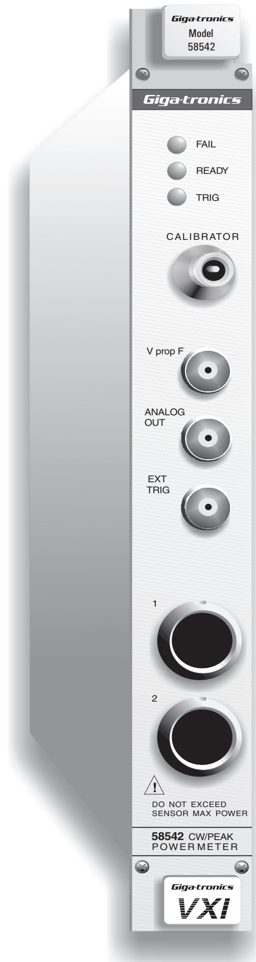
Giga-tronics

P/N: 21555
Revision: F
Publication Date: March 2007



58542 VXIbus Universal Power Meter

Operation & Maintenance Manual

58542

ISO 9001.. Certified Process
Registrar: BSI, Certification No. FM 34226 v Registered 04 June 1996 v Amended 01 March 2000

Giga-tronics Incorporated v 4650 Norris Canyon Road v San Ramon, California 94583
925.328.4650 or 800.726.4442 v 925.328.4700 (Fax) v 800.444.2878 (Customer Service) v 925.328.4702 (CS Fax)
www.gigatronics.com

Printed in the USA.

WARRANTY

Giga-tronics 58542 instrument is warranted against defective materials and workmanship for one year from date of shipment. Giga-tronics will at its option repair or replace products that are proven defective during the warranty period. This warranty DOES NOT cover damage resulting from improper use, nor workmanship other than Giga-tronics service. There is no implied warranty of fitness for a particular purpose, nor is Giga-tronics liable for any consequential damages. Specification and price change privileges are reserved by Giga-tronics.

Illustrations

Contents

3

Theory of Operation

4

Calibration & Testing

5

Maintenance

6

Parts Lists

7

Diagrams

A

Program Examples

B

Power Sensors

C

Options

Index

Tables

About This Manual

The following is contained within the Operation & Maintenance Manual of the 58542 VXIbus Universal Power Meter:

Preface:

In addition to a comprehensive Table of Contents and general information about the manual, the Preface also contains a record of changes made to the manual since its publication, and a description of Special Configurations. If you have ordered a user-specific manual, please refer to page xv for a description of the special configuration.

Chapters:

1 - Introduction

Brief introduction to the instrument and its performance parameters.

2 – Operation

Guide to instrument operation using the SCPI command set.

3 – Theory of Operation

Block diagram level description of the instrument and its circuits for maintenance and applications.

4 – Calibration & Testing

Procedures for inspection, calibration and performance testing of the instrument.

5 – Maintenance

Procedures for maintenance and troubleshooting.

6 – Parts Lists

All components and parts, and their sources.

7 – Diagrams

Schematics and component diagrams for all circuits.

Appendices:

A - Program Examples

Examples of programs for controlling the 58542.

B - Power Sensors

Specifications and technical data for the selection and application of power sensors.

C – Options

Options available for the 58542.

Index:

A subject listing of contents.

Changes that occur after publication of the manual, and Special Configuration data will be inserted as loose pages in the manual binder. Please insert and/or replace the indicated pages as detailed in the Technical Publication Change Instructions included with new and replacement pages.

Conventions

The following conventions are used in this product manual. Additional conventions not included here will be defined at the time of usage.

Warning

WARNING

The WARNING statement is encased in gray and centered in the page. This calls attention to a situation, or an operating or maintenance procedure, or practice, which if not strictly corrected or observed, could result in injury or death of personnel. An example is the proximity of high voltage.

Caution

CAUTION

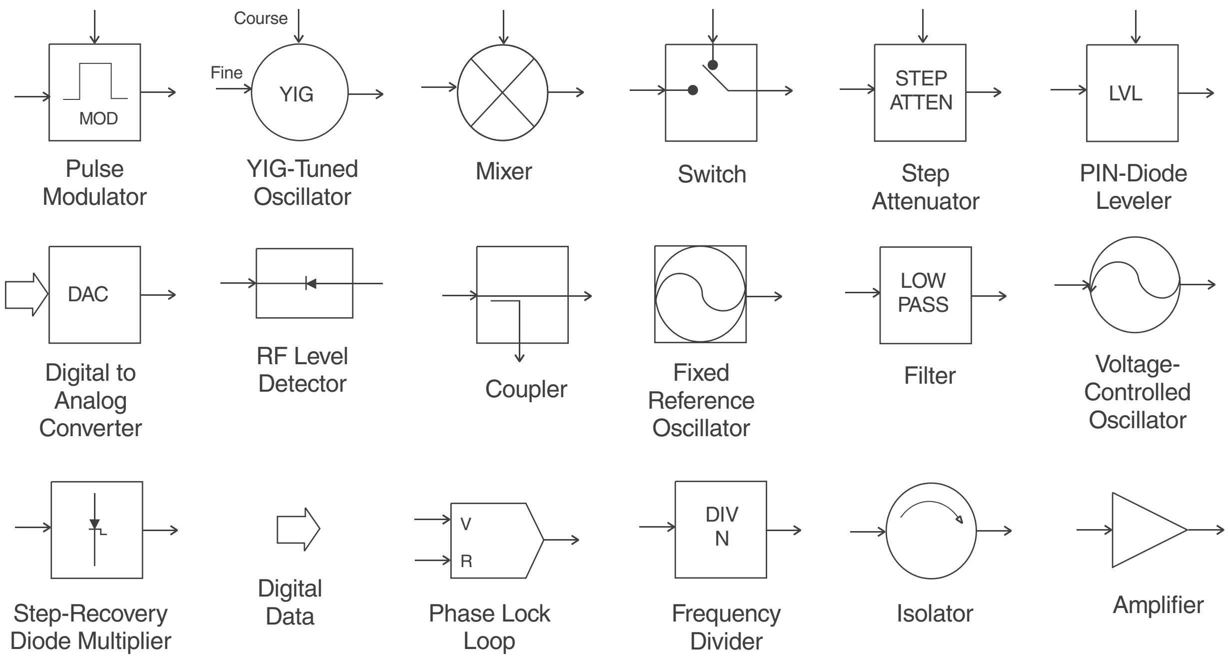
The CAUTION statement is enclosed with single lines and centered in the page. This calls attention to a situation, or an operating or maintenance procedure, or practice, which if not strictly corrected or observed, could result in temporary or permanent damage to the equipment, or loss of effectiveness.

Notes

✲ ***NOTE:*** *A NOTE Highlights or amplifies an essential operating or maintenance procedure, practice, condition or statement.*

Symbols

Block diagram symbols frequently used in the manual are illustrated below.



Record of Manual Changes

This table is provided for your convenience to maintain a permanent record of manual change data. Corrected replacement pages will be issued as Technical Publication Change Instructions, and will be inserted at the front of the binder. Remove the corresponding old pages, insert the new pages, and record the changes here.

Change Instruction Number	Change Instruction Date	Date Entered	Comments

Special Configurations

When the accompanying product has been configured for user-specific application(s), supplemental pages will be inserted at the front of the manual binder. Remove the indicated page(s) and replace it (them) with the furnished Special Configuration supplemental page(s).

1

Introduction

1.1 Description

The Giga-tronics 58542 VXI Universal Power Meter offers high accuracy with an ultra-fast reading rate and dual power inputs. Depending on the sensor used, the frequency range is from 10 MHz to 40 GHz for CW signals, and 50 MHz to 40 GHz for pulsed signals. The unit is optimized for fast measurements over the VXI bus. Maximum measurement speed is achieved with an internal data buffer, which logs a user-specified number of measurements for subsequent transfer over the VXI bus.

The 58542 is optimized for the fast measurements required by ATE systems. The 58542 diode-based sensors have response times significantly faster than thermal sensors. The high measurement speeds are available over a wider dynamic range than with thermal sensors. The power meter uses the Standard Commands for Programmable Instruments (SCPI) language. It is a message-based instrument that responds to high level ASCII character SCPI commands. The commands are parsed and interpreted by the power meter. Their standardized (English) language format makes SCPI program development fast and easy.

The power sweep calibrator incorporates a built-in thermistor-based power meter bridge. The thermistor oven stability provides a standard 1 mW power reference at 50 MHz, traceable to the National Institute of Standards and Technology (NIST). The thermistors inherent linearity produces 51 precisely controlled power levels from -30 to +20 dBm for linearizing the diodes in the power sensors.

The excellent resulting linearity of –0.5% (–0.02 dB) complements the low VSWR of the sensors and tightly specified Cal Factor uncertainty. The Cal Factors are stored in EEPROMs in each sensor so that you need only to enter a frequency of operation to the power meter to obtain frequency corrected power readings.

The 58542 power meter is a standard C-size (single-width) VXI module. It weighs 2.5 kg (5.5 lbs). Power requirements are +5 Vdc @ 800 mA, +24 Vdc @ 250 mA and -24 Vdc @ 250 mA.

The 58542 uses Giga-tronics Series 80300 Schottky diode-based sensors for fast measurements with a dynamic range of up to 90 dB, depending on the sensor. Excellent linearity of – 0.5% is assured through a built-in power sweep calibrator. Special purpose CW sensors are available, including low VSWR sensors, four high-power versions (1W, 5W, 25W and 50W), and True RMS sensors with 50 dB dynamic range. Peak power sensors are available for making true instantaneous power readings on pulsed signals such as radar and digital communications. The sensors sample pulsed RF signal amplitude directly, therefore a duty cycle correction is not required.

Refer to Appendix B in this manual for power sensor selection and specifications data. Appendix C contains technical data for options available for the 58542.

1.1.1 Items Furnished

In addition to options and/or accessories specifically ordered, items furnished with the instrument are as follows:

- Operation & Maintenance Manual, P/N 21555
- Two sets of power sensor cables

1.1.2 Items Required

A VXI mainframe which meets the power and cooling requirements of the modules is required. Two sensor cables are furnished with the 58542 to fit the dual-channel female output connectors.

1.1.3 Tools and Test Equipment

Test equipment required for calibration and testing is described in Chapter 4 of this manual.

