of warranty for this product may be superceded 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.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause at 52.227–7013. Hewlett–Packard Company; 3000 Hanover Street; Palo Alto, California 94304

Hazardous Material

This product contains Mercury. When disposing of this product follow the local law for disposal of articles which contain liquid mercury. There is approximately 3 grams of liquid mercury in this product. The relays may be removed from the board by carefully unsoldering the leads, then lifting the relay out of the holes. After ALL relays are removed the board may be disposed of. The relays must be disposed as hazardous goods or sent to a qualified mercury reclaimer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Getting Started **1–1**
Using This Chapter 1–1
Instrument Definition 1–1
Switch Description 1–1
Basic Operation 1–1
Typical Configuration 1–3
Programming the Switch 1–3
Specifying SCPI Commands 1–4
The Channel List 1–4
The Card Number 1–4
The Channel Number 1–4
Initial Operation 1–5

Configuring the Form C Switch Module **2–1**
Using This Chapter 2–1
Warnings and Cautions 2–1
Setting the Module Address Switch 2–2
Selecting the Interrupt Priority 2–3
Connecting User Inputs 2–5
Typical Switch Configurations 2–7
General Purpose Relay Configuration 2–7
Example: General Purpose Relay Configuration 2–7
Digital Output Configuration 2–8
Example: Digital Output Configuration 2–8
Protecting your Relays and Circuits 2–9
Relay Reliability 2–9
Adding Relay and Circuit Protection 2–10
Maximum Allowable Module Switch Current 2–11

Using the Form C Switch **3–1**
Using This Chapter 3–1
Form C Switch Commands 3–1
Power–On and Reset Conditions 3–2
Module Identification 3–2
Switching Channels 3–4
Example: Voltage Switching 3–5
Example: Controlling RF Switches/Step Attenuators 3–6
Example: Digital Output Configuration 3–7
Example: Matrix Switching 3–8
Scanning Channels 3–9
Example: Scanning Channels With an External Instrument Using Trig In and Out Ports 3–9
Example: Scanning Channels with System Multimeter Using TTL Trigger 3–11
Querying the Form C Switch Module 3–13
Using the Scan Complete Bit 3–13
Recalling and Saving States 3–14
Detecting Error Conditions 3–15
Synchronizing the Form C Switch 3–17

Form C Switch Command Reference **4–1**
Using This Chapter 4–1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Types 4-1
Common Command Format 4-1
SCPI Command Format 4-2
Command Separator 4-2
Abbreviated Commands 4-2
Implied Commands 4-2
Parameters 4-3
Linking Commands 4-3
SCPI Command Reference 4-4
ABORt 4-5
ARM 4-6
:COUNt 4-6
:COUNt? 4-7
DISPlay 4-8
:MONitor[:STATe] 4-8
:MONitor:CARD 4-9
INITiate 4-10
:CONTinuous 4-10
:CONTinuous? 4-11
[:IMMediate] 4-11
OUTPut 4-12
:EXTernal[:STATe] 4-12
:EXTernal[:STATe]? 4-13
[:STATe] 4-13
[:STATe]? 4-13
:TTLTrg <line>[:STATe] 4-14
:TTLTrg <line>[:STATe]? 4-14
[ROUTe:] 4-15
CLOSe 4-15
CLOSe? 4-16
OPEN 4-16
OPEN? 4-17
SCAN 4-17
STATus 4-18
:OPERation[:EVENt]? 4-18
:OPERation:ENABle 4-18
:OPERation:ENABle? 4-19
SYSTem 4-20
:ERRor? 4-20
:CDEScription? 4-20
:CTYPe? 4-21
:CPON 4-21
TRIGger 4-22
[:IMMediate] 4-22
:SOURce 4-23
:SOURce? 4-25
IEEE 488.2 Common Commands 4-26
Command Quick Reference 4-27

Specifications **A-1**

Form C Switch Registers **B-1**
Using this Appendix B-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Addressing the Registers B-1
The Base Address B-3
Address Space Outside the Command Module B-3
Address Space Inside the Command Module B-3
Register Offset B-3
Register Descriptions B-4
Reading and Writing to the Registers B-4
The Manufactures Identification Register B-4
The Device Identification Register B-4
The Status/Control Register B-4
Reading the Status/ Control Register B-4
Enable (bit 6) B-5
Busy (bit 7) B-5
Modid Select (bit 14) B-5
Writing to the Status/ Control Register B-5
Soft Reset (bit 0) B-5
Enable (bit 6) B-5
The Relay Control Registers B-5
Register-Based Programming Examples B-7
Reading the Registers B-7
HP BASIC B-7
HP-UX, C B-8
Making Measurements B-8
HP BASIC B-8
HP-UX, C B-9
Scanning Channels B-11
HP BASIC B-11
HP-UX,C B-12

Form C Switch Error Messages C-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1

Getting Started

Using This Chapter

This chapter includes a Form C switch description, addressing guidelines, and an example program to check initial operation. Chapter contents are as follows:

- Instrument Definition
- Switch Description
- Programming the Switch
- Initial Operation

Instrument Definition

HP plug–in modules installed in an HP mainframe are treated as independent instruments each having a unique secondary HP–IB address. Each instrument is also assigned a dedicated error queue, input and output buffers, status registers and, if applicable, dedicated mainframe memory space for readings or data. An instrument may be composed of a single plug–in module (such as a counter) or multiple plug–in modules (for a Switchbox or Scanning Voltmeter Instrument).

Switch Description

The HP Z2417A 32–Channel Mercury–Wetted, Form C Switch module is a C–Size VXIbus and VMEbus register–based product which can be used for switching, scanning, and control. The switch can operate in a C–Size VXIbus or VMEbus mainframe. The switch has 32 channels of Form C relays. Each channel includes a relay with common (C), normally open (NO), and normally closed (NC) contacts.

For the Form C switch, switching consists of opening or closing a channel relay to provide alternate connections to user devices. Scanning consists of closing a set of relays, one at a time.

Basic Operation

As shown in figure 1–1, the Form C switch module consists of 32 channels (channels 00 through 31). Each channel uses a nonlatching relay. Resistors or fuses can be added for circuit protection.

Each channel is switched by opening or closing the appropriate channel relay. Since the relays are nonlatching, the relays are all open during power–up or power–down. When a reset occurs, all channel commons (C) are connected to the corresponding normally closed (NC) contacts. When a channel is closed, the common contact (C) is connected to the normally open contact (NO). User inputs and outputs to each channel are via the NO, NC, and C terminal connectors on the terminal block.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Warning

The relays are mercury wetted. This means that there may not be break before make or make before break. During the switching of a relay all 3 contacts may be connected, or they may be separate, or only 2 of the contacts may be connected.

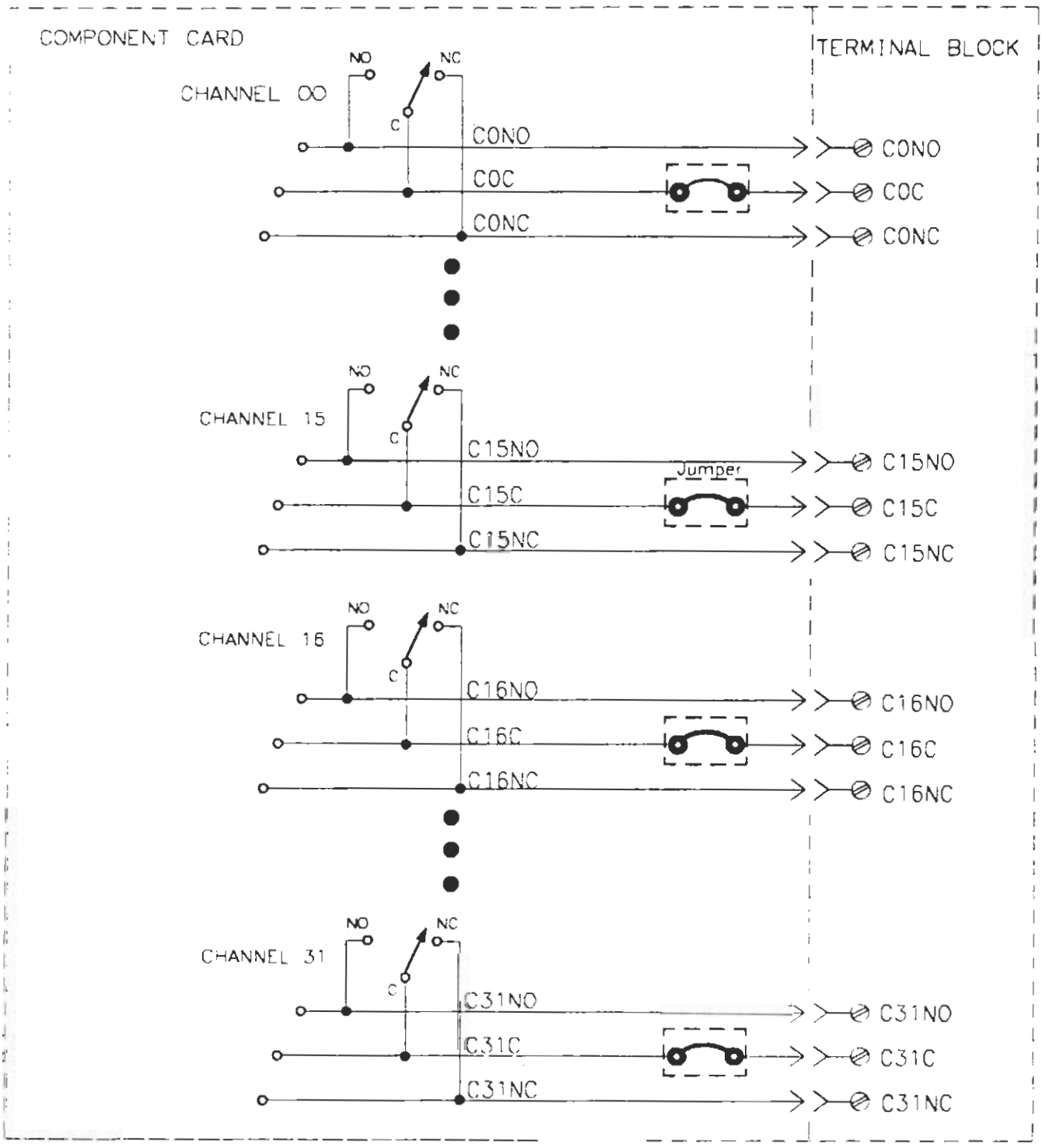


Figure 1–1. Form C Switch Simplified Diagram

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Typical Configuration

The Form C switch accepts user inputs up to 100 V dc or 100 V ac peak. Maximum rated power capacity (external load) is 50 W dc or 50 VA ac per channel. Per module, you can switch 320 W dc or 320 VA ac.

As noted, the switch may be configured for general purpose switching/scanning or digital output applications. For general purpose switching or scanning, no additional configuration is required. To configure the switch for digital output applications, install external pullup resistors as required. See Chapter 2 for configuration details.

Multiple Form C switch modules can be configured as a switchbox instrument. When using a switchbox instrument, multiple Form C switch modules within the switchbox instrument can be addressed using a single interface address. This configuration, however, requires the use of SCPI (Standard Commands for Programmable Instruments, formerly TMSL) commands. SCPI commands are discussed throughout this manual and information on the switchbox instrument configuration can be found in the *C–Size VXIbus System Installation and Getting Started Guide*.

Programming the Switch

There are several ways you can program the Form C switch module. One way is to write directly to the registers. This method can provide better throughput speed, however, it requires more knowledge of the Form C switch design. See appendix B, "Form C Registers" for information on register programming.

Another way to program the Form C switch module is to use an HP E1405A/B Command Module and SCPI commands. With SCPI commands the HP E1405A/B Command Module parses the commands and writes to the appropriate Form C switch register. The examples in this manual use the SCPI programming language. See appendix B, "Form C Registers," for examples on writing directly to the registers.

You can use different controllers and different programming languages. The examples in this manual, however, are based on the following configurations:

- HP 9000 Series 200/300 Computer running HP BASIC
- HP Vectra Computer (or compatible) with an HP 82335A HP–IB Interface Card (with command library) running Borland® Turbo C
- HP E1480A V/360 Computer running HP BASIC/HP–UX,C (This configuration is used in appendix B for register based programming).

See the *C–Size VXIbus System Installation and Getting Started Guide* for information on additional configurations.

Note

Most examples in this manual use SCPI commands. See Appendix B, "Form C Switch Registers," for information on writing directly to the registers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Specifying SCPI Commands

To address specific channels (relays) within a Form C switch module, you must specify the SCPI command and switch channel list. The following are the most commonly used commands:

SCPI Command	Command Description
CLOSEe <channel_list>	Connects the normally open (NO) terminal to the common (C) terminal for the specified channels.
OPEN <channel_list>	Connects the normallu closed (NC) terminal to the common (C) terminal for the channels specified.
SCAN <cahnnel_list>	Closed the set of Form C relays, one at a time.

The Channel List

The channel list is a combination of the switch card number and the channel numbers. The channel list takes the form of @ccnn where,

cc = switch card number (01–99)

nn = channel number (00–31)

The Card Number

The card number (cc of the channel list) depends on the configuration used. If you have multiple modules addressed as a switchbox, this number determines what card of the switchbox will be addressed. If you only have one module, your Form C card number is always 01. (Leading zeroes can be ignored for the card number.)

For a multiple–module switchbox, the card numbers are 01, 02,...,nn. The module with the lowest logical address is card number 01, the module with the next–lowest logical address is card number 02, etc. See the *C–Size VXIbus System Installation and Getting Started Guide* for additional switchbox instrument information.

The Channel Number

The channel number (**nn** of the channel list) determines which relay on the selected card will be addressed. Form C switch channel numbers are 00 through 31. The channels can be addressed using channel numbers or channel ranges.

Use a comma (,) to form a channel list or a colon (:) to form a channel range. Only valid channels can be accessed in a channel list or channel range. Also, the channel range must be from a lower channel number to a higher channel number. For example, CLOS (@100:215) is acceptable, but CLOS (@215:100) generates an error.

You can address the following:

- single channels (@ccnn)
- multiple channels (@ccnn,ccnn,...)
- sequential channels (@ccnn:ccnn)
- groups of sequential channels (@ccnn:ccnn,ccnn:ccnn)
- or any combination of the above

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Initial Operation

Two example programs follow which use Hewlett–Packard BASIC and TURBO C languages to get you started using the Form C switch module. The first example assumes an HP 9000 Series 200/300 controller and a Hewlett–Packard Interface Bus (HP–IB). (HP–IB is the Hewlett–Packard implementation of the IEEE 488.2–1987 standard.) The second example assumes an HP Vectra Computer (or compatible) with an HP 82335A HP–IB Interface Card (with command library) running Borland® Turbo C.

This program closes channel 02 of a Form C switch module at logical address 120 (secondary address = 120/8 = 15) and queries the channel closure state. The result is returned to the computer and displayed (1 = channel closed, 0 = channel open). See Chapter 4 for information on the SCPI commands.

HP BASIC

```
10  OUTPUT 70915;"*RST"              ! Reset the module
20  OUTPUT 70915;"CLOSE (@102)"      ! Close channel 02
30  OUTPUT 70915;"CLOS? (@102)"      ! Query channel 02 state
40  ENTER 70915;Value                ! Enter result into Value
50  PRINT Value                      ! Display result (should print a
                                     "1" to indicate that the channel
                                     is closed)
60  END
```

