

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 *Microsemi / Symmetricom / Datum bc350VXI* at our website: ***Click HERE***

HP 75000 SERIES B

BANC-350
IRIGB Software Drivers

User's Manual

HEWLETT PACKARD

Copyright© Hewlett-Packard Company, 1992

Manual Part Number: E1300-90012
Microfiche Part Number: E1300-99012

Printed: July 1992 Edition 1
Printed in U.S.A.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.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 Duration and conditions of warranty for this product may be superceded when the product is integrated into (becomes a part of) other HP products.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

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.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 E1300-90012) ..... July 1992

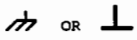
Safety Symbols



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 Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Table of Contents

1. Getting Started

Chapter Contents . . . 1-1
Preparation for Use . . . 1-1
Installing the Module . . . 1-1
Device Drivers . . . 1-1
Logical Address . . . 1-2
Instrument and Programming Languages . . . 1-2
Example Programs . . . 1-2
Sample Programs . . . 1-3
SCPI Command Structure . . . 1-3
Compiled SCPI Language Program . . . 1-4
C Language Program . . . 1-7
HP BASIC/IBASIC Language Program . . . 1-12
QuickBASIC Language Program . . . 1-15

2. Using the BANC-350 IRIGB Module

Chapter Contents . . . 2-1
Module Description . . . 2-1
Examples . . . 2-1
Reading the Time Registers (rd_time.scpi) . . . 2-2
Comments . . . 2-3
Triggering the Module and Packed Outputs (trg_time.scpi) . . . 2-5
Comments . . . 2-6

3. Command Reference

About this Chapter . . . 3-1
Command Types . . . 3-1
Parameter and Return Value Types and Sizes . . . 3-4
Linking Commands . . . 3-4
ABORt . . . 3-5
DIAGnostic . . . 3-6
DIAGnostic:SMODe . . . 3-6
DIAGnostic:SMODe? . . . 3-7
DIAGnostic:SMAJor . . . 3-7
DIAGnostic:SMAJor? . . . 3-8
DIAGnostic:SMINor . . . 3-8
DIAGnostic:SMINor? . . . 3-9
FETCh? . . . 3-10
FETCh[:TIME]? . . . 3-10
FORMat . . . 3-11
FORMat[:DATA] . . . 3-11
FORMat[:DATA]? . . . 3-12
INITiate . . . 3-13
INITiate[:IMMediate] . . . 3-13
MEASure . . . 3-14
MEASure:TIME? . . . 3-14
READ? . . . 3-15

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

READ[:TIME]? 3-15
STATus 3-16
STATus:OPERation:CONDition? 3-16
STATus:OPERation:ENABle 3-17
STATus:OPERation:ENABle? 3-17
STATus:OPERation[:EVENt]? 3-17
STATus:QUEStionable:CONDition? 3-18
STATus:QUEStionable:ENABle 3-18
STATus:OPERation:ENABle? 3-19
STATus:QUEStionable[:EVENt]? 3-19
SYSTem 3-20
SYSTem:ERRor? 3-20
SYSTem:IRIG:TIME 3-21
SYSTem:VERSion? 3-21
TRIGger 3-22
TRIGger[:IMMediate] 3-22
TRIGger:SLOPe 3-22
TRIGger:SLOPe? 3-22
TRIGger:SOURce 3-23
TRIGger:SOURce? 3-24
BANC-350 IRIGB Module Command Quick Reference 3-25
SCPI Conformance Information 3-29

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Chapter 1
Getting Started

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Chapter 1
Getting Started

Chapter Contents

This chapter shows how to begin using the HP BANC-350 IRIGB Module. The BANC-350 IRIGB Module consist of a Bancocm IRIGB B-size VXIbus module with the Hewlett-Packard supplied Compiled SCPI (C-SCPI) and SCPI Drivers (see "Instrument and Programming Languages" on page 1-2). The sections are as follows:

- Preparation for Use 1-1
  - Installing the Module 1-1
  - Device Drivers 1-1
  - Logical Address 1-2
- Instrument and Programming Languages 1-2
  - Example Programs 1-2
  - Sample Programs 1-3
  - SCPI Command Structure 1-3
  - Compiled SCPI Language Program. 1-4
  - C Language Program 1-7
  - HP BASIC/IBASIC Language Program 1-12
  - QuickBASIC Language Program. 1-15

Preparation for Use

This section shows how to prepare the module for use. Included is information on installation, downloading device drivers, and addressing of the module.

Installing the Module

The module can be installed in any mainframe slot, except slot 0. For installation instructions, see the HP B-size or C-size " Installation and Getting Started Guide". For additional instructions, refer to the Bancom IRIGB hardware manual.

Device Drivers

The module can be controlled by an HP-UX type Embedded Computer (e.g., HP V/382). The module can also be controlled by the HP E1405 C-size Command Module, or the HP E1300/E1301 B-size Mainframe connected to an external HP-IB1 computer (e.g., HP Series 200/300 computers).

1 HP-IB is Hewlett-Packard's implementation of the IEEE Standard 488.1-1978

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

HP-UX Drivers

These drivers are the Compiled SCPI (C-SCPI) drivers for the module (see "Instrument and Programming Languages" below for information on C-SCPI). The drivers are called "IRIGB" and are located on the same storage medium as the Example Programs (see "Example Programs" below).

The drivers operate on computers with HP-UX version 8.0 and above. For the C-SCPI driver installation instructions, see the "HP Compiled SCPI Getting Started Guide", HP part number: E1570-90002.

Command Module/Mainframe Device Drivers

These drivers are called "IRIGB" and are located on the Driver Disk formatted for DOS or the Driver Disk formatted for LIF (HP BASIC/IBASIC). Depending on the instrument option ordered, the drivers can be downloaded from an externally connected MS-DOS® computer, an externally connected HP BASIC computer, or using IBASIC in an HP E1405 Command Module or HP E1300/E1301 Mainframe.

The drivers operate in a Command Module/Mainframe version A.06.00 and above. For driver installation instructions, refer to the "Downloading Device Drivers" Installation Note, HP part number E1400-90021. The "HP C-size Installation Guide" and the "HP E1405 Command Module User's Guide" may also contain the applicable information.

Logical Address

The module is shipped with a logical address setting of 200. Refer to the IRIGB hardware manual to change the setting, if needed.

Instrument and Programming Languages

The BANC-350 Module uses SCPI (Standard Commands for Programmable Instruments) as the instrument programming language. Chapter 3, "Command Reference" documents the applicable SCPI commands. Chapter 2, "Using the BANC-350 Module" shows how to program the card using the commands.

Example Programs

Shipped with the module is a storage medium that contains the example programs shown in Chapter 2. The medium type depends on the instrument option ordered (tape, or LIF or DOS formatted disk).

The example programs use Compiled SCPI (C-SCPI) as a means to execute the SCPI commands within a C program in an HP-UX based system. C-SCPI replaces the SCPI commands with driver calls that are compiled with the C program. This allows for faster executable files. For more information on C-SCPI, refer to the C-SCPI User manuals.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Sample Programs

Included with the example program, there are also listed in this chapter sample programs consisting of a C program (for a PC computer), and HP BASIC/IBASIC program, and a QuickBASIC program.

SCPI Command Structure

A SCPI command structure is a hierarchal 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:

```
SYSTem
    :ERRor?
    :IRIG
        :TIME <days>,<hours>,<minutes>,<seconds>
    :VERSion?
```

SYSTem is the root keyword of the command, :ERRor?, :IRIG, and :VERSion? are the second level keyword, :TIME is the third level keyword, and so on. A colon (:) always separates a command keyword from a lower level keyword as shown below:

```
SYSTem:IRIG:TIME
```

A semicolon (;) separates two commands within the same subsystem. For example, this command message:

```
STATus:OPERation:ENABle 1;:CONDition?
```

Is the same as these two commands:

```
STATus:OPERation:ENABle 1
STATus:OPERation:CONDition?
```

SCPI Command Format

The typical format of commands listed in the command reference and throughout this manual is:

```
FORMat[:DATA] <type>,[<length>]
```

Command headers enclosed in brackets are optional. Required are the UPPER CASE letters in the header; you can omit the lower case letters. **To aid in learning the command sets, all command headers are included in the example programs; however, the command headers are abbreviated.** In the example programs, the previous command with an *type* parameter of ASCii for [:DATA] appears as:

```
FORM:DATA ASC
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Compiled SCPI Language Program

The following identifies the system on which the C-SCPI example programs for the BANC-350 IRIGB Module were developed. Also listed is an example program that is used in this manual to show a typical program format.

System Configuration

The C-SCPI example programs were developed on the following system:

Mainframe: HP 75000 Series C

Slot 0/Resource Manager: HP V/382 Computer

BANC-350 IRIGB Module Logical Address: 200

Instrument Language: SCPI

Compiler Used

The C-SCPI programs were compiled using the ANSI-C compiler of the HP-UX version 8.0 operating system with the HP V/382 computer. Also used were the libraries and drivers supplied by C-SCPI and the BANC-350 IRIGB Module.

For more information on how to compile with C-SCPI, see the Compiled SCPI User Manuals.

Typical C-SCPI Example Program

A typical C-SCPI example program is shown below. This program sets the module's internal timer and then reads the initial time. After a time determined by you, the program reads the new time. The program also has an error checking function (i.e., check_error(func_tion)). This function is unnecessary, but may be useful for debugging errors the pre-processor does not catch.

```
/*****************************************************************************/
/* rd_time.scpi - This program resets the module, reads the time, waits for  */
/* an input, and then reads the new time.                                    */
/*****************************************************************************/

#include <stdio.h>
#include <cscpi.h>

      /* Defines the module's logical address */
#define LADD "200"

      /* Specifies the module as a register device */
INST_DECL(irig, "IRIG", REGISTER);
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
/*****************************************************************************/
main()
{
        /* Start the register-based operating system */
    INST_STARTUP();

        /* Open the I/O path to the module */
    INST_OPEN(irig, "vxi," LADD);

        /* Check if the path is good */
    if ( !irig )
    {
        printf( "INST_OPEN failed (ladd = %d). %d\n", LADD, cscpi_open_error);
        exit(1);
    }

    rst_cls();      /* Reset and clear the module */

    select_time(); /* Run function to set and read the time */

    exit(0);
}
/*****************************************************************************/
select_time()
{
    int16   day = 1,    /* Set the day */
            hour = 0,   /* Set the hour */
            min = 0,    /* Set the minutes */
            sec = 0,    /* Set the seconds */
            rd_day[2],
            rd_hour[2],
            rd_min[2],
            i;
    float64 calc_time[2],
            rd_sec[2];
    char    data_form[10] = "REAL",     /* Select the return data type */
            *show_day[2] = {"  Initial Day is:","      New Day is:"},
            *show_hour[2] = {"  Initial Hour is:","     New Hour is:"},
            *show_min[2] = {"Initial Minute is:","   New Minute is:"},
            *show_sec[2] = {"Initial Second is:","   New Second is:"};

        /* Select the data format */
    INST_SEND(irig, "FORM:DATA %s\n", data_form);
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
        /* Set the day and time */
    INST_SEND(irig, "SYST:IRIG:TIME %d, %d, %d, %d", day, hour, min, sec);

        /* Call function to check for errors */
    check_error("select_time(initial)");

        /* Read the initial day and time */
    INST_QUERY(irig, "READ?", "", &calc_time[0]);

        /* Wait for an action */
    printf("\n\tPress 'Return' to read the new time!\n");
    getchar();

        /* Read the new time */
    INST_QUERY(irig, "READ?", "", &calc_time[1]);

        /* Call function to check for errors */
    check_error("select_time(new)");

        /* Get day and time of both the initial and new time value */
    for (i = 0; i < 2; i ++)
    {
        rd_day[i] = (int)(calc_time[i] / 1e6);
        rd_hour[i] = ((int)calc_time[i] / 1e4) - (rd_day[i] *  1e2);
        rd_min[i] = ((int)calc_time[i] / 1e2) - (rd_day[i] * 1e4) - (rd_hour[i] * 1e2);
        rd_sec[i] = (calc_time[i] - (rd_day[i] * 1e6) - (rd_hour[i] * 1e4) - (rd_min[i] *
1e2));

        printf("\n %s %d", show_day[i], rd_day[i]);
        printf("\n %s %d", show_hour[i], rd_hour[i]);
        printf("\n %s %d", show_min[i], rd_min[i]);
        printf("\n %s %.7f", show_sec[i], rd_sec[i]);
        printf("\n");
    }