1.1.4 Storage

Giga-tronics VXIbus modules should be stored in an environment free from excessive dust and dirt and in the temperature range of -40°C to +70°C.

1.1.5 Cooling

No special cooling is required. If the module is to be operated outside of a properly ventilated VXI frame, auxiliary air circulation is required, such as a small fan directed at the module.

1.1.6 Receiving Inspection

Use care in removing the instrument from the carton and check immediately for physical damage, such as bent or broken connectors on the front and rear panels, dents or scratches on the panels, broken extractor handles, etc. Check the shipping carton for evidence of physical damage and immediately report any damage to the shipping carrier.

Each Giga-tronics instrument must pass rigorous inspections and tests prior to shipment. Upon receipt, the instrument’s performance should be promptly checked to ensure that operation has not been impaired during shipment.

1.1.7 Safety Precautions

When installing modules into the mainframe, be sure that the connectors are properly aligned before pushing the modules into place. Apply gentle but firm pressure to insert the modules and make sure they are fully seated for proper operation.

1.1.8 Preparation for Reshipment

Follow these instructions if it is necessary to return the product to the factory.

To protect the instrument during reshipment, use the best packaging materials available. If possible use the original shipping container. If this is not possible, a strong carton or a wooden box should be used Wrap the instrument in heavy paper or plastic before placing it in the shipping container. Completely fill the areas on all sides of the instrument with packaging material. Take extra precautions to protect the front and rear panels.

Seal the package with strong tape or metal bands. Mark the outside of the package "FRAGILE — DELICATE INSTRUMENT". If corresponding with the factory or local Giga-tronics sales office regarding reshipment, please reference the full model number and serial number. If the instrument is being reshipped for repair, enclose all available pertinent data regarding the problem that has been found.

*

NOTE: *If you are returning an instrument to Giga-tronics for service, first contact Customer Service so that a return authorization number (RMA) can be assigned via e-mail at* ***repairs@gigatronics.com*** *or at 800.726.4442 (The 800 number is only valid within the US). You may also try our domestic line at 925.328.4650 or Fax at 925.328.4702.*

1.2 Performance Specifications

Performance specifications describe the 58542 warranted performance, and apply when using the Series 80300A Power Sensors. Typical performance (shown in italics) is non-warranted.

1.2.1 Range

Frequency Range	10 MHz to 40 GHz[1]
Power Range	-70 dBm to +47 dBm (100 pW to 50 Watt)[1]
Single Sensor Dynamic Range	CW Sensors: 90 dB[1] Peak Power Sensors: 40 dB, Peak 50 dB, CW[1]

1.2.2 Accuracy

Calibrator	Power Sweep calibration signal to dynamically linearize the sensors.
Frequency	50 MHz nominal
Settability	The 1 mW (0.0 dBm) level in the Power Sweep Calibrator is factory set to – 0.7% traceable to the National Institute of Standards and Technology (NIST). Measure within 15 seconds of setting calibrator to 0.0 dBm.
Accuracy	–1.2% worst case for one year, over temperature range of 5°C to 35°C
Connector	Type N(f) connector, 50 W
VSWR	<1.05 dB (Return Loss >33 dB)
System Linearity (at 50 MHz for Standard CW Sensors)	–0.02 dB over any 20 dB range from -70 to +16 dBm –0.02 dB + (0 dB, -0.05 dB/dB) from +16 to +20 dBm –0.04 dB from -70 to +16 dBm
Temperature Coefficient of Linearity	<0.1%/ °C temperature change following Power Sweep Calibration, 24-hour warm-up required.For 80350A Series Sensors, <0.3%/ °C temperature change following Power Sweep Calibration, 24-hour warm-up required.

1.2.3 Zeroing Accuracy (Standard CW Sensors)

Zero Set	*<–50 pW*[2]
Zero Drift	*<–100 pW* during 1 hour[2]
Noise	*<–50 pW* measured over any 1 minute interval[2]
Averaging	Auto-averaging or user-selectable averaging from 1 to 512 readings per measurement

Notes:

1. Depending on sensor used.
2. Specified performance applies with maximum averaging and 24-hour warm-up at constant temperature.

Figure 1-1 illustrates the instrument Linearity plus Typical Noise and Zeroing Error vs Input Power. (The X-axis scale is sensor dependent)

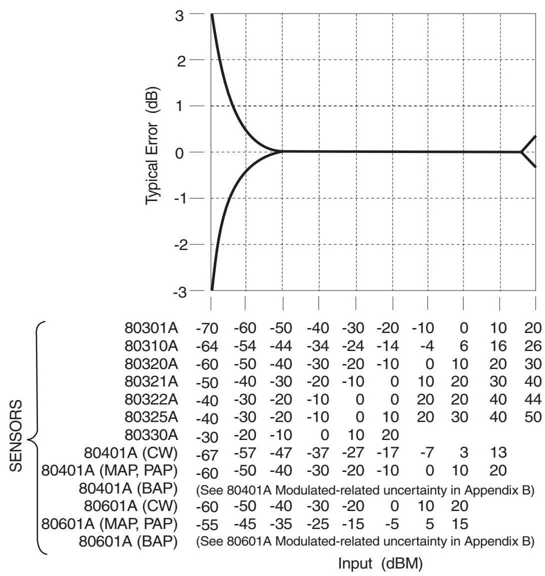


Figure 1-1: Instrument Linearity

1.2.4 Meter Functions

dB Offset and Relative: Allows both relative readings and offset readings. Power readings can be offset by -99.999 to +99.999 dB to account for external loss/gain.

Configuration Storage Registers: Up to ten instrument configurations can be stored and recalled from non-volatile memory for fast configuration changes.

Power Measurements: Any two of the following channel configurations, simultaneously: 1, 2, 1/2, 2/1, 1-2, 2-1.

1.2.5 Measurement Speed

Measurement speed increases significantly using the meter's data storage capabilities. Storing data in memory for later downloading to the controller reduces word serial protocol and protocol conversion overhead. Up to 128,000 readings can be buffered. The measurement rate depends on several factors, including controller speed and number of averages. Burst Mode speed does not include bus communication time. The following lists typical maximum measurement rates for CW power sensors or Series 80340 Peak Power Sensors.

Normal Mode *Non-Buffered*	Swift Mode *Buffered Data*	Burst Mode *Buffered Data*
55 rdgs/sec	150 rdgs/sec	5100 rdgs/sec

Individual data points are read immediately after measurement in the Normal Mode. Swift Mode allows triggering of individual data points, and stores the data in the 58542 memory. Burst Mode also buffers measurement data: measurement timing of individual data points is controlled by setting the time interval (0.001 to 5.000 sec) between the data points following a single group burst trigger event.

1.2.6 Inputs/Outputs

Analog Output	Provides an output voltage (at the Analog Out BNC) that is configurable from -10 to +10 V from either Channel 1 or Channel 2 in either Lin or Log units[1]
Accuracy	*1.0% –32 mV*, -10 to +10 V
Linearity	<0.3%
Trigger Input	Connects EXT trigger (at the EXT TRIG BNC). TTL level input signal for fast reading of buffered data modes
Voltage proportional to Frequency (in GHz)	Automated Cal Factor correction. Input the analog VpropF signal level from the microwave signal source to the V_{PROP}F IN BNC[1]
Input Range	0 to 10 V
Accuracy	*1.0% –32 mV* (14 bit) (0.6 mV resolution)

Note:

1. Operates in Normal Mode only.

1.2.7 Power Requirements

+5 Vdc @ 800 mA
+24 Vdc @ 250 mA
-24 Vdc @ 250 mA

1.2.8 Environmental Specifications

Temperature Range

Operating	0°C to 50°C (32°F to 122°F)
Storage	-40°C to 70°C (-40°F to 158°F)

Physical Characteristics

Dimensions	C-size, single slot VXI standard 30 mm (1.2 in) wide, 234 mm (9.2 in) high, 340 mm (13.4 in) deep
Weight	2.5 kg (5.5 lbs)

Accessories Included

Two detachable sensor cables.

2

Operation

2.1 Preparation for Use

This chapter describes how to operate the 58542 VXI Universal Power Meter. The first part of the chapter explains how to set up and install the unit. This is followed by operating procedures using the General Purpose Interface Bus (GPIB) command reference, with the Standard Commands for Programmable Instruments (SCPI) command language.

The following topics are presented in this chapter:

- Initial Setup
- Installation
- Sensor Precautions
- Operation
- Error Messages

2.2 Installation

Before installing the power meter, ensure that the logical address and data transfer bus arbitration have been set according to the procedures in Section 2.3.

The power meter installs into any slot of a VXI mainframe except slot 0 (zero). When inserting the instrument into the mainframe, rock it gently back and forth to fully seat the connectors into the backplane receptacles. The ejectors will be at right angles to the front panel when the instrument is properly seated. The two captive screws above and below the ejectors secure the instrument into the chassis.

The power meter contains three printed circuit assemblies: (1) The Analog board, (2) VXI Processor board, and (3) Digital board.

The Analog board contains the front panel connectors, the Power Sweep Calibrator (beneath the top metal cover), and the dual measurement channels (beneath the lower metal cover). The memory backup battery is attached to this lower metal cover.

The VXI Processor board contains the 68000 processor, connection to the VXIbus backplane, and two EEPROMS containing communications software. The operational software must be compatible with the EEPROMs on the Digital board. The EEPROMs will be replaced whenever operational software upgrades are performed.

The Digital board connects to the VXI backplane for access to TTL triggering and the main operational software.

2.3 Initial Setup

The logical address and data transfer bus arbitration must be set up in the power meter before installing the unit in a VXI mainframe and applying power. The following procedures define how to complete the initial setup.

2.3.1 Logical Address

The VXI chassis Resource Manager identifies each unit in the system by it's logical address. The VXI logical address can range from 0 to 255. Addresses 0 and 255 are reserved for special functions: Address 0 identifies the Resource Manager (slot and controller); address 255 permits the Resource Manager to dynamically address the unit based on the chassis VXI slot.

To change the logical address, set the respective sections on the eight position DIP switch. The switch is accessible after removing the right cover (see Figure 2-1).