TURBO C

```
#include <stdio.h>>
#include <chpib.h>>                  /*Include file for HP–IB*/

#define ISC 7L
#define FORMC 70915L                 /*Form C default address*/
#define TASK1 "*RST"                 /*Command for a reset*/
#define TASK2 "CLOSE (@102)"         /*Command to close channel 02*/
#define TASK3 "CLOS? (@102)"         /*Command to query channel 02*/

main()
 {
   char into[257];
     int length = 256;
                                     /*Output commands to Form C*/
   error_handler (IOTIMEOUT (7L,5.0), "TIMEOUT");

   error_handler(IOOUTPUTS (FORMC, TASK1, 4), "OUTPUT com-
mand");
   error_handler(IOOUTPUTS(FORMC, TASK2, 12),"OUTPUT com-
mand");
   error_handler(IOOUTPUTS(FORMC, TASK3, 12),"OUTPUT com-
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
mand");

                                        /*Enter from Form C*/

    error_handler(IOENTERS (FORMC, into, &length), "ENTER com-
mand");

    printf("Now let's see if the switch is closed: %s",into);

    return;
}
int error_handler (int error, char *routine)
{
    char ch;
    if (error != NOERR)
    {
        printf ("\n Error %d %s \n", error, errstr(error));
        printf (" in call to HP-IB function %s \n\n", routine);
        printf ("Press 'Enter' to exit: ");
        scanf ("%c", &ch);
        exit(0);
    }
    return 0;
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2

Configuring the Form C Switch Module

Using This Chapter

This chapter shows how to make user connections and configure the Form C switch module. Chapter contents are as follows:

- Warnings and Cautions
- Setting the Module Address Switch
- Selecting the Interrupt Priority
- Connecting User Inputs
- Typical Switch Configurations
- Protecting your Relays and Circuits

Warnings and Cautions

Warning

SHOCK HAZARD. Only qualified, service–trained personnel who are aware of the hazards involved should install, configure, or remove the Form C switch module. Use only wire rated for the highest input voltage and remove all power sources from the mainframe and installed modules before installing or removing a module.

Caution

MAXIMUM VOLTAGE/CURRENT. Maximum allowable voltage per channel for the Form C switch is 100 V dc or 100 V ac peak. Maximum current per channel is 2 A dc or ac peak (non–inductive). Maximum power of an external load is 50 W dc or 50 VA ac per channel or 320 W dc or 320 VA ac per module. Exceeding any limit may damage the Form C switch. **STATIC–SENSITIVE DEVICE.** Use anti–static procedures when removing, configuring, and installing a module. The Form C switch is susceptible to static discharges. Do not install the Form C switch module without its metal shield attached.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Setting the Module Address Switch

The logical address switch (LADDR) factory setting is 120. Valid addresses are from 1 to 255. The Form C switch module can be configured as a single instrument, or as a switchbox. Refer to the *C–Size VXIbus System Installation and Getting Started Guide* for addressing information. Refer to figure 2–1 for switch position information.

Note

When using an HP E1405A/B, the address switch value must be a multiple of 8 if the module is the first module in a "switchbox" used with a VXIbus command module using SCPI commands.

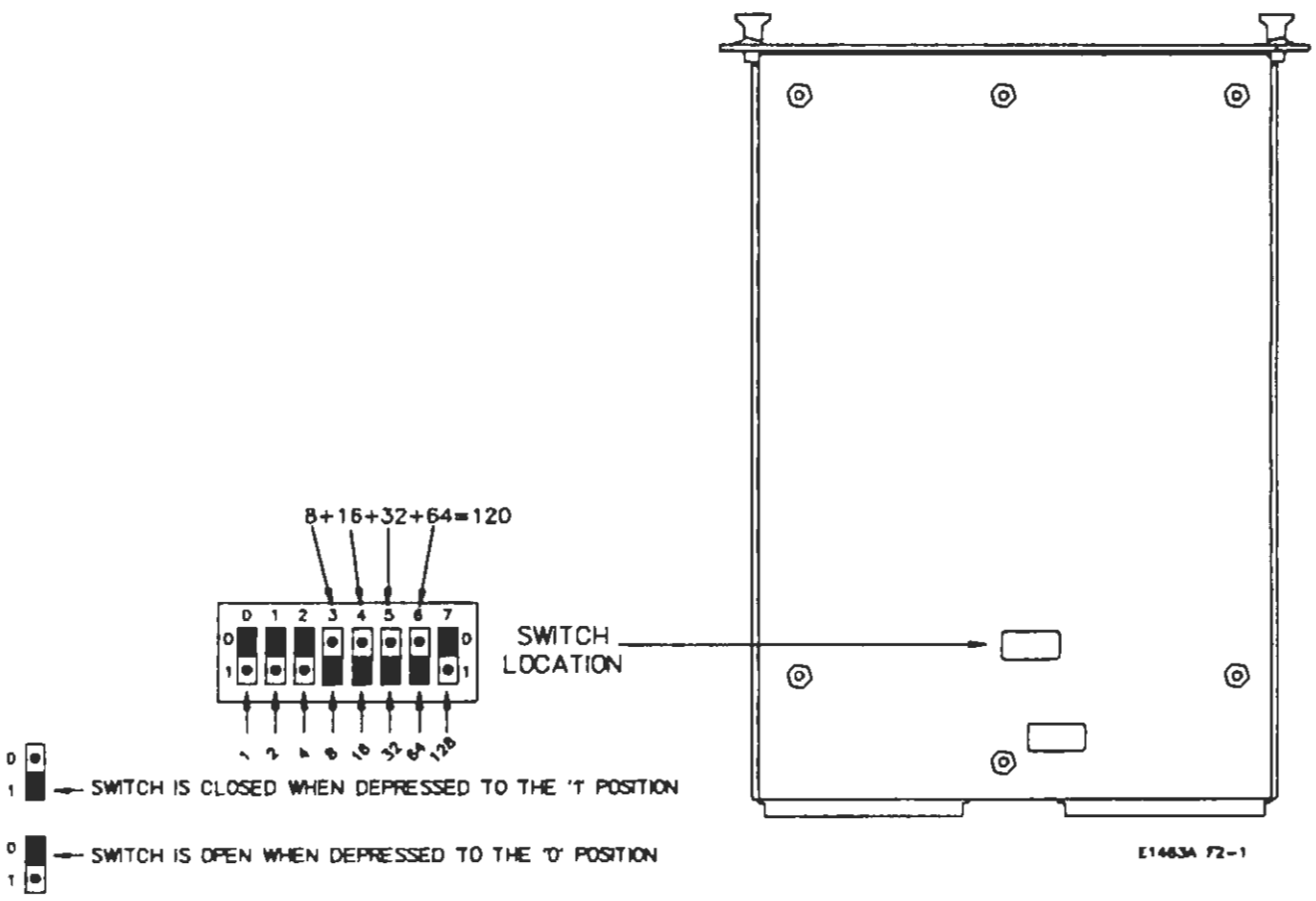


Figure 2–1. Setting the Address Switch

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Selecting the Interrupt Priority

The Form C switch modules generate an interrupt after a channel has been closed. These interrupts are sent to, and acknowledgements are received from, the command module (HP E1405A/B) via the VXIbus backplane interrupt lines.

For most applications where the Form C switch module is installed in an HP 75000 Series C mainframe, the interrupt priority jumper does not have to be moved. This is because the VXIbus interrupt lines have the same priority, and interrupt priority is established by installing the modules in slots numerically closest to the E1405A/B Command Module. Thus slot 1 has a higher priority than slot 2, slot 2 has a higher priority than slot 3, and so on.

You can select eight different interrupt priority levels. Level 1 is the lowest priority and level 7 is the highest priority. Level X disables the interrupt. The module's factory setting is level 1. To change, remove the four−pin jumper (HP part number 1258−0247) from the old priority location and reinstall in the new priority location (figure 2−2 shows a priority change from 1 to 7). If you are setting the interrupt priority level to something other than 1, see the *HP E1405A/B Command Module User's Manual* for additional information (Chapter 2). If the four pin jumper is not used, the two jumper locations must have the same interrupt priority level selected.

Note

When using the HP E1405A/B as the resource manager, the interrupt priority jumper MUST be installed in position 1. However, if you are using an HP E1480A (V/360) with the HP E1405A/B Command Module, an interrupt level of 2 is recommended. See the *C−Size VXIbus Systems Installation and Getting Started Guide*. Level X interrupt priority should not be used under normal operating conditions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

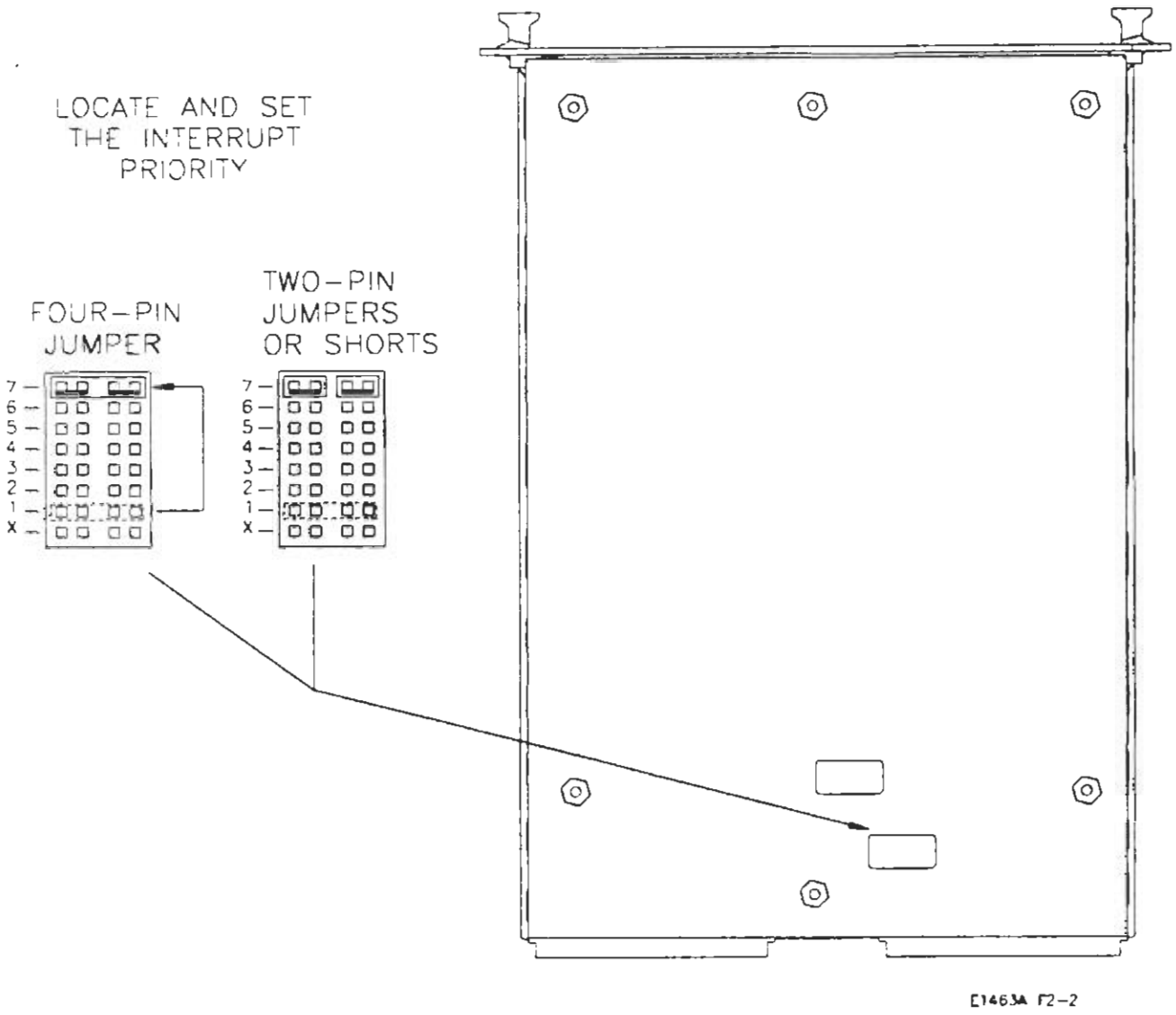


Figure 2–2. Setting the Interrupt Priority

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connecting User Inputs

The Form C switch module consists of a component module and a terminal block. User inputs to the Form C switch are to the normally open (NO), normally closed (NC), and common (C) terminal connectors on the terminal block. Figure 2–3 shows the terminal block and associated channel numbers. Figure 2–4 shows how to connect user (field) wiring to the terminal block.

Maximum terminal wire size is 18 AWG. Wire ends should be stripped 6 mm (0.25 in.) and tinned. When wiring all channels, use a smaller gauge wire (No. 20 – 22 Awg).

HP Z2417A TERMINAL BLOCK

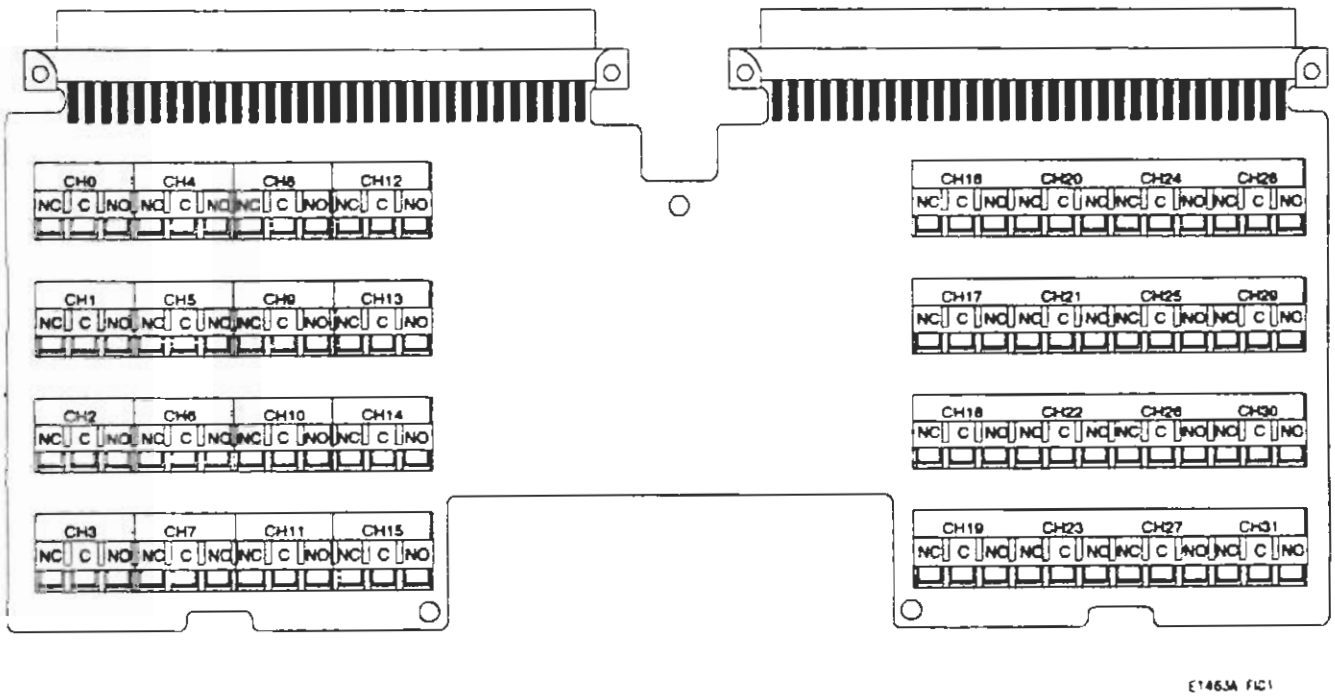


Figure 2–3. Form C Switch Terminal Block

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

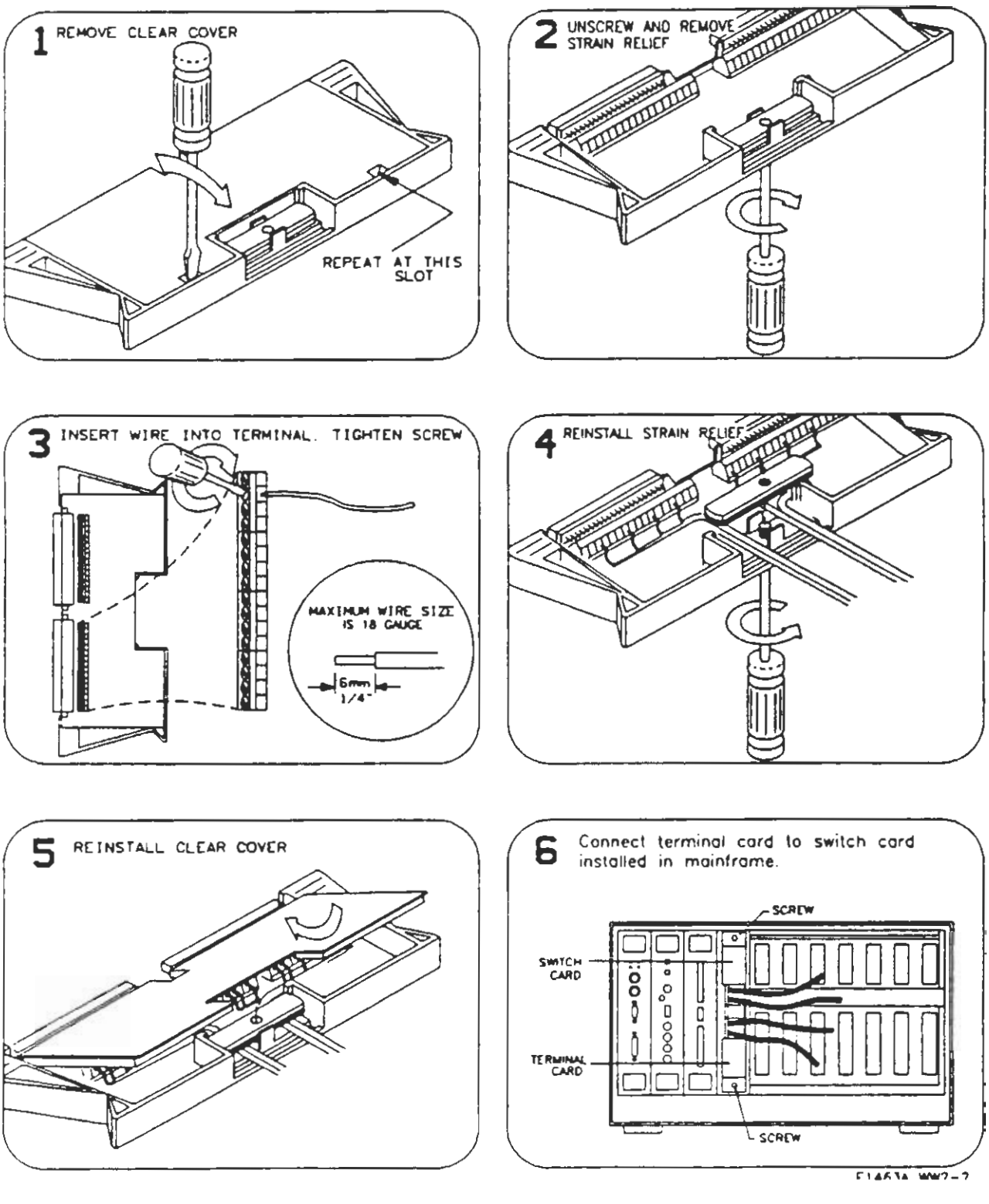


Figure 2–4. Wiring the Form C Switch Terminal Block

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Typical Switch Configurations

Typical Form C switch configurations are as follows:

- General Purpose Relay
- Digital Output

General Purpose Relay Configuration

As factory–configured, the Form C switch module is set for general purpose relay configuration. For this configuration, you can switch channels by opening or closing channel relays or you can scan a set of channels.

Example: General Purpose Relay Configuration

Figure 2–5 shows a typical general purpose relay configuration for channel 00. When the relay is open (NC terminal is connected to the C terminal), load 1 is connected. When the relay is closed, (NO terminal is connected to the C terminal), load 2 is connected.

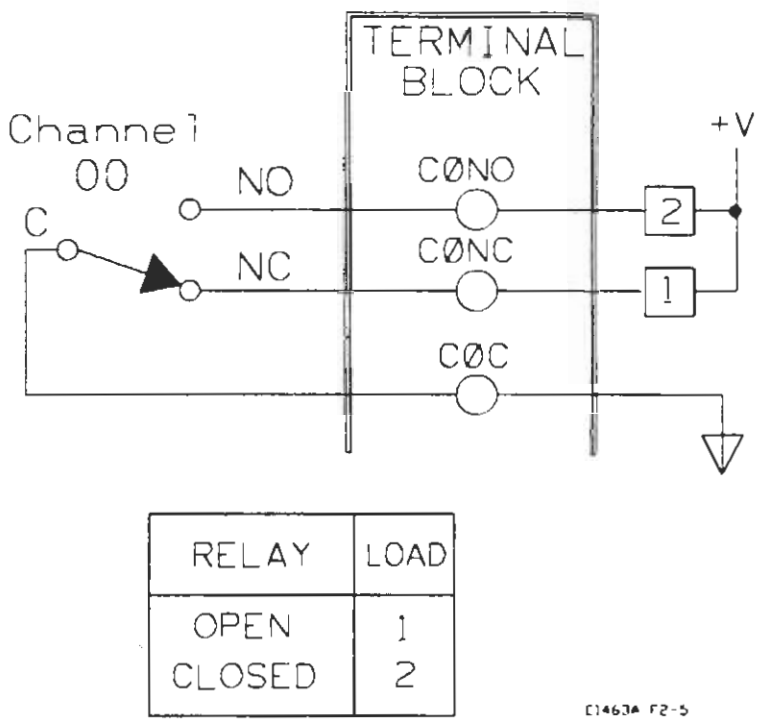


Figure 2–5. Example General Pupose Relay Configuration

Note

Mercury wetted relays are NOT Break Before Make or Make Before Break. The order of contact closure and contact openning is not constant. Evaluate the effects of the relays on your own circuit before installation.

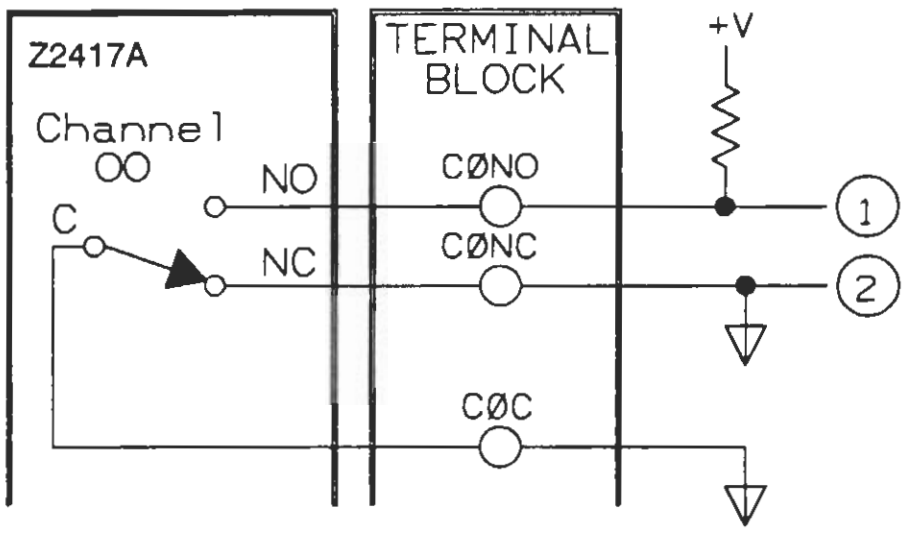
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Digital Output Configuration

By installing external pullup resistors, you can configure the switch as a digital output device.

Example: Digital Output Configuration

Figure 2–6 shows channel 00 configured for digital output operation. When the channel 00 relay is open (NC connected to C), point 1 is at + 5 V. When the channel 00 relay is closed (NO connected to C), point 1 is at 0 V.



RELAY	1	2
OPEN	+V	OV
CLOSED	OV	OV

E1463A F3-4

Figure 2–6. Example Digital Output Configuration

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Protecting your Relays and Circuits

Mercury wetted relays have a shorter lifespan than other electronic parts such as ICs. Because of their mechanical nature, relays usually have about 100 million operations (at 30 operations per second) which isn't quite 1000 hours. Therefore, to get the full life out of a relay in a switching module, you must protect the relay. The following sections provide additional information about protecting your relays and circuits.

Relay Reliability

To make sure you get the full life of your relays, keep the following in mind:

- **Be aware of non-resistive loads.** When switching inductive loads, high voltages (thousands of volts) are produced across the relay contacts. This causes arcing and transfer of material between contacts. Oxides and carbides from components of the atmosphere coat the contacts and cause high contact resistance. The transfer of material creates hills and valleys which lock together to "weld" contacts. Motor loads, for example, produce large inrush currents that can be 5 to 10 times greater than the steady state current. The following table summarizes how many times greater the inrush current can be for different types of loads:

Type of Load	Inrush Current times Steady State
Resistive	1
Solenoid	10 – 20
Motor	5 – 10
Incandescent Lamp	10 – 15
Mercury Lamp	3
Sodium Vapor Lamp	1 – 3
Capacitive	20 – 40
Transformer	5 – 15

Capacitors are not to be placed across the load or relay contacts. Capacitors may suppress arcs, but the energy stored in the capacitors will flow through the relay contacts, welding them.

You can also add circuit protection within the HP Z2417A component module. See the sections following.

Adding Relay and Circuit Protection

Space has been made available on the HP Z2417A Form C Module for adding relay and circuit protection. Relay protection can be added by placing a protective device across the contacts. This is done by adding metal oxide varistors (MOVs) between the common (C) and normally open (NO) or normally closed (NC) terminals. Now as the voltage goes up, the varistor draws current to protect the relay.

Circuit protection can be added by placing a protective device in series with the common lead. This is done by adding a resistor or fuse between the common (C) terminal and your circuit. Figure 2-7 shows the locations where these protective devices can be added. When installing circuit protection, a jumper must be removed first.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To install these protective devices it maybe necessary to remove the sheet metal covers from the module. The locations for installing the devices are labeled as follows:

Circuit Protection	
JMxx	The resistor or fuse location in series with common (C)

Where xx is the channel number

Again, do not install a capacitor in any of these locations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

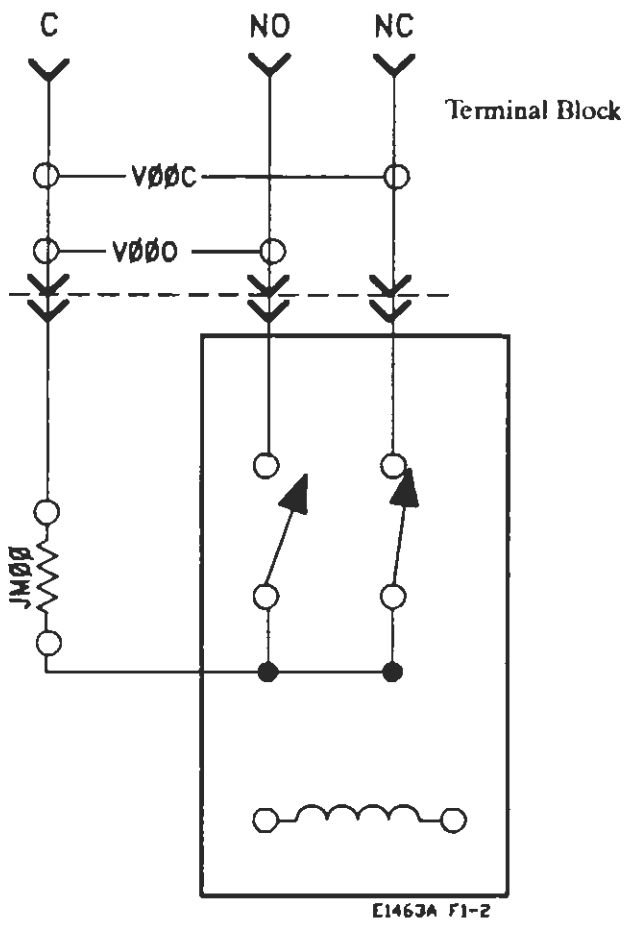


Figure 2–7. Adding Relay and Circuit Protection

Maximum Allowable Module Switch Current

The HP Z2417A Form C Switch Module has an individual channel current specification of 2 A. However, if you apply the 2 A to all the channels with a relay contact resistance of .25 ohms, the power dissipation would be 80 W. Since the HP E1400B Mainframe can only provide cooling for 45 W per slot (keeps the temperature rise to 10°C), this can not be allowed to happen.

A reasonable current for the entire module is 50 A. That is, 25 channels each carrying 2 A or some combination of channels and currents that total 50 A. This will produce about 67.5 W of internal dissipation, leading to a 15 C temperature rise.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3

Using the Form C Switch

Using This Chapter

This chapter uses typical examples to show how to use the Form C switch module for switching channels and scanning channels. See Chapter 4, "Form C Switch Command Reference," for command information. Chapter contents are as follows:

- Form C Switch Commands
- Power–On and Reset Conditions
- Module Identification
- Switching Channels
- Scanning Channels
- Querying the Form C Switch Module
- Using the Scan Complete Bit
- Recalling and Saving States
- Detecting Error Conditions
- Synchronizing the Form C Switch

All examples in this chapter use the HP–IB select code of 7, primary address of 09, and a secondary address of 15 (LADDR=120).

Form C Switch Commands

Table 3–1 explains some of the commands used in this chapter. Refer to Chapter 4 for more information on these commands.

Table 3–1. Form C Switch Commands used in Chapter 3

Command	Description
[ROUT:]CLOS <*channel_list*>	Closes the channels in the channel list.
[ROUT:]CLOS? <*channel_list*>	Queries the state of the channels in the channel list.
[ROUT:]OPEN <*channel_list*>	Opens the channels in the channel list.
[ROUT:]OPEN? <*channel_list*>	Queries the state of the channels in the channels list.
[ROUT:]SCAN <*channel_list*>	Closes the channels in the channel list, one at a time.
INIT[:IMM]	Starts the scan sequence and closes the first channel in the channel list.
TRIG:SOUR BUS|EXT|HOLD|IMM|TTLT	Selects the trigger source to advance the scan.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Power–On and Reset Conditions

Since the Form C switch module has nonlatching relays, all relays are in the normally closed (NC) position at power–down and power–up.

The ***RST** command opens all channels, invalidates the current channel list for scanning, and sets the following:

Paramter	Default	Description
ARM:COUNt	1	Number of scanning cycles is 1.
TRIGger:SOURce	IMM	Will advance scanning cycles automatically.
INITiate:CONTinous	OFF	Number of scanning cycles set by ARM:COUNt
OUTPut:STATe	OFF	Trigger output from EXT or TTL sources is disabled

Module Identification

The following short programs use the *RST, *CLS, *IDN?, CTYP?, and CDES? commands to reset and identify the Form C switch module.

HP BASIC