}
/*****************************************************************************************/
rst_cls()
    /* This function resets and clears the module */
{
    int16    rst_cls;

        /* Reset the module to its turn-on state; wait to complete */
    INST_SEND(irig, "*RST");
    INST_QUERY(irig, "*OPC?", "", &rst_cls);
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
        /* Clear the module; wait to complete */
    INST_SEND(irig, "*CLS");
    INST_QUERY(irig, "*OPC?", "", &rst_cls);
}
/*****************************************************************************/
check_error(func_tion)
char *func_tion;
{
    int16    error;
    char     err_out[256];

        /* Check for any errors */
    INST_QUERY(irig, "SYST:ERR?", "", &error, err_out);

        /* If error is found, print out the error(s) and and exit */
    if (error)
    {
        while(error)
        {
            printf("Error %d,%s in function %s\n", error, err_out, func_tion);
            INST_QUERY(irig, "SYST:ERR?", "", &error, err_out);
        }
        exit(1);
    }
}
```

C Language Program

The following C Language sample program is written for an MS-DOS® computer. The program requires the HP 82335 HP-IB Interface Card and the HP-IB Command Library for C to operate. The library functions in the program is compatible with the ANSI-C standard.

The following lists the system requirements and configuration, lists the libraries to compile the programs, and lists the program.

System Configuration

The C program can run on the following typical system:

Controller: HP Vectra 386/25 personal computer (386 processor operated at 25 MHz)

HP-IB Interface Card: HP 82335 HP-IB Interface with Command Library

Mainframe: HP 75000 Series B or C

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Slot 0/Resource Manager: HP E1300/E1301 Mainframe or HP E1405 Command Module

BANC-350 IRIGB Module Logical Address: 200

Instrument Language: SCPI

Running and Compiling the Program

To run the program, first compile and link the program to make an executable file. To compile a program, be sure to:

- Link the appropriate HP-IB C library (located on the HP-IB Command Library disk that came with the HP-IB Interface Card). Use the following libraries:
  - **Microsoft® C and QuickC©:** clhpib.lib
  - **Turbo C© and C++©:** tchhpib.lib
- The necessary paths have been added to the AUTOEXEC.BAT file for the compilers to find the library and header files (see the appropriate C Language manual to set the proper paths).

HP-IB Addressing

To run the program in a B or C size mainframe, the module must have a unique HP-IB address. This address consists of a Select Code, Primary Address, and Secondary Address.

Select Code (7): The default select code of the HP-IB card is 7. Use this code to access the devices in the mainframe.

Primary HP-IB Address (9): The default primary address of the HP E1300/E1301 Mainframe and the HP E1405 Command Module is 9. Valid addresses are from 0 to 30.

Secondary HP-IB Address (25): The default address of the BANC-350 IRIGB Module is 25. This address is derived from the module's logical address divided by 8 (.e.g., 200 / 8 = 25).

Typical C Program

A typical C program is shown below. This program sets the module's internal timer and then reads the initial time. After a time determined by you, the program reads the new time. The program also has an error checking function (i.e., check_error(func_tion)). This function is useful for debugging.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
/*****************************************************************************/
/* RD_TIME.C - This program resets the module, reads the time, waits for   */
/* an input, and then reads the new time.                                    */
/*****************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <cfunc.h>        /* This file is from the HP-IB Command Library Disk */

#define ADDR 70925L    /* Assigns the module's HP-IB address */

      /* Functions */
void select_time(void);
void cmd_exe(char *commands);
void check_error(char *func_tion);
void rst_clr(void);

/*****************************************************************************/
void main(void)
{
      rst_clr();/* Reset and clear the module */

      select_time(); /* Run function to set and read the time */

      exit(0);
}
/*****************************************************************************/
void select_time(void)
{
      int       rd_day[2],
                rd_hour[2],
                rd_min[2],
                length,
                i;
      double    calc_time[2],
                rd_sec[2];
      char      rd_time[18],
                *show_day[2] = {"  Initial Day is:","      New Day is:"},
                *show_hour[2] = {"  Initial Hour is:","     New Hour is:"},
                *show_min[2] = {"Initial Minute is:","   New Minute is:"},
                *show_sec[2] = {"Initial Second is:","   New Second is:"},
                rd;

        /* Select the data format */
      cmd_exe("FORM:DATA ASC");
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
    /* Set the day and time */
    cmd_exe("SYST:IRIG:TIME 1, 0, 0, 0");

    /* Call function to check for errors */
    check_error("select_time(initial)");

    /* Read the initial day and time; change to real number */
    length = 18;
    cmd_exe("READ?");
    IOENTERS(ADDR, rd_time, &length);
    calc_time[0] = atof(rd_time);

    /* Wait for an action */
    printf("\n\tPress 'Enter' to read the new time!\n");
    scanf("%c", &rd);
    fflush(stdin);

    /* Read the new time and day; change to real number */
    length = 18;
    cmd_exe("READ?");
    IOENTERS(ADDR, rd_time, &length);
    calc_time[1] = atof(rd_time);

    /* Call function to check for errors */
    check_error("select_time(new)");

    /* Get day and time of both the initial and new time value */
    for (i = 0; i < 2; i ++)
    {
      rd_day[i] = (int)(calc_time[i] / 1e6);
      rd_hour[i] = (int)((calc_time[i] / 1e4) - (rd_day[i] *  1e2));
      rd_min[i] = ((int)(calc_time[i] / 1e2) - (rd_day[i] * 1e4) - (rd_hour[i] * 1e2));
      rd_sec[i] = (calc_time[i] - (rd_day[i] * 1e6) - (rd_hour[i] * 1e4) - (rd_min[i] *
1e2));

      printf("\n %s %d", show_day[i], rd_day[i]);
      printf("\n %s %d", show_hour[i], rd_hour[i]);
      printf("\n %s %d", show_min[i], rd_min[i]);
      printf("\n %s %.7f", show_sec[i], rd_sec[i]);
      printf("\n");
    }

}
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
/*****************************************************************************/
void rst_clr(void)
{
        float      into;

           /* Executes the commands that resets the module and clears its error
register */
        IOOUTPUTS(ADDR, "*RST;*OPC?", 10);
        IOENTER(ADDR, &into);
        IOOUTPUTS(ADDR, "*CLS;*OPC?", 10);
        IOENTER(ADDR, &into);
}
/*****************************************************************************/
void cmd_exe(char *commands)
{
        IOOUTPUTS(ADDR, commands, strlen(commands));
}
/*****************************************************************************/
void check_error(char *func_tion)
{
        char    into[161];
        int     length = 160;

        IOOUTPUTS(ADDR, "SYST:ERR?", 9);          /* Query error register */
        IOENTERS(ADDR, into, &length);            /* Enter error message */

        if (atoi(into) != 0)                      /* Determine if error is present */
                                                  /* If errors present, print and exit */
        {
           while (atoi(into) != 0)
           {
                printf("Error %s in function %s\n\n", into, func_tion);
                length = 160;
                IOOUTPUTS(ADDR, "SYST:ERR?", 9);
                IOENTERS(ADDR, into, &length);
           }

           exit(1);
        }
}
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

HP BASIC/IBASIC Language Program

The following HP BASIC sample program is written for an HP Series 200/300 computer. However, the program is easily converted using HP IBASIC in an HP E1300/E1301 Mainframe or HP E1405 Command Module (Options 020 - 022). To use with IBASIC, change the HP-IB Select Code to 8 instead of 7 (see "HP-IB Addressing" below).

The following lists the system requirements and configuration, and lists the program.

System Configuration

The HP BASIC program can run on the following typical system:

Controller: HP 9000 Series 200 or 300

Mainframe: HP 75000 Series B or C

Slot 0/Resource Manager: HP E1300/E1301 Mainframe or HP E1405 Command Module

BANC-350 IRIGB Module Logical Address: 200

Instrument Language: SCPI

HP-IB Addressing

To run the program in a B or C size mainframe, the module must have a unique HP-IB address. This address consists of a Select Code, Primary Address, and Secondary Address.

Select Code (7): The default select code of the computer is 7. Use this code to access the devices in the mainframe. To change the the program to run in IBASIC, use select code "8".

Primary HP-IB Address (9): The default primary address of the HP E1300/E1301 Mainframe and the HP E1405 Command Module is 9. Valid addresses are from 0 to 30.

Secondary HP-IB Address (25): The default address of the BANC-350 IRIGB Module is 25. This address is derived from the module's logical address divided by 8 (.e.g., 200 / 8 = 25).

Typical HP BASIC Program

A typical HP BASIC program is shown below. This program sets the module's internal timer and then reads the initial time. After a time determined by you, the program reads the new time. The program also has an error checking routine (i.e., Errmsg). This routine may be useful for debugging errors.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
10 ! RD_TIME - This program resets the module, reads the time, waits for
20 ! an input, and then reads the new time.
30 !
40 ! Assign I/O paths between the computer and the module
50 ASSIGN @Addr TO 70925
60 !
70 COM @Addr
80 !
90 ! Set up error checking
100 ON INTR 7 CALL Errmsg
110 ENABLE INTR 7;2
120 !
130 ! Clear all bits in the standard event register and enable for error
140 ! checking by setting the error bits in the register
150 OUTPUT @Addr;"*CLS"
160 OUTPUT @Addr;"*SRE 32"
170 OUTPUT @Addr;"*ESE 60"
180 !
190 ! Call subprogram to read time
200 CALL Select_time
210 WAIT .1    ! allow interrupt to be serviced
220 OFF INTR 7
230 END
240 !
250 SUB Select_time
260  Select_time:  ! Subprogram that sets and reads time
270   !
280   COM @Addr
290   !
300   DIM Rd_time$(2)[17],Show_day$(2)[20],Show_hour$(2)[20],Show_min$(2)[20],
Show_sec$(2)[20]
310   !
320   Show_day$(1)="   Initial Day is: "
330   Show_day$(2)="       New Day is: "
340   Show_hour$(1)="  Initial Hour is: "  !
350   Show_hour$(2)="      New Hour is: "  !
360   Show_min$(1)="Initial Minute is: "  !
370   Show_min$(2)="    New Minute is: "  !
380   Show_sec$(1)="Initial Second is: "  !
390   Show_sec$(2)="    New Second is: "  !
400   !
410   OUTPUT @Addr;"FORM:DATA ASC"  ! Select the data format
420   OUTPUT @Addr;"SYST:IRIG:TIME 1, 0, 0, 0"  ! Set the day and time
430   !
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
440  OUTPUT @Addr;"READ?"  ! Read the initial day; change to real number
450  ENTER @Addr;Rd_time$(1)
460  !
470  ! Wait for an action
480  INPUT "Press 'Return' to read the new time",A$
490  !
500  OUTPUT @Addr;"READ?"  ! Read the new time and day; change to real number
510  ENTER @Addr;Rd_time$(2)
520  !
530  ! Get day and time of both the initial and new time values
540  !
550  FOR I=1 TO 2
560     Calc_val=(VAL(Rd_time$(I)))-FRACT(VAL(Rd_time$(I)))
570     Rd_day(I)=(Calc_val/1.E+6)-(FRACT(Calc_val/1.E+6))
580     Rd_hour(I)=(Calc_val/10000.)-(Rd_day(I)*100)-(FRACT(Calc_val/10000.))
590     Rd_min(I)=(Calc_val/100.)-(Rd_day(I)*10000)-(Rd_hour(I)*100)-
(FRACT(Calc_val/100.))
600     Rd_sec(I)=(FRACT(VAL(Rd_time$(I))/100)*100)
610     !
620     PRINT Show_day$(I);Rd_day(I)
630     PRINT Show_hour$(I);Rd_hour(I)
640     PRINT Show_min$(I);Rd_min(I)
650     PRINT USING "K,DDD.DDDDDDD";Show_sec$(I),Rd_sec(I)
660     PRINT
670  NEXT I
680 SUBEND
690 !
700 SUB Errmsg
710  Errmsg:  ! Subprogram that displayes errors
720  !
730  COM @Addr
740  DIM Messages$[256]
750  ! Read status byte and clear service request bit
760  B=SPOLL(@Addr)
770  OUTPUT @Addr
780  REPEAT
790     OUTPUT @Addr;"SYST:ERR?"  ! Read error queue
800     ENTER @Addr;Code,Messages$
810     PRINT Code,Messages$
820  UNTIL Code=0
830  STOP
840 SUBEND
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

QuickBASIC Language Program

The following QuickBASIC Language sample program is written for an MS-DOS® computer. The program requires the HP 82335 HP-IB Interface Card and the HP-IB Command Library for QuickBASIC to operate.

The following lists the system requirements and configuration, lists the libraries to compile the programs, and lists the program.

System Configuration

The QuickBASIC program can run on the following typical system:

Controller:	HP Vectra 386/25 personal computer (386 processor operated at 25 MHz)
HP-IB Interface Card:	HP 82335 HP-IB Interface with Command Library
Mainframe:	HP 75000 Series B or C
Slot 0/Resource Manager:	HP E1300/E1301 Mainframe or HP E1405 Command Module
BANC-350 IRIGB Module Logical Address:	200
Instrument Language:	SCPI

Running and Compiling the Program

To run the program, first compile and link the program to make an executable file. You can either compile and link the program in the QuickBASIC environment or the command line. Be sure to link the "qbhpib" library file to the program. Also be sure to include the "QBSETUP" include file in the program, as shown.

HP-IB Addressing

To run the program in a B or C size mainframe, the module must have a unique HP-IB address. This address consists of a Select Code, Primary Address, and Secondary Address.

Select Code (7): The default select code of the HP-IB card is 7. Use this code to access the devices in the mainframe.

Primary HP-IB Address (9): The default primary address of the HP E1300/E1301 Mainframe and the HP E1405 Command Module is 9. Valid addresses are from 0 to 30.

Secondary HP-IB Address (25): The default address of the BANC-350 IRIGB Module is 25. This address is derived from the module's logical address divided by 8 (.e.g., 200 / 8 = 25).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Typical QuickBASIC Program

A typical QuickBASIC program is shown below. This program sets the module's internal timer and then reads the initial time. After a time determined by the you, the program reads the new time. The program also has an error checking routine (i.e., CheckError (FuncType$)). This routine may be useful for debugging errors.