The address is set with binary values of 0 to 255. Switch position 1 is the least significant bit of the address. Figure 2-1 illustrates logical address values of 3 (binary 00000011) and 255 (binary 11111111). Giga-tronics ships the power meter with a logical address of 255 for dynamic configuration.

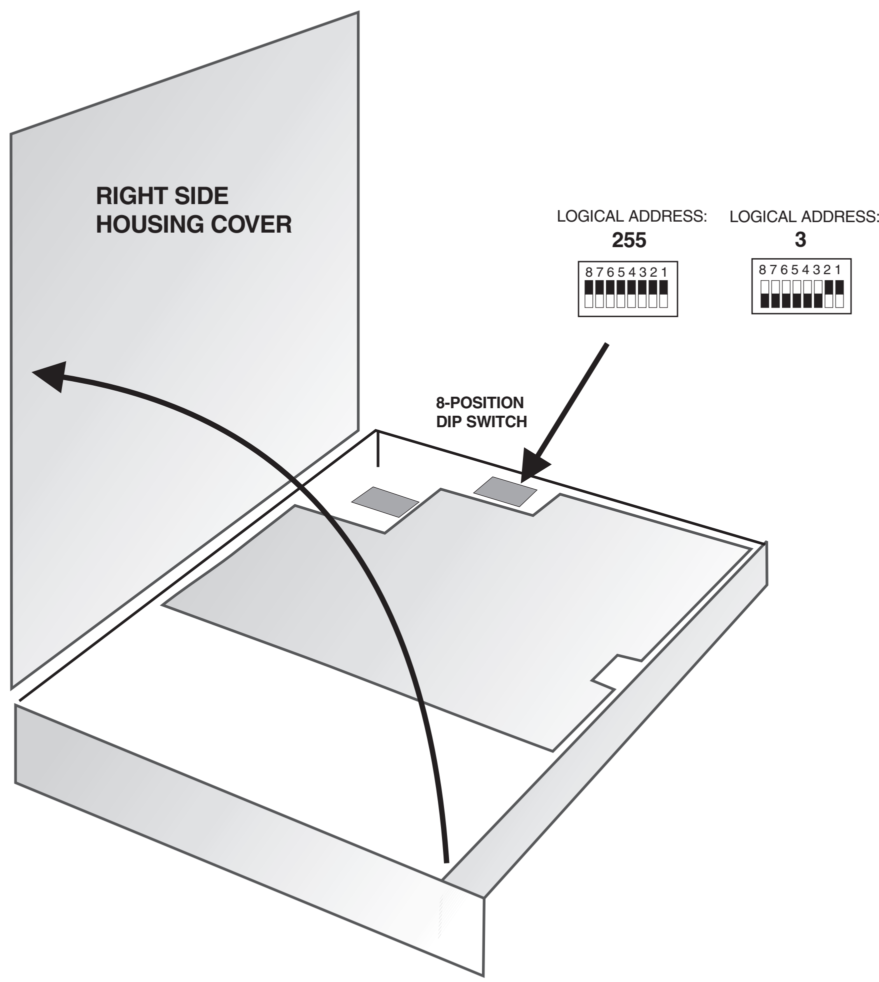


Figure 2-1: Setting the Logical Address

2.3.2 Data Transfer Bus Arbitration

The power meter has VMEbus Mastership capability. When enabled, it sends responses and events as signals (software interrupts) to its Commander Signal Register. The power meter cannot drive the interrupt lines.

The power meter is configured as a level 3 requester by the factory. The level 3 Bus Request and Bus Grant lines (BR3*, BG3IN* and BG3OUT*) are used. The other Bus Grant lines are daisy-chained by jumpers (see Figure 2-2).

The VMEbus specifications describe three priority schemes: (1) Prioritized, (2) round-robin, and (3) single level. Prioritized arbitration assigns the bus according to a fixed priority scheme where each of four bus lines has a priority from highest (BR3*) to lowest (BR0*). Round-robin arbitration assigns the bus on a rotating basis. Single level arbitration accepts requests only on BR3*.

The jumpers must be changed if a different requester level is required. Figure 2-2 will aid in reconfiguring the power meter to a new level. Refer to the VMEbus specification for more information on data transfer bus arbitration.

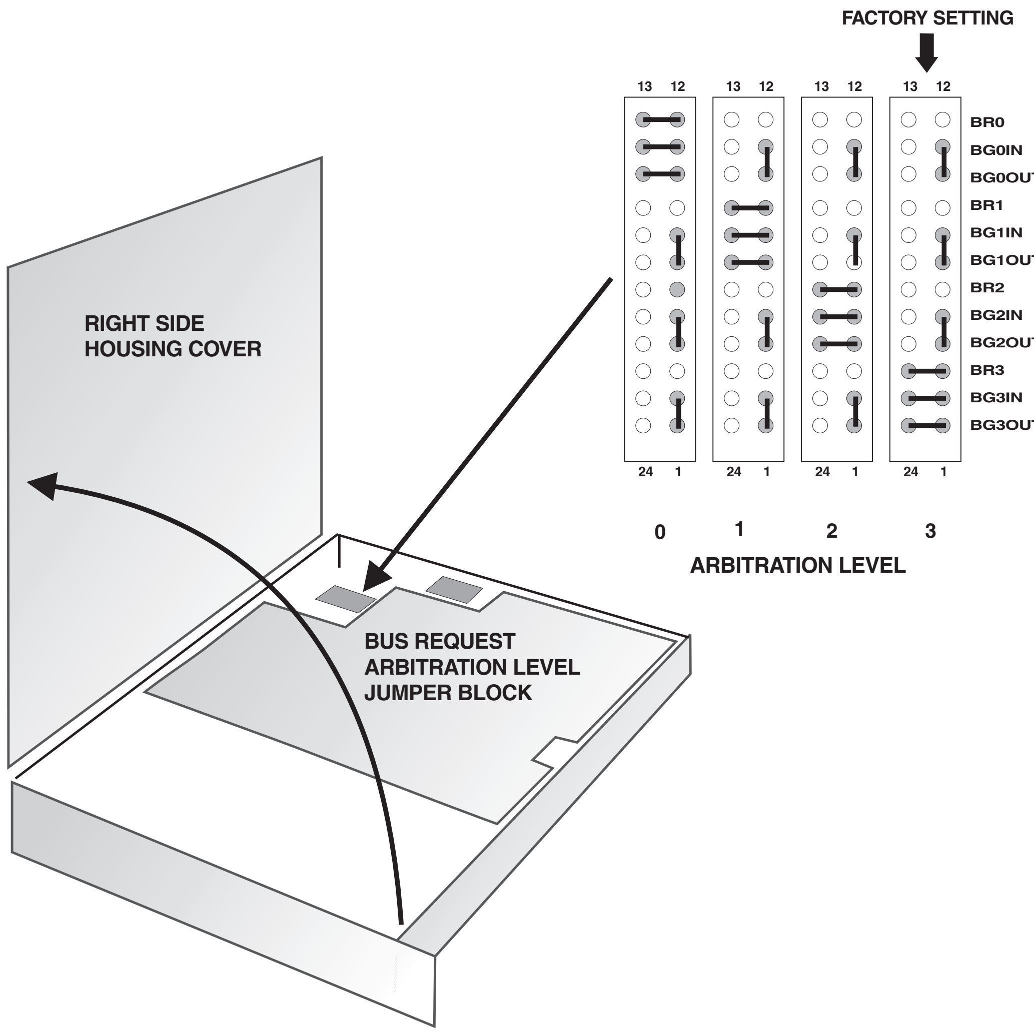


Figure 2-2: Default Bus Arbitration Settings

2.4 Sensor Precautions

Sensors used with the 58542 are configured in metal housings for superior mechanical performance as well as excellent shielding.

CAUTION

When connecting the sensors to other devices or components, the body of the sensor should never be turned to tighten the RF connection. Mechanical damage to the connector can result if improperly handled when connecting the sensors. Scratched or damaged connector mating surfaces can lead to inaccurate measurements.

If a sensor is connected to CW or peak power devices with power output in excess of +23 dBm (200 mW), degradation or destruction of the diode can occur.

Diodes degraded or destroyed in this manner will not be replaced under warranty. Destructive signal levels are higher for High Power, $True_{RMS}$, and Low VSWR sensors.

2.5 Operation

2.5.1 SCPI Command Interface

This section details operation of the power meter using the SCPI interface commands. A SCPI Command Reference is presented in Table 2-2 and the sections that follow.

The power meter is compatible with the Standard Commands for Programmable Instruments (SCPI) standard. SCPI promotes consistency in its definition of a common instrument control and measurement command language. The structured approach of the SCPI standard offers test system design engineers a number of system integration advantages that achieve considerable efficiency gains during control program development.

SCPI compatible instrument commands are structured from a common functional organization or model of a test instrument (see Figure 2-3). Most of the power meter configuration and measurement functions fall within the Measurement Function Block and the Trigger Subsystem of the SCPI instrument model.

The 58542 uses the SENSe Subsystem of the Measurement Function Block to implement commands that apply specifically to the individual power sensors; sensor 1, and sensor 2. For example, the SENSe2:CORRection:OFFSet command corrects for the attenuation of a signal that passes through an attenuator or coupler before it is measured by power sensor 2. Figure 2-3 illustrates the SCPI subsystem model.

*

NOTE: *Throughout this manual, some commands will be in both upper and lowercase, such as CALCulate and MEMory. The uppercase is the required input. The whole word can be used if desired.*

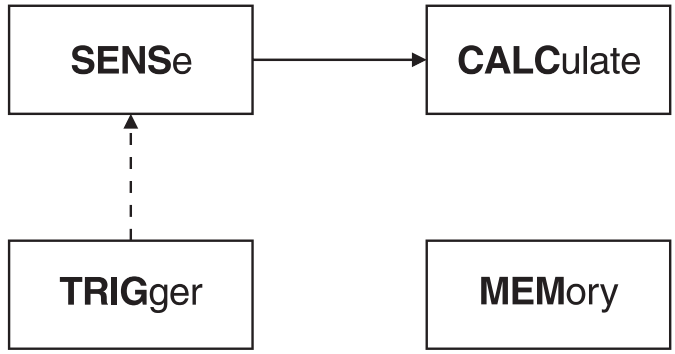


Figure 2-3: SCPI Subsystem Model

2.5.2 Sensor and Channel Configuration

The CALCulate Subsystem of the Measurement Function Block contains commands that define the form of the measured data from sensor 1 and sensor 2. CALCulate commands define the configuration of the two Software Calculation Channels (channels). For example, the CALC1:POW 1 command configures channel 1 to report the power level as measured by sensor 1. CALC2: RAT 2,1 configures channel 2 to report the ratio of power levels, sensor 2 over sensor 1.