```
10  DIM A$[50], B$[50], C$[50]      ! Dimensions three string variables to fifty characters
20  OUTPUT 70915; "*RST; *CLS"      ! Outputs the commands to reset and clear the status register
30  OUTPUT 70915; "*IDN?"           ! Queries for module identification
40  ENTER 70915; A$                 ! Enters the results into A$
50  OUTPUT 70915; "SYST:CDES? 1"    ! Output the command for a card description
60  ENTER 70915; B$                 ! Enters the results into B$
70  OUTPUT 70915; "SYST:CTYP? 1"    ! Outputs the command for the card type
80  ENTER 70915; C$                 ! Enters the results into C$
90  PRINT A$, B$, C$                ! Prints the contents of variables A$, B$, and C$

100 END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TURBO C

```
#include <stdio.h>
#include <chpib.h>                    /*Include file for HP-IB*/

#define ISC 7L
#define FORMC 70915L                  /*Form C default address*/
#define TASK1 "*RST;*CLS;*IDN?"       /*Reset, clear, and query identifica-
tion*/
#define TASK2 "SYST:CDES? 1"          /*Command for card description*/
#define TASK3 "SYST:CTYP? 1"          /*Command for card type*/

main()
 {
    char into1[51], into2[51], into3[51];
      int length = 50;
                                      /*Output and enter commands to Form C*/

    error_handler (IOTIMEOUT (7L,5.0), "TIMEOUT");

    error_handler(IOOUTPUTS(FORMC, TASK1, 15),"OUTPUT com-
mand");
    error_handler(IOENTERS(FORMC, into1, &length),"ENTER com-
mand");

    error_handler(IOOUTPUTS(FORMC, TASK2, 12),"OUTPUT com-
mand");
    error_handler(IOENTERS(FORMC, into2, &length),"ENTER com-
mand");

    error_handler(IOOUTPUTS(FORMC, TASK3, 12),"OUTPUT com-
mand");
    error_handler(IOENTERS(FORMC, into3, &length),"ENTER com-
mand");

    printf("IDENTIFICATION: %s",into1);
    printf("CARD DESCRIPTION: %s",into2);
    printf("CARD TYPE: %s",into3);

    return;
}
int error_handler (int error, char *routine)
{
    char ch;
    if (error != NOERR)
    {
        printf ("\n Error %d %s \n", error, errstr(error));
        printf (" in call to HP-IB function %s \n\n", routine);
        printf ("Press 'Enter' to exit: ");
        scanf ("%c", &ch);
        exit(0);
    }
    return 0;
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A typical print for the HP Z2417A will look like the following:

```
HEWLETT-PACKARD,SWITCHBOX,0,A.04.00
32 Channel General Purpose Relay
HEWLETT-PACKARD,Z2417A,0,A.04.00
```

Switching Channels

For general purpose relay operation, you can connect or disconnect a load by opening or closing specified channel relays. By adding external pullup resistors, the switch can be configured for digital output operations.

Use CLOSe *<channel_list>* to connect a channel's normally open (NO) terminal to its common (C) terminal, or use OPEN *<channel_list>* to connect a channel's normally closed (NC) contact to its common (C) terminal. The *<channel_list>* has the form (@ccnn) where,

cc = card number (01–99)

nn = channel number (00–31)

To OPEN or CLOSe multiple channels, place a comma (,) between the channel numbers. For example, to close channels 101 and 103, execute CLOS (@101,103). To OPEN or CLOSe a continuous range of channels place a colon (:) between the first and last channel numbers.

The following HP BASIC program shows how to close and open channel 2 on an HP Z2417A Form C module (card #1):

```
120 OUTPUT 70915; "ROUT:CLOSE (@102)"
130 OUTPUT 70915; "ROUT:OPEN (@102)"
140 END
```

Note

When using any of the commands that are preceded by [ROUTe:], the ROUTe portion of the command can be eliminated. For example, as in the program above, ROUTe can be eliminated and just the CLOSe command can be used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example: Voltage Switching

This example closes channel 00 of a Form C switch module to switch the load voltage (E) from load 1 to load 2. When the channel relay is open, the load voltage is applied to load 1. When the relay is closed, the voltage is applied to load 2. See figure 3–1 for typical user connections.

The following program shows how to close channel 00 of the HP Z2417A Form C Switch:

```
10  DISP "Testing the HP Z2417A"
20  OUTPUT 70915; "CLOSe (@100)" ! Close channel 00 relay (connect NO to C). 1 is the card number and 00 is the channel number.
30  END
```

To open channel 00, use OPEN (@100).

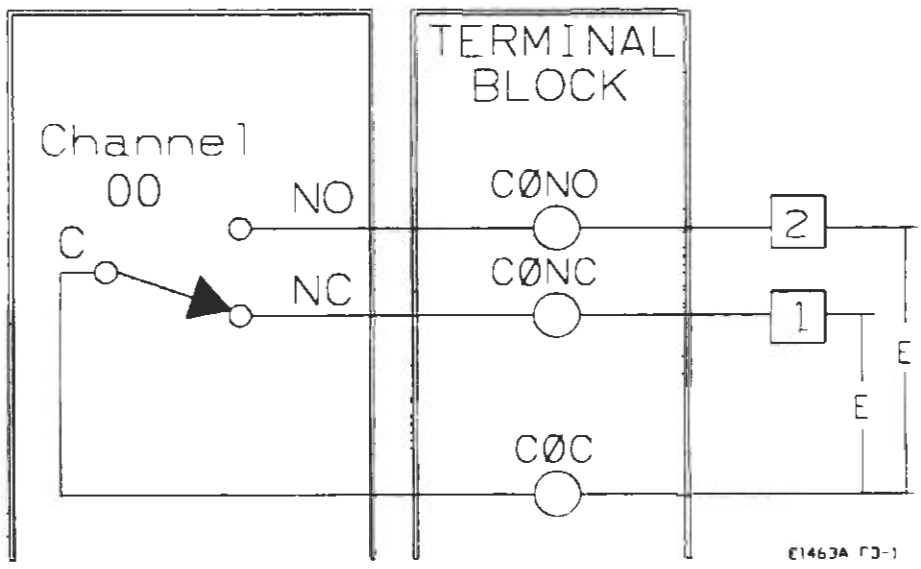


Figure 3–1. Example Voltage Switching

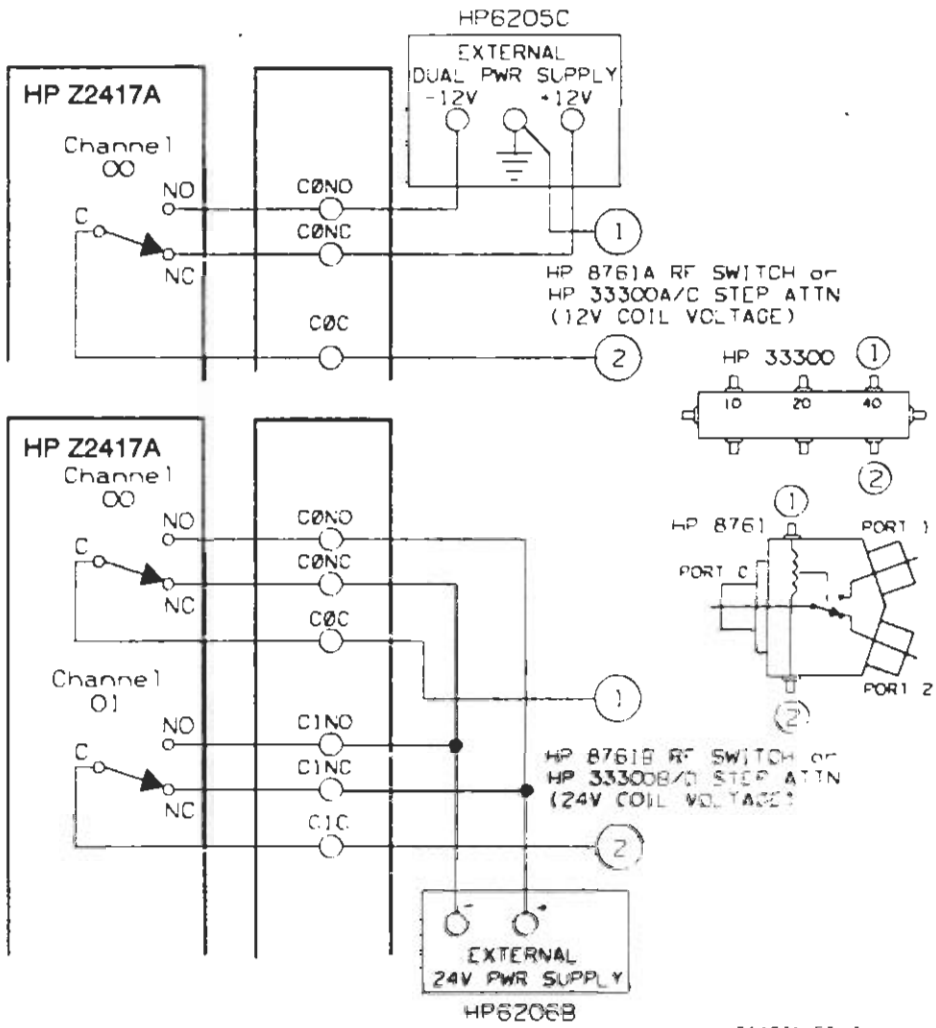
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example: Controlling RF Switches/Step Attenuators

Figure 3−2 shows one way to drive the HP 8761 SPDT RF Switches or HP 33300 Series Programmable Step Attenuators. (This figure only shows control for the HP 33300 40 db step. Additional drive relays are required for the 10 dB and 20 dB steps.) The HP 8761A and HP 33300A/C operate from a 12−15V coil voltage, while the HP 8761B and HP 33300B/D operate from a 24V−30V coil voltage. To close channel 00, execute:

```
10  DISP "Applying -12V"
20  OUTPUT 70915; "CLOSe (@100)" ! Close channel 00 relay (connect NO to C). 1 is the card number and 00 is the channel number
30  END
```

To open channel 00, use OPEN (@100).



E1463A F3-3

Figure 3−2. Example: Controlling RF Switches/Step Attenuators

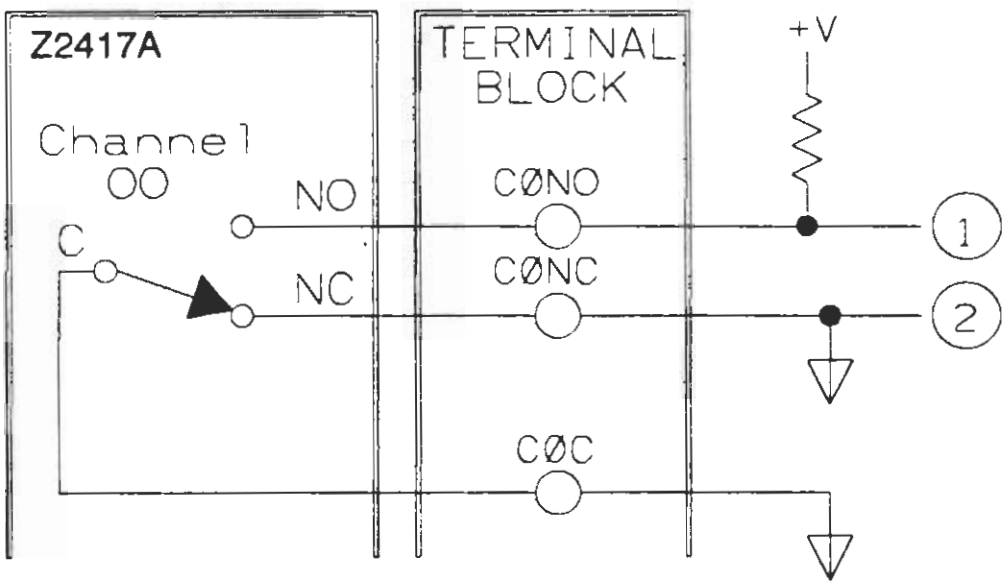
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example: Digital Output Configuration

Figure 3–3 shows channel 00 configured for digital output operation. When the channel 00 relay is open (NC connected to C), point 1 is at +V and point 2 is at 0 V. When the channel 00 relay is closed (NO connected to C), points 1 and 2 are both at 0 V. To close channel 00, execute the following:

```
10   DISP "Closing channel 0"
20   OUTPUT 70915; "CLOS (@100)" ! Close channel 00 relay (connect NO to C). 1 is the card number and 00 is the channel number.
30   END
```

To open channel 00, use OPEN (@100).



RELAY	1	2
OPEN	+V	OV
CLOSED	OV	OV

D1463A F3-4

Figure 3–3. Example: Digital Output Configuration

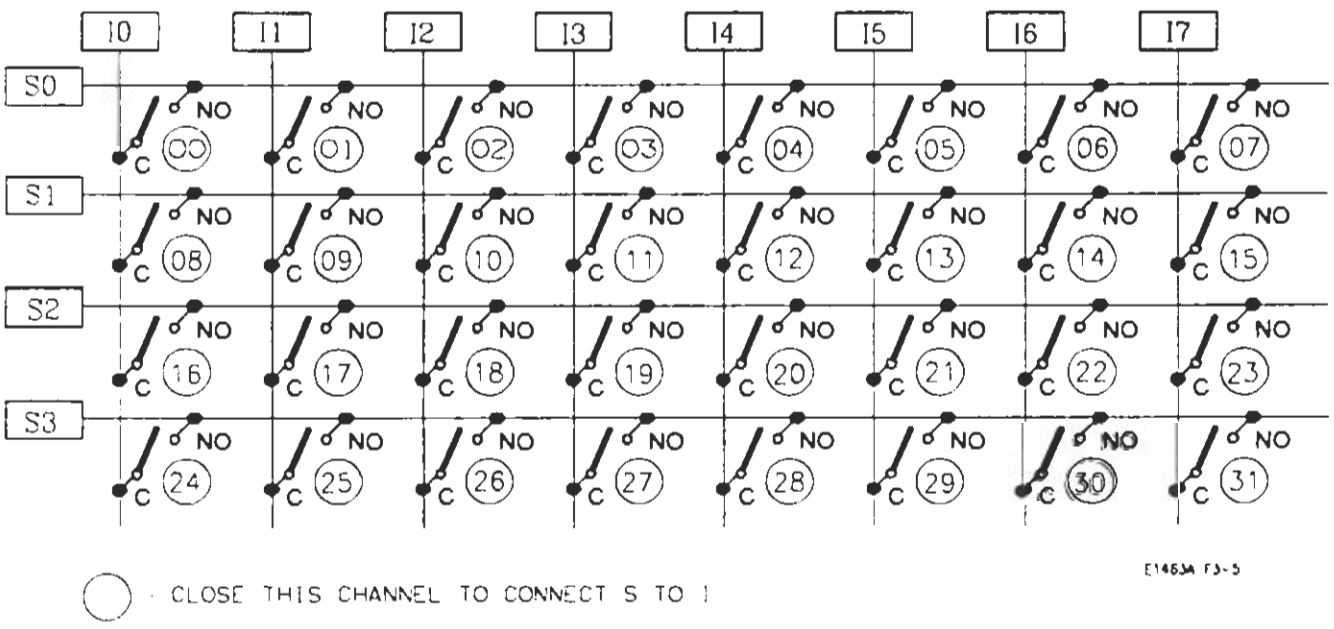
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example: Matrix Switching

The Form C switch module can be configured as a 4 x 8 single–wire matrix to connect any combination of up to four user sources (S1, S2, S3, S4) to any combination of up to eight user instruments (I1, I2, I3 . . . I8) at a time. To do this you must make the following connections:

Connect Common (C) channel numbers together	Connect Normally open (NO) Channel Numbers together
0, 8, 16, and 24	0 – 7
1, 9, 17 and 25	8 – 15
2, 10, 18, and 26	16 – 23
3, 11, 19, and 27	24 – 31
4, 12, 20, and 28	
5, 13, 21, and 29	
6, 14, 22, and 30	
7, 15, 23, and 31	

When this is done, you'll get the following matrix:



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Close the channel number enclosed in the circle to connect the corresponding row and column. The following example closes channel 25 to connect S3 to I1 and closes channel 20 to connect S2 to I4. To close channels 20 and 25, execute the following:

```
10  DISP "Testing Switch Matrix"
20  OUTPUT 70915; "CLOS (@120,125)" ! Close channels 20 and
                                     25. 1 is card number and 20
                                     and 25 are channel numbers.
30 END
```

To open the channels, use OPEN (@120,125).

Scanning Channels

For the Form C switch module, scanning channels consists of closing a specified set of channels, one channel at a time. You can scan any combination of channels for a single−module or a multiple−module switchbox. Single, multiple, or continuous scanning modes are available. See Chapter 4 for additional information on scanning Form C switch channels.

Channel lists can extend across boundaries. For multiple−module switchbox instruments, the channels to be scanned can extend across switch modules. For example, for a two−module switchbox instrument, SCAN (@100:231) will scan all channels of both Form C switch modules.

Use ARM:COUNt *number* to set multiple/continuous scans (from 1 to 32767 scans). Use INITiate:CONTinuous ON to set continuous scanning. See Chapter 4 for details.

Example: Scanning Channels With an External Instrument Using Trig In and Out Ports

This example shows one way to synchronize instrument measurements of a devices under test (DUT) with Form C switch channel closures. For measurement synchronization, the HP E1405A/B Trig In and Out BNC ports are connected to the instrument Voltmeter Complete and External Trigger ports. See figure 3−5 for typical user connections.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

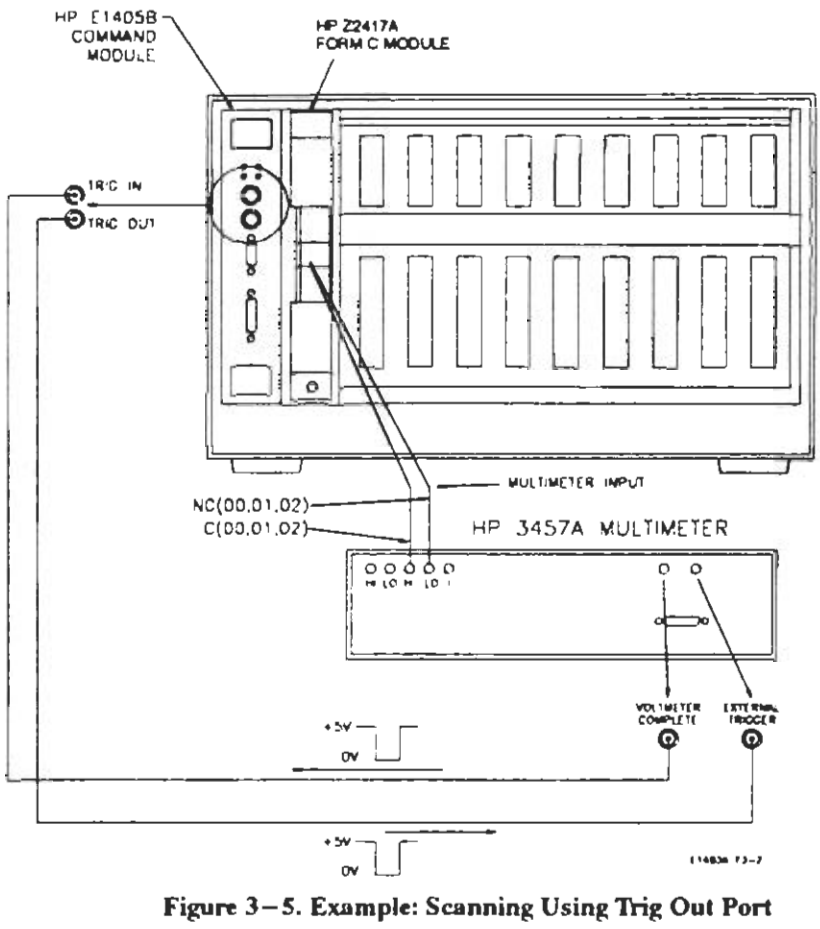


Figure 3–5. Example: Scanning Using Trig Out Port

For this example, the normally closed (NC) contacts (channels 00–02) are connected to ground, and the measurements are made on the common (C) contacts. The command module and instrument are connected via HP–IB. The Form C switch module has a logical address 120 (secondary address 15), and the external instrument has an address of 722.

```
10  OUTPUT 70915; "*RST;*CLS"       ! Reset and Clear the module
20  OUTPUT 722;"TRIG EXT;DCV"       ! External trigger, DC volts
30  OUTPUT 722;"MEM FIFO"           ! Memory first in first out
40  OUTPUT 70915;"OUTP ON"          ! Enable Trig Out
50  OUTPUT 70915;"TRIG:SOUR EXT"    ! Bus triggering
60  OUTPUT 70915;"SCAN (@100:102)"  ! Scan channels 00–02
70  OUTPUT 70915;"INIT"             ! Enable scan
80  WAIT 2                          ! Wait for switch closures
90  FOR Channel=1 TO 3              ! Start loop
100      ENTER 722;Result           ! Enter result
110     PRINT Result                ! Display result
120 NEXT Channel                    ! Increment count
130 END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example: Scanning Channels with System Multimeter Using TTL Trigger

This example uses the HP E1405A/B Command Module's TTL Trigger Bus Lines to synchronize Form C channel closures to a system multimeter (HP E1410A). For measurement synchronization:

- HP E1405A/B TTL Trigger Bus Line 0 is used by the Form C module to trigger the multimeter to perform a measurement.
- HP E1405A/B TTL Trigger Bus Line 1 is used by the multimeter to advance the Form C scan.

These trigger bus lines are not actual hardware connections. The triggering is accomplished by the HP E1405A/B's firmware. The measurement is taken from the common (C) terminal. The common terminals for channels 0 through 2 are connected together for this example. When one of these switches is closed (C connected to NO), different DUTS are switched in for a measurement. Figure 3–6 shows how to connect the Form C module to the multimeter module. The connections shown with dotted lines are not actual hardware connections. These connections indicate how the firmware operates to accomplish the triggering.

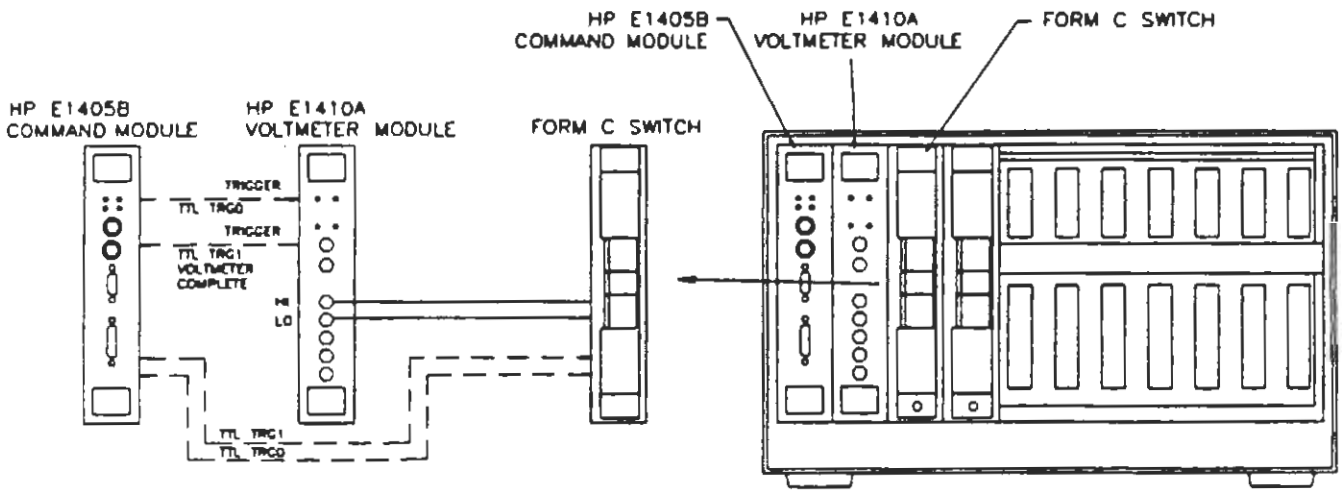


Figure 3–6. Example: Scanning Using TTL Trig

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following HP BASIC program sets up the voltmeter (HP–IB address 70903) to scan making 2–wire resistance measurements.

```
10 ALLOCATE REAL Rdgs(1:3)          ! reset and clear the modules
20 OUTPUT 70915; "*RST;*CLS"
30 OUTPUT 70903; "*RST;*CLS"        ! Set From C to be triggered by
                                    TTL Trigger line 1
40 OUTPUT 70915;"TRIG:SOUR TTLT1"
                                    ! Voltmeter Triggers on TTL
                                    Trigger line 0, VM pulses TTL
                                    Trigger line 1 on measurement
                                    complete, Set VM function to
                                    Resistance, range, NPLC
50 OUTPUT 70903;"ABORT;:TRIG:SOUR TTLTRG0"
60 OUTPUT 70903; "OUTP:TTLTRG1:STAT ON"
70 OUTPUT 70903; ":FUNC:RES;:RES:RANGE 1E6;:RES:NPLC
                                    1"
80 OUTPUT 70903; ":TRIG:DEL 0; COUN 3;:CAL:ZERO:AUTO
                                    ON;:TRIG:BUFF ON" !
                                    Check to see VM ready; when
                                    ready, initialize trigger 1
90 OUTPUT 70903; "*OPC?"
100 ENTER 70903; Check
110 OUTPUT 70903; "INIT"            ! Setup the Form C: Form C
                                    pulses TTL trigger line 0 on
                                    channel closed
120 OUTPUT 70915; "OUTPUT:TTLT0:STATE ON"
130 OUTPUT 70915; "SCAN (@100:102)"
140 OUTPUT 70915; "INIT"
150 OUTPUT 70903; "FETCH?"
160 ENTER 70903; Rdgs(*)
170 PRINT Rdgs(*)
180 END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Querying the Form C Switch Module

All query commands end with a "?". These commands are used to determine a specific state of the module. The data is sent to the output buffer where you can retrieve it into your computer. See Chapter 4 for more information on these commands.

Use CLOSe? <*channel_list*> or OPEN? <*channel_list*> to query the channel state (open/closed). CLOS? returns a 1 for channel(s) closed and a 0 for channel(s) open. OPEN? returns a 0 for channel(s) closed and a 1 for channel(s) open. (Commands are software queries and do not account for relay hardware failures.)

The following example closes a range of channels and queries for the results:

```
10 DIM Channels$[32]                    ! Dimensions a string variable
                                        to 32 characters
20 OUTPUT 70915;"CLOS (@100:131)"
                                        ! Closes channels 00 through 31
30 OUTPUT 70915;"CLOS? "(@100:131)"! Queries to see if the chan-
                                        nels are closed
40 ENTER 70915; Channels$               ! Enters the results from the
                                        switch card into the variable
                                        Channels$
50 PRINT "Channels Closed:";Channels$ ! Prints the channels
                                        closed (should print 1's)
60 END
```

Using the Scan Complete Bit

You can use the Scan Complete bit (bit 8) in the Operation Status Register (in the command module) of a switchbox to determine when a scanning cycle is complete (no other bits in the register apply to the switchbox). Bit 8 has a decimal value of 256 and you can read it directly with the STAT:OPER? command. See the STATe:OPERation[:EVENt]? command in Chapter 4 for an example.

When enabled by the STAT:OPER:ENAB 256 command, the Scan Complete bit will be reported as bit 7 of the Status Register. Use the HP–IB Serial Poll or the IEEE 488.2 Common Command *STB? to read the Status Register.

When bit 7 of the Status Register is enabled by the *SRE 128 Common Command to assert an HP–IB Service Request (SRQ), you can interrupt the computer when the Scan Complete bit is set, after a scanning cycle completes. This allows the computer to do other operations while the scanning cycle is in progress.

The following example monitors bit 7 in the Status Register to determine when the scanning cycle is complete. The computer used in this example is an HP 9000 Series 200/300 running HP BASIC as the programming language. The

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

computer interfaces with an HP E1405A/B Command Module over HP−IB. The HP−IB select code is 7, the HP−IB primary address is 09, and the HP−IB secondary address is 15.

```
20 OUTPUT 70915;"*RST; *CLS"        ! Resets and Clears the module
                                    ! Enable Scan Complete Bit
30 OUTPUT 70915;"STAT:OPER:ENAB 256"
40 OUTPUT 70915; "TRIG:SOUR IMM"    ! Set the Form C switch up for
                                      continuous triggering
50 OUTPUT 70915; "SCAN (@100:115)"! Select channels to scan
60 OUTPUT 70915; "*OPC?"            ! Wait for operation complete
70 ENTER 70915; A$                  ! Query the contents in the op-
                                      eration status register
80 PRINT "*OPC? = ",A$              ! Print the contents of the oper-
                                      ation status register
90 OUTPUT 70915;"STAT:OPER:ENAB?"
                                    ! Query the contents of the sta-
                                      tus byte register
100 ENTER 70915;A$
110 PRINT "STAT:OPER:ENAB? =",A$ ! Print the contents of the status
                                    byte register
120 OUTPUT 70915;"*STB?"            ! Query the contents of the sta-
                                      tus byte register
130 ENTER 70915;A$
140 PRINT "Switch Status =",A$      ! Print the contents of the status
                                      byte register
150 OUTPUT 70915; "INIT"            ! Start scan cycle
160 I = 0                           ! Initialize the value of the
                                      counter
170 WHILE (I=0)                     ! Stay in loop until some value
                                      is returned from the SPOLL
                                      (70915) command
180     I = SPOLL(70915)
190     PRINT "Waiting for scan to complete: SPOOL =
                                      ";I
200 END WHILE
210 I = SPOLL(70915)
220 PRINT "Scan complete: spool = ";I
230 END
```

Recalling and Saving States

The *SAV <*numeric_state*> command saves the current instrument state. The state number (0−9) is specified by the <*numeric_state*> parameter. The settings saved by this command are as follows:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Channel relay states (open or closed)
- ARM:COUNt
- TRIGger:SOURce
- OUTput:STATe
- INITiate:CONTinuous

The ***RCL** <*numeric_state*> command recalls the state when the last ***SAV** was executed for the specified <*numeric_state*> parameter (0–9). If no ***SAV** was executed for the <*numeric_state*>, ***RST** default settings are used. Refer to the ***SAV** settings list for the settings recalled by ***RCL.**

The following program shows how to save and recall Form C switch states.

```
10 DIM A$[150]                          ! Dimension a string variable
                                        for 150 characters
20 OUTPUT 70915; "CLOSE (@100:131)"! Close channels 00 – 31
                                        on the form c
30 OUTPUT 70915; "*SAV 5"               ! Save as numeric state 5
40 OUTPUT 70915 "*RST;*CLS"             ! Reset and Clear the module
50 OUTPUT 70915;"CLOS? (@100:131)"! Query the channels closed
60 ENTER 70915;A$
70 PRINT "Channels Closed:";A$          ! Prints closed channels (should
                                        print 0's)
80  OUTPUT 70915; "*RCL 5"              ! Recall numeric state 5
90  OUTPUT 70915 "CLOS? (100:131)"! Query to see what channels
                                        are closed
100  ENTER 70915;A$
110  PRINT "Channels Closed:";A$ ! Print the closed channels
                                        (should print 1's)
100 END
```

Detecting Error Conditions

The **SYSTem:ERRor?** query requests a value from instrument's error register. This register contains an integer in the range [–32768 to 32767]. The response takes the following form:

Where, <*err_number*> is the value of the instrument's error, and <*err_message*> is a short description of the error.

The following programs attempt an illegal channel closure and polls for an error message:

HP BASIC

```
10   DIM Err_num$[256]                  ! Dimension a string variable
                                        for 256 characters
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
20  OUTPUT 70915; "CLOS (@135)"  ! Try to close an illegal channel
30  OUTPUT 70915; "SYST:ERR?"    ! Query for a system error
40  ENTER 70915; Err_num$
50  PRINT Err_num$               ! Prints error +2001, "Invalid
                                 channel number"
60  END
```

TURBO C

```
#include <stdio.h>
#include <chpib.h>                          /*Include file for HP-IB*/

#define ISC 7L
#define FORMC 70915L                        /*Form C default address*/
#define TASK1 "CLOSE (@135)"                /*Command for illegal switch clo-
sure*/
#define TASK2 "SYST:ERR?"                   /*Command for system error*/

main()
 {
    char into[257];
      int length = 256;

                                            /*Output commands to Form C*/

    error_handler (IOTIMEOUT (7L,5.0), "TIMEOUT");

    error_handler (IOOUTPUTS (FORMC, TASK1, 12), "OUTPUT com-
mand"); error_handler (IOOUTPUTS (FORMC, TASK2, 9), "OUTPUT
command");

                                            /*Enter from Form C*/

    error_handler (IOENTERS (FORMC, into, &length), "ENTER com-
mand");

    printf("Now let's print the errors: %s",into);

    return;
}
int error_handler (int error, char *routine)
{
    char ch;
    if (error != NOERR)
    {
        printf ("\n Error %d %s \n", error, errstr(error));
        printf (" in call to HP-IB function %s \n\n", routine);
        printf ("Press 'Enter' to exit: ");
        scanf ("%c", &ch);
        exit(0);
    }
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
    return 0;
}
```

If no error occurs, the switchbox responds with 0,"No error". If there has been more than one error, the instrument will respond with the first one in its error queue. Subsequent queries continue to read the error queue until it is empty. The maximum *<err_message>* string length is 255 characters.

Synchronizing the Form C Switch

The following example shows how to synchronize a Form C switch module with a measurement instrument. In this example, the Form C switch module switches a signal to a multimeter. The program then verifies that the channel is closed before the multimeter begins its measurement.

```
10  OUTPUT 70915; "CLOS (@105)"     ! Closes channel 5
20  OUTPUT 70915; "*OPC?"           ! Wait for operation complete
30  ENTER 70915; Opc_value
40  OUTPUT 70915; "CLOS? (@105)"    !Check to see if channel is
                                    closed
50  ENTER 70915;A
60  OUTPUT 70914; MEAS:VOLT:DC?"    ! When channel is closed, mea-
                                    sure the voltage
70  ENTER 70914; Meas_value
80  PRINT Meas_value                ! Print the measured voltage
90  END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4

Form C Switch Command Reference

Using This Chapter

This chapter summarizes SCPI (Standard Commands for Programmable Instruments, formerly TMSL) commands IEEE 488.2 Common (*) commands used in this manual.