```
" RD_TIME.BAS - This program resets the module, reads the time, waits for
' an input, and then reads the new time.

DECLARE SUB SelectTime ()
DECLARE SUB CmdExe (Commands$)
DECLARE SUB CheckError (FuncType$)
DECLARE SUB RstClr ()

COMMON SHARED ADDR&, ISC&

REM $INCLUDE: 'QBSETUP'

ISC& = 7       'Assigns the HP-IB select code
ADDR& = 70925  'Assign an I/O path between the computer and the module

CALL RstClr  ' Reset the module
CALL SelectTime  ' Run routine to read time

END

SUB CheckError (FuncType$)

Max.Length% = 160: Actual.Length% = 0
Info$ = SPACE$(Max.Length%)

CALL IOOUTPUTS(ADDR&, "SYST:ERR?", 9)  ' Query error register
CALL IOENTERS(ADDR&, Info$, Max.Length%, Actual.Length%)

IF VAL(Info$) <> 0 THEN
      CLS
      WHILE VAL(Info$) <> 0
         Info$ = LEFT$(Info$, Actual.Length%)
         PRINT : PRINT Info$; "  in function: "; FuncType$
         Info$ = SPACE$(Max.Length%)
         CALL IOOUTPUTS(ADDR&, "SYST:ERR?", 9)
         CALL IOENTERS(ADDR&, Info$, Max.Length%, Actual.Length%)
      WEND
      END
END IF

END SUB
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SUB CmdExe (Commands$)

CALL IOOUTPUTS(ADDR&, Commands$, LEN(Commands$))

END SUB

SUB RstClr

CALL IOOUTPUTS(ADDR&, "*RST;*OPC?", 10)
CALL IOENTER(ADDR&, Into!)

CALL IOOUTPUTS(ADDR&, "*CLS;*OPC?", 10)
CALL IOENTER(ADDR&, Into!)

END SUB

SUB SelectTime

DIM CalcTime$(1 TO 2)
DIM RdDay%(1 TO 2)
DIM RdHour%(1 TO 2)
DIM RdMin%(1 TO 2)
DIM RdSEc#(1 TO 2)
DIM ShowDay$(1 TO 2)
DIM ShowHour$(1 TO 2)
DIM ShowMin$(1 TO 2)
DIM ShowSec$(1 TO 2)

ShowDay$(1) = "  Initial Day is: "
ShowDay$(2) = "      New Day is: "
ShowHour$(1) = "  Initial Hour is: "
ShowHour$(2) = "     New Hour is: "
ShowMin$(1) = "Initial Minute is: "
ShowMin$(2) = "   New Minute is: "
ShowSec$(1) = "Initial Second is:  "
ShowSec$(2) = "   New Second is:  "

      ' Select the data format
CALL CmdExe("FORM:DATA ASC")

      ' Set the day and time
CALL CmdExe("SYST:IRIG:TIME 1, 0, 0, 0")

      ' Call routine to check for errors
CALL CheckError("SelectTime(initial)")
```

Continued on Next Page

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
        ' Read the initial day and time
Max.Length% = 18: Actual.Length% = 0
CalcTime$(1) = SPACE$(Max.Length%)

CALL CmdExe("READ?")
CALL IOENTERS(ADDR&, CalcTime$(1), Max.Length%, Actual.Length%)

        ' Wait for an action
PRINT "Press 'Enter' to read the new time"
I$ = INPUT$(1)

        ' Read the new time
Max.Length% = 18: Actual.Length% = 0
CalcTime$(2) = SPACE$(Max.Length%)

CALL CmdExe("READ?")
CALL IOENTERS(ADDR&, CalcTime$(2), Max.Length%, Actual.Length%)

        ' Call routine to check for errors
CALL CheckError("SelectTime(new)")

PRINT

        ' Get day and time of both the initial and new time value
FOR I = 1 TO 2
        RdDay%(I) = (VAL(CalcTime$(I)) / 1000000!)
        RdHour%(I) = (VAL(CalcTime$(I)) / 10000!) - (RdDay%(I) * 100!)
        RdMin%(I) = (VAL(CalcTime$(I)) / 100!) - (RdDay%(I) * 10000!) - (RdHour%(I) * 100!)
        RdSEc#(I) = (VAL(CalcTime$(I)) - (RdDay%(I) * 1000000!) - (RdHour%(I) * 10000!) -
(RdMin%(I) * 100!))
        PRINT ShowDay$(I); RdDay%(I)
        PRINT ShowHour$(I); RdHour%(I)
        PRINT ShowMin$(I); RdMin%(I)
        PRINT USING "&##.#######"; ShowSec$(I); RdSEc#(I)
        PRINT
NEXT I

END SUB
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Chapter 2
Using the BANC-350 Module

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Chapter 2
Using the BANC-350 IRIGB Module

Chapter Contents

This chapter shows how to use the BANC-350 IRIGB Module using the SCPI instrumentation language. The sections are as follows:

- Module Description 2-1
- Examples Programs 2-1
- Reading the Time Registers 2-2
- Triggering the Module and Packed Outputs 2-5

Module Description

The BANC-350 IRIGB Module can use two different time code generators to generate time stamps: an external IRIGB signal or its own internal time code generator. The module automatically uses the external IRIGB signal, if present. If the module cannot detect an external IRIGB signal, it automatically switches over to the internal time code generator.

The internal time code generator provides a backup time source, if the external IRIGB source connection fails. The module always synchronizes its internal time code generator to the external IRIGB source, to keep track of the correct time in case of an external source failure. When the card operates from its internal time code generator and an the external IRIGB signal connects to the module, the module automatically switches back to the external signal and re-synchronize its internal time code generator.

The module also allows an external trigger connection to generate time measurements whenever it receives a trigger.

Examples

This chapter explains module operation using examples that only show the execution of the SCPI commands. Computer program language dependent commands are not listed.

However, each example has corresponding Compiled SCPI (C-SCPI) programs written in C for the HP-UX version 8.0 and above operating system. The example program names are shown in parenthesis in the example program heading. For example, the C-SCPI program name for the "Reading the Time Registers (rd_time.scpi)" example is "rd_time.scpi".

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Reading the Time Registers (rd_time.scpi)

This program sets the module's internal time code generator to a specified value and then reads the time. After a time period determined by you, the program reads the new time. This program uses the internal time code generator to generate the time stamps. The commands are:

1. **Reset the Module** -
   *RST

   The *RST command sets the module to a defined state.

2. **Select the Data Format** -
   FORMat:DATA REAL

   This command selects the format in which the time readings are output (see "Comments" below).

3. **Set the Initial Day and Time Values (Optional)** -
   SYSTem:IRIG:TIME 1, 0, 0, 0

   This command sets the day to 1, and the hours, minutes, and seconds to 0 (see "Comments" below).

4. **Read the Time Registers for the Initial Time Value** -
   READ?

   This command initiates a time measurement and places the value into the output buffer for the computer to read. Instead of the READ? command, you can also use the MEASure:TIME? command to make the time measurement.

5. **Wait for a New Measurement** -

   Execute the next command for the new time measurement.

6. **Read the Time Registers for the Initial Time Value** -
   READ?

   This command initiates a time measurement and places the value into the output buffer for the computer to read.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Comments

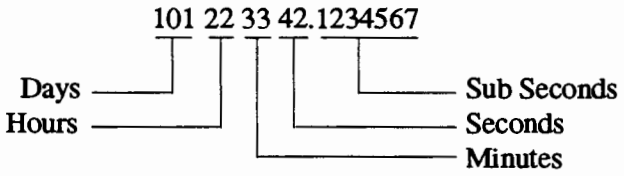
The following comments explains the module's different data formats.

Data Formats

The module can be set to output time measurements in the following three formats: ASCii, REAL, and PACKed. They are explained as follows:

ASCii

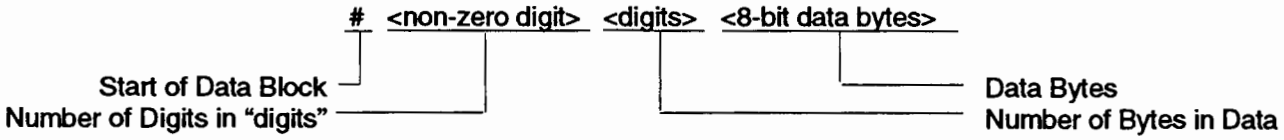
This format outputs an ASCII string containing the time measurements. The day can have a maximum of 3 characters, the hours and minutes are 2 characters each, the seconds have 2 characters, and the sub seconds have 8 characters. The ASCII string has a total of 17 characters (includes the decimal point between the seconds and sub seconds). For example, a reading of 101223342.1234567 is where:



and shows a reading of 101 days, 22 hours, 33 minutes, 42 seconds, and 1234567 sub seconds.

REAL

This format outputs a 64-bit (8 byte) real or floating point number that is output as Definite Length Arbitrary Block Data. For example, the Real number is output as:



where:

- "#" shows that the data to be sent is in an arbitrary block format.
- "<non-zero digit>" is a single digit number that shows the number of digits contained in <digits>. For the BANC-350 module, this vlaue is always "1".
- "<digits>" shows the number of data bytes to be sent. For the BANC-350 module, this value is always "8".
- "<8-bit data bytes>" is the time data output.

Since the the module's block data is output as a 64-bit real number and the IEEE 488.2 standard requires an 8-bit code, the module ouputs eight 8-bit data bytes.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Reading REAL Data Using C-SCPI - The following example shows how to read and print a Real number using C-SCPI.

```
double  real_result;                                    /* Select variable */
INST_SEND(irig, "FORM:DATA  REAL");                     /* Select data format */
INST_QUERY(irig, "READ?", "", &real_result);            /* Read time and get data */
printf("Time value is: %lf", real_result);              /* Print data */
```

Reading REAL Data Using C - The following example shows how to read and print a Real number using C.

```
double  readings[1];                                    /* Select array */
int   elements = 8;                                     /* Set byte size */
IOOUTPUT(70925L, "FORM:DATA  REAL", 14);                /* Select data format */
IOOUTPUT(70925L, "READ?", 5);                           /* Read the time */
IOENTERAB(70925L, readings, &elements, sizeof(double)); /* Get data */
printf("Time value is: %lf", readings[0]);              /* Print data */
```

Reading REAL Data Using HP BASIC/IBASIC - The following example shows how to read and print a Real number using HP BASIC/IBASIC.

```
20   ASSIGN@Irig TO 70925
30   REAL Time_real  ! Select variable type
40   DIM Ndig$[1],Count$[9]  ! Dimension arrays
50   OUTPUT @Irig;"FORM   REAL"  ! Select format
60   OUTPUT @Irig;"READ?"  ! Read time
70   ASSIGN @Irig;FORMAT  OFF  ! Turn format off
80   ! Enter and print readings
90   ENTER @Irig USING "#,X,K,K";Ndig$;Count$[1;VAL(Ndig$)]
100   ENTER @Irig;Time_real
110   PRINT USING "9D.7D";Time_real
```

Reading REAL Data Using QuickBASIC - The following example shows how to read and print a Real number using QuickBASIC.