2.5.3 Measurement TRIGgering

The power meter uses the TRIGger Subsystem to trigger measurements in two different operational modes - a normal mode which maximizes the instrument's functionality, and swift and burst modes that maximize the power measurement rate.

2.5.4 MEMory Functions

The MEMory commands control the configuration of the automated Voltage-Proportional-to-Frequency ($V_{PROP}F$), sensor Cal Factor correction, and the Analog Output on the front panel. Each of these connectors is used with external devices. The $V_{PROP}F$ can be configured to match the $V_{PROP}F$ output of your microwave source. The Analog Output is used with a variety of devices including chart recorders, oscilloscopes, voltmeters, and microwave source leveling inputs.

2.5.5 IEEE 488.2 Commands

Consistent with SCPI compliance criteria, the power meter implements all the common commands declared mandatory by IEEE 488.2. These commands are listed in Table 2-1.

Table 2-1: VXI GPIB Command Syntax

Mnemonic	Name
*CLS	Clear Status Command
*ESE	Standard Event Status Enable Command
*ESE?	Standard Event Status Enable Query
*ESR?	Standard Event Status Register Query
*IDN?	Identification Query
*OPC	Operation Complete Command
*OPC?	Operation Complete Query
*RST	Reset Command
*SRE	Service Request Enable Command
*SRE?	Service Request Enable Query
*STB?	Read Status Byte Query
*TST?	Self-Test Query
*WAI	Wait-to-Continue Command

2.5.6 DIAGnostic Commands

DIAGnostic commands are used for a variety of instrument specific maintenance and calibration functions. Unless you are performing instrument calibration functions, it is unlikely that you will need to use the DIAGnostic command sets. For calibration laboratory metrology professionals, the DIAG commands will allow you to completely automate instrument and sensor calibration functions. The commands program EEPROMs inside the meter and individual sensors. If desired, a password function is incorporated to prevent unauthorized personnel from altering calibration information. Table 5-1 in Chapter 5 lists available Diagnostic commands.

2.5.7 CALCulate Subsystem Commands

CALCulate: commands specify and query the configuration of power measurement channels, known in SCPI references as Software Configuration Channels, and in this manual as “channels.” (For sensor specific configuration and measurement function control, see Section 2.5.8).

The query form of CALC#? with the appropriate modifier (1 or 2) inserted ahead of the ? will return the current configuration status for that channel. This can be used to verify configuration commands or return current status information following data acquisition or power measurements.

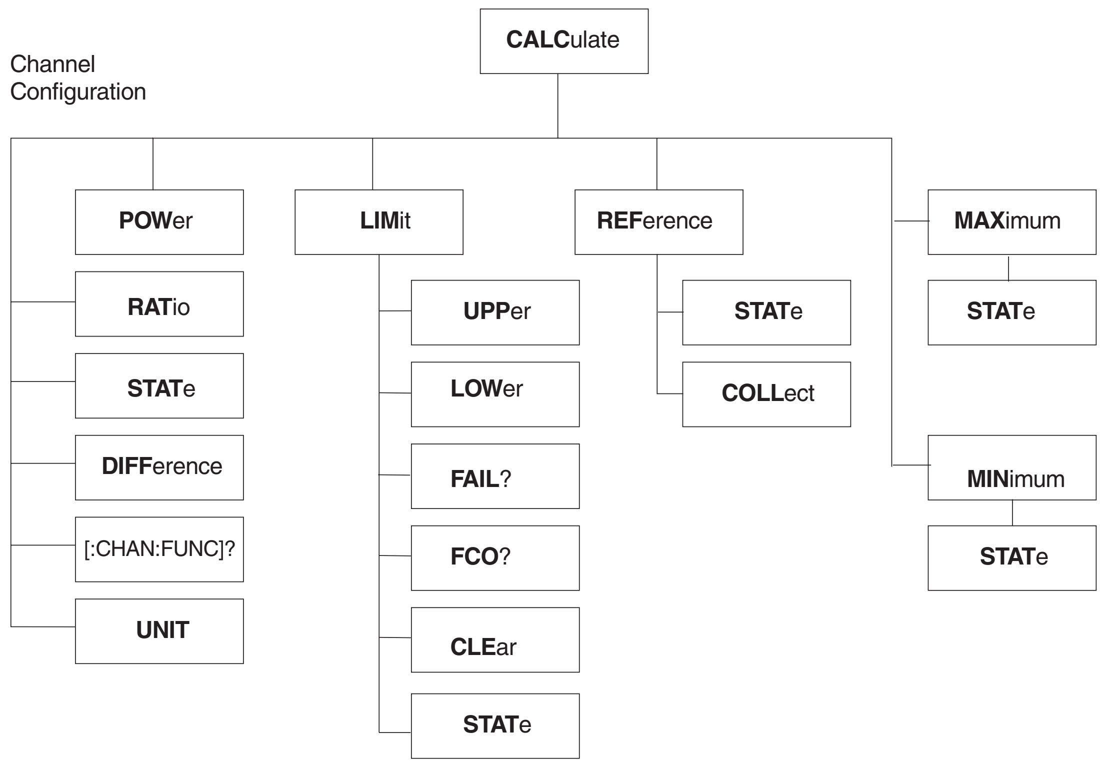


Figure 2-4: CALculate Subsystem Command Tree

Limit Lines are set on a channel basis. The LIMit commands set limits, monitor the number of violations, and allow the violation counter to be cleared.

The REFerence command allows channel based offset values. For example, using CALC#:REF:COLL automatically converts the inverse of the current channel measurement value to an offset - simplifying the 1 dB compression testing of amplifiers.

MIN and MAX commands monitor deviation of measured values over a user controllable time period.

The two software calculation channels can individually and simultaneously perform the internal instrument functions that calculate final measurement data for output over the VXI bus to the controller. The final measurement data is calculated from SENSe Subsystem processed sensor data as well as the CALC subsystem channel configuration data.

This means that only two measurement configurations can be obtained from the 58542 simultaneously. For example, the controller can obtain measurements for sensor 1 plus sensor 2/sensor 1 simultaneously, but not sensor 1 plus sensor 2 plus sensor 2/sensor 1 simultaneously.

2.5.8 SENSe Subsystem Commands

The SENSe subsystem configuration commands, illustrated in Figure 2-5, apply specifically to individual sensors. These commands alter the value of the measured power level according to the sensor's characteristics. For example, measured power levels can be offset for attenuators or couplers in the measurement path so that the power data reading reflects the power level at the measurement point of interest.

Use SENSe for:

- Averaging power measurements in the 58542 rather than the controller
- Offsetting power measurements for attenuation or amplification
- Entering the operating frequency of the measured signal (automatically computes and applies sensor specific Calibration Factor corrections) which compensates for sensor frequency response characteristics
- Controlling Peak Power Sensor triggering

SENSe subsystem commands control functions that are related directly to the individual power sensors. For example, these commands control items that would not apply to numerical alteration of a ratio measurement of Sensor1/Sensor2. Controls that would apply to that type of a configuration are channel functions, not sensor functions, and would therefore be located in the CALCulate subsystem.

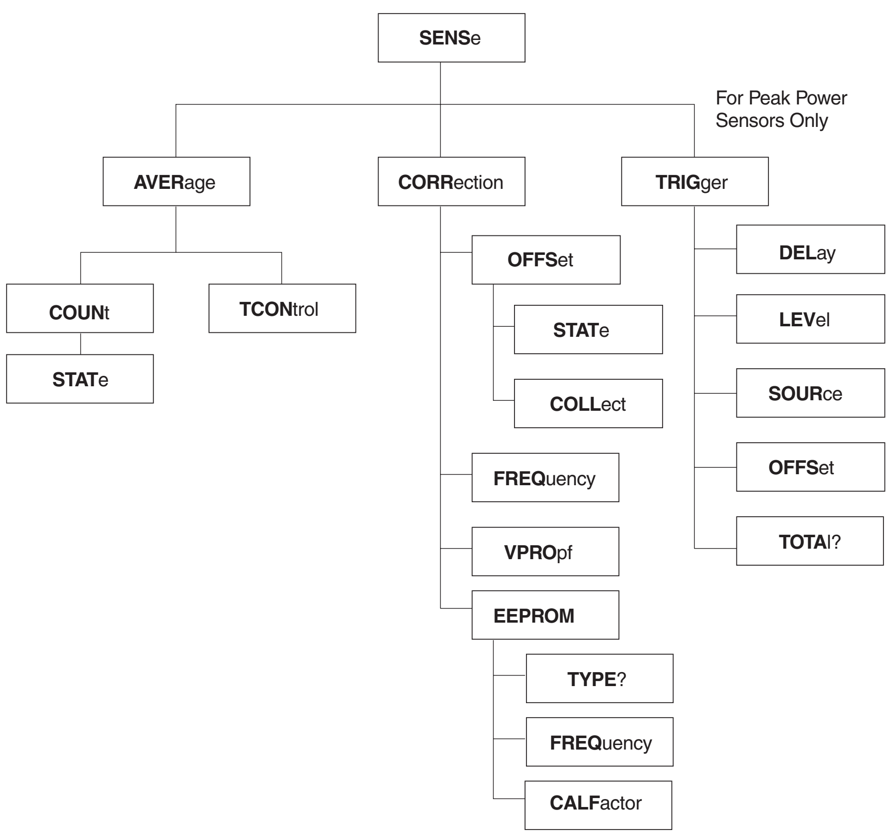


Figure 2-5: SENSe Subsystem Command Tree

SENSe:AVERage functions control the number of data samples for each measurement and the manner in which those numbers are accumulated. COUNt determines the averaging number or AUTO-averaging. TCONtrol determines whether each new sample is added to previous COUNt # of samples or if COUNt # of samples are taken each time the 58542 instrument is triggered. Please note that the SENSe:TRIGger commands are not instrument triggers, but Peak Power Sensor configuration controls.