See the *HP E1405A/B Command Module User's Manual* for additional information on SCPI and common commands. Chapter contents are as follows:

- Command Types
- SCPI Command Format
- SCPI Command Reference
- Common Command Reference
- IEEE 488.2 Common Commands
- Command Quick Reference

Command Types

Commands are separated into two types:

- IEEE 488.2 Common Commands
- 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 as follows:

```
*RST

*ESR 32

*STB?
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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>
        SCAN <channel_list>
```

ROUTe: is the root command and CLOSe and SCAN are second level commands.

Note

There must be a space between the second level command (CLOSe, for example) and the *<channel list>*.

Command Separator

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

```
ROUTe:SCAN
```

Abbreviated Commands

The command syntax shows most commands as a mixture of upper and lower case letters. The upper case letters indicate the abbreviated spelling for the command. For shorter program lines, send the abbreviated form, and for better program readability, send the entire command. The instrument will accept either the abbreviated form or the entire command.

For example, if the command syntax shows TRIGger, then TRIG and TRIGGER are both acceptable forms. Other forms of TRIGger, such as TRIGG or TRIGGE will generate an error. You may use upper or lower case letters. Therefore, TRIGGER, trigger, and TrIgGeR 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.) You can send a second level command, but you do not need to send the preceding implied command.

For example, the instrument assumes you intend to use the implied command and it responds as if you had sent it. Examine the INITiate subsystem shown below:

```
INITiate
    :CONTinuous ON | OFF | 1 | 0
    :CONT inuous?
    [:IMMediate]
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The root command INITiate is sent without a secondary command. IMMediate is implied. You can send either of the following commands to start the scanning cycle:

```
INIT:IMM
        or
INIT
```

Parameters

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

Table 4–1. SCPI Parameter Types

Parameter Type	Explanations and Examples
Numeric	Accepts all commonly used decimal representations of number including optional signs, decimal points, and scientific notation. 123, 123E2, −123, −1.23e2, 1.23Ee−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.
Discrete	Selects from a finite set of values. These parameters use mnemonics to represent each valid setting. An example is TRIGger:SOURce <*source*> where <*source*> can be BUS, EXT, HOLD, or IMM.

Optional Parameters. Parameters shown within square brackets ([ ]) are optional parameters. (Note that the brackets are not part of the command and are not sent to the instrument.) If you do not specify a value for an optional parameter, the instrument chooses a default value.

For example, consider the ARM:COUNt? [<FIN|MAX>] command. If you send the command without specifying a parameter, the present ARM:COUNt value is returned. If you send the MIN parameter, the command returns the minimum count available. If you send the MAX parameter, the command returns the maximum count available. Be sure to place a space between the command and the parameter.

Linking Commands

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

```
*RST;OUTP ON
   or
TRIG:SOUR HOLD;*RST
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

```
ARM:COUN 1;:TRIG:SOUR EXT
```

SCPI Command Reference

This section describes the Standard Commands for Programmable Instruments (SCPI) commands for the Form C switch module. Commands are listed alphabetically by subsystem and also within each subsystem. Command guides are printed in the top margin of each page. The left guide indicates the first command listed on that page. The right guide indicates the last command listed on that page. Where only a single command appears on a page, the left and right guides will be the same.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ABORt

The ABORt command subsystem stops a scan in progress when the scan is enabled via the interface and the trigger source is TRIGger:SOURce BUS or TRIGger:SOURce HOLD.

Subsystem Syntax ABORt

Comments

- **ABORt Actions:** ABORt aborts the scan and invalidates the current channel list.
- **Stopping Scan Enabled Via Interface:** When a scan is enabled via an interface, an interface CLEAR command can be used to stop the scan. When the scan is enabled via the interface and TRIG:SOUR BUS or HOLD is set you can use ABORt to stop the scan.
- **Related Commands:** ARM, INITiate:CONTinuous, [ROUTe:]SCAN, TRIGger

Example **Stopping a Scan With ABORt**

This example stops a (continuous) scan in progress.

```
TRIG:SOUR BUS          *TRG command is trigger source
INIT:CONT ON           Set continuous scanning
SCAN (@100:103)        Scan channels 00–03
INIT                   Start scan, close channel 00
 .
 .
ABOR                   Abort scan in progress
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ARM

The ARM subsystem selects the number of scanning cycles (1 to 32767) for each INITiate command.

Subsystem Syntax

```
ARM
    :COUNt <number> MIN|MAX
    :COUNt? [MIN|MAX]
```

:COUNt

ARM:COUNt <*number*>MIN|MAX allows scanning cycles to occur a multiple of times (1 to 32767) with one INITiate command when INITiate:CONTinuous OFF|0 is set. MIN sets 1 cycle and MAX sets 32767 cycles.

Parameters

Parameter Name	Parameter Type	Range of Values
number	numeric	1–32767|MIN|MAX

Comments

- **Number of Scans:** Use only values between 1 and 32767 for the number of scanning cycles.
- **Related Commands:** ABORt, INITiate:IMMediate
- ***RST Condition:** ARM:COUNt 1

Example **Setting Ten Scanning Cycles**

This example sets a Form C switch for 10 scans of channels 00 through 03. When the scan sequence completes, channels 00 through 03 (relays 00 through 03) are closed.

```
ARM:COUN 10              Set 10 scans per INIT command
SCAN (@100:103)          Scan channels 00–03
INIT                     Start scan, close channel 00
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

:COUNt?

ARM:COUNt? [MIN|MAX] returns the current number of scanning cycles set by ARM:COUNt. The current number of scan cycles is returned when MIN or MAX is not supplied. With MIN or MAX as a parameter, MIN returns 1 and MAX returns 32767.

Parameters

Parameter Name	Parameter Type	Range of Values
MIN | MAX	numeric	MIN=1, MAX=32767

Comments

- **Related Commands:** INITiate[:IMMediate]

Example **Query Number of Scans**

This example sets a switchbox for 10 scanning cycles of channels 00 through 03 and queries the number of scan cycles set. The ARM:COUN? command returns 10.

```
ARM:COUN 10          Set 10 scans per INIT command
ARM:COUN?            Query number of scans
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DISPlay

The DISPlay subsystem monitors the channel state of the selected module in a switchbox. This subsystem operates with an HP E1405B when a terminal is connected.

Subsystem Syntax

```
DISPlay
    :MONitor
                [:STATe] <boolean>
                :CARD <number | AUTO>
```

:MONitor[:STATe]

DISPlay:MONitor[:STATe] *<boolean>* turns the monitor mode on or off.

Parameters

Parameter Name	Parameter Type	Range of Values
ON|OFF|1|0	boolean	ON|OFF|1|0

Comments

- **Monitoring Switchbox Channels:** DISPlay:MONitor[STATe] ON or DISPlay:MONitor[:STATe] 1 turns the monitor mode ON to show the channel state of the selected module. DISPlay:MONitor[:STATe] OFF or DISPlay:MONitor[:STATe] 0 turns the channel monitor OFF.
- **Selecting the Module to be Monitored:** Use the DISPlay:MONitor:CARD *<number>* AUTO command to select the module.
- **Monitor Mode with an HP Z2417A:** When monitoring mode is turn on, decimal numbers representing the channels closed will be displayed at the bottom of the terminal. For example, if channels 3, 7, and 12 are closed, the bottom of the display will read as follows:

  Chan , , ,3, , , ,7, , , , ,12, , , , , , . . . etc.

Where the channel numbers represent the channels that are closed.

- ***RST Condition:** DISPlay:MONitor[:STATe] OFF | 0

Example **Enabling the Monitor Mode**

```
DISP:MON:CARD 2          Selects module #2 in a switch-box
DISP:MON 1               Turns monitor mode ON
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

:MONitor:CARD

DISPlay:MONitor:CARD *<number>* | AUTO selects the module in a switchbox to be monitored.

Parameters

Parameter Name	Parameter Type	Range of Values
number | AUTO	numeric	1–99

Comments

- **Selecting a Specific Module to be Monitored:** Use the DISPlay:MONitor:CARD command to send the card number for the switchbox to be monitored.
- **Selecting the Present Module to be Monitored:** Use the DISPlay:MONitor:CARD AUTO command to select the last module addressed by a switching command (for example, [ROUTe:]CLOSe).
- ***RST Conditions:** DISPlay:MONitor:CARD AUTO

Example

Select Module #2 in a Switchbox for Monitoring

```
DISP:MON:CARD 2
```

Selects module #2 in a switchbox

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INITiate

The INITiate command subsystem selects continuous scanning cycles and starts the scanning cycle.

Subsystem Syntax

```
INITiate
        :CONTinuous ON|OFF|1|0
        :CONTinuous?
        [:IMMediate]
```

:CONTinuous

INITiate:CONTinuous ON|OFF|1|0 enables or disables continuous scanning cycles for the switchbox.

Parameters

Parameter Name	Parameter Type	Range of Values
ON|OFF|1|0	boolean	ON|OFF|1|0

Comments

- **Continuous Scanning Operation:** Continuous scanning is enabled with the INITiate:CONTinuous ON or INITiate:CONTinuous 1 command. Sending the INITiate:IMMediate command closes the first channel in the channel list. Each trigger from a trigger source selected by the TRIGger:SOURce command advances the scan through the channel list. A trigger at the end of the channel list closes the first channel in the channel list and the scan cycle repeats.
- **Non-Continuous Scanning Operation:** Non-continuous scanning is enabled with the INITiate:CONTinuous OFF or INITiate:CONTinuous 0 command. Sending the INITiate:IMMediate command closes the first channel in the channel list. Each trigger from a trigger source selected by the TRIGger:SOURce command advances the scan through the channel list. At the end of the scanning cycle, the last channel in the channel list is opened.
- **Stopping Continuous Scan:** See the ABORt command.
- **Related Commands:** ABORt, ARM:COUNt, TRIGger, TRIGger:SOURce
- ***RST Condition:** INITiate:CONTinuous OFF

Example **Enabling Continuous Scanning**

This example enables continuous scanning of channels 00 through 03 of a single-module switchbox. Since TRIGger:SOURce IMMediate (default) is set, use an interface clear command (such as CLEAR) to stop the scan.

```
INIT:CONT ON          Enable continuous scanning
SCAN (@100:103)       Defines channel list
INIT                  Start scan cycle, close channel 00
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

:CONTinuous?

INITiate:CONTinuous? queries the scanning state. With continuous scanning enabled, the command returns 1. With continuous scanning disabled, the command returns 0.

Example **Query Continuous Scanning State**

This example enables continuous scanning of a switchbox and queries the state. Since continuous scanning is enabled, INIT:CONT? returns 1.

```
INIT:CONT ON          Enable continuous scanning
INIT:CONT?            Query continuous scanning state
```

[:IMMediate]

INITiate[:IMMediate] starts the scanning process and closes the first channel in the channel list. Successive triggers from the source selected by the TRIGger:SOURce command advance the scan through the channel list.

Comments

- **Starting the Scanning Cycle:** The INITiate:IMMediate command starts scanning by closing the first channel in the channel list. Each trigger received advances the scan to the next channel in the channel list. An invalid channel list definition causes an error (see [ROUTe:]SCAN).
- **Stopping Scanning Cycles:** See ABORt.

Example **Enabling a Single Scan**

This example enables a single scan of channels 00 through 03 of a single−module switchbox. The trigger source to advance the scan is immediate (internal) triggering set with (default) TRIGger:SOURce IMMediate.

```
SCAN (@100:103)       Scan channels 00−03
INIT                  Begin scan, close channel 00
                      (use immediate triggering).
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OUTPut

The OUTPut command subsystem enables or disables the different trigger lines of the HP E1405A/B Command Module.

OUTPut
:EXTernal[:STATe] ON | OFF | 1 | 0
:EXTernal[:STATe]?
[:STATe] ON | OFF | 1 | 0
[:STATe]?
:TTLTrg <*line*>[:STATe] ON | OFF | 1 | 0
:TTLTrg <*line*>[:STATe]?

:EXTernal[:STATe]

OUTPut:EXTernal[:STATe] ON | OFF | 1 | 0 enables or disables the "Trig Out" BNC port on the HP E1405A/B Command Module.
OUTPut:EXTernal[:STATe] ON | 1 enables the port and
OUTPut:EXTernal[:STATe] OFF | 0 disables the port.

Parameters

Parameter Name	Parameter Type	Range of Values
ON|OFF|1|0	boolean	ON|OFF|1|0

Comments

- **Enabling Trig Out Port:** When enabled, a pulse is output from the "Trig Out" BNC port after each scanned switchbox channel is closed. If disabled, a pulse is not output from the BNC port after channel closures. The output pulse is a + 5 V negative−going pulse.
- **Trig Out Port Shared by Switchboxes:** When enabled, the "Trig Out" port is pulsed by any switchbox each time a scanned channel is closed. To disable the output for a specific module send the OUTPut:EXTernal[:STATe] OFF or OUTPut:EXTernal[:STATe] 0 command for that module.
- **One Output Selected at a Time:** Only one output (TTLTrg or EXTernal) can be enabled at one time. Enabling a different output source will automatically disable the active output.
- **Related Commands:** [ROUTe:]SCAN, TRIGger:SOURce
- ***RST Condition:** OUTPut:EXTernal[:STATe] OFF (port disabled)

Example

```
OUTP:EXT ON
```

Enable "Trig Out" BNC port to output pulse after each scanned channel is closed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

:EXTernal[:STATe]?

OUTPut:EXTernal[:STATe]? queries the present state of the "Trig Out" BNC port. The command returns 1 if the port is enabled or 0 if the port is disabled.

Example **Query Trig Out BNC Port Enable State**

This example enables the "Trig Out" Port and queries the enable state. The OUTPut:EXTernal[:STATe]? command returns 1 since the port is enabled.

```
OUTP:EXT ON          Enable "Trig Out" BNC Port
OUTP:EXT ?           Query port enable state
```

[:STATe]

OUTPut[:STATe] ON | OFF | 1 | 0 enables or disables the "Trig Out" BNC port on the HP E1405A/B Command Module. OUTPut[:STATe] ON | 1 enables the port and OUTPut[:STATe] OFF | 0 disables the port. This command functions the same as the OUTPut:EXTernal[:STATe] command.

Parameters

Parameter Name	Parameter Type	Range of Values
ON|OFF|1|0	boolean	ON|OFF|1|0

Comments

- ***RST Condition:** OUTPut[:STATe] OFF (port disabled)

Example

```
OUTP ON              Enable "Trig Out" BNC port to
                     output pulse after each scanned
                     channel is closed
```

[:STATe]?

OUTPut[:STATe]? queries the present state of the "Trig Out" BNC port. The command returns 1 if the port is enabled or 0 if the port is disabled. This command functions the same as the OUTPut:EXTernal[:STATe]? command.

Example **Query Trig Out BNC Port Enable State**

This example enables the "Trig Out" Port and queries the enable state. The OUTPut[:STATe]? command returns 1 since the port is enabled.

```
OUTP ON              Enable "Trig Out" BNC Port
OUTP ?               Query port enable state
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

:TTLTrg <line>[:STATe]

OUTPut:TTLTrg <*line*> [:STATe] ON | OFF | 1 | 0 selects and enables which TTL trigger bus line (0 – 7) will output a trigger when a channel is closed during a scan. ON|1 enables the specified bus line, and OFF|0 disables the specified bus line.

Parameters

Parameter Name	Parameter Type	Range of Values
ON|OFF|1|0	boolean	ON|OFF|1|0

Comments

- **Enabling TTL Trigger Bus:** When enabled, a pulse is output from the selected TTL trigger bus line (0 – 7) after each channel in the switchbox is closed. If disabled, a pulse is not output. The output is a negative going pulse.
- **One Output Selected at a Time:** Only one output (TTLTrg or EXTernal) can be enabled at one time. Enabling a different output source will automatically disable the active output.
- **Related Commands:** [ROUTe:]SCAN, TRIGger:SOURce, OUTPut:TTLTrg <*line*>[:STATe]?
- ***RST Condition:** OUTP:TTLT <*line*>:STAT 0

Example **Enable the TTLT1 trigger bus**

```
OUTP:TTLT1:STAT ON
```

:TTLTrg <line>[:STATe]?

OUTPut:TTLTrg <*line*> [:STATe]? queries the present state of the specified TTL trigger bus line. The command returns 1 if the specified bus line is enabled, or 0 if the specified bus line is disabled.

Example **Query for the state of TTLTrg1**

```
OUTP:TTLT1:STAT?
enter statement                    !Returns the current state
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

[ROUTe:]

The ROUTe command subsystem controls switching and scanning operations for Form C switch modules in a switchbox.

Subsystem Syntax

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

Note

There must be a space between the second level command (CLOS, for example) and the *<channel list>*.

CLOSe

[ROUTe:]CLOSe *<channel_list>* closes the Form C switch channels specified by *<channel_list>*. *<channel_list>* has the form (@ccnn) where cc = card number (01–99) and nn = channel number (00–31).

Parameters

Parameter Name	Parameter Type	Range of Values
channel list	numeric	cc00 – cc31

Comments

- **Closing Channels:** To close a single channel, use [ROUTe:]CLOSe (@ccnn); for multiple channels use [ROUTe:]CLOSe (@ccnn,ccnn,...); for sequential channels, use [ROUTe:]CLOSe (@ccnn:ccnn); for groups of sequential channels, use [ROUTe:]CLOSe (@ccnn:ccnn,ccnn:ccnn); or any combination. Closure order for multiple channels with a single command is not guaranteed.
- **Related Commands:** [ROUTe:]OPEN, [ROUTe:]CLOSe?
- ***RST Condition:** All channels open

Example **Closing Form C Switch Channels**

This example closes channels 100 and 213 of a two–module switchbox (card numbers 01 and 02).

```
CLOS (@100,213)          Close channels 100 and 213.
                         100 closes channel 00 of card
                         #1 and 213 closes channel 13
                         of card 2.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CLOSe?

[ROUTe:]CLOSe? <*channel_list*> returns the current state of the channel(s) queried. <*channel_list*> has the form (@ccnn) (see [ROUTe:]CLOSe for definition). The command returns 1 if channel(s) are closed or returns 0 if channel(s) are open.

Comments

- **Query is Software Readback:** The [ROUTe:]CLOSe? command returns the current software state of the channel(s) specified. It does not account for relay hardware failures. A maximum of 128 channels at a time can be queried for a multi−module switchbox.

Example Query Channel Closure

This example closes channels 100 and 213 of a two−module switchbox and queries channel closure. Since the channels are programmed to be closed 1,1 is returned as a string.

```
CLOS (@100,213)          Close channels 100 and 213
CLOS? (@100,213)         Query channels 100 and 213
                         state
```

OPEN

[ROUTe:]OPEN <*channel_list*> opens the Form C switch channels specified by <*channel_list*>. <*channel_list*> has the form (@ccnn) where cc = card number (01−99) and nn = channel number (00−31).

Parameters

Parameter Name	Parameter Type	Range of Values
channel list	numeric	cc00 − cc31

Comments

- **Opening Channels:** To open a single channel, use [ROUTe:]OPEN (@ccnn); for multiple channels use [ROUTe:]OPEN (@ccnn,ccnn,...); for sequential channels, use [ROUTe:]OPEN (@ccnn:ccnn); for groups of sequential channels, use [ROUTe:]OPEN (@ccnn:ccnn,ccnn:ccnn); or any combination. Opening order for multiple channels with a single command is not guaranteed.
- **Related Commands:** [ROUTe:]CLOSe, [ROUTe:]OPEN?
- ***RST Condition:** All channels open.

Example Opening Channels

This example opens channels 100 and 213 of a two−module switchbox (card numbers 01 and 02).

```
OPEN (@100,213)          Open channels 100 and 213.
                         100 opens channel 00 of #1
                         and 213 opens channel 13 of
                         card #2.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OPEN?

[ROUTe:]OPEN? <*channel_list*> returns the current state of the channel(s) queried. <*channel_list*> has the form (@ccnn) (see [ROUTe:]OPEN for definition). The command returns 1 if channel(s) are open or returns 0 if channel(s) are closed.

Comments

- **Query is Software Readback:** The ROUTe:OPEN? command returns the current software state of the channels specified. It does not account for relay hardware failures. A maximum of 128 channels at a time can be queried for a multi−module switchbox.