```
DIM Reading#(1)   ' Dimension array
CALL IOOUTPUTS(70925, "FORM  REAL", 9)   ' Select the data format

SwapSize% = 8: Max.Byte% = 8: Act.Byte% = 0   ' Set byte sizes
CALL IOOUTPUTS(70925, "READ?", 5)   ' Read the time
CALL IOENTERAB(70925, SEG Reading#(0), Max.Byte%, Act.Byte%,
    SwapSize%)   ' Get the data
PRINT Reading#(0)   ' Print reading
```

PACKed This format outputs eight 8-bit byte blocks of data (see "Triggering the Module and Packed Outputs" on page 2-5 for information).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Triggering the Module and Packed Outputs (trg_time.scpi)

This program sets up the module for a BUS trigger source to generate a time output. It then sets the internal time code generator to a specified value. The module then receives a BUS trigger and reads the initial time. After a time determined by you, the program triggers the module again to read the new time. This program uses the internal time code generator to generate the time stamps. The program outputs the time values in the PACKed data format. The commands are:

1. **Reset the Module** -
   *RST

   The *RST command sets the module to a defined state.

2. **Select the Data Format** -
   FORMat:DATA PACK

   This command selects the PACKed data format to output the time readings (see "Comments" below).

3. **Set the Trigger Source** -
   TRIGger:SOURce BUS

   This command selects the trigger source (see "Comments" below for the different sources).

4. **Set the Trigger Slope to Positive (optional)** -
   TRIGger:SLOPe POS

   This command sets the trigger slope to positive to allow any positive going signal to generate a trigger. This command is only used for an EXTernal trigger source; it is given here for good programming practice.

5. **Set the Initial Day and Time Values (Optional)** -
   SYSTem:IRIG:TIME 1, 0, 0, 0

   This command sets the day to 1, and the hours, minutes, and seconds to 0.

6. **Place the Module into the Wait-for-Trigger State** -
   INITiate[:IMMediate]

   This command places the module in the wait for trigger state. Once the module is triggered, it reads the time and places the reading into the time registers.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

7. **Trigger the Module -**
   *TRG

   This command triggers the module and places the time measurement into the time registers.

8. **Read the Time Registers for the Initial Time Value -**
   FETCh[:TIME]?

   This command reads the time registers and places the time value into the output buffer for the computer to read. Unlike the READ? command, this command is used with the INITiate[:IMMediate] command and the applied trigger to output the time measurement.

9. **Wait for a New Measurement -**
   Execute the next commands for the new time measurement.

10. **Place the Module into the Wait-for-Trigger State -**
    INITiate[:IMMediate]

    This command places the module in the wait for trigger state.

11. **Trigger the Module -**
    *TRG

    This command triggers the module and places the time measurement into the time registers.

12. **Read the Time Registers for the New Time Value -**
    FETCh[:TIME]?

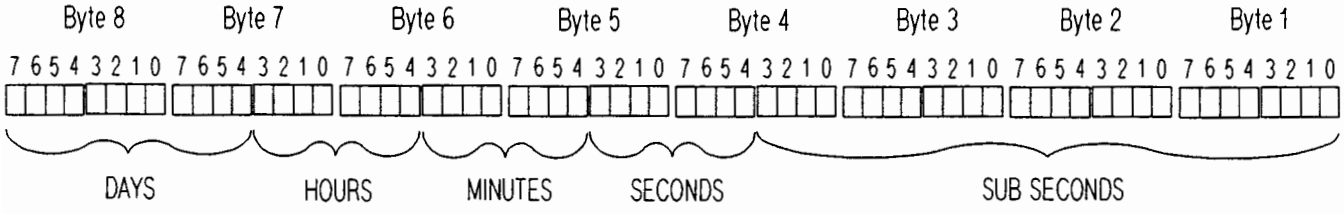
    This command reads the time registers and places the time value into the output buffer for the computer to read.

Comments

The following explain the PACKed data format and trigger parameters.

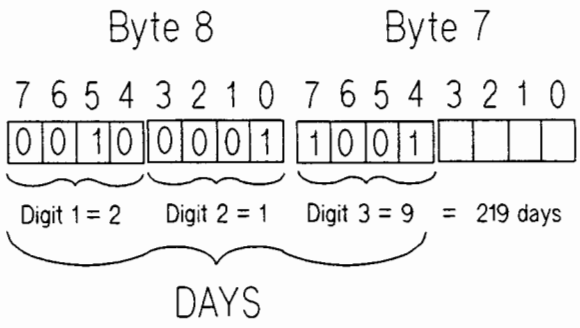
PACKed Data Format

This format outputs eight, 8-bit byte blocks of data output as Definite Length Arbitray Block Data. Each byte contains a certain part of the time value, illustrated as follows:

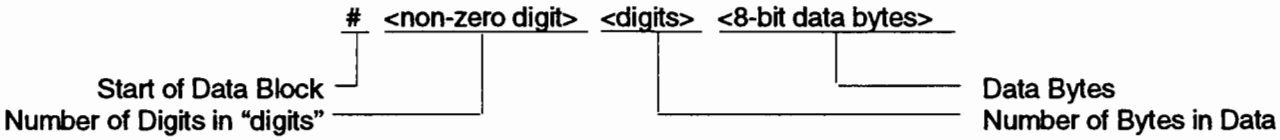


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The decimal equivalent of each half of a byte contains one digit of the day and time values. For example, if bits 4 and 7 of byte 7, and bits 1 and 5 of byte 8 are high, the value is 219 days, shown as follows:



The Definite Length Arbitrary Block Data consists of:



where:

- "#" shows that the data to be sent is in an arbitrary block format.
- "<non-zero digit>" is a single digit number that shows the number of digits contained in <digits>. For the BANC-350 module, this vlaue is always "1".
- "<digits>" shows the number of data bytes to be sent. For the BANC-350 module, this value is always "8".
- "<8-bit data bytes>" is the data sent to the module.

Reading PACKed Data Using C-SCPI - The following example shows how to read and print packed data using C-SCPI.

```
unsigned char   pack_result;                /* Select variables */
int16    i;
INST_SEND(irig, "FORM:DATA  PACK");         /* Select data format */
INST_QUERY(irig,"READ?","",pack_result);    /* Read time and get data */
printf("\nTime value is: ");
for (i=0; i<8; i++)                         /* Loop to print data */
    printf("%2x", pack_result);             /* Print data */
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Reading PACKed Data Using C - The following example shows how to read and print packed data using C.

```
long     packread[2];                                    /* Select array */
int  elements = 8;                                       /* Set byte size */
IOOUTPUT(70925L, "FORM:DATA  PACK", 14);                 /* Select data format */
IOOUTPUT(70925L, "READ?", 5);                            /* Read the time */
IOENTERAB(70925L, packread, &elements, sizeof(long));    /* Get data */
printf("Time value is: %lx%lx", packread[0], packread[1]);    /* Print data */
```

Reading PACKed Data Using HP BASIC/IBASIC - The following example shows how to read and print packed data using HP BASIC/IBASIC.

```
20   ASSIGN@Irig TO 70925
30   INTEGER Pack_time(4)  ! Select array type
40   DIM Ndig$[1],Count$[9]  ! Dimension arrays
50   OUTPUT @Irig;"FORM  PACK"  ! Select format
60   OUTPUT @Irig;"READ?"  ! Read time
70   ASSIGN @Irig;FORMAT  OFF  ! Turn format off
80   ! Enter and print readings
90   ENTER @Irig USING "#,X,K,K";Ndig$;Count$[1;VAL(Ndig$)]
100   ENTER @Irig;Pack_time(*)
110   PRINT "Time Value is: "
135   FOR I=0 TO 3
140      Pack_value$=IVAL$(Pack_time(I),16)  ! Get Hex value
150      PRINT USING "#,ZZZZ";VAL(Pack_value$)  ! Print time value
150   NEXT I
```

Reading PACKed Data Using QuickBASIC - The following example shows how to read and print packed data using QuickBASIC.

```
DIM  HexRead&(2)   ' Dimension array
CALL IOOUTPUTS(70925, "FORM  PACK", 9)   ' Select the data format
SwapSize% = 4: Max.Byte% = 8: Act.Byte% = 0   ' Set byte size
CALL IOOUTPUTS(70925, "READ?", 5)   ' Read the time
CALL IOENTERAB(70925, SEG HexRead&(0), Max.Byte%, Act.Byte%,
    SwapSize%)   ' Get the data
PRINT "Time value is: ";HEX$(HexRead&(0)); HEX$(HexRead&(1))
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Trigger Sources

The following explains the module's different trigger sources.

BUS - This trigger source it he same as applying either a VXIbus trigger using the IEEE 488.2 Common Command *TRG or a Group Execute Trigger (GET). However, you can also trigger the module using the TRIGger[:IMMediate] command when using the this trigger source.

EXTernal - This trigger source requires that a trigger signal is applied to the module's External Trigger input (pin 6 of the front panel's J1 connector, see the module's hardware manual). If using this trigger source, select the slope of the trigger signal using the TRIGger:SLOPe *<edge>* command.

HOLD - This trigger source suspends all triggering. However, you can override this command using the TRIGger[:IMMediate] command to trigger the module.

IMMediate - This source continuously triggers the module to make new measurements.

Trigger Slope

The trigger slope is only applicable for the EXTernal trigger source. The TRIGger:SLOPe commands selects the edge at which the module accepts a trigger input from an external source. The available values are:

POSitive - This is for positive going (rising edge) of the external trigger signal.

NEGative - This is for negative going (falling edge) of the external trigger signal.

EITHer - This is for either positive going (rising edge) or negative going (falling edge) of the external trigger signal.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Chapter 3
Command Reference

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Chapter 3
Command Reference

About this Chapter

This chapter describes the Standard Commands for Programmable Instruments (SCPI) command set and the IEEE 488.2 Common Commands for the BANC-350 IRIGB Module.

Command Types

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

Common Command Format

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

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

SCPI Command Format

The SCPI commands perform functions like enabling the module, receiving data, closing switches, making measurements, and querying instrument states or retrieving data. A SCPI command structure is a hierarchal 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:

```
SYSTem
    :IRIG
        :TIME  1,0,0,0
```

SYSTem is the root command, :IRIG is a second level command, :TIME is the third level commands with parameters.

Compound Command Separator

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

```
SYSTem:IRIG:TIME  1,0,0,0
```

Colons separate the root command from the second level command (SYSTem:IRIG) and the second level from the third level (IRIG:TIME).

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. For

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

better program readability, you may send the entire command. The module accepts either the abbreviated from or the entire command.

For example, if the command syntax shows SYSTem, then SYSTem and SYST are both acceptable forms. Other forms of SYSTem, such as SY or SYs generates an error. You may use upper or lower case letters. Therefore, SYSTEM, system, and SyStEm are all acceptable.

Implied Commands

Implied Commands are those which appear in square brackets ([ ]) in the command syntax. (Note that the brackets are not part of the command, and are not sent to the instrument.) Suppose you send a third level command but do not send the preceding implied command. In this case, the instrument assumes you intend to use the implied command and it responds as if you had sent it. Examine the FETCh subsystem shown below:

```
FETCh
    [:TIME]?
```

The second level command [:TIME] is an implied command. To read the time, you could send either of the following command statements:

```
FETCh:TIME?
```

or

```
FETCh?
```

Parameters

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

Parameter Types	Explanations and Examples
Numeric	Accepts all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation (NR1, NR2, NR3, and NRf numeric representations as defined in ANSI X3.42-1975 [5]). See the following definitions for NR1, NR2, NR3, and NRf. 123, 123E2, -123, -1.23E2, .123, 1.23E-2, 1.23000E-01. Special cases include MIN and MAX.
NR1	NR1 Numeric Response Data has the form ± <digits>; for example 123, -123.
NR2	NR2 Numeric Response Data has the form ± <digits>.<digits>; for example 0.123, 12.123.
NR3	NR3 Numeric Response Data has the form ± <digits>.<digits>E±<digits>; for example 123E2, -1.23E2, 1.23E-2, 1.23000E-01.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Parameter Types	Explanations and Examples
NRf	NRf is a flexible version of the three numeric representations (NR1, NR2, or NR3) as defined in ANSI X3.42-1975 [5].
Boolean	Represents a single binary condition that is either true or false. ON, OFF, 1, 0
Discrete	Selects from a finite number of values. These parameters use mnemonics to represent each valid setting. An example is the FORMat command where the formats can be FORMat[:DATA] ASCii | REAL | PACKed.
Character	12 characters maximum, first character must be an alpha character, the character set is alpha-numeric (A-Z, 0-9) and includes the underscore.
Block	A block of data consisting of a data type specified by NR1. The data can be either bytes or words.

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 FORMat[:DATA] <*type*>[<*length*>] command. If you send the command without specifying the parameter <length>, the module sets the length as is. If you send the <length> parameter, the command selects the data size set by <length>. A space is required between the command and the parameter.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Parameter and Return Value Types and Sizes

The following table contains the allowable type and sizes of the parameter and query data to sent to the module and returned by the module.

Data Types/Sizes	Description
int16	Signed 16-bit integer number.
int32	Signed 32-bit integer number.
uint16	Unsigned 16-bit integer number.
uint32	Unsigned 32-bit integer number.
float32	32-bit floating point number.
float64	64-bit floating point number.
string	String of characters.

Linking Commands

Linking IEEE 488.2 Common Commands With SCPI Commands

Individual commands can be linked together in a single "program message". These commands will execute, in order, when a "program message terminator" is encountered, i.e. a new-line character, or an EOI or both. Combining commands in this way provides some efficiency improvements for the programmer; a more compact program listing and faster execution. To link IEEE 488.2 Common Commands with SCPI Commands, use a semicolon between the commands. For example:

```
*RST;FORMat  ASCii
```

Linking Multiple SCPI Commands From Different Subsystems

Use both a semicolon and a colon between the commands. For example:

```
FORMat  ASCii;:SYSTem:IRIG:TIME  1,0,0,0
```

Linking Multiple SCPI Commands at the Same Hierarchical Level

Use a semicolon to separate commands and omit higher level commands from the header. For example, linking STATus:OPERation:ENABle 1 with the command STATus:OPERation:CONDition? produces the following shorter form when the two commands are linked:

```
STATus:OPERation:ENABle  2;:CONDition?
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The ABORt Subsystem resets the trigger system and places the module into an idle state. It aborts any actions related to the trigger systems such as acquiring the current time.

Subsystem Syntax

ABORt	[no query]

Example **Placing the Module in its Idle State**

ABOR	*Places the card in an idle state.*

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The DIAGnostic Subsystem controls the Strobe output.

Substem Syntax

```
DIAGnostic
    :SMODe <mode>
    :SMAJor <hours>,<minutes>,<seconds>,<milliseconds>
    :SMINor <milliseconds>
```

DIAGnostic:SMODe

DIAGnostic:SMODe ***<mode>*** sets the mode for the strobe output on pin 3 of the front panel J1 connector. The available modes are:

MINor: Minor strobe mode
MAJor: Major strobe mode
OFF: Strobe mode is off

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
mode	discrete	string	MINor | MAJor | OFF	N/A

Comments

- The Minor strobe mode generates an output pulse once per second when the minor time (milliseconds) match the programmed strobe time. You must set the Minor time using the DIAGnostic:SMINor command.
- The Major strobe mode generates an output pulse once per 24 hours when the Major time (hours, minutes, seconds, milliseconds) matches the programmed strobe time. You must set the Major time using the DIAGnostic:SMAJor command.
- You can use the strobe output to generate an external trigger event at a precise time (see the example below).
- **Related Commands:** DIAGnostic:SMAJor, DIAGnostic:SMINor
- ***RST Condition:** OFF
- ***** **Power-ON Condition:** OFF

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Example **Selecting the Strobe Mode**

This example assumes that the strobe output (pin 3 of J1) is connected to the external trigger event input (pin 6 of J1).

TRIG:SOUR HOLD	*Suspends triggering*
DIAG:SMOD MIN	*Set strobe output mode to minor*
DIAG:SMIN 345	*Strobe output pulse every 345 milliseconds*
TRIG:SOUR EXT	*External trigger*
INIT	*Wait-for-trigger state*
FETC?	*Places the reading in the output buffer.*

DIAGnostic:SMODe?

DIAGnostic:SMODe? returns the current mode of the strobe output.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
mode	discrete	string	MIN I MAJ I OFF	N/A

Comments

- ***RST Condition:** OFF
- **Power-ON Condition:** OFF

Example **Returning the Srobe Mode**

DIAG:SMOD?	*Returns strobe output mode.*

DIAGnostic:SMAJor

DIAGnostic:SMAJor ***<hours>,<minutes>,<seconds>,<milliseconds>*** sets the strobe output pulse which will occur once every 24 hours.

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
hours	numeric	int16	0 to 23	N/A
minutes	numeric	int16	0 to 59	N/A
seconds	numeric	int16	0 to 59	N/A
milliseconds	numeric	int16	0 to 999	N/A

Comments

- If the strobe output mode has not been set to MAJor with the DIAGnostic:SMODe command, the module generates error -221,"Settings conflict".

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

- ***RST Condition:** 0
- **Power-ON Condition:** 0

Example **Outputting a Major Strobe Pulse**

```
DIAG:SMO MAJ                    Select the strobe mode
DIAG:SMAJ 12,15,33,678          Module outputs a strobe pulse when the
                                IRIG-B time matches 12 hours, 15 minutes,
                                33 seconds and 678 milliseconds
```

DIAGnostic:SMAJor?

DIAGnostic:SMAJor? returns four int16 values for the hours, minutes, seconds and milliseconds that were set by the DIAGnostic:SMAJor command.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
hours	numeric	int16	0 to 23	N/A
minutes	numeric	int16	0 to 59	N/A
seconds	numeric	int16	0 to 59	N/A
milliseconds	numeric	int16	0 to 999	N/A

Example **Returning the MAJor Strobe Values**

```
DIAG:SMAJ?                      Returns four, comma separated, int16
                                numbers.
```

DIAGnostic:SMINor

DIAGnostic:SMINor ***<milliseconds>*** sets the strobe output pulse which will occur once every second.

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
milliseconds	numeric	int16	0 to 999	N/A

Comments

- If the strobe output mode has not been set to MINor with the DIAGnostic:SMODe command, the module generates error -221,"Settings conflict".
- ***RST Condition:** 0
- **Power-ON Condition:** 0

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Example **Outputting a Minor Strobe Pulse**

DIAG:SMO MIN	*Select the strobe mode*
DIAG:SMIN 750	*Module outputs a strobe pulse when the IRIG-B time matches 750 milliseconds*

DIAGnostic:SMINor?

DIAGnostic:SMINor? returns the int16 value for the milliseconds that were set by the DIAGnostic:SMINor command.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
milliseconds	numeric	int16	0 to 999	N/A

Example **Returning the MINor Strobe Value**

DIAG:SMIN?	*Returns single int16 numbers:*

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The FETCh? subsystem retrieves the time measurement previously stored in the module's time registers.

Subsystem Syntax

```
FETCh
    [:TIME]?                    [query only]
```

FETCh[:TIME]?

FETCh[:TIME]? retrieves the time measurement previously stored in the time registers by the most recent INITiate[:IMMediate] command. The command places the time measurement into the output buffer.

Comments

- The module generates error -230, "Data corrupt or stale". if there is no valid reading (e.g., *RST;TRIGger:SOURce HOLD followed by FETCh?).
- The module generates a Data invalid error if INITiate[:IMMediate] is not executed prior to executing the FETCh[:TIME]? command.
- The FORMat command determines the format of the time data in the output buffer.
- **Related Command:** INITiate[:IMMEdiate]

Example **Placing the Time Reading into the Output Buffer**

```
INIT                    Places a reading in the module's time
                        registers.
FETC?                   Places the reading into the output buffer.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The FORMat subsystem selects the format of the data transferred from the module to the computer by the FETCh?, MEASure?, or READ? commands.

Subsystem Syntax

```
FORMat
    [:DATA] <type>[,<length>]
```

FORMat[:DATA]

FORMat[:DATA] ***<type>*[,*<length>*]** determines the data type and the number of bits used for a transmission (see Chapter 2 for more information on data types).

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
type	discrete	string	ASCii | PACKed | REAL	ASCii
length	numeric	int16	17 | 32 | 64	N/A

Comments

- **ASCii** - outputs the time value as a 17 digit ascii number representing the value as "*ddhhmmss.xxxxxxxx*", where *dd*=days, *hh*=hours, *mm*=minutes, *ss*=seconds, and *xxxxxxxx*=sub seconds.
- **REAL** - outputs the time value as either a 64-bit or 32-bit Real or floting point number that is output as Definite Length Arbirtrary Block Data, as defined by the IEEE-488.2 standard.
- **PACKed** - outputs the time value as a block of data consisting of eight 8-bit bytes that are output as Definite Length Arbitray Block, as defined by the IEEE-488.2 standard.
- Sending *<type>* as **ASCii** with a value in *<length>* requires that *<length>* must be 17, or the module generates error -224,"Illegal parameter".
- Sending *<type>* as **PACKed** ignores *<length>*, even if sending a *<length>* parameter.
- ***RST Condition:** ASCii
- **Power-ON Condition:** ASCii

Example **Setting the Data Format**

```
FORM REAL,32          Format is set to REAL,32
READ?                 Takes a reading and places it into the
                      output buffer using 4 bytes.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Example **Setting the Data Format**

```
FORM REAL                    Format is set to REAL,64
READ?                        Takes a reading and places it into the
                             output buffer using 8 bytes.
```

FORMat[:DATA]?

FORMat[:DATA]? returns the format of the data transferred from the module to the computer by the FETCh?, MEASure?, or READ? commands.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
type	discrete	string	ASCii | PACKed | REAL	ASCii
length	numeric	int16	17 | 0 | 32 | 64	N/A

Comments

- The FORMat? command returns one of the following values:
  - ASC,17
  - PACK,0
  - REAL,32
  - REAL,64

Example **Returning the Data Format**

```
FORM?                        Get Data format.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The INITiate Subsystem places the module in a wait-for-trigger state.

Subsystem Syntax

```
INITiate
    [:IMMediate]          [no query]
```

INITiate[:IMMediate]

INITiate[:IMMediate] places the module in a wait-for-trigger state. When a trigger is received, the time measurment data is placed in the module's time registers.

Comments

- ABORt places the module in the idle state.
- If the trigger source is immediate (TRIGger:SOURce IMMediate), the time measurement starts and the time data is placed in the module's time registers as soon as the INITiate command is executed.
- To transfer the time reading from the module's time registers to the output buffer, use the FETCh? command.
- ***RST Condition:** module in the idle state
- **Power-ON Condition:** module in the idle state

Example **Placing a Reading into the Time Registers**

TRIG:SOUR HOLD	*Suspends triggering*
INIT	*Select wait-for-trigger state*
TRIG	*Triggers the module*
FETC?	*Places the reading into the output buffer*

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The MEASure subsystem requests for the module to make a time measurement. The command then makes the measurement and places the reading into the output buffer.

Subsystem Syntax

```
MEASure
    :TIME?                                    [query only]
```

MEASure:TIME?

MEASure:TIME? initiates a time measurement and sends it to the computer.

Comments

- Executing MEASure:TIME? is equivalent to programming with both the INITiate[:IMMediate] and FETCh[:TIME]? commands.
- This command reads the current time and places the results into the output buffer.
- The FORMat command determines the format of the time data in the output buffer.
- This command is equivalent to the READ? command.

Example **Placing the Time Reading into the Output Buffer**

```
MEAS:TIME?          Takes a time measurement and places the
                    reading into the output buffer.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The READ? subsystem requests for the module to make a time measurement. The command then makes the measurement and places the reading into the output buffer.

Subsystem Syntax

```
READ
    [:TIME]?                    [query only]
```

READ[:TIME]?

READ[:TIME]? initiates a time measurement and sends it to the computer.

Comments

- Executing READ[:TIME]? is equivalent to programming with both the INITiate[:IMMediate] and FETCh[:TIME]? commands.
- The READ[:TIME]? command reads the current time and places the results into the output buffer.
- The FORMat command determines the format of the time data in the output buffer.
- The READ[:TIME]? command is equivalent to the MEASure:TIME? command.