SENSe:TRIGger functions apply only to Giga-tronics Peak Power Sensors. The DELay and LEVel functions of these Peak Power Sensor controls apply to the 80350A Peak Power Sensors, not the 80340 Peak Power (Triggerable Pulse) Sensors. The AVERage and CORRection commands apply to all 80300 Series CW & Peak Power Sensors.

STATe ON OFF controls for COUNt and DELay, above, are not shown.

2.5.9 TRIGger Subsystem Commands

The TRIGger Subsystem is divided into two sections; Instrument Measurement Event Triggering, and Special Triggering Configuration commands for the fast reading buffered data modes, Burst Mode and Swift Mode. The TRIGger command tree is illustrated in Figure 2-6.

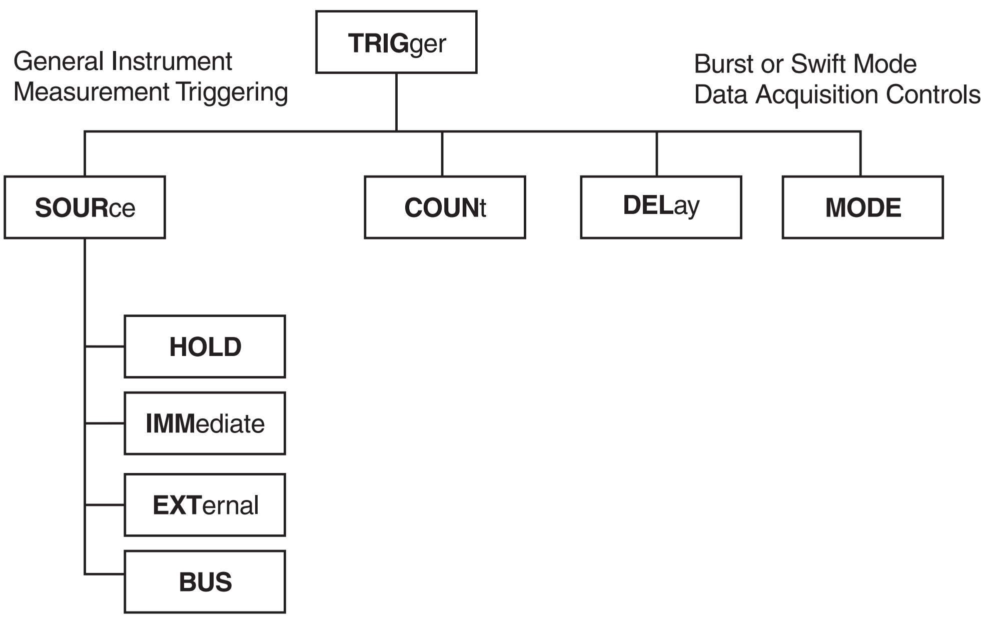


Figure 2-6: TRIGger Subsystem Command Tree

The query form of these commands, TRIGger? with the appropriate modifier inserted ahead of the ?, will return the instrument's current configuration status. This can be used to verify triggering configuration or return status information following command errors which are commonly caused by using illegal configuration commands during SWIFt or BURSt Modes.

SOURce:IMMediate triggering allows the 58542 to control measurement triggering; this is the default configuration. EXTernal triggering is performed using a TTL signal into the front panel connector. There is provision for connecting this trigger signal to the backplane TTL Triggering, but the ASCII software command syntax remains EXT. BUS allows software controlled triggering.

COUNt refers to the number of data points to store in the meter's 5000 reading buffer (128,000 with option 02) before the measurement data is requested by the controller. DELay controls the time interval between Burst Mode data samples, and the MODE command controls whether the data is taken after receipt of the instrument trigger or if data is collected (in a FIFO buffer) immediately preceding receipt of an instrument trigger.

2.5.10 GPIB Command Syntax

The following conventions are used with the GPIB commands in this manual. Throughout this manual, some commands will be in both upper- and lowercase, such as CALCulate and MEMory. The uppercase is the required input. The whole word can be used if desired. Table 2-2 lists in alphabetical order all of the VXI GPIB commands supported by the power meter. A typical example is shown, the basic function performed by that command is given, and the page numbers of this chapter where descriptions of each command can be located.

2.5.10.1 Commands in Brackets [ ]

Commands and command separators within brackets, such as [COMMand:], are optional. These portions of the commands may be used in your program command strings, but are not required for proper operation of the power meter.

2.5.10.2 Programmer Selectable Parameters < >

Command descriptions enclosed in angle brackets (< and >) show the syntax placement of configurable or settable parameters. A description of the necessary parameter and the range of values or mnemonics which are valid for that parameter are located enclosed in angle brackets.

2.5.10.3 Italics in Syntax Descriptions

Some command syntax descriptions show certain words in italics such as *space* and *comma* to indicate where the character must be included within a command string.

2.5.10.4 Query Format ?

Except where specifically noted, all query commands are formed by adding a question mark (?) to the command header. Be sure to omit command parameters when you are using the query format. Some commands have only a query format. With the exception of the CALibration queries, sending query commands will not change the status of the power meter. The CALibration1? and CALibration1:ZERO? queries automatically begin the power meter’s calibration and zeroing process, respectively.

2.5.10.5 Linking Command Strings

The 58542 uses ASCII strings for commands. When sending more than one command in a single string, a semi-colon must be used as a delimiter between commands. No spaces or other characters are necessary. Use only a semi-colon (;) to link commands in a string.

2.5.10.6 Measurement Data Output Format

The examples shown in this chapter are written in HTBasic™ format. Different languages will use different commands, but the string sent or received will always be the same. In HTBasic, the OUTPUT command sends a string to the GPIB bus. The number or variable after the word OUTPUT is the GPIB address of the power meter.

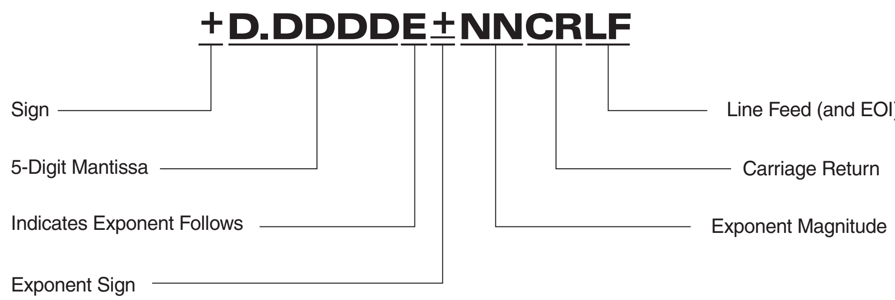


Figure 2-7: Measurement Data Output Format

™HTBasic is a trademark of TransEra Corporation

Table 2-2: VXI GPIB Command Syntax

Command Syntax Typical Example	Function	Ref. Page
ABORt **OUTPUT @Pwr_mtr;ABOR**	Halts measurement & triggering	2-60
CALCulate<channel 1 or 2>:DATA? **OUTPUT @Pwr_Mtr;CALC1:DATA?** Response is comma delimited data. Number of points equals # from TRIG:COUN #. 0<#<128000	Query channel 1 & 2 burst mode data	2-34
CALCulate<channel 1 or 2>[:CHANnel]:DIFFerence*space***<sensor 1 or 2***comma***sensor 2 or 1>** **OUTPUT @Pwr_Mtr;CALC1:DIFF 2,1**	Configures channel 1 to measure Sensor 2 minus Sensor 1 power	2-28
CALCulate<channel 1 or 2>[:FUNCtion]? **OUTPUT @Pwr_mtr;CALC2?** Response is: POW 1, POW 2, RAT 2,1, RAT 2,1, DIFF 1,2, or DIFF 2,1	Query channel 2 sensor mode	2-30
CALCulate<channel 1 or 2>:LIMit:CLEar[:IMMediate] OUTPUT @Pwr_mtr;CALC1:LIM:CLE	Reset channel 1 limit violation indicator to 0	2-55
CALCulate<channel 1 or 2>:LIMit:FCOunt? **OUTPUT @Pwr_mtr;CALC1:LIM:FCO?** Response is a single number - the number of times the limit lines were exceeded	Query number of channel 1 limit failures	2-55
CALCulate<channel 1 or 2>:LIMit:FAIL? **OUTPUT @Pwr_mtr;CALC1:LIM:FAIL?**	Check for channel 1 limit line violation (Output 0 = OK; 1 = fail)	2-56
CALCulate<channel 1 or 2>:LIMit:LOWer*space***<numeric value in dB from -299.99 to 299.99>** **OUTPUT @Pwr_mtr;CALC1:LIM:LOW -50.0**	Set channel 1 lower limit line to -50 dBm	2-56
CALCulate<channel 1 or2>:LIMit:LOWer? **OUTPUT @Pwr_mtr;CALC1:LIM:LOW?** Response is a single number - the lower limit setting	Query channel 1 lower limit line setting	2-56
CALCulate<channel 1 or 2>:LIMit:STATe*space***<ON or OFF>** **OUTPUT @Pwr_mtr;CALC1:LIM:STAT ON**	Enable channel 1 upper and lower limit checking. (0 = off; 1 = on)	2-56
CALCulate<channel 1 or 2>:LIMit:STATe? **OUTPUT @Pwr_mtr;CALC1:LIM:STAT?** Response is 1 for ON or 0 for OFF	Query channel 1 ON or OFF status	2-56
CALCulate<channel 1 or 2>:LIMit:UPPer*space***<numeric value in dB from -299.99 to 299.99>** **OUTPUT @Pwr_mtr;CALC1:LIM:UPP 17.0**	Set channel 1 upper limit line to 17 dBm	2-57
CALCulate<channel 1 or 2>:LIMit:UPPer? **OUTPUT @Pwr_mtr;CALC1:LIM:UPP?** Response is a single number - the upper limit setting	Query channel 1 upper limit line setting	2-57
CALCulate<channel 1 or 2>[:CHANnel]:POWer*space***<sensor 1 or 2>** **OUTPUT @Pwr_Mtr;CALC2:POW 1**	Configures channel 2 to measure Sensor 1 power	2-28
CALCulate<channel 1 or 2>[:CHANnel]:RATio*space***< sensor 1 or 2***comma***sensor 2 or 1>** **OUTPUT @Pwr_Mtr;CALC1:RAT 2,1**	Configures channel 1 to measure Sensor 2 over Sensor 1 power	2-28
CALCulate<channel 1 or 2>:MODE*space* **<NORMal, BURSt or SWIFt Mode Selection>** **OUTPUT @Pwr_mtr;CALC2:MODE SWIF** ***CAUTION:*** This command sets the operating mode for both channels regardless of CALC# 1 or 2	Set measurement mode to SWIFt	2-35
CALCulate<channel 1 or 2>:MODE? **OUTPUT @Pwr_mtr;CALC1:MODE?** Response is a NORM, BURS or SWIF	Query channel 1 measurement mode	2-35
CALCulate<channel 1 or 2>:MAXimum:STATe*space***<ON or OFF>** **OUTPUT @Pwr_mtr;CALC2:MAX:STAT ON**	Enable channel 2 max. value monitoring	2-54