Example

Query Channel Open State

This example opens channels 100 and 213 of a two−module switchbox and queries channel 213 state. Since channel 213 is programmed to be open, 1 is returned.

```
OPEN (@100,213)          Open channels 100 and 213
OPEN? (@213)             Query channel 213 state
```

SCAN

[ROUTe:]SCAN <*channel_list*> defines the channels to be scanned. <*channel_list*> has the form (@ccnn) where cc = card number (01−99) and nn = channel number (00−31).

Parameters

Parameter Name	Parameter Type	Range of Values
channel list	numeric	cc00 − cc31

Comments

- **Defining Scan List:** When [ROUTe:]SCAN is executed, the channel list is checked for valid card and channel numbers. An error is generated for an invalid channel list.
- **Scanning Channels:** You can scan single channels (@ccnn); multiple channels (@ccnn,ccnn,...); sequential channels (@ccnn:ccnn); groups of sequential channels (@ccnn:ccnn,ccnn:ccnn); or any combination.
- **Scanning Operation:** When a valid channel list is defined, INITiate[:IMMediate] begins the scan and closes the first channel in the channel list. Successive triggers from the source specified by TRIGger:SOURce advances the scan through the channel list.
- **Stopping Scan:** See ABORt.
- **Related Commands:** TRIGger, TRIGger:SOURce
- ***RST Condition:** All channels open.

See Chapter 3, for example scanning programs using external instruments.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

STATus

The STATus subsystem reports the bit values of a Standard Operation Status Register. This register enables the Status Byte Register to set a bit after a bit is set to 1 by the Standard Operation Status Register.

Subsystem Syntax

```
STATus
        :OPERation
                [:EVENt]?
                :ENABle <number>
                :ENABle?
```

:OPERation[:EVENt]?

STATus:OPERation[:EVENt]? returns the bit value of the Standard Operation Status Register (only bit 8 is used by the Form C switch module).

Comments

- **Setting Bit 8 of the Standard Operation Status Register:** Bit 8 (Scan Complete) is set to 1 after a scanning cycle completes. Bit 8 returns to 0 after sending the STATus:OPERation[:EVENt]? command.
- **Returned Data After Sending the STATus:OPERation[:EVENt]? Command:** The command returns +256 if bit 8 of the Standard Operation Status Register is set to 1. The command returns +0 if bit 8 of the Standard Operation Status Register is set to 0.
- **Related Commands:** [ROUTe:] SCAN

Example **Reading the Standard Operation Status Register After a Scanning Cycle**

```
STAT:OPER?
```

Returns the bit values of the Standard Operation Status Register. Returns +256 if bit 8 is set to 1 or returns +0 if bit 8 is set to 0.

:OPERation:ENABle

STATus:OPERation:ENABle *<number>* enables the Standard Operation Status Register to set a bit in the Status Byte Register. For the Form C switch, when bit 8 in the Standard Operation Status Register is set to 1, bit 7 in the Status Byte Register is set to 1.

Parameters

Parameter Name	Parameter Type	Range of Values
number	numeric	0–32,767

Comments

- **Setting Bit 7 of the Status Byte Register:** STATus:OPERation:ENABle 256 sets bit 7 of the Status Register to 1 after bit 8 of the Standard Operation Status Register is set to 1.
- **Related Commands:** [ROUTe:]SCAN

Example **Enabling Standard Operation Status Register Bit 8**

```
STAT:OPER:ENAB 256
```

Enables bit 8 of the Standard Operation Status Register and

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

sets bit 7 in the Status Byte Register

:OPERation:ENABle?

STATus:OPERation:ENABle? returns the bit value of the Standard Operation Status Register.

Comments

- **Output Format:** Returns a decimal weighted value from 0 to 65535 indicating which bits are set to true.
- **Maximum Value Returned:** The maximum decimal weighted value used in this module is 256 (bit 8 set to true).

Example

Query the Standard Operation Status Enable Register

```
STAT:OPER:ENAB?
```

Queries the Standard Operation Status Enable register

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SYSTem

The SYSTem subsystem returns the error numbers and error messages in the error queue of a switchbox. It can also return the types and descriptions of modules (cards) in a switchbox.

Subsystem Syntax

```
SYSTem
        :ERRor?
        :CDEScription? <number>
        :CTYPe? <number>
        :CPON <number>|ALL
```

:ERRor?

SYSTem:ERRor? returns the error numbers and corresponding error messages in the error queue of a switchbox. See Appendix C for a listing of some switchbox error numbers and messages.

Comments

- **Error Numbers/Messages in the Error Queue:** Each error generated by a switchbox stores an error number and corresponding error message in the error queue. 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? command 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 for each switchbox. 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.

Example **Reading the Error Queue**

```
SYST:ERR?                    Query the error queue
```

:CDEScription?

SYSTem:CDEScription? *number*| **returns the description of a selected module (card) in a switchbox.**

Parameters

Parameter Name	Parameter Type	Range of Values
number	numeric	1–99

Comments

- **Form C Switch Module Description:** The SYSTem:CDEScription? <*number*> command returns: "32 Channel General Purpose Relay"

Example **Reading the Description of a Card #1 Module**

```
SYST:CDES? 1                 Return the description
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

:CTYPe?

SYSTem:CTYPe? *<number> returns the module (card) type of a selected module in a switchbox.*

Parameters

Parameter Name	Parameter Type	Range of Values
number	numeric	1–99

Comments

- **Form C Switch Module Model Number:** The SYSTem:CTYPe? *<number>* command returns:

  HEWLETT–PACKARD,Z2417A,0,A.04.00

  where the 0 after Z2417A is the module serial number (always 0) and A.04.00 is an example of the module revision code number.

Example **Reading the Model Number of a Card #1 Module**

```
SYST:CTYP? 1                    Return the model number
```

:CPON

SYSTem:CPON *<number>* | ALL sets the selected module (card) in a switchbox to its power–on state.

Parameters

Parameter Name	Parameter Type	Range of Values
number	numeric	1–99

Comments

- **Form C Switch Module Power–On State:** The power–on state is all channels (relays) open. Note that *RST opens all channels of all modules in a switchbox, while SYSTem:CPON *<number>* opens the channels in only the module (card) specified in the command.

Example **Setting Card #1 Module to its Power–On State**

```
SYST:CPON 1                     Sets module #1 to power–on
                                state
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TRIGger

The TRIGger command subsystem controls the triggering operation of a Form C switch module in a switchbox.

Subsystem Syntax

```
TRIGger
        [:IMMediate]
        :SOURce BUS | EXTernal | HOLD | IMMediate | TTLTrg <line>
        :SOURce?
```

[:IMMediate]

TRIGger[:IMMediate] causes a trigger event to occur when the defined trigger source is TRIGger:SOURce BUS or TRIGger:SOURce HOLD.

Comments

- **Executing the TRIGger[:IMMediate] Command:** A channel list must be defined with [ROUTe:]SCAN *<channel_list>* and an INITiate[:IMMediate] command must be executed before TRIGger[:IMMediate] will execute.
- **BUS or HOLD Source Remains:** If selected, the TRIGger:SOURce BUS or TRIGger:SOURce HOLD commands remain in effect after triggering a switchbox with the TRIGger[:IMMediate] command.
- **Related Commands:** INITiate, [ROUTe:]SCAN

Example **Advancing Scan Using TRIGger Command**

This example uses the TRIGger command to advance the scan of a single-module switchbox from channel 00 through 03. Since TRIGger:SOURce HOLD is set, the scan is advanced one channel each time TRIGger is executed.

```
TRIG:SOUR HOLD          Sets trigger source to HOLD
SCAN (@100:103)         Defines channel list
INIT                    Begin scan, close channel 00
loop statement          Start count loop
TRIG                    Advance scan to next channel
increment loop          Increment loop count
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

:SOURce

TRIGger:SOURce BUS|EXTernal|HOLD|IMMediate|TTLTrg specifies the trigger source to advance the channel list during scanning.

Parameters

Parameter Name	Parameter Type	Parameter Description
BUS	discrete	*TRG or GET command
EXTernal	discrete	"EVENT In" BNC port
HOLD	discrete	Hold triggering
IMMediate	discrete	Immediate triggering
TTLTrg	discrete	Trigger lines 0–7

Comments

- **Enabling the Trigger Source:** The TRIGger:SOURce command only selects the trigger source. The INITiate[:IMMediate] command enables the trigger source.
- **Using the TRIG Command:** You can use TRIGger[:IMMediate] to advance the scan when TRIGger:SOURce BUS or TRIGger:SOURce HOLD is selected.
- **Using External Trigger Inputs:** With TRIGger:SOURce EXTernal selected, only one switchbox at a time can use the external trigger input at the HP E1405A/B Command Module. The trigger input is assigned to the first switchbox that requested the external trigger source (with a TRIGger:SOURce EXTernal command).
- **Assigning External Trigger:** A switchbox assigned with TRIGger:SOURce EXTernal remains assigned to that source until the switchbox trigger source is changed to BUS, HOLD, or IMMediate. When the source is changed, the external trigger source is available to the next switchbox which requests it (with a TRIGger:SOURce EXTernal command). If a switchbox requests an external trigger input already assigned to another switchbox an error is generated.
- **Using Bus Triggers:** To trigger the switchbox with TRIGger:SOURce BUS selected, use the IEEE 488.2 common command *TRG or the HP–IB Group Execute Trigger (GET) command.
- **Trig Out Port Shared by Switchboxes:** See the OUTPut command.
- **Related Commands:** ABORt, [ROUTe:]SCAN, TRIGger
- ***RST Condition:** TRIGger:SOURce IMMediate

Examples

Scanning Using External Triggers

This example uses external triggering (TRIG:SOUR EXT) to scan channels 00 through 03 of a single–module switchbox. The trigger source to advance the scan is the input to the "Event In" BNC on an HP E1405A/B Command Module. When INIT is executed, the scan is started and channel 00 is closed. Then each trigger received at the "Event In" BNC advances the scan to the next channel.

```
TRIG:SOUR EXT                 Select external triggering
SCAN (@100:103)               Scan channels 00 through 03
INIT                          Begin scan, close channel 00
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
trigger externally              Advance scan to next channel
```

Scanning Using Bus Triggers

This example uses bus triggering (TRIG:SOUR BUS) to scan channels 00 through 03 of a single–module switchbox. The trigger source to advance the scan is the *TRG command (as set with TRIGger:SOURce BUS). When INIT is executed, the scan is started and channel 00 is closed. Then, each *TRG command advances the scan to the next channel.

```
TRIG:SOUR BUS                   Select interface (bus) triggering
SCAN (@100:103)                 Scan channels 00 through 03
INIT                            Start scan, close channel 00
loop statement                  Loop to scan all channels
*TRG                            Advance scan using bus trigger-
                                ing
increment loop                  Increment loop count
```

:SOURce?

TRIGger:SOURce? returns the current trigger source for the switchbox. Command returns BUS, EXT, HOLD, IMM, or TTLT for sources BUS, EXTernal, HOLD, IMMediate, or TTLTrg respectively.

Example **Query Trigger Source**

This example sets external triggering and queries the trigger source. Since external triggering is set, TRIG:SOUR? returns EXT.

```
TRIG:SOUR EXT                   Set external trigger source
TRIG:SOUR?                      Query trigger source
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IEEE 488.2 Common Commands

The following table lists the IEEE 488.2 Common (*) Commands that apply to the Form C switch module. The operation of some of these commands is described in Chapter 3 of this manual. For more information on Common Commands, refer to the *HP E1405A/B Command Module User's Manual* or the ANSI/IEEE Standard 488.2–1987.

Command	Title	Description
*IDN?	Identification	Returns identification sting of the switchbox
*RST	Reset	Opens all channels, and sets the module to a know state
*TST?	Self–Test	Returns +0 is self–test passes Returns +cc01 for firmware error Returns +cc02 for bus error Returns +cc10 if an interrupt was expected but not received Returns +cc11 if the busy bit was not held for 3 msec
*OPC	Operation Complete	Sets the request for OPC flag when all pending operations have completed. Also sets OPC bit in the Standard Event Register.
*OPC?	Operation Complete Query	Returns a 1 to the output query when all pending operations have completed. Used to synchronize between multiple instruments.
*WAI	Wait to Complete	Prevents an instrument from executing another command until the operation caused by the previous command is finished. Since all instruments normally preform sequential operations, executing this command causes no change.
*CLS	Clear status	Clear all statue registers (see STATus:OPERation[:EVENt]?)
*ESE	Event status enable	Enables statue registers
*ESE?	Event status enable query	Queries for the current contents of the Event Status Register
*ESR?	Event status register query	Queries and clears the current contents in the Event Status Register
*SRE	Service request enable	Sets the Service Request Enable register bits and corresponding serial pool status register bits, to generate a service request.
*SRE?	Service request enable query	Queries the current contents in the Service Request Enable Register
*STB?	Read status byte query	Queries the current contents in the Status Byte Register
*TRG	Trigger	Trigger the switchbox to advance the scan when scan is enabled and trigger source is TRIGger:SOURce BUS
*RCL	Recall instrument state	Recalls previously stored configuration
*SAV	Store instrument state	Stores the current configuration in specified memory

Note: These commands apply to many instruments and are not documented in detail here. See the *HP E1405A/B Command Module User's Manual* or the ANSI/IEEE Standard 488.2–1987 for more information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Quick Reference

The following table summarizes the SCPI commands for the Form C switch.

Command Subsystem	Command/Parameter	Description
ABORt	ABORt	Abort a scan in progress
ARM	:COUNt <number> MIN | MAX :COUNT? [MIN | MAX]	Multiple scans per INIT command. Query number of scans.
DISPlay	:MONitor[:STATe] <number> MONitor:CARD <number>	Turns monitor mode on or off. Selects the module in a switchbox to be monitored.
INITiate	:CONTinuous ON | OFF | 1| 0 :CONTinuous? [:IMMediate]	Enables/Disables continuous scanning. Query continuous scan state. Starts a scanning cycle.
OUTPut	:EXTernal[:STATe] ON | OFF | 1 | 0 :EXTernal[:STATe?] [:STATe] ON | OFF | 1 | 0 [:STATe]? :TTLTrg <line>[:STATe] ON | OFF | 1 | 0 :TTLTrg <line>[:STATe]?	Enables/Disables "TRIG OUT" pulse. Query port enable state. Enables/Disables "TRIG OUT" pulse. Query port enable state. Enables/Disables TTL Trigger bus line pulse. Query TTL Trigger Bus line state.
[ROUTe:]	CLOSe <channel_list> CLOSe? <channel_list> OPEN <channel_list> OPEN? <channel_list> SCAN <channel_list>	Close channel(s). Query channel(s) closed. Open channel(s). Query channel(s) opened. Define channels for scanning.
STATus	:OPERation[:EVENt]? :OPERation:ENABle :OPERation:ENABle?	Returns status of Operation Status Register. Enables the Operation Status Register to set a bit in the Status Register. Query the contents in the Operation Status Register.
SYSTem	:CDEScription? <number> :CPON <number | ALL :CTYPe? :ERRor?	Returns description of module in switchbox. Sets specified module to its power−on state. Returns the module type. Returns error number/message to error queue.
TRIGger	[:IMMediate] :SOURce BUS :SOURce EXTernal :SOURce HOLD :SOURce IMMediate :SOURce TTLTrg <line> :SOURce?	Causes a trigger to occur. Trigger source is *TRG. Trigger source is "TRIG IN" BNC. Hold off triggering. Continuous (internal) triggering. Trigger source is TTL trigger bus line (0 − 7). Query scan trigger source.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Specifications

General

Module Size/ Device Type:
C–Size VXIbus, Register based, A16/D16, Interrupter (levels 1–7, jumper selectable)

Relay Life:

Condition	Number of Operations
No Load	5×10^7

NOTE: Relays are subject to normal wear based on the number of operations.

Terminals:
Screw type, maximum wire size 18 AWG or Solder Eye connections.

Power Requirements:
Voltage: +5
Peak Module Current (A): 0.70

Watts/slot: 10 W

Cooling/slot: .08 mm H_2O @ 1.0 Liter/sec for 10°C rise

Operating Temperature: 0 – 55°C

Operating Humidity: 65% RH, 0 – 40°C

Maximum Voltage:
100V dc; 100 V ac peak Terminal to Terminal and Terminal to Chassis

Maximum Current per Channel (non–inductive):
2A dc or ac peak

Maximum Power per Channel:
50 W dc; 50 VA ac Per Switch
320 W dc; 320 VA ac Per Module

DC Performance

Thermal Offset per Channel:
<20µV

Closed Channel Resistance:
<0.250 Ω (typical)

Installation Resistance (Between any Two Points):
> 10^7 Ω (at <= 40°C,95% RH)
10^8 Ω (at <= 25°C, 40% RH)

AC Performance

Capacitance
<2 pF (channel to channel)
<5 pF (common to NO or NC)
<70pF (channel to chassis)

Input Characteristics

Bandwidth (–3dB) 50 ohm load: > 10MHz (typical)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B

Form C Switch Registers

Using this Appendix

The HP Z2417A 32–Channel Mercury–Wetted Form C Switch Module is a register–based device. When a SCPI command is sent to the Form C module, the HP E1405A/B Command Module parses the command and programs the Form C switch at the register level.

Register–based programming is a series of reads and writes directly to the Form C switch module registers. Writing directly to the registers can increase throughput speed since it eliminates the command parsing and allows the use of an embedded controller. This appendix includes information on the following:

- Addressing the Registers
- Register Descriptions
- Register–Based Programming Examples

Addressing the Registers

Register addresses for register–based modules are located in the upper 25% of VXI A16 address space. Every VXI module (up to 256) is allocated a 64 byte block of addresses.

Figure B–1 shows the register address location within A16. Figure B–2 shows the location of A16 address space in the HP E1405A/B Command Module.