Example **Placing the Time Reading into the Output Buffer**

```
READ?                       Takes a time measurement and places the
                            reading into the output buffer.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The STATus subsystem returns the module's status information and sets the masks that specify the events that are reported to the Device Status Byte.

Subsystem Syntax

```
STATus
    :OPERation
        :CONDition?                    [query only]
        :ENABle <mask>
        [:EVENt]?                      [query only]
    :QUEStionable
        :CONDition?                    [query only]
        :ENABle <mask>
        [:EVENt]?                      [query only]
```

STATus:OPERation:CONDition?

STATus:OPERation:CONDition? returns the value in the Operation Condition Register. Use this value to determine if the module currently uses an external IRIGB source or the internal time code generator.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
cond reg	NR1	uint16	0-288	N/A

Comments

- Bit 8 of the Operation Status Register contains the IRIGB Mode bit.
- A low (0) in bit 8 indicates that the module uses an external IRIGB source, and a high (1) in bit 8 indicates that the module uses the internal time code generator.
- Bit 5 of the Operation Status Register contains "Waiting for TRG" state. A high (1) means the module is waiting for a trigger.
- **Related Commands:** STATus:OPERation:ENABle
  STATus:OPERation[:EVENt]?
  STATus:QUEStionable[:EVENt]?
- ***RST Condition:** all bits in the Operation Status Register are set to 0; if the module fails to detect an external IRIGB source, it automatically selects the internal time code generator and set bit 8 to a high (1)
- **Power-ON Condition:** all bits in the Operation Status Register are set to 0; if the module fails to detect an external IRIGB source, it automatically selects the internal time code generator and set bit 8 to a high (1)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Example **Returning the Operation Status Value**

STAT:OPER:COND? *Returns a copy of the Operation status.*

STATus:OPERation:ENABle

STATus:OPERation:ENABle *<mask>* sets the enable mask to allow reporting true conditions in the Event Register by the summary bit.

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
mask	numeric	int16	0 to 32767	N/A

Comments

- ***RST Condition:** all bits in the Enable Register are set to 0
- **Power-ON Condition:** all bits in the Enable Register are set to 0

Example **Reading the Operation Status Value**

STAT:OPER:ENAB 512 *Enables the external trigger bit.*

STAT:OPER:COND? *Returns the Operation Status value to the output buffer.*

STATus:OPERation:ENABle?

STATus:OPERation:ENABle? returns the enable mask value to the output buffer.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
enab reg	NR1	uint16	0 to 32767	N/A

Example **Reading the Enable Register**

STAT:OPER:ENAB? *Read the register.*

STATus:OPERation[:EVENt]?

STATus:OPERation[:EVENt]? returns the contents of the Operation Event Register to the output buffer.

Comments

- Reading the Operation Event Register clears it.
- Bit 9 of the Operation Event Register is the "External Trigger" bit. This bit goes high (1) when the module receives an external trigger (the module must be in TRIGger:SOURce EXTernal and initialized with INITiate[:IMMediate]).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

- **Related Commands:** STATus:OPERation:ENABle
  STATus:QUEStionable[:EVENt]?
- ***RST Condition:** all bits in the Enable Register are set to 0
- **Power-ON Condition:** all bits in the Enable Register are set to 0

Example **Reading the Operation Event Register**

```
STAT:OPER?
```
Returns the Operation Event value to the output buffer.

STATus:QUEStionable:CONDition?

STATus:OPERation:CONDition? returns the value of the Condition Register in the Questionable Data Group.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
cond reg	NR1	uint16	0 to 32767	N/A

Example **Reading the Questionable Condition Register**

```
STAT:QUES:COND?
```
Read the register

STATus:QUEStionable:ENABle

STATus:QUEStionable:ENABle ***<mask>*** sets the enable mask to allow reporting true conditions in the Event Register by the summary bit.

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
mask	numeric	int16	0 to 32767	N/A

Comments

- ***RST Condition:** all bits in the Enable Register are set to 0
- **Power-ON Condition:** all bits in the Enable Register are set to 0

Example **Reading the Operation Status Value**

```
STAT:QUES:ENAB 4096
```
Enables the bit.

```
STAT:QUES:COND?
```
Returns the condition register value to the output buffer.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

STATus:OPERation:ENABle?

STATus:OPERation:ENABle? returns the enable mask value to the output buffer.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
enab reg	NR1	uint16	0 to 32767	N/A

Example **Reading the Enable Register**

```
STAT:QUES:ENAB?          Read the register.
```

STATus:QUEStionable[:EVENt]?

STATus:QUEStionable[:EVENt]? returns the bit value of the Event Register in the Questionable Data Group.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
ev reg	NR1	uint16	0 | 4	N/A

Comments Bit 2 of the Questionable Event Register is the TIME Questionable Event. This bit is high (1) if the time data in the time registers is questionable.

Example **Reading the Event Register**

```
STAT:QUES?               Get event register value
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The SYSTem subsystem returns error messages and the SCPI version number to which the BANC-350 IRIGB Module complies. The subsystem also sets the time for the internal time code generator.

Subsystem Syntax

```
SYSTem
    :ERRor?                                                    [query only]
    :IRIG
        :TIME <days>,<hours>,<minutes>,<seconds>               [no query]
    :VERSion?                                                  [query only]
```

SYSTem:ERRor?

SYSTem:ERROR? returns the error messages in the error queue.

Comments

- The module places any generated errors into the error queue. The queue is first-in, first out. With several errors waiting in the queue, the SYSTem:ERRor? returns the oldest unread error message first.
- The error queue can hold 30 error messages. If the module generates more than 30 messages that are not read, it replaces the last error message in the queue with error -350,"Too many errors". No additional messages are placed into the queue until SYSTem:ERRor? reads some messages or the *CLS (clear status) command clears the queue.
- When the error queue is empty, SYSTem:ERRor? returns: +0,"No error".
- ***RST Condition:** unaffected
- **Power-On Condition:** no errors are in the error queue

Note For a listing of the error messages, refer to the "HP E1405 Command Module or the HP E1406 Command Module" User's Manual.

Example **Reading the error queue**

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


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SYSTem:IRIG:TIME

SYSTem:IRIG:TIME *<days>,<hours>,<minutes>,<seconds>* sets the time of the internal time code generator.

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
days	numeric	int16	1 to 365	N/A
hours	numeric	int16	0 to 23	N/A
minutes	numeric	int16	0 to 59	N/A
seconds	numeric	int16	0 to 59	N/A

Comments

- The SYSTem:IRIGB:TIME command only sets the time if there is no external time source, and the module has automatically switched over to the internal time code generator.
- If there is an external IRIGB source and this command is executed, it generates error -221, "Settings conflict".
- ***RST Condition:** with no external time source, the module switches to the internal time code generator with all time values set to 0; the module locks to an external IRIGB source if there is a source connected to the module
- **Power-ON Condition:** with no external time source, the module switches to the internal time code generator with all time values set to 0

Example

Setting the Day and Time

```
SYST:IRIGB:TIME 122,12,30,00        Sets the time to 122 days, 12:30 p.m.
```

SYSTem:VERSion?

SYSTem:VERSion? returns the SCPI version number to which the module complies: "1990.0".

Comment

- ***RST Condition:** none

Example

Querying the SCPI revision

```
SYST:VERS?                          Query SCPI revision
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The TRIGger subsystem synchronizes the time measurement acquisition using the module.

Subsystem Syntax

```
TRIGger
    [:IMMediate]                    [no query]
    :SLOPe <edge
    :SOURce <source
```

TRIGger[:IMMediate]

TRIGger[:IMMediate] causes the module to make a time measurement if in the BUS or HOLD trigger source and an INITiate[:IMMediate] command was executed.

Comments

- **Related Commands:** TRIGger:SOURce, INITiate[:IMMediate]

Example **Triggering the Module**

```
TRIG:SOUR HOLD      Suspends triggering
INIT                Wait-for-trigger state
TRIG                Trigger the module
FETC?               Places the reading into the output buffer
```

TRIGger:SLOPe

TRIGger:SLOPe ***<edge>*** selects the edge (rising, falling or either edge) of the trigger applied to the module.

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
edge	discrete	string	POSitive | NEGative | EITHer	N/A

Comments

- ***RST Condition:** POSitive
- **Power-ON Condition:** POSitive

Example **Selecting the Trigger Slope**

```
TRIG:SLOP NEG       Sets the trigger to occur on a falling edge
TRIG:SOUR EXT       Enables an external trigger
```

TRIGger:SLOPe?

TRIGger:SLOPe? returns the current trigger edge of the module.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
edge	discrete	string	POS | NEG | EITH	N/A

Example Returning the Trigger Edge

```
TRIG:SLOP?                    Get the current trigger edge
```

TRIGger:SOURce

TRIGger:SOURce ***<source>*** specifies the trigger source for the module. The available sources are:

BUS: Group Execute Trigger (GET) bus command or *TRG common command.
EXTernal: The IRIGB module's external Event Input (pin 6 of front panel connector J1).
HOLD: Suspends triggering.
IMMediate: The trigger system is always true (continuous triggering).

Parameters

Parameter Name	Parameter Type	Parameter Data Type	Range of Values	Default Units
source	discrete	string	BUS | EXTernal | HOLD | IMMediate	N/A

Comments

- The TRIGger:SOURce command only selects the trigger source. You must use the INITiate[:IMMediate] command to place the module in a wait-for-trigger state.
- ***RST Condition:** IMMediate
- **Power-ON Condition:** IMMediate

Example Selecting the Trigger Source

```
TRIG:SLOP  NEG          Sets the trigger to occur on a falling edge
TRIG:SOUR  BUS          Enables a bus trigger
INIT                    Places the card in a wait-for-trigger state;
                        measure on an external trigger
*TRG                    Trigger the module to store reading
FETC?                   Places the reading into the output buffer
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

TRIGger:SOURce?

TRIGger:SOURce? returns the current trigger source.

Return Values

Return Name	Return Type	Return Data Type	Return Values	Default Units
source	discrete	string	BUS | EXT | HOLD | IMM	N/A

Example

Returning the Trigger Source

```
TRIG:SOUR?                    Get the current trigger source
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

BANC-350 IRIGB Module Command Quick Reference

The following tables summarize the SCPI commands and IEEE 488.2 Common (*) commands for the BANC-350 IRIGB Module.

IEEE 488.2 Common Commands Quick Reference

Command	Title	Description
*CLS	Clear Status	Clears all Event registers. Forces the system to the Operation Complete, Command Idle state and Query Idle state. Also clears the SYSTem:ERRor queue.
*ESE	Enable Standard Event	*ESE<*numeric_value*> specifies an 8-bit mask that determines which event bits from the Standard Event Register will be summarized by bit 5 (ESB) in the Standard Status Byte.
*ESE?	Enable Standard Event Query	Returns the current Enable Standard Event mask byte.
*ESR?	Standard Event Register Query	Returns the current contents of the Standard Event Register which is cleared when read.
*IDN?	Instrument ID Query	Returns the instrument ID as four groups separated by commas e.g. BANCOM DIVISION OF DATUM INC,BANC-350,0,A.00.02
*OPC	Operation Complete	Sets the Operation Complete bit in the Standard Event register when all pending operations have finished.
*OPC?	Operation Complete Query	Returns a "1" when all pending operations have finished.
*SRE	Service Request Enable	*SRE<*numeric_value*> specifies an 8-bit mask that enables bits in the Status Byte register that cause the generation of a Service Request. All bits enabled are summarized in bit 6 (MSS) of the Status Byte Register.
*SRE?	Service Request Enable Query	Returns the current Service Request Enable mask.
*STB?	Status Byte Register Query	Returns the current contents of the Status Byte register.
*RST	Reset	• *RST Conditions: ABORt DIAGnostic:SMODe OFF FORMat[:DATA] ASCii STATus:OPERation:ENABle 0 SYSTem:IRIG:TIME 0 TRIGger:SLOPe POSitive TRIGger:SOURce IMMediate

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Command	Title	Description
*TRG	Bus Trigger	Produces a trigger event.
*TST?	Test Query	Performs a self-test and responds with the test result. A 0 response indicates the test passed. A non-zero response indicates the test failed.
*WAI	Wait To Continue	Prevents further operations until the No-Operation-Pending flag is true.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SCPI Commands Quick Reference

In the following quick reference, "co" means command only (i.e., command has no corresponding query command) and "qo" means query only (command is a query command). No designator means the command has a corresponding query command.

ABORt	co	Resets the trigger system and places the module into an idle state.
DIAGnostic :SMODe MINor | MAJor | OFF :SMAJor *<hours>,<minutes>,* *<seconds>,<milliseconds>* :SMINor *<milliseconds>*		 Set strobe output mode. Set major strobe output time Set minor strobe output time.
FETCh?	qo	Retrieves the time measurement previously stored in the module's time registers.
FORMat [:DATA] ASCii | PACKed | REAL,17 | 32 | 64		Determines the data type and the number of bits used for a transmission.
INITiate [:IMMediate]	co	Transfers the time measurement data into the module's time registers.
MEASure :TIME?	qo	Requests for the module to make a time measurement (same as READ[:TIME]?).
READ? [:TIME]?	qo	Requests for the module to make a time measurement (same as MEASure:TIME?).
STATus :OPERation :CONDition? :ENABle *<mask>* [:EVENt]? :QUEStionable :CONDition? :ENABle *<mask>* [:EVENt?]	 qo qo qo qo	 Returns value in the operation condition register. Sets enable mask of operation enable register Returns value of the operation event register. Returns value in the questionable condition register. Sets enable mask of questionable enable register. Returns value in the questionable data register.
SYSTem :ERRor? :IRIG :TIME *<days>,<hours>,<minutes>,<seconds>* :VERSion?	qo qo	 Returns the error messages in the error queue. Sets the time of the internal time code generator. Returns the SCPI version number to which the module complies: "1990.0".

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

TRIGger [:IMMediate]	co	Causes the module to make a time measurement (for BUS or HOLD trigger sources only).
:SLOPe POSitive | NEGative | EITHer		Selects the edge (rising, falling or either edge) of the trigger applied to the module.
:SOURce BUS | EXTernal | HOLD | IMMediate		Specifies the trigger source for the module.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SCPI Conformance Information

The BANC-350 IRIGB Module conforms to the SCPI-1990.0 standard.

The folloiwng tables list all the SCPI confirmed, approved, and non-SCPI commands that the module can execute.

SCPI Confirmed Commands