Table 2-2: VXI GPIB Command Syntax (Continued)

Command Syntax Typical Example	Function	Ref. Page
CALCulate<channel 1 or 2>:MAXimum:STATe? **OUTPUT @Pwr_mtr;CALC2:MAX:STAT?** Response is 1 for ON or 0 for OFF	Query channel 2 max. mon. ON/OFF status	2-54
CALCulate<channel 1 or 2>:MAXimum[:MAGnitude]? **OUTPUT @Pwr_mtr;CALC2:MAX?** Response is highest power reading since CALC2:MAX:STAT ON was sent	Query channel 2 maximum value in dBm	2-54
CALCulate<channel 1 or 2>:MINimum:STATe*space***<ON or OFF>** **OUTPUT @Pwr_mtr;CALC1:MIN:STAT ON**	Enable channel 1 min. value monitoring	2-54
CALCulate<channel 1 or 2:MINimum:STATe? **OUTPUT @Pwr_mtr;CALC1:MIN:STAT?** Response is 1 for ON or 0 for OFF	Query channel 1 min. mon. ON/OFF status	2-54
CALCulate<channel 1 or 2>:MINimum[:MAGnitude]? **OUTPUT @Pwr_mtr;CALC1:MIN?** Response is lowest power reading since CALC2:MIN:STAT ON was sent	Query channel minimum value in dBm	2-54
CALCulate<channel 1 or 2>:REFerence:STATe*space***<ON or OFF>** **OUTPUT @Pwr_mtr;CALC2:REF:STAT ON**	Activate level reference for relative measurements	2-45
CALCulate<channel 1 or 2>:REFerence:STATe? **OUTPUT @Pwr_mtr;CALC2:REF:STAT?** Response is 1 for ON or 0 for OFF	Query ON or OFF status	2-45
CALCulate<channel 1 or 2>:REFerence[:MAGnitude]<dB Offset value from -299.999 to 299.999> OUTPUT @Pwr_mtr;CALC2:REF -30.11 OUTPUT @Pwr_mtr;CALC2:REF 0.00	Set channel 2 reference offset value in dB Reset channel 2 Relative meas. operation	2-45
CALCulate<channel 1 or 2>:REFerence[:MAGnitude]? **OUTPUT @Pwr_mtr;CALC1:REF?** Response is a single number reporting the reference value	Query current channel 1 reference value	2-45
CALCulate<channel 1 or 2>:REFerence:COLLect **OUTPUT @Pwr_mtr;CALC2:REF:COLL !** Current Rdg.==> Ref level	Take channel 2 reading as reference value. Reset with CALC:REF 0.0	2-45
CALCulate<channel 1 or 2>:STATe*space***<ON or OFF>** **OUTPUT @Pwr_mtr;CALC1:STAT OFF**	Disable channel 1 measurement	2-30
CALCulate<channel 1 or 2>:STATe? **OUTPUT @Pwr_mtr;CALC1:STAT?** Response is 1 for ON or 0 for OFF	Query channel 1 measurement status	2-30
CALCulate<channel 1 or 2>:UNIT[:POWer]*space***<data units selection, DBM or Watt>** **OUTPUT @Pwr_mtr;CALC1:UNIT W** **OUTPUT @Pwr_mtr;CALC1:UNIT DBM**	Selects channel 1 linear units in Watts Selects channel 1 Log Units in dBm	2-30
CALCulate<channel 1 or 2>:UNIT[:Power]? **OUTPUT @Pwr_mtr;CALC1:UNIT?** Response is DBM or W	Query channel 1 Units Configuration	2-30
CALibrate<sensor 1 or 2>[:SENSor] **OUTPUT @Pwr_mtr;CAL1**	Calibrate Sensor 1	2-18
CALibrate<sensor 1 or 2>[:SENSor]? **OUTPUT @Pwr_mtr;CAL1?** Response is 1 or 0 immediately upon power sweep calibration failure or completion, respectively	Cal. Sensor 1 & return Pass/Fail status (0 = pass; 1 = fail)	2-18
CALibrate<sensor 1 or 2>:ZERO **OUTPUT @Pwr_mtr;CAL1:ZERO**	Zero Sensor 1	2-19
CALibrate<sensor 1 or 2>:ZERO? **OUTPUT @Pwr_mtr;CAL1:ZERO?** Response is a 1 or 0 immediately upon power sensor zeroing failure or completion, respectively	Zero Sensor 1 & return Pass/Fail status	2-19
CALibrate<sensor 1 or 2>:STATe? **OUTPUT @Pwr_mtr;CAL1:STAT?** Response is 1 if sensor is calibrated or 0 if sensor is not calibrated	Query sensor for calibration status	2-18

Table 2-2: VXI GPIB Command Syntax (Continued)

Command Syntax Typical Example	Function	Ref. Page
***CLS** **OUTPUT @Pwr_mtr;*CLS**	Clear SRQ and status byte registers	2-49
ESEspace***<event status register value, 0 to 255>** **OUTPUT @Pwr_Mtr;*ESE 16**	Enable bit 4 of event status register mask	2-49
***ESE?** **OUTPUT @Pwr_Mtr;*ESE?** Response is the sum of the enabled bit numbers n from 0 to 7 expressed as the sum of the enabled 2*n*	Query currently enabled bits of event status register mask	2-49
***ESR?** **OUTPUT @Pwr_Mtr;*ESR?** Response is the sum of the active bit numbers n from 0 to 7 expressed as the sum of the enabled 2*n*	Return event status register value	2-49
FETCh<channel 1 or 2>? **OUTPUT @Pwr_Mtr;FETC1?**	Read previously triggered channel 1 power, Normal, Swift or Burst mode data	2-20
***IDN?** **OUTPUT @Pwr_Mtr;*IDN?** Response is Giga-tronics 58542,0,#.## where #.## is the software version	Query instrument mfgr & model #	2-63
INITiate[:IMMediate] **OUTPUT @Pwr_Mtr;INIT**	Initiate instrument triggering cycle	2-27
INITiate:CONTinuous*space***<ON or OFF>** **OUTPUT @Pwr_mtr;INIT:CONT ON**	58542 self-triggers continuously	2-27
INITiate:CONTinuous? **OUTPUT @Pwr_mtr;INIT:CONT?** Response is 1 for ON or 0 for OFF	Query ON or OFF status;	2-27
MEASure<channel 1 or 2>[:SCALar:POWer]? **OUTPUT @Pwr_Mtr;MEAS2?**	Configure triggering and measure Channel 2 power in NORMal mode	2-20
MEMory[:TABLe]:CHANnel*space***<channel 1 or 2>** **OUTPUT @Pwr_mtr;MEM:CHAN 2**	Set analog out to channel 2	2-57
MEMory[:TABLe]:CHANnel? **OUTPUT @Pwr_mtr;MEM:CHAN?** Response is channel 1 or channel 2	Query analog out channel number	2-57
MEMory[:TABLe]:FREQuency < Start frequency> **OUTPUT @Pwr_mtr;MEM:FREQ 1e9**	Set v_prop_f start frequency to 1 GHz	2-33
MEMory[:TABLe]:FREQuency? **OUTPUT @Pwr_mtr;MEM:FREQ?** Response is the frequency value you set that corresponds to your source V_{PROP}F output at 0.0V	Query v_prop_f start frequency	2-33
MEMory[:TABLe]:POWer*space***<start value from -80 to +20 dBm>***comma***<stop value from -80 to +20 dBm>** **OUTPUT @Pwr_mtr;MEM:POW -70,20**	Set analog out power range (in dBm) from -70 to 20 dBm	2-58
MEMory[:TABLe]:POWer? **OUTPUT @Pwr_mtr;MEM:POW?** Response is two dBm values separated by a comma indicating analog output range	Query analog out power range	2-59
MEMory[:TABLe]:SELect*space***<ANALOGout,VPROPF1, VPROPF2>** **OUTPUT @Pwr_mtr;MEM:SEL VpROPF1**	Select memory table VpROPF1 for editing in following lines	2-58
MEMory[:TABLe]:SELect? **OUTPUT @Pwr_mtr;MEM:SEL?** Response is the table currently editable ANALOGout, VPROPF1 or VpROPF2	Query selected memory table	2-58
MEMory[:TABLe]:SLOPe*space***<Volts per Hz>** **OUTPUT @Pwr_mtr;MEM:SLOP 1e-9**	Set v_prop_f slope in 1V/GHz	2-33
MEMory[:TABLe]:SLOPe? **OUTPUT @Pwr_mtr;MEM:SLOP?** Response is the voltage to frequency value you set that corresponds to your source VpropF output	Query v_prop_f slope value	2-33
MEMory[:TABLe]:UNIT*space***<Units for analog output configuration, DBM or Watt>** **OUTPUT @Pwr_mtr;MEM:UNIT DBM**	Set analog out power unit	2-58

Table 2-2: VXI GPIB Command Syntax (Continued)