When you are reading or writing to a Form C switch module register, a hexadecimal or decimal register address needs to be specified. This address consists of a base address plus a register offset:

Register Address = Base Address + Register Offset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

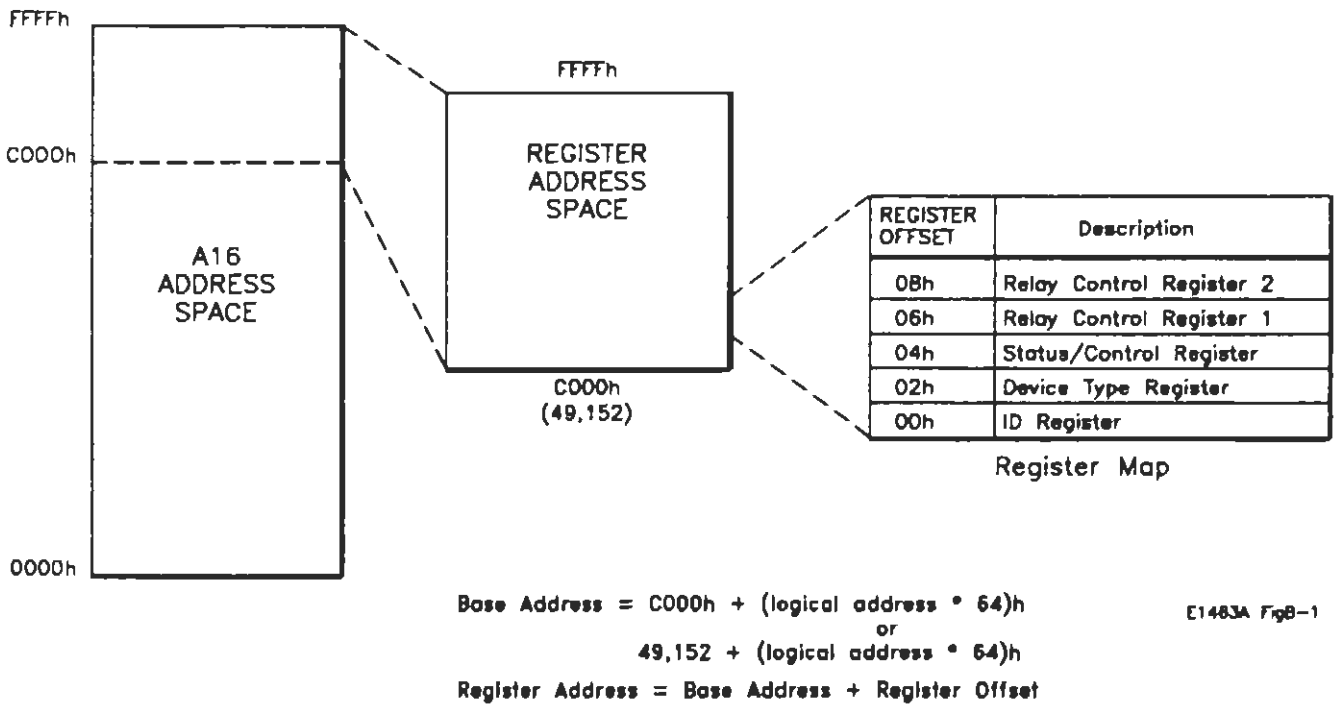


Figure B–1. Register Address Location within A16

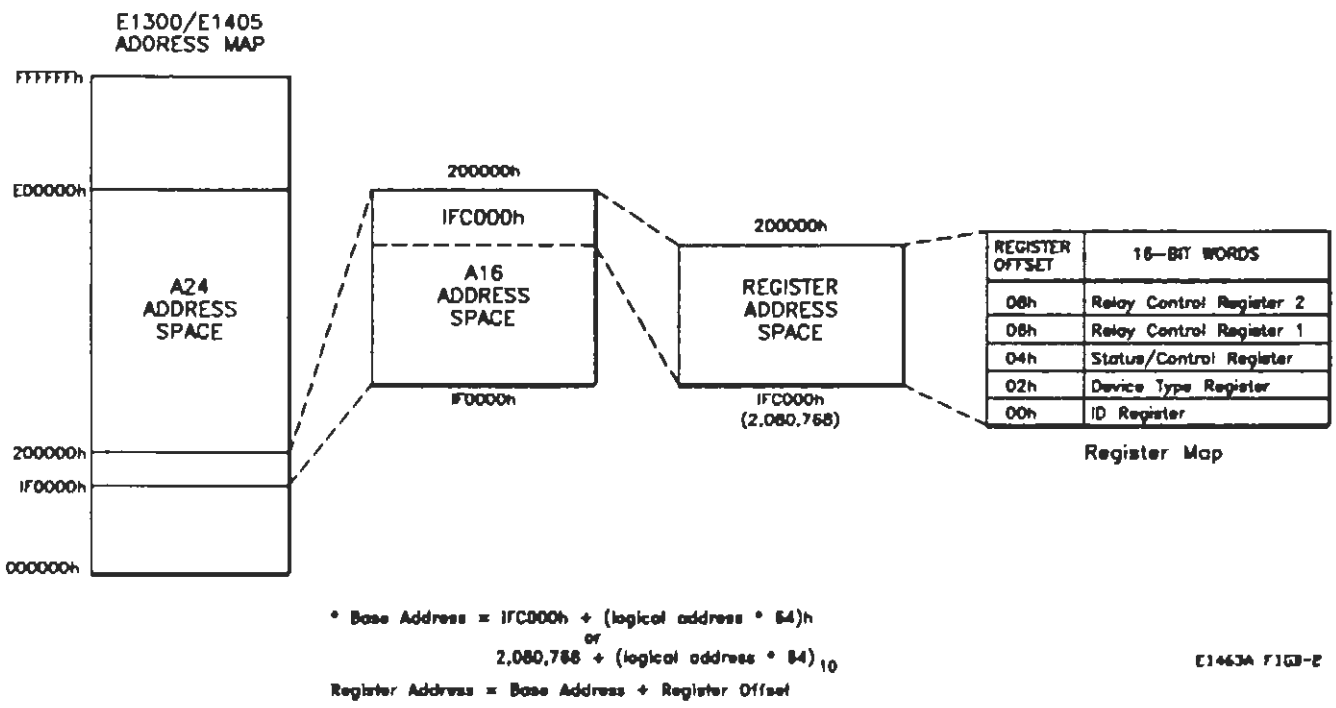


Figure B–2. A16 Register Address Location in the HP E1405A

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Base Address

The base address used in register–based programming depends on whether the A16 address space is outside or inside the HP E1405A/B Command Module.

Address Space Outside the Command Module

When the command module is not part of your VXIbus system (figure B–1), the Form C switch module's base address depends on the command module used:

$$\text{Command Module} + C000_h + (\text{LADDR} * 64)_h$$
$$\text{OR}$$
$$\text{Command Module} + 49{,}152 + (\text{LADDR} * 64)$$

Where $C000_h$ (49,152) is the starting location of the VXI A16 addresses, LADDR is the Form C switch module's logical address, and 64 is the number of address bytes per register–based module. For example, the Form C module's factory set logical address is 120. If the address is not changed and an HP E1480A (V/360) controller is used, the Form C module will have the following base address:

$$C000_h + (120 * 64)_h = DE00_h$$
$$\text{OR}$$
$$49{,}152 + (120 * 64) = 56{,}832$$

Address Space Inside the Command Module

When the A16 address space is inside the command module (figure B–2), the Form C module's base address is computed as follows:

$$1FC000_h + (\text{LADDR} * 64)_h$$
$$\text{OR}$$
$$2080{,}768 + (\text{LADDR} * 64)$$

Where $1FC000_h$ (2080,768) is the starting location of the register addresses, LADDR is the Form C module's logical address, and 64 is the number of address bytes per VXI module. Again, the matrix module's factory set logical address is 120. If the address is not changed, the matrix modules will have the following base address:

$$1FC000_h + (120 * 64)_h = 1FDE00_h$$
$$\text{OR}$$
$$2{,}080{,}768 + (120 * 64) = 2088{,}448$$

Register Offset

The register offset is the register's location in the block of 64 address bytes. For example, the Form C module's Status Register has an offset of 04_h (see the next section). When you write a command to this register, the register offset is added to the base address to form the following register address:

$$1FDE00_h + 04_h = 1FDE04_h$$
$$\text{OR}$$
$$2088{,}448 + 04 = 2088452$$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Register Descriptions

The Form C switch modules contain 2 read registers, 1 read/write register, and 2 write registers. This section describes each Form C module register.

Reading and Writing to the Registers

Example programs are provided at the end of this appendix that show how to read and write to these registers. You can read or write to the following Form C switch module registers:

- The Manufactures Identification Register (read only)
- The Device Type Register (read only)
- The Status/Control Register (read or write)
- Relay Control Register for Channels 00 – 15 (write)
- Relay Control Register for Channels 16 – 31 (write)

Each of these registers is discussed in the following sections.

The Manufactures Identification Register

The manufactures identification register is at offset address 00_h and returns $FFFF_h$. This shows Hewlett–Packard as the manufacture and that the module is an A16 register–based module. This register can not be written to.

B–1. Manufacturer ID Register

b + 4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read*	Manufacturer ID															

* Returns FFFFh = Hewlett–Packard A16 only register–based device

The Device Identification Register

The device identification register is at offset address 02_h and returns 0121_h if you have an HP Z2417A Form C Switch Module. The device identification register can not be written to.

B–2. Device Type Register

b + 4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read	0123h															

The Status/Control Register

The status/control register is at offset address 04_h and informs the user about the module's status and configuration.

When using an HP E1405A/B, the address switch value must be a multiple of 8 if the module is the first module in a "switchbox" used with a VXIbus command module using SCPI commands.**B–3. Status/Control Register**

b + 4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	X	MS	Undefined						B	E	X	X	1	1	X	X
Read**	Undefined									E	Undefined					R

Reading the Status/ Control Register

When reading the status/control register, the following bits are of importance:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Enable (bit 6) 0 indicates that the interrupt is enabled. The interrupt generated after a channel has been closed can be disabled. Bit 6 of this register is used to inform the user of the interrupt status.

Busy (bit 7) 0 indicates that the module is busy. Each relay requires about 10 ms execution time during which time the Form C switch is busy. Bit 7 of this register is used to inform the user of a busy condition.

ModId Select (bit 14) 0 indicates that the module has been selected by MODID (module ID), and a 1 indicates it has not.

For example, if the Form C switch module is not busy (bit 7 = 1) and the interrupt is enabled (bit 6 = 0), then a read of the Status/Control register (base + $04_{h)}$ returns FFBF.

Writing to the Status/ Control Register You can write to bits 0 and 6. See the following explanations:

Soft Reset (bit 0) Writing a 1 to this bit soft resets the module.

Enable (bit 6) Writing a 1 to this bit disables the interrupt function of the module.

Note

When writing to the registers it is necessary to write a "0" to bit 0 after the reset has been performed before any other commands can be programmed and executed. SCPI commands take care of this automatically.Typically, interrupts are only disabled to "peek−poke" a module. Refer to the operating manual of the Command Module used before disabling the interrupt. Writing a "1" to bit 0 resets the switch (all channels open).

The Relay Control Registers There are two relay control registers. These registers are used to connect the common (C) to the normally open (NO) terminal. When reading these registers, $FFFF_h$ is always returned.

- Relay Control Register 1 (base + 06_h)
- Relay Control Register 2 (base + 08_h)

The numbers shown in the register maps indicate the channel number to be written to. Writes to the relay control registers enable you to open or close the desired channel . For example, write a "1" to bit 2 of the relay control register 06_h to close channel 02:

```
WRITEIO −16, (2088448 + 06);4
```

Where, the 4 represents 100 in hexadecimal.

B−4. Relay Control Register (Channels 0 − 15)

b + 6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	Ch15	Ch14	Ch13	Ch12	Ch11	Ch10	Ch9	Ch8	Ch7	Ch6	Ch5	Ch4	Ch3	Ch2	Ch1	Ch0
Read	Always Returns FFFFh															

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B–5. Relay Control Register (Channels 16 – 31)

b + 8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write*	Ch31	Ch30	Ch29	Ch28	Ch27	Ch26	Ch25	Ch24	Ch23	Ch22	Ch21	Ch20	Ch19	Ch18	Ch17	Ch16
Read	Always Returns FFFFh															

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Register–Based Programming Examples

The following sections provide examples in both HP BASIC and HP–UX, C. These examples support the following configuration:

- Mainframe: HP 75000 Series C (HP E1400B)
- Controller: HP V/360 (HP E1480A) w/Resource Manager and Slot 0
- Programming Language: HP BASIC/HP–UX, C
- Switch Card: HP Mercury–Wetted Form C Switch Module (HP Z2417A)

Reading the Registers

The following examples read the module ID, DEVICE ID, and STATUS registers on the Form C module.

HP BASIC

```
10!*************************************************************
**
20 !****                READREG
*****
30 !*************************************************************
*
40  OPTION BASE 1
50  ! Set up arrays to store register names and addresses
60  DIM Reg_name$(1:3)[32], Reg_addr(1:3)
70  !
80   ! Read register names and addresses into the arrays
90  READ Reg_name$(*)
100 READ Reg_addr(*)
110 !
120 ! Set base address variable
130 Base_addr = DVAL("DE00",16)
140 !
150 ! Map the A16 address space in the V/360
160 !
170 CONTROL 16,25;2
180 ! Call the subprogram Read_regs
190 Read_regs(Base_addr, Reg_name$(*), Reg_addr(*))
200 !
210 DATA Identification register, Device register, Status
register
220 DATA 00, 02, 04
230 END

300 ! This subprogram steps through a loop that reads
each register and prints its  contents
310 !
320 SUB Read_regs(Base_addr, Reg_name$(*), Reg_addr(*))
330 !
340 For Number = 1 to 3
350   Register = READIO(-16,Base_addr + Reg_addr(number))
360   PRINT Reg_name$(number); " = "; IVAL$(Register,16)
370 Next Number
380 SUBEND
```

Note

When running HP BASIC on the HP V/360 (HP E1480A), you must include an **–L vxi** option to load the vxi drivers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

HP–UX, C

```
/*****************************************************************/
/***                    readreg.c                                ***/
/*****************************************************************/
#include <sys/vxi.h>>                    /*source file for HP V/360 VXI driv-
ers*/
#include <fcntl.h>>
#include <stdio.h>>
#define logical_address 120              /*logical address of the matrix mod-
ule*/
int fd;
typedef unsigned short word;
typedef struct dev_regs {                /*set up pointers*/
                        unsigned short id_reg;
                        unsigned short device_type;
                        unsigned short status_reg;
                        unsigned short bank0_channels;
} DEV_REGS;
main()
{
                                         /*open the HP V/360 VXI interface*/
fd=open("/dev/vxi/primary",O_RDWR);
if (fd) {
          perror("open");
          exit(1);
 }
                                         /*retrieve the A16 pointers*/
dev=(struct dev_regs *)vxi_get_a16_addr(fd,logical_address);
                                         /*sub to read the registers*/
read_reg(dev);
}                                        /*END of main program*/

/*SUB READ_REG*/

int read_reg(reg_ptr)
DEV_REGS *reg_ptr;
{                                        /*read the id register*/

printf("\n ID Register = 0x%x\n",reg_ptr->>id_reg);
                                         /*read the device type register*/
printf("\n DEVICE TYPE Register = 0x%x\n",reg_ptr->>device_type);
                                         /*read the status register*/
printf("\n STATUS Register = 0x%x\n",reg_ptr->>status_reg);
return;
}
```

Making Measurements

The following examples close bit 1 on bank 0, wait for a measurement to be made, and then open the channel.

You must insert your own programming code for the measurement part of this program. If you are using the HP E1411B, see the *HP E1326B/E1411B 5 1/2–Digit Multimeter User's Manual* for programming examples.

HP BASIC

```
10
!*************************************************************
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
20 !****                    MAKEMEAS                          *****
30
!*************************************************************
40  OPTION BASE 1
50  !Set up arrays to store register names and addresses
60  DIM Reg_name$(1:1)[32], Reg_addr(1:1)
70  !
80  !Read register names and address into the arrays
90  READ Reg_name$(*)
100 READ Reg_addr(*)
110 !
120 !Set base address variable
130 Base_addr = DVAL("DE00",16)
140 !
150 !Map the A16 address space in the V/360
160 CONTROL 16,25;2
170 !Call the subprogram Make_meas
180 Make_meas(Base_addr, Reg_addr(*))
190 !
200 DATA Bank0 channels register
210 DATA 06
220 END

280 ! This subprogram closes bit 1 of bank0 channels,
waits for the channel to be closed, makes a measurement,
and then opens the relay

300 SUB Make_meas(Base_addr, Reg_addr(*))
310 !
320 WRITEIO -16, Base_addr + Reg_addr(1); 1
330   REPEAT
340   UNTIL BIT(READIO(-16,Base_addr+4),7)
            .
            .      ! Make Measurements
            .
380 WRITEIO -16, Base_addr + Reg_addr(1);0
390 SUBEND
```

HP–UX, C

```
/*************************************************************/
/***                                                  makemeas.c
      ***/
/*************************************************************/
#include <time.h>>
#include <sys/vxi.h>>              /*source file for HP V/360 VXI drivers*/
#include <fcntl.h>>
#include <stdio.h>>
#define logical_address 120        /*logical address of the matrix module*/
int fd;
typedef unsigned short word;
typedef struct dev_regs {                    /*set up pointers*/
            unsigned short id_reg;
            unsigned short device_type;
            unsigned short status_reg;
            unsigned short bank0_channels;
} DEV_REGS;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
main()
{
                                                    /*open the HP V/360 VXI interface*/
fd=open("/dev/vxi/primary",O_RDWR);
if (fd) {
          perror("open");
          exit(1);
 }
                                                    /*retrieve the A16 pointers*/
dev=(struct dev_regs *)vxi_get_a16_addr(fd,logical_address);
                                                  /*sub to verify the time to close the switch*/
ver_time();

                                                /*sub to close switch and make measurement*/
make_meas(dev);
}                                                   /*END of main program*/

/*SUB VER_TIME*/

ver_time()
{
sturct timeval first,
              second,
              lapsed;

struct timezone tzp;

getimteofday(&first,&tzp);
for (j=0; j<=10000; j++);
gettimeofday ($second,&tzp);

if (first.tv_usec >> second.tv_usec)
    {
    second.tv_usec +=1000000;
    second.tv_sec--;
}

lapsed.tv_usec = second.tv_usec - first.tv_usec;
lapsed.tv_sec = second.tv_sec - first.tv_sec;

printf("Elapsed time for closing a channel is:  %ld sec %ld usec \n",
lapsed.tv_sec, lapsed.tv_usec);
}

/*SUB MAKE_MEAS*/

int make_meas(reg_ptr)
DEV_REGS *reg_ptr;
{
                                                    /*close bit 1 of bank0 */
reg_ptr->>bank0_channels=0x0001;
for (j=0; j<=10000; j++);                           /*wait for switch to close*/
printf("\n Making Measurement");
          .
          .                                         /*make measurements*/
          .
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
                                                    /*open bit 1 of bank0*/
reg_ptr->>bank0_channels=0x0000;
return;
}
```

Note

The sub **ver_time** allows time for switch closures. This sub should print a time around 10 ms. If the time is less, you must change the value of j in the for loop. For example, instead of 10000, you might have to use 12000.

Scanning Channels

The following examples scan through the bank 0 channels (closing one switch at a time) and make measurements between switch closures.

Again, you must insert your own programming code for the measurement part of this program. If you are using the HP E1411B, see the *HP E1326B/E1411B 5 1/2–Digit Multimeter User's Manual* for programming examples.

HP BASIC

```
10
!*********************************************************
20  !****                  SCANNING                    *****
30
!*********************************************************
40  OPTION BASE 1
50  !Set up arrays to store register names and addresses
60  DIM Reg_name$(1:1)[32], Reg_addr(1:1)
70  !
80  !Read register names and addresses into the arrays
90  READ Reg_name$(*)
100 READ Reg_addr(*)
110 !Set base address variable
110 Base_addr = DVAL("DE00",16)
120 !
130 !Map the A16 address space in the V/360
140 CONTROL 16,25;2
150 !Call the subprogram Scan_meas
160 Scan_meas(Base_addr, Reg_addr(*))
170 !
180 DATA Bank0 channels register
190 DATA 06
200 END

280 ! This subprogram sets all bits in bank0 open then
scan through bank 0, closing one channel at a time (waits
for the channel to be closed) so a measurement can be
made
300 SUB Scan_meas(Base_addr, Reg_addr(*))
310 !
320 WRITEIO -16, Base_addr + Reg_addr(1);0
330 FOR I= 0 to 15
340         WRITEIO -16, Base_addr + Reg_addr(1);2^I
350   REPEAT
360         UNTIL BIT(READIO(-16,Base_addr+4),7)
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
370          PRINT "Making Measurement"
                     .
                     .      ! Make Measurements
                     .
420 NEXT I
430 WRITEIO -16,Base_addr + Reg_addr(1);0
440 SUBEND
```

HP-UX,C

```
/*****************************************************************/
/***                                              scanning.c
 ***/
/*****************************************************************/
#include <time.h
#include <math.h>>              /*file to perform math functions*/
#include <sys/vxi.h>>           /*source file for HP V/360 VXI drivers*/
#include <fcntl.h>>
#include <stdio.h>>

#define logical_address 120     /*logical address of the matrix module*/
#define lastch                  15

int fd, i, reg;
double y;
typedef unsigned short word;
typedef struct dev_regs {       /*set up pointers*/
            unsigned short id_reg;
            unsigned short device_type;
            unsigned short status_reg;
            unsigned short bank0_channels;
} DEV_REGS;
main()
{
                                /*open the HP V/360 VXI interface*/
fd=open("/dev/vxi/primary",O_RDWR);
if (fd) {
      perror("open");
      exit(1);
 }                              /*retrieve the A16 pointers*/
dev=(struct dev_regs *)vxi_get_a16_addr(fd,logical_address);

                                /*sub to verify the time to close the switch*/
ver_time();
                    /*sub to close a set of switches and make measurements*/
scan_meas(dev);
}                               /*END of main program*/

/*SUB VER_TIME*/
ver_time()
{
sturct timeval first,
              second,
              lapsed;

struct timezone tzp;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
getimteofday(&first,&tzp);
for (j=0; j<=10000; j ++);
gettimeofday ($second,&tzp);
if (first.tv_usec > > second.tv_usec)
    {
    second.tv_usec +=1000000;
    second.tv_sec--;
}

lapsed.tv_usec = second.tv_usec - first.tv_usec;
lapsed.tv_sec = second.tv_sec - first.tv_sec;

printf("Elasped time for closing a channel is:  %ld sec %ld usec \n",
lapsed.tv_sec, lapsed.tv_usec);
}

/*SUB SCAN_MEAS*/
int scan_meas(reg_ptr)
DEV_REGS *reg_ptr;
{
                                        /*set bank0 to 000 */
reg_ptr->>bank0_channels=0x000;
i=0;
for (i=0;i=lastch;i ++)
    {
    y=i;
    reg=pow(2.0,y);
    reg_ptr-bank0_channels=reg;
    for (j=0; j<=10000; j ++);          /*wait for switch to be closed*/
    printf("\n Making Measurement");
        .
        .                               /*make measurements*/
        .
    }
return;
}
```

Note

The sub **ver_time** allows time for the switches to close. The program should print a time around 10 ms. If the time is less, you must change the value of j in the for loop. For example, instead of 10000, you might have to use 12000.