```
ABORt                          STATus
                                 :OPERation
FETCh?                             :CONDition?
                                   :ENABle <mask>
FORMat                             [:EVENt]?
  [:DATA]                        :QUEStionable
                                   [:EVENt]?
INITiate
  [:IMMediate]                 SYSTem
                                 :ERRor?
MEASure                          :VERSion?
  :TIME?
                               TRIGger
READ                             [:IMMediate]
  [:TIME]?                       :SLOPe
                                 :SOURce
```

Non-SCPI Commands

```
SYSTem
  :IRIG
    :TIME
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Index

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Index

A

Abbreviated commands, 3-1
Address, logical, 1-2
Addressing, HP-IB, 1-8, 1-12, 1-15
ASCii data format, 2-3

C

C language program, 1-7
Command module/mainframe device drivers, 1-2
Command types, 3-1
Commands
    abbreviated, 3-1
    implied, 3-2
    linking, 3-4
    linking IEEE 488.2 common commands
        with SCPI, 3-4
Common command format, 3-1
Compiled SCPI language program, 1-4
Compiler used, 1-4
Compound command separator, 3-1
Configuration, system, 1-4, 1-7, 1-12, 1-15

D

Data formats, 2-3
    ASCii, 2-3
    PACKed, 2-6
    REAL, 2-3
Description, module, 2-1
Device drivers, 1-1
Drivers
    command module/mainframe device, 1-2
    device, 1-1
    HP-UX, 1-2

E

Example programs, 1-2

F

Formats
    common command, 3-1
    data, 2-3
    SCPI command, 1-3, 3-1

H

HP BASIC/IBASIC language program, 1-12
HP-IB addressing, 1-8, 1-12, 1-15
HP-UX drivers, 1-2

I

Implied commands, 3-2
Installing the module, 1-1
Instrument and programming languages, 1-2 - 1-18

L

Languages
    instrument, 1-2 - 1-18
    programming, 1-2 - 1-18
Level, linking multiple SCPI commands at the
    same hierarchical, 3-4
Linking commands, 3-4
Linking IEEE 488.2 common commands
    with SCPI commands, 3-4
Linking multiple SCPI commands at the
    same hierarchical level, 3-4
Linking multiple SCPI commands from
    different subsystems, 3-4
Logical address, 1-2

M

Module description, 2-1
Module, installing the, 1-1

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

O

Outputs (trg_time.scpi)
    packed, 2-5 - 2-8
    triggering the module, 2-5 - 2-8

P

PACKed data format, 2-6
Parameter and return value types and sizes, 3-4
Preparation for use, 1-1
Program
    C language, 1-7
    compiled SCPI language, 1-4
    compiling, 1-8, 1-15
    example, 1-2
    HP BASIC/IBASIC language, 1-12
    QuickBASIC language, 1-15
    running, 1-8, 1-15
    sample, 1-3
    typical C, 1-8
    typical C-SCPI example, 1-4
    typical HP BASIC, 1-12
    typical QuickBASIC, 1-16

Q

QuickBASIC language program, 1-15

R

Reading the time registers (rd_time.scpi), 2-2 - 2-4
REAL data format, 2-3
Running and compiling the program, 1-8, 1-15

S

Sample programs, 1-3
SCPI command format, 1-3, 3-1
SCPI command structure, 1-3
Separator, compound command, 3-1
Slope, trigger, 2-9
Sources, trigger, 2-97
Structure, SCPI command, 1-3
Subsystems, linking multiple SCPI commands
    from different, 3-4
System configuration, 1-4, 1-7, 1-12, 1-15

T

Trigger slope, 2-9
Trigger sources, 2-9
Triggering the module and packed
    outputs (trg_time.scpi), 2-5 - 2-9
Types, command, 3-1
Typical,
    C program, 1-8
    C-SCPI example program, 1-4
    HP BASIC program, 1-12
    QuickBASIC program, 1-16

U

Use, preparation for, 1-1
Used, compiler, 1-4

V

Value types and sizes
    parameter, 3-4
    return, 3-4

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SALES & SUPPORT OFFICES
Arranged alphabetically by country

HEADQUARTERS OFFICES
If there is no sales office listed for your area, contact one of these headquarters offices.

NORTH/CENTRAL AFRICA
Hewlett-Packard S.A.
7, rue du Bois-du-Lan
CH-1217 **MEYRIN** 1, Switzerland
Tel: (022) 83 12 12
Telex: 27835 hmea
Cable: HEWPACKSA Geneve

ASIA
Hewlett-Packard Asia Ltd.
47/F, 26 Harbour Rd.,
Wanchai, **HONG KONG**
G.P.O. Box 863, Hong Kong
Tel: 5-8330833
Telex: 76793 HPA HX
Cable: HPASIAL TD

EASTERN EUROPE
Hewlett-Packard Ges.m.b.h.
Lieblgasse 1
P.O.Box 72
A-1222 **VIENNA**, Austria
Tel: (222) 2500-0
Telex: 1 3 4425 HEPA A

NORTHERN EUROPE
Hewlett-Packard S.A.
V. D. Hooplaan 241
P.O.Box 999
NL-1183 AG **AMSTELVEEN**
The Netherlands
Tel: 20 547999
Telex: 18 919 hpner

SOUTH EAST EUROPE
Hewlett-Packard S.A.
World Trade Center
110 Avenue Louis Casai
1215 Cointrin, **GENEVA**, Switzerland
Tel: (022) 98 96 51
Telex: 27225 hpser

EASTERN USA
Hewlett-Packard Co.
4 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 948-6370

MIDWESTERN USA
Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800

SOUTHERN USA
Hewlett-Packard Co.
2000 South Park Place
ATLANTA, GA 30339
Tel: (404) 955-1500

WESTERN USA
Hewlett-Packard Co.
5161 Lankershim Blvd.
NORTH HOLLYWOOD, CA 91601
Tel: (818) 505-5600

MEDITERRANEAN AND MIDDLE EAST
Hewlett-Packard S.A.
Mediterranean and Middle East Operations
Atrina Centre
32 Kifissias Ave.
Paradissos-Amarousion, **ATHENS**
Greece
Tel: 682 88 11
Telex: 21-6588 HPAT GR
Cable: HEWPACKSA Athens

OTHER INTERNATIONAL AREAS
Hewlett-Packard Co.
Intercontinental Headquarters
3495 Deer Creek Road
PALO ALTO, CA 94304
Tel: (415) 857-1501
Telex: 034-8300
Cable: HEWPACK

ARGENTINA
Hewlett-Packard Argentina S.A.
Montaneses 2140/50
1428 **BUENOS AIRES**
Tel: 781-4059/69
Cable: HEWPACKARG

AUSTRALIA
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
P.O. Box 221
BLACKBURN, Victoria 3130
Tel: 895-2895
Telex: 31-024
Cable: HEWPARD Melbourne

Hewlett-Packard Australia Ltd.
17-23 Talavera Road
P.O. Box 308
NORTH RYDE, N.S.W. 2113
Tel: 888-4444
Telex: 21561
Cable: HEWPARD Sydney

AUSTRIA
Hewlett-Packard Ges.m.b.h.
Lieblgasse 1
P.O. Box 72
A-1222 **VIENNA**
Tel: (0222) 2500-0
Telex: 134425 HEPA A

BELGIUM
Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 **BRUSSELS**
Tel: (02) 762-32-00
Telex: 23-494 paloben bru

BRAZIL
Hewlett-Packard do Brasil
I.e.C. Ltda.
Alameda Rio Negro, 750
ALPHAVILLE
06400 Barueri SP
Tel: (011) 421.1311
Telex: (011) 33872 HPBR-BR
Cable: HEWPACK Sao Paulo

Hewlett-Packard do Brasil
I.e.C. Ltda.
Praia de Botafago 228
6° Andar-conj 614
Edificio Argentina - Ala A
22250 **RIO DE JANEIRO**, RJ
Tel: (021) 552-6422
Telex: 21905 HPBR-BR
Cable: HEWPACK Rio de Janeiro

CANADA
Hewlett-Packard (Canada) Ltd.
11120-178th Street
EDMONTON, Alberta T5S 1P2
Tel: (403) 486-6666

Hewlett-Packard (Canada) Ltd.
17500 Trans Canada Highway
South Service Road
KIRKLAND, Quebec H9J 2X8
Tel: (514) 697-4232
Telex: 058-21521

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Telex: 069-8644

Hewlett-Packard (Canada) Ltd.
2670 Queensview Dr.
OTTAWA, Ontario K2B 8K1
Tel: (613) 820-6483

CHINA, People's Republic of
China Hewlett-Packard Co., Ltd.
P.O. Box 9610, Beijing
4th Floor, 2nd Watch Factory Main Bldg.
Shuang Yu Shou, Bei San Huan Road
Hai Dian District
BEIJING
Tel: 28-0567
Telex: 22601 CTSHP CN
Cable: 1920 Beijing

DENMARK
Hewlett-Packard A/S
Kongevejen 25
DK-3460 **BIRKEROED**
Tel: (02) 81-66-40
Telex: 37409 hpas dk

FINLAND
Hewlett-Packard Oy
Pilspankalliontie 17
02200 **ESPOO**
Tel: 00358-0-88721
Telex: 121563 HEWPA SF

FRANCE
Hewlett-Packard France
Chemin des Mouilles
Boite Postale 162
69131 **ECULLY** Cedex (Lyon)
Tel: (78) 133-81-25
Telex: 310617F

Hewlett-Packard France
Parc d'activités du Bois Briard
Avenue du Lac
91040 **EVRY** Cedex
Tel: (60) 77-83-83
Telex: 692315F

Hewlett-Packard France
Zone Industrielle de Courtaboeuf
Avenue des Tropiques
91947 **LES ULIS** Cedex (Orsay)
Tel: (69) 07-78-25
Telex: 600048F

GERMAN FEDERAL REPUBLIC
Hewlett-Packard GmbH
Vertriebszentrum Mitte
Hewlett-Packard-Strasse
D-6380 **BAD HOMBURG**
Tel: (06172) 16-0
Fax: (06172) 16-1309

Hewlett-Packard GmbH
Vertriebszentrun Südwest
Schickardstrasse 2
D-7030 **BÖBLINGEN**
Tel: (07031) 14-0
Fax: (07031) 14-6429

Hewlett-Packard GmbH
Vertriebszentrum Süd
Eschenstrasse 5
D-8028 **TAUFKIRCHEN**
Tel: (089) 61207-0
Fax: (089) 61207-300

GREECE
Hewlett-Packard A.E.
178, Kifissias Avenue
6th Floor
Halandri-**ATHENS**
Greece
Tel: 6471543, 6471673, 6472971
Telex: 221 286 HPHLGR

HONG KONG
Hewlett-Packard Hong Kong, Ltd.
G.P.O. Box 795
5th Floor, Sun Hung Kai Centre
30 Harbour Road
HONG KONG
Tel: 5-8323211
Telex: 66678 HEWPA HX
Cable: HEWPACK HONG KONG