Command Syntax Typical Example	Function	Ref. Page
MEMory[:TABLe]:UNIT? **OUTPUT @Pwr_mtr;MEM:TABL:UNIT?** Response is the units used for analog output configuration, DBM or W	Query analog out power unit	2-58
MEMory[:TABLe]:VOLTage_space_**<start value from -10 to +10 V>**_comma_**<stop value from -10 to +10 V>** **OUTPUT @Pwr_mtr;MEM:VOLT -8,2**	Set analog out voltage range from -8 to 2 volts corresponding to power	2-59
MEMory[:TABLe]:VOLTage? **OUTPUT @Pwr_mtr;MEM:VOLT?** Response is two Voltage values separated by a comma indicating analog output range	Query analog out voltage range	2-59
***OPC** **OUTPUT @Pwr_mtr;*OPC**	*OPC allows one time SRQ enable	2-50
***OPC?** **OUTPUT @Pwr_mtr;*OPC?** Response to this one time operation complete query is a 0 or a 1	send a 1 upon operation complete	2-50
OUTPut[:BNC]:ANAlog[:STATe]_space_**<ON or OFF>** **OUTPUT @Pwr_mtr;OUTP:ANA ON**	Enable analog out function	2-59
OUTPut[:BNC]:ANAlog[:STATe]? **OUTPUT @Pwr_mtr;OUTP:ANA?** Response is a 1 for ON or 0 for OFF	Query analog out function status	2-59
OUTPut:ROSCillator[:STATe]_space_**<ON or OFF>** **OUTPUT @Pwr_mtr;OUTP:ROSC ON**	Turn ON 0.0 dBm Calibrator Oscillator	2-64
OUTPut:ROSCillator[:STATe]? **OUTPUT @Pwr_mtr;OUTP:ROSC?** Response is 1 for ON or 0 for OFF	Query ON or OFF status	2-64
***RCL**_space_**<memory location number 0 to 20>** **OUTPUT @Pwr_mtr;*RCL 19**	Recall 58542 register 19	2-60
READ<channel 1 or 2>[:POWer]? **OUTPUT @Pwr_Mtr;READ2?**	Trigger measurement and read Channel 2	2-21
***RST** **OUTPUT @Pwr_mtr;*RST**	Reset 58542 configuration	2-61
***SAVe**_space_**<memory location number 1 to 20>** **OUTPUT @Pwr_mtr;*SAV 20**	Save at 58542 register 20	2-60
SENSe<sensor 1 or 2>:AVERage:COUNt_space_**<averaging value 1,2,4,8,16,32,64,128,256,or512>** **OUTPUT @Pwr_mtr;SENS1:AVER:COUN 16**	Set sensor 1 average number to 16	2-43
SENSe<sensor 1 or 2>:AVERage:COUNt? **OUTPUT @Pwr_mtr;SENS1:AVER:COUN?** Response is a 1, 2, 4, 8, 16, 32, 64, 128, 256 or 512	Query sensor 1 current averaging value	2-43
SENSe<sensor 1 or 2>:AVERage:COUNt:AUTO_space_**<ON or OFF>** **OUTPUT @Pwr_mtr;SENS2:AVER:COUN:AUTO ON**	Select sensor 2 Auto-average mode	2-43
SENSe<sensor 1 or 2>:AVERage:COUNt:AUTO? **OUTPUT @Pwr_mtr;SENS2:AVER:COUN:AUTO?** Response is 1 for ON or 0 for OFF	Query sensor 2 Auto-Average mode ON or OFF status	2-43
SENSe<sensor 1 or 2>:AVERage:TCONtrol_space_**<data acquisition averaging method, MOVing or REPeat>** **OUTPUT @Pwr_mtr;SENS1:AVER:TCON REP**	Set sensor 1 to acquire fresh measurement data before averaging	2-44
SENSe<sensor 1 or 2>:AVERage:TCONtrol? **OUTPUT @Pwr_mtr;SENS1:AVER:TCON?** Response is MOV or REP	Query sensor 1 average method	2-44
SENSe<sensor 1 or 2>:CORRection:FREQuency[:CW-FIXed] 5e7 **OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 5e7**	Set sensor 2 frequency to 50 MHz	2-31
SENSe<sensor 1 or 2>:CORRection:FREQuency? **OUTPUT @Pwr_mtr;SENS1:CORR:FREQ?** Response is a single frequency in Hz which is the frequency of the signal incident upon the sensor	Query sensor 1 current frequency value	2-31
SENSe<sensor 1 or 2>:CORRection:OFFSet:COLLect **OUTPUT @Pwr_mtr;SENS1:CORR:OFFS:COLL**	Take sensor 1 current reading as offset value	2-47

Table 2-2: VXI GPIB Command Syntax (Continued)

Command Syntax Typical Example	Function	Ref. Page
SENSe<sensor 1 or 2>:CORRection:OFFSet [:MAGnitude]*space***<offset value in dB -99.99 to 99.99>** **OUTPUT @Pwr_mtr;SENS2:CORR:OFFS 10.2**	Compensate sensor 2 for 10.2 dB attenuation	2-47
SENSe<sensor 1 or 2>:CORRection:OFFSet[:MAGnitude]? **OUTPUT @Pwr_mtr;SENS2:CORR:OFFS?** Response is a single value in dB for the sensor offset value	Query sensor 2 offset value	2-47
SENSe<sensor 1 or 2>:CORRection:OFFSet:STATe*space***<ON or OFF>** **OUTPUT @Pwr_Mtr;SENS2:CORR:OFFS:STAT ON**	Enable sensor 2 offset correction	2-47
SENSe<sensor 1 or 2>:CORRection:OFFSet:STATe? **OUTPUT @Pwr_Mtr;SENS2:CORR:OFFS:STAT?** Response is 1 for ON or 0 for OFF	Query sensor 2 offset function status	2-47
SENSe<sensor 1 or 2>:CORRection:VPROpf[:STATe]*space***<ON or OFF>** **OUTPUT @Pwr_mtr;SENS1:CORR:VPRO:ON**	Enable sensor 1 v_prop_f function	2-33
SENSe<sensor 1 or 2>:CORRection:VPROpf[:STATe]? **OUTPUT @Pwr_mtr;SENS1:CORR:VPRO?** Response is 1 for ON or 0 for OFF	Query v_prop_f function status	2-33
SENSe<sensor 1 or 2>:TEMPerature? **OUTPUT @Pwr_mtr;SENS1:TEMP?**	Query sensor 1 temperature (in centigrade)	2-19
SENSe<sensor 1 or 2>:TRIGger:SOURce*space***<PEak Sensor triggering INTernal, EXTernal or CW>** **OUTPUT @Pwr_mtr;SENS1:TRIG:SOUR CW**	Set sensor 1 peak trigger mode to CW mode	2-40
SENSe<sensor 1 or 2>:TRIGger:SOURce? **OUTPUT @Pwr_mtr;SENS1:TRIG:SOUR?** Response is INT, EXT or CW	Query sensor 1 peak trigger mode	2-40
SENSe<sensor 1 or2>:TRIGger:DELay[:MAGnitude]:*space* **<Peak Sensor sample delay in seconds from -20e-9 to 104e-3>** OUTPUT @Pwr_mtr;SENS1:TRIG:DEL 1e6	Set sensor 2 peak delay value to 1e6 seconds	2-40
SENSe<sensor 1 or 2>:TRIGger:DELay[:MAGnitude]? **OUTPUT @Pwr_mtr;SENS1:TRIG:DEL?** Response is Peak Sensor sample delay in seconds from 20e-9 to 104e-3	Query sensor 1 peak delay value	2-40
SENSe<sensor 1 or 2>:TRIGger:DELay:STATe*space***<ON or OFF>** **OUTPUT @Pwr_mtr;SENS1:TRIG:DEL:STAT ON**	Enable peak sensor 1 delay function	2-41
SENSe<sensor 1 or 2>:TRIGger:DELay:STATe? **OUTPUT @Pwr_mtr;SENS1:TRIG:DEL:STAT?** Response is 1 for ON or 0 for OFF	Query peak sensor 1 delay function	2-41
SENSe<sensor 1or2>:TRIGger:LEVel[:MAGnitude]*space* **<trigger level of peak power sensor.** **In EXT triggering use 0.1 to 5.1 V.** **In INT triggering use -30 to +20 dBm>** **OUTPUT @Pwr_mtr;SENS1:TRIG:LEV 1.7**	Set peak sensor 1 trigger level to 1.7 volts	2-41
SENSe<sensor 1 or 2>:TRIGger:LEVel[:MAGnitude]? **OUTPUT @Pwr_mtr;SENS1:TRIG:LEV?** Response is the current peak power sensor trigger level setting - a single value in volts or dBm	Query peak sensor 1 trigger level	2-42
SENSe<sensor 1 or 2>:TRIGger:OFFSet[:MAGnitude] **OUTPUT @Pwr_mtr;SENS1:TRIG:OFFS 0**	Set peak sensor 1 trigger offset time to 0 second	2-42
SENSe<sensor 1 or 2>:TRIGger:OFFSet[:MAGnitude]? **OUTPUT @Pwr_mtr;SENS1:TRIG:OFFS?** Response is the current peak power sensor trigger offset setting - a single value in seconds	Query peak sensor 1 trigger offset time	2-42
SENSe<sensor 1 or 2>:TRIGger:TOTAl[:MAGnitude]? **OUTPUT @Pwr_mtr;SENS1:TRIG:TOTA?** Response is the current peak power sensor total trigger delay setting - a single value in seconds	Query peak sensor 1 total trigger delay time	2-42
SREspace***<event status register value, 128, 64, 32, 16, 8, 4, 2, 1>** **OUTPUT @Pwr_mtr;*SRE 32**	Enable bit 5 of Status Byte register	2-50
***SRE?** **OUTPUT @Pwr_mtr;*SRE?** Response is the status register mask value	Query Bits Enabled of Status Byte register	2-50

Table 2-2: VXI GPIB Command Syntax (Continued)

Command Syntax Typical Example	Function	Ref. Page
STATus:OPERation[:EVENt]? **OUTPUT @Pwr_mtr;STAT:OPER?** Response is Operation Status Register value. Sum of on bits 0 to 14 expressed as 2^n	Query operation event register result	2-50
STATus:PRESet **OUTPUT @Pwr_mtr;STAT:PRES**	Clears all the status register value	2-51
***STB?** **OUTPUT @Pwr_mtr:*STB?** Response is status byte value. Sum of on bits 0 to 7 expressed as 2^n	Query status byte register	2-52
SYSTem:ERRor? **OUTPUT @Pwr_mtr;SYST:ERR?** Response is an error number value followed by a comma followed by an error information string in quotes	Query system error message	2-68
SYSTem:PRESet **OUTPUT @Pwr_mtr;SYST:PRES**	Reset 58542 configuration	2-61
SYSTem:VERSion? **OUTPUT @Pwr_mtr;SYST:VERS?** Response is the SCPI command set version, 1990.0	Query SCPI version	2-64
***TRG** **OUTPUT @Pwr_mtr;*TRG**	VXI cont. BUS Trigger Event	2-23
TRIGger[:IMMediate] **OUTPUT @Pwr_mtr;TRIG:IMM**	Trigger measurement cycle	2-23
TRIGger:SOURce*space***<instrument triggering source IMMediateBUS, HOLD, or EXTernal>** **OUTPUT @Pwr_mtr;TRIG:SOUR IMM**	Set instrument measurement trigger source to IMMediate	2-23
TRIGger:SOURce? **OUTPUT @Pwr_mtr;TRIG:SOUR?** Response is instrument triggering source IMM, BUS, HOLD, or EXT	Set instrument measurement trigger source	2-23
TRIGger:MODE*space***<burst mode data gathering POST trigger receipt or PRE trigger receipt>** **OUTPUT @Pwr_mtr;TRIG:MODE POST**	Set Burst or Swift mode data collection relative to Timing of event trigger post	2-24
TRIGger:MODE? **OUTPUT @Pwr_mtr;TRIG:MODE?** Response is PRE or POST	Query Burst or Swift mode returns PRE or POST	2-24
TRIGger:DELay*space***<delay time between buffered readings in sec 0.000 to 5.000>** **OUTPUT @Pwr_mtr;TRIG:DEL 5e-3** **OUTPUT @Pwr_mtr;TRIG:DEL 0**	Set Burst mode measurement time to 5 ms between measurements (resolution, one msec)	2-24
TRIGger:DELay? **OUTPUT @Pwr_mtr;TRIG:DEL?** Response is a single number indicating the number of seconds between readings. Resolution to 0.001	Query Burst mode measurement time returns delay value	2-24
TRIGger:COUNt*space***<number of data values to buffer in memory from 1 to 5000 for standard 58542s** **from 1 to 128,000 on one channel with option 02;** **from 1 to 64000 for two channels with option 02>** **OUTPUT @Pwr_mtr;TRIG:COUN 100**	Set Burst or Swift mode buffer reading number to 100	2-26
TRIGger:COUNt? **OUTPUT @Pwr_mtr;TRIG:COUN?** Response is a single number - the number of data points stored in memory during Burst or Swift modes	Query Burst or Swift mode buffer number	2-26
***TST?** **OUTPUT @Pwr_mtr;*TST?** Response is a 0 when self test passes	Query self-test result	2-64
***WAI** **OUTPUT @Pwr_mtr;*WAI**	Wait, following command completion	2-26

2.5.11 Sensor Calibration and Zeroing

CALibrate<sensor 1 or 2>
CALibrate<sensor 1 or 2>:STATe?
CALibrate<sensor 1 or 2>:ZERO
SENSe<sensor 1 or 2>:TEMPerature?

Sensor Calibration

The CALibration commands for sensor calibration and zeroing are important for accurate power measurement results. Be sure to perform the sensor calibration prior to beginning measurement operation or channel configuration. Sensors must be calibrated to the meter before performing measurements.

Zeroing of all active sensors should always be performed whenever a second sensor (whether calibrated or not) is added or removed. Zeroing should also be performed prior to measurement of low signal levels, generally within the lower 15 dB of a sensor's dynamic range. For standard sensors, this is -55 dBm.

2.5.11.1 CAL#

Syntax: CALibration<sensor 1 or 2>[:SENSor]

Example: OUTPUT @Pwr_mtr;CAL1 ! Calibrate Sensor 1

Description: This command begins the sensor power sweep calibration process. Power sweep calibration can be performed only during Normal mode. The sensor must be attached to the front panel CALIBRATOR (use an adapter for high frequency sensors with Type K connectors). If the sensor is not connected or if the sensor is disconnected during the power sweep calibration procedure, the calibration automatically fails and the sensor power sweep calibration table is restored to its previous values.

2.5.11.2 CAL#?

Syntax: CALibration<sensor 1 or 2>[:SENSor]?

Example: OUTPUT @Pwr_mtr;CAL1 ! Calibrate sensor 1 & Return Pass/Fail status

Response: 0 = pass; 1 = fail

Description: This command executes the same procedure as above except a pass fail flag is automatically output at either the completion of power sweep calibration or detection of failure. A 0 indicates a successful completion of the power sweep calibration for the sensor attached to the calibrator port or a cable leading to the calibrator port. A 1 indicates that something has caused a failure. The most common cause is forgetting to connect the sensor to the calibrator.

2.5.11.3 CALibrate: STATe?

Syntax: CALibrate<sensor 1 or 2>:STATe?

Example: OUTPUT @Pwr_mtr;CAL1:STAT? ! Query sensor for calibration status

Response: 1 if sensor is calibrated, or 0 if sensor is not calibrated.

Description: This command queries whether or not a sensor has been calibrated.

Sensor Zero

Zeroing automatically accounts for ground noise and other noise in your measurement system. Measurements will be sensitive to noise-induced errors only in the lowest 15 dB of the sensor dynamic range. *Be sure to turn off the signal going into the sensor during zeroing, otherwise a failure will be indicated*.

2.5.11.4 CAL#:ZERO

Syntax: CALibration<sensor 1 or 2>:ZERO

Example: OUTPUT @Pwr_mtr;CAL1:ZERO ! Zero Sensor 1

Description: This command begins the zeroing process. Since you want to offset for measurement circuit noise during zeroing, you want the sensor to remain attached to the measurement circuit during zeroing. Disable the signal source into the sensor. If you cannot disable the signal source, connect the sensor to a grounded connector (preferably RF grounded) or leave the sensor disconnected. Do not connect the sensor to the power meter front panel calibrator port. If the meter detects excessive RF power at the start of the zeroing process (generally above -50 dBm for standard sensors), zeroing will automatically fail.

2.5.11.5 CAL#:ZERO?

Syntax: CALibration<sensor 1 or 2>:ZERO?

Example: OUTPUT @Pwr_mtr;CAL1:ZERO ! Zero Sensor 1 & Return Pass/Fail status

Response: 0 = pass; 1 = fail

Description: This command executes the same procedure as above except a pass/fail flag is automatically output at either the completion of zeroing or detection of failure. The most common cause is not turning off the source power before beginning the zeroing process.

2.5.11.6 SENS#:TEMP?

Syntax: SENSe<sensor 1 or 2>:TEMPerature?

Example: OUTPUT @Pwr_mtr;SENS1:TEMP? ! Query sensor 1 temperature (in centigrade)

Response: Current sensor temperature in degrees centigrade

Description: This command reports sensor temperature. If the temperature varies in your operating environment, monitor the relative temperature variations. If the variation since the previous power sweep calibration exceeds ±5°C (±9°F), perform the power sweep calibration for that sensor.

Example programs for Sensor Calibration and Zeroing are contained in Appendix A.

2.5.12 Reading Power Measurements

FETCh?
MEASure?
READ?
***OPC**

These commands return measurement data from the 58542. During Normal mode the data will be single measurement values. During Swift or Burst modes, the data will be an array of values. Generally, it is a single array if one sensor is connected and calibrated, and a dual array if two sensors are connected and calibrated.

2.5.12.1 FETC?

Syntax: FETCh<channel 1 or 2>?

Example: OUTPUT @Pwr_Mtr;FETC1? ! Read previously triggered channel 1 power
! Normal, Swift, or Burst mode data.

Response: Whatever is in the output buffer whether it is a new measurement or an old one.

Description: The FETCh? query command returns post-processed measurement data from the active channels of the 58542. After receiving FETCh? the 58542 will output the contents of it's active data output buffer. The data size and output format of the buffer are dependent upon channel and measurement mode configuration.

For example, if channel 1 is defined as sensor 1 and channel 2 is defined as sensor 2/1, the FETCh? query command will return the power level incident upon sensor 1 and the ratio of the power levels incident upon sensor 2 and 1. FETCh? is used with NORMal mode measurements when INITiate:IMMediate is active, and for reading SWIFt and BURSt Mode measurement data from the meter's data buffer.

2.5.12.2 MEAS?

Syntax: MEASure<channel 1 or 2>[:SCALar:POWer]?

Example: OUTPUT @Pwr_Mtr;MEAS2? ! Arm, Trigger, and Measure Channel 2
! power in NORMal mode.

Response: A power measurement.

Description: The MEAS? command returns measured data from the active software calculation channels of the 58542. The MEAS? command will also initiate the trigger cycle and will turn on the Auto Averaging mode; that is, measurement will be triggered and the data will be transmitted from the power meter. This is in contrast to the FETCh? command, which is capable of causing immediate output of the measurement data but not initiating the triggering cycle.

2.5.12.3 READ?

Syntax: READ<channel 1 or 2>[:POWer]?

Example: OUTPUT @Pwr_Mtr;READ2? ! Trigger measurement and read Channel 2

Response: A power measurement.

Description: The READ[:POWer]? query command triggers measurement and sends the data to the controller. The READ? command enables the power meter to acquire data at the next instrument trigger and return post-processed measurement data from the active channels of the 58542. READ uses the current sensor averaging setup.

2.5.12.4 *OPC

Example: OUTPUT @Pwr_mtr;*OPC ! *OPC allows one-time SRQ enable

Description: The *OPC command determines when an operation is completed. This command is generally used to monitor the completion of long measurement sequences. It sets the operation complete bit in the event status register upon completion of operation.

2.5.12.5 *OPC?

Example: OUTPUT @Pwr_mtr;*OPC? ! send a 1 upon operation complete

Response: This one time operation complete query is a 0 or a 1.

Description: The *OPC command determines when an operation is completed. This command is generally used to monitor the completion of long measurement sequences. It sets the operation complete bit in the event status register upon completion of operation.

2.5.12.6 Special Errors

An unusual or non-sense numeric response, such as 9e+40, indicates that you are getting an error response. For instance, if you have not performed the power sweep calibration procedure to calibrate the sensor to the power meter, the response to a MEAS#? command will be 9.0000e+40.

Selected basic SCPI syntax and execution errors apply to these commands.

If you are using the READ#? measurement