Note

The **math.h** include file requires a **-lm** option when compiling the above program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Form C Switch Error Messages

Table C–1 lists the error messages associated with the Form C switch module programmed by SCPI. See the appropriate mainframe manual for a complete list of error messages.

Table C–1. Form C Switch Error Messages

No.	Title	Potential Cause(s)
–109	Missing Parameter	Sending a command requiring a channel list without the channel list
–211	Trigger ignored	Trigger received when scan not enabled. Trigger received after scan complete. Trigger too fast.
–213	Init Ignored	Attempting to execute an INIT command when a scan is already in progress.
–224	Illegal parameter value	Attempting to execute a command with a parameter not applicable to the command
–310	System Error	Too many characters in the channel list expression.
1500	External trigger source already allocated	Assigning an external trigger source to a switchbox when the trigger source has already allocated already been assigned to another switchbox.
2000	Invalid card number	Addressing a module (card) in a switchbox that is not part of the switchbox.
2001	Invalid channel number	Attempting to address a channel of a module in a switchbox that is not supported by the module (e.g., channel 99 of a multiplexer module).
2006	Command not supported on this card	Sending a command to a module (card) in a switchbox that is unsupported by on this card the module.
2008	Scan list not initialized	Executing a scan without the INIT command.
2009	Too many channels in channel list	Attempting to address more channels than available in the switchbox.channel list
2012	Invalid Channel Range	Invalid channel(s) specified in SCAN <*channel_list*>> command. Attempting to begin scanning when no valid channel list is defined.
2600	Function not supported on this card	Sending a command to a module (card) in a switchbox that is not supported by on this card the module or switchbox.
2601	Channel list required	Sending a command requiring a channel list without the channel list.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

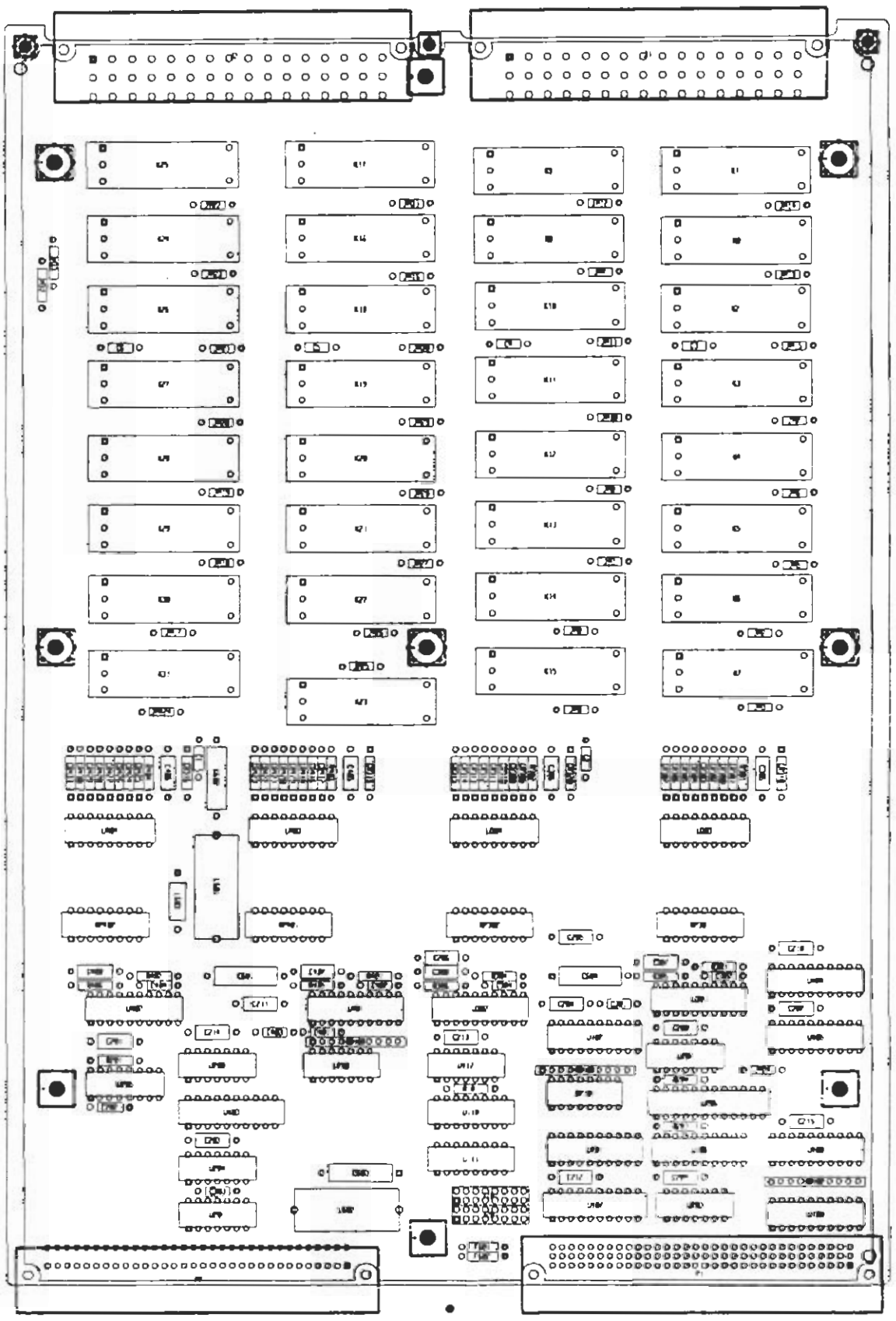
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D

Service Information

The figure below shows the component locator for the Z2417A Switch Module, a parts list follows the figure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Parts Available Through HP Service Centers

A1	Main relay PC Bd	Z2417-66501
K0-K31	Relay Hg	0490-1229
F601,F602	4A	2110-0712

Complete Parts Listing for the Z2417-66501 PC Assembly

LOGIC AND DRIVERS

U404	ULN2803A	1858-0069
U403	ULN2803A	1858-0069
U402	74HCT273	1820-4086
U401	74HCT273	1820-4086
U304	ULN2803A	1858-0069
U303	ULN2803A	1858-0069
U302	74HCT273	1820-4086
U301	74HCT273	1820-4086
U209B	74HC74	1820-3081
U209A	74HC74	1820-3081
U208D	74HC08	1820-3097
U208C	74HC08	1820-3097
U208B	74HC08	1820-3097
U208A	74HC08	1820-3097
U207D	74F38	1820-4057
U207C	74F38	1820-4057
U207B	74F38	1820-4057
U207A	74F38	1820-4057
U206	PAL	1820-6731
U205B	74HC423	1820-4590
U205A	74HC423	1820-4590
U204B	74HC74	1820-3081
U204A	74HC74	1820-3081
U203F	74HCT14	1820-4242
U203E	74HCT14	1820-4242
U203D	74HCT14	1820-4242
U203C	74HCT14	1820-4242
U203B	74HCT14	1820-4242
U203A	74HCT14	1820-4242
U201D	74HCT02	1820-4643
U201C	74HCT02	1820-4643
U201B	74HCT02	1820-4643
U201A	74HCT02	1820-4643
U112C	74HCT02	1820-4643
U112B	74HCT02	1820-4643
U112A	74HCT02	1820-4643
U111	74HC138	1820-3079
U110	74HC138	1820-3079
U109	74ALS245	1820-3714
U108	74ALS245	1820-3714
U107	74HCT573	1820-4147
U106	74HCT688	1820-3631
U105	74HC541	1820-3975
U104	74HC541	1820-3975
U103	74HCT541	1820-4586
U102	74HC541	1820-3975
U101	74HCT688	1820-3631
SP101		3101-2094

RESISTORS and RESISTOR PACKS

RP402	1810-0265	680 ohms	2%
RP401	1810-0265	680 ohms	2%
RP302	1810-0265	680 ohms	2%
RP301	1810-0265	680 ohms	2%
RP103	1810-0279	4.7K	2%
RP102	1810-0279	4.7K	2%
RP101	1810-0279	4.7K	2%
R406	0757-0346	10.0	1%
R405	0757-0346	10.0	1%
R404	0757-0442	10.0k	1%
R403	0757-0442	10.0k	1%
R402	0757-0407	200	1%
R401	0757-0407	200	1%
R306	0757-0346	10.0	1%
R305	0757-0346	10.0	1%
R304	0757-0407	200	1%
R303	0757-0407	200	1%
R302	0757-0442	10.0k	1%
R301	0757-0442	10.0k	1%
R204	0757-0417	562	1%
R203	0757-0453	30.1K	1%
R201	0757-0469	150K	1%
R101	0757-0421	825	1%
P2 DIN	1251-7892		
P1 DIN	1252-1596		

INDUCTORS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

L603	9100-1623	27uH	170mA	
L602	9140-1354	47uH	3.98A	
L601	9140-1354	47uH	3.98A	

RELAYS

K9	relay Hg	0490-1229
K8	relay Hg	0490-1229
K7	relay Hg	0490-1229
K6	relay Hg	0490-1229
K5	relay Hg	0490-1229
K4	relay Hg	0490-1229
K31	relay Hg	0490-1229
K30	relay Hg	0490-1229
K3	relay Hg	0490-1229
K29	relay Hg	0490-1229
K28	relay Hg	0490-1229
K27	relay Hg	0490-1229
K26	relay Hg	0490-1229
K25	relay Hg	0490-1229
K24	relay Hg	0490-1229
K23	relay Hg	0490-1229
K22	relay Hg	0490-1229
K21	relay Hg	0490-1229
K20	relay Hg	0490-1229
K2	relay Hg	0490-1229
K19	relay Hg	0490-1229
K18	relay Hg	0490-1229
K17	relay Hg	0490-1229
K16	relay Hg	0490-1229
K15	relay Hg	0490-1229
K14	relay Hg	0490-1229
K13	relay Hg	0490-1229
K12	relay Hg	0490-1229
K11	relay Hg	0490-1229
K10	relay Hg	0490-1229
K1	relay Hg	0490-1229
K0	relay Hg	0490-1229

JUMPERS AND CONNECTORS

JM9	wire	8159-0005
JM8	wire	8159-0005
JM7	wire	8159-0005
JM6	wire	8159-0005
JM5	wire	8159-0005
JM424	wire	8159-0005
JM4	wire	8159-0005
JM31	wire	8159-0005
JM30	wire	8159-0005
JM3	wire	8159-0005
JM29	wire	8159-0005
JM28	wire	8159-0005
JM27	wire	8159-0005
JM26	wire	8159-0005
JM25	wire	8159-0005
JM24	wire	8159-0005
JM23	wire	8159-0005
JM22	wire	8159-0005
JM21	wire	8159-0005
JM20	wire	8159-0005
JM2	wire	8159-0005
JM19	wire	8159-0005
JM18	wire	8159-0005
JM17	wire	8159-0005
JM16	wire	8159-0005
JM15	wire	8159-0005
JM14	wire	8159-0005
JM13	wire	8159-0005
JM12	wire	8159-0005
JM11	wire	8159-0005
JM10	wire	8159-0005
JM1	wire	8159-0005
JM0	wire	8159-0005
J201	PINS	1251-4927
J2	DIN	1252-1575
J101	PINS	1251-4927
J1	DIN	1252-1575
P2	DIN	1251-7892
P1	DIN	1252-1596

FUSES

F602	2110-0712	4A
F601	2110-0712	4A

DIODES

CR431	1N967B	1902-0964	18.0V	5%
CR430	1N967B	1902-0964	18.0V	5%
CR429	1N967B	1902-0964	18.0V	5%
CR428	1N967B	1902-0964	18.0V	5%
CR427	1N967B	1902-0964	18.0V	5%
CR426	1N967B	1902-0964	18.0V	5%
CR425	1N967B	1902-0964	18.0V	5%
CR424	1N967B	1902-0964	18.0V	5%
CR423	1N967B	1902-0964	18.0V	5%
CR422	1N967B	1902-0964	18.0V	5%
CR421	1N967B	1902-0964	18.0V	5%
CR420	1N967B	1902-0964	18.0V	5%

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CR419	1N967B	1902-0964	18.0V	5%
CR418	1N967B	1902-0964	18.0V	5%
CR417	1N967B	1902-0964	18.0V	5%
CR416	1N967B	1902-0964	18.0V	5%
CR320	1N967B	1902-0964	18.0V	5%
CR319	1N967B	1902-0964	18.0V	5%
CR318	1N967B	1902-0964	18.0V	5%
CR316	1N967B	1902-0964	18.0V	5%
CR315	1N967B	1902-0964	18.0V	5%
CR314	1N967B	1902-0964	18.0V	5%
CR313	1N967B	1902-0964	18.0V	5%
CR312	1N967B	1902-0964	18.0V	5%
CR311	1N967B	1902-0964	18.0V	5%
CR310	1N967B	1902-0964	18.0V	5%
CR309	1N967B	1902-0964	18.0V	5%
CR308	1N967B	1902-0964	18.0V	5%
CR307	1N967B	1902-0964	18.0V	5%
CR306	1N967B	1902-0964	18.0V	5%
CR305	1N967B	1902-0964	18.0V	5%
CR304	1N967B	1902-0964	18.0V	5%
CR303	1N967B	1902-0964	18.0V	5%
CR302	1N967B	1902-0964	18.0V	5%
CR301	1N967B	1902-0964	18.0V	5%
CR300	1N967B	1902-0964	18.0V	5%

CAPACITORS

C604	0180-1746	15uF	10%
C603	0180-1746	15uF	10%
C602	0180-1746	15uF	10%
C601	0180-1746	15uF	10%
C6	0160-4822	1000pF	5%
C5	0160-4822	1000pF	5%
C408	0160-4844	1uF	+80-20
C407	0160-4844	1uF	+80-20
C406	0160-4835	0.1uF	10%
C405	0160-4835	0.1uF	10%
C404	0160-4801	100pF	5%
C403	0160-4822	1000pF	5%
C402	0160-4801	100pF	5%
C401	0160-4822	1000pF	5%
C4	0160-4822	1000pF	5%
C308	0160-4844	1uF	+80-20
C307	0160-4844	1uF	+80-20
C306	0160-4835	0.1uF	10%
C305	0160-4835	0.1uF	10%
C304	0160-4801	100pF	5%
C303	0160-4822	1000pF	5%
C302	0160-4801	100pF	5%
C301	0160-4822	1000pF	5%
C3	0160-4822	1000pF	5%
C215	0160-4835	0.1uF	10%
C214	0160-4835	0.1uF	10%
C213	0160-4835	0.1uF	10%
C212	0160-4835	0.1uF	10%
C211	0160-4835	0.1uF	10%
C210	0160-4835	0.1uF	10%
C209	0160-4835	0.1uF	10%
C208	0160-4835	0.1uF	10%
C207	0160-4835	0.1uF	10%
C206	0160-4835	0.1uF	10%
C205	0160-4835	0.1uF	10%
C204	0160-4835	0.1uF	10%
C203	0160-4835	0.1uF	10%
C202	0160-4801	100pF	5%
C201	0160-4835	0.1uF	10%
C2	0160-4822	1000pF	5%
C1	0160-4822	1000pF	5%

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Symbols

[ROUTe:], 4–15
[ROUTe:]CLOSe, 4–15
[ROUTe:]CLOSe?, 4–16
[ROUTe:]OPEN, 4–16
[ROUTe:]OPEN?, 4–17
[ROUTe:]SCAN, 4–17
*RCL, 3–15
*SAV, 3–14

A

ABORt, 4–5
Addresses, switch channel, 1–4
ARM, 4–6
ARM:COUNt, 4–6
ARM:COUNt?, 4–7

C

card numbers, switch, 1–4
CAUTIONS, 2–1
command
  abbreviated, 4–2
  implied, 4–2
  parameters, 4–3
  separator, 4–2
Command Reference, 4–1
  [ROUTe:], 4–15
  [ROUTe:]CLOSe, 4–15
  [ROUTe:]CLOSe?, 4–16
  [ROUTe:]OPEN, 4–16
  [ROUTe:]OPEN?, 4–17
  [ROUTe:]SCAN, 4–17
  ABORt, 4–5
  ARM, 4–6
  ARM:COUNt, 4–6
  ARM:COUNt?, 4–7
  INITiate, 4–10
  INITiate:CONTinuous, 4–10
  INITiate:CONTinuous?, 4–11
  INITiate[:IMMediate], 4–11
  STATus, 4–18
  STATus:OPERation:ENABle, 4–18
  STATus:OPERation[:EVENt]?, 4–18
  SYSTem, 4–20
  SYSTem:CPON, 4–21
  SYSTem:CTYPe?, 4–21
  SYSTem:ERRor?, 4–20
  TRIGger, 4–22
  TRIGger:SOURce, 4–23
  TRIGger:SOURce?, 4–25
  TRIGger[:IMMediate], 4–22
Command Types, 4–1
Commands, IEEE 488.2, 4–26
commands, linking, 4–3
Common Command Format, 4–1
Common Commands, 4–26
connecting user inputs, 2–5

D

definition, instrument, 1–1
description, switch, 1–1
Digital Output Configuration, 2–8

E

Error Messages, C–1
Error Numbers, C–1
Example: Controlling RF Switches/Step Attenuators, 3–6
Example: Digital Output Configuration, 2–8
Example: Digital Output Configuration , 3–7
Example: General Purpose Relay Configuration, 2–7
Example: Matrix Switching , 3–8
Example: Scanning Using Trig Out Port, 3–9
Example: Voltage Switching, 3–5

F

Form C Switch, error messages, C–1
Form C Switch Commands, 3–1
Format
  common command, 4–1
  SCPI, 4–2

G

General Purpose Relay Configuration, 2–7

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

I

INITiate, 4–10
INITiate:CONTinuous, 4–10
INITiate:CONTinuous?, 4–11
INITiate[:IMMediate], 4–11
instrument definition, 1–1

O

operation
  basic, 1–1
  initial, 1–5

P

parameters
  Boolean, 4–3
  numeric, 4–3
  optional, 4–3
programming, basic, 1–3

R

Recalling States, 3–14
root command, 4–2

S

Scanning Channels , 3–9
SCPI Command Format, 4–2
SCPI specifying commands, 1–4
STATus, 4–18
STATus:OPERation:ENABle, 4–18
STATus:OPERation[:EVENt]?, 4–18
subsystem, 4–2
  ABORt, 4–5
  ARM, 4–6
subystem
  [ROUTe:], 4–15
  [ROUTe:]CLOSe, 4–15
  [ROUTe:]CLOSe?, 4–16
  [ROUTe:]OPEN, 4–16
  [ROUTe:]OPEN?, 4–17
  [ROUTe:]SCAN, 4–17
  ARM:COUNt, 4–6
  ARM:COUNt?, 4–7
  INITiate, 4–10
  INITiate:CONTinuous, 4–10
  INITiate:CONTinuous?, 4–11
  INITiate[:IMMediate], 4–11
  STATus, 4–18
  STATus:OPERation:ENABle, 4–18
  STATus:OPERation[:EVENt]?, 4–18
  SYSTem, 4–20
  SYSTem:CPON, 4–21
  SYSTem:CTYPe?, 4–21
  SYSTem:ERRor?, 4–20
  TRIGger, 4–22
  TRIGger:SOURce, 4–23
  TRIGger:SOURce?, 4–25
  TRIGger[:IMMediate], 4–22
Switching Channels, 3–4
SYSTem, 4–20
SYSTem:CPON, 4–21
SYSTem:CTYPe?, 4–21
SYSTem:ERRor?, 4–20

T

TRIGger, 4–22
TRIGger:SOURce, 4–23
TRIGger:SOURce?, 4–25
TRIGger[:IMMediate], 4–22
Typical Switch Configurations, 2–7

U

Using the Form C Relay, 3–1
Using the Scan Complete Bit, 3–13

W

WARNINGS, 2–1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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