ICELAND
Hewlett-Packard Iceland
Hoefdabakka 9
110 **REYKJAVIK**
Tel: (1) 67 1000

INDIA
Blue Star Ltd.
13 Community Center
New Friends Colony
NEW DELHI 110 065
Tel: 633182, 636674
Telex: 031-61120
Cable: BLUEFROST

INDONESIA
BERCA Indonesia P.T.
P.O.Box 2497/Jkt
Antara Bldg., 11th Floor
Jl. Medan Merdeka Selatan 17
JAKARTA-PUSAT
Tel: 343989
Telex: 46748 BERSAL IA

IRELAND
Hewlett-Packard Ireland Ltd.
82/83 Lower Leeson Street
DUBLIN 2
Tel: 0001 608800
Telex: 30439

ISRAEL
Computation and Measurement Systems (CMS) Ltd.
11 Masad Street
67060
TEL-AVIV
Tel: 388 388
Telex: 33569 Motil IL

ITALY
Hewlett-Packard Italiana S.p.A.
Via G. di Vittorio 9
I-20063 **CERNUSCO SUL NAVIGLIO**
(Milano)
Tel: (02) 923691
Telex: 334632

Hewlett-Packard Italiana S.p.A.
Viale C. Pavese 340
I-00144 **ROMA EUR**
Tel: (06) 54831
Telex: 610514

JAPAN
Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg.,
4-20 Nishinakajima, 5 Chome
Yodogawa-ku
OSAKA, 532
Tel: (06) 304-6021
Telex: YHPOSA 523-3624

Yokogawa-Hewlett-Packard Ltd.
29-21 Takaido-Higashi, 3 Chome
Suginami-ku **TOKYO** 168
Tel: (03) 331-6111
Telex: 232-2024 YHPTOK

Yokogawa-Hewlett-Packard Ltd.
Yasuda Seimei Nishiguchi Bldg.
30-4 Tsuruya-cho, 3 Chome
Kanagawa-ku, **YOKOHAMA** 221
Tel: (045) 312-1252

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SALES & SUPPORT OFFICES

Arranged alphabetically by country

KOREA
Samsung Hewlett-Packard Co. Ltd.
Dongbang Yeoeuido Building
12-16th Floors
36-1 Yeoeuido-Dong
Youngdeungpo-Ku
SEOUL
Tel: 784-4666, 784-2666
Telex: 25166 SAMSAN K

MALAYSIA
Hewlett-Packard Sales (Malaysia) Sdn. Bhd.
9th Floor
Chung Khiaw Bank Building
46, Jalan Raja Laut
50350 **KUALA LUMPUR**
Tel: 2986555
Telex: 31011 HPSM MA

MEXICO
Hewlett-Packard de Mexico, S.A. de C.V.
Monte Pelvoux No. 111
Lomas de Chapultepec
11000 **MEXICO, D.F.**
Tel: 5-40-62-28, 72-66, 50-25
Telex: 17-74-507 HEWPACK MEX

NETHERLANDS
Hewlett-Packard Nederland B.V.
Startbaan 16
NL-1187 XR **AMSTELVEEN**
P.O. Box 667
NL-1180 AR **AMSTELVEEN**
Tel: (020) 547-6911
Telex: 13 216 HEPA NL

NORWAY
Hewlett-Packard Norge A/S
Osterndalen 16-18
P.O. Box 34
N-1345 **OESTERAAS**
Tel: 0047/2/24 60 90
Telex: 76621 hpnas n

PUERTO RICO
Hewlett-Packard Puerto Rico
101 Muñoz Rivera Av
Esu. Calle Ochoa
HATO REY, Puerto Rico 00918
Tel: (809) 754-7800

SAUDI ARABIA
Modern Electronics Establishment
Hewlett-Packard Division
P.O. Box 1228
Redec Plaza, 6th Floor
JEDDAH
Tel: 644 96 28
Telex: 4027 12 FARNAS SJ
Cable: ELECTA JEDDAH

SINGAPORE
Hewlett-Packard Singapore (Sales) Pte. Ltd.
#08-00 Inchcape House
450-2 Alexandra Road
Alexandra P.O. Box 58
SINGAPORE, 9115
Tel: 4731788
Telex: 34209 HPSGSO RS
Cable: HEWPACK, Singapore

SOUTH AFRICA
Hewlett-Packard So Africa (Pty.) Ltd.
9 Eastern Service Road
Eastgate Ext. 3
SANDTON 2144
Tel: 802-5111, 802-5125
Telex: 4-20877 SA
Cable: HEWPACK Johannesburg

SPAIN
Hewlett-Packard Española, S.A.
Crta. de la Coruña, Km. 16, 400
Las Rozas
E-MADRID
Tel: (1) 637.00.11
Telex: 23515 HPE

SWEDEN
Hewlett-Packard Sverige AB
Skalholtsgatan 9, Kista
Box 19
S-16393 **SPÅNGA**
Tel: (08) 750-2000
Telex: (854) 17886
Telefax: (08) 7527781

SWITZERLAND
Hewlett-Packard (Schweiz) AG
7, rue du Bois-du-Lan
Case postale 365
CH-1217 **MEYRIN** 1
Tel: (0041) 22-83-11-11
Telex:27333 HPAG CH

TAIWAN
Hewlett-Packard Taiwan Ltd.
8th Floor, Hewlett-Packard Building
337 Fu Hsing North Road
TAIPEI
Tel: (02) 712-0404
Telex: 24439 HEWPACK
Cable:HEWPACK Taipei

TURKEY
Teknim Company Ltd.
Iran Caddesi No. 7
Karaklidere
ANKARA
Tel: 275800
Telex: 42155 TKNM TR

UNITED KINGDOM

ENGLAND
Hewlett-Packard Ltd.
Heathside Park Road
Cheadle Heath
STOCKPORT
Cheshire
SK3 ORB
Tel: 061-428-0828
Telex: 668068

Hewlett-Packard Ltd.
King Street Lane
Winnersh, **WOKINGHAM**
Berkshire RG11 5AR
Tel: 0734 784774
Telex: 847178

SCOTLAND
Hewlett-Packard Ltd.
SOUTH QUEENSFERRY
West Lothian, EH30 9TG
Tel: 031 331 1188
Telex: 72682

UNITED STATES

Alabama
Hewlett-Packard Co.
420 Wynn Drive
HUNTSVILLE, AL 35805
Tel: (205) 830-2000

Arizona
Hewlett-Packard Co.
8080 Pointe Parkway West
PHOENIX, AZ 85044
Tel: (602) 273-8000

California
Hewlett-Packard Co.
1421 S. Manhattan Av.
FULLERTON, CA 92631
Tel: (714) 999-6700

Hewlett-Packard Co.
5651 West Manchester Ave.
LOS ANGELES, CA 90045
Tel: (213) 337-8000
Telex: 910-325-6608

Hewlett-Packard Co.
9606 Aero Drive
SAN DIEGO, CA 92123
Tel: (619) 279-3200

Hewlett-Packard Co.
3003 Scott Boulevard
SANTA CLARA, CA 95054
Tel: (408) 988-7000
Telex: 910-338-0586

Colorado
Hewlett-Packard Co.
24 Inverness Place, East
ENGLEWOOD, CO 80112
Tel: (303) 649-5000

Connecticut
Hewlett-Packard Co.
47 Barnes Industrial Road South
WALLINGFORD, CT 06492
Tel: (203) 265-7801

Florida
Hewlett-Packard Co.
2901 N.W. 62nd Street
FORT LAUDERDALE, FL 33309
Tel: (305) 973-2600

Hewlett-Packard Co.
6177 Lake Ellenor Drive
ORLANDO, FL 32809
Tel: (305) 859-2900

Georgia
Hewlett-Packard Co.
2000 South Park Place
ATLANTA, GA 30339
Tel: (404) 955-1500
Telex: 810-766-4890

Illinois
Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800
Telex: 910-687-1066

Indiana
Hewlett-Packard Co.
11911 N. Meridian St.
CARMEL, IN 46032
Tel: (317) 844-4100

Louisiana
Hewlett-Packard Co.
160 James Drive East
ST. ROSE, LA 70087
P.O. Box 1449
KENNER, LA 70063
Tel: (504) 467-4100

Maryland
Hewlett-Packard Co.
3701 Koppers Street
BALTIMORE, MD 21227
Tel: (301) 644-5800
Telex: 710-862-1943

Hewlett-Packard Co.
2 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 948-6370

Massachusetts
Hewlett-Packard Co.
1775 Minuteman Road
ANDOVER, MA 01810
Tel: (617) 682-1500

Michigan
Hewlett-Packard Co.
39550 Orchard Hill Place Drive
NOVI, MI 48050
Tel: (313) 349-9200

Minnesota
Hewlett-Packard Co.
2025 W. Larpenteur Ave.
ST. PAUL, MN 55113
Tel: (612) 644-1100

Missouri
Hewlett-Packard Co.
1001 E. 101st Terrace Suite 120
KANSAS CITY, MO 64131-3368
Tel: (816) 941-0411

Hewlett-Packard Co.
13001 Hollenberg Drive
BRIDGETON, MO 63044
Tel: (314) 344-5100

New Jersey
Hewlett-Packard Co.
120 W. Century Road
PARAMUS, NJ 07653
Tel: (201) 265-5000

New Mexico
Hewlett-Packard Co.
7801 Jefferson N.E.
ALBUQUERQUE, NM 87109
Tel: (505) 823-6100

New York
Hewlett-Packard Co.
9600 Main Street
CLARENCE, NY 14031
Tel: (716) 759-8621

Hewlett-Packard Co.
7641 Henry Clay Blvd.
LIVERPOOL, NY 13088
Tel: (315) 451-1820

Hewlett-Packard Co.
3 Crossways Park West
WOODBURY, NY 11797
Tel: (516) 682-7800

North Carolina
Hewlett-Packard Co.
5605 Roanne Way
GREENSBORO, NC 27420
Tel: (919) 852-1800

Ohio
Hewlett-Packard Co.
15885 Sprague Road
CLEVELAND, OH 44136
Tel: (216) 243-7300

Hewlett-Packard Co.
9080 Springboro Pike
MIAMISBURG, OH 45342
Tel: (513) 433-2223

Hewlett-Packard Co.
675 Brooksedge Blvd.
WESTERVILLE, OH 43081
Tel: (614) 891-3344

Oklahoma
Hewlett-Packard Co.
3525 N.W. 56th St.
Suite C-100
OKLAHOMA CITY, OK 73112
Tel: (405) 946-9499

Oregon
Hewlett-Packard Co.
9255 S. W. Pioneer Court
WILSONVILLE, OR 97070
Tel: (503) 682-8000

Pennsylvania
Hewlett-Packard Co.
111 Zeta Drive
PITTSBURGH, PA 15238
Tel: (412) 782-0400

Hewlett-Packard Co.
2750 Monroe Boulevard
VALLEY FORGE, PA 19482
Tel: (215) 666-9000

Texas
Hewlett-Packard Co.
1826-P Kramer Lane
AUSTIN, TX 78758
Tel: (512) 835-6771

Hewlett-Packard Co.
10535 Harwin Drive
HOUSTON, TX 77036
Tel: (713) 776-6400

Hewlett-Packard Co.
930 E. Campbell Rd.
RICHARDSON, TX 75081
Tel: (214) 231-6101

Hewlett-Packard Co.
1020 Central Parkway South
SAN ANTONIO, TX 78232
Tel: (512) 494-9336

Utah
Hewlett-Packard Co.
3530 W. 2100 South St.
SALT LAKE CITY, UT 84119
Tel: (801) 974-1700

Virginia
Hewlett-Packard Co.
4305 Cox Road
GLEN ALLEN, VA 23060
Tel: (804) 747-7750

Washington
Hewlett-Packard Co.
15815 S.E. 37th Street
BELLEVUE, WA 98006
Tel: (206) 643-4000

Wisconsin
Hewlett-Packard Co.
275 N. Corporate Dr.
BROOKFIELD, WI 53005
Tel: (414) 794-8800

VENEZUELA
Hewlett-Packard de Venezuela C.A.
3A Transversal Los Ruices Norte
Edificio Segre 2 & 3
Apartado 50933
CARACAS 1050
Tel: (582) 239-4133
Telex: 251046 HEWPACK

YUGOSLAVIA
Do Hermes
General Zdanova 4
YU-11000 **BEOGRAD**
Tel: (011) 342 641
Telex: 11433

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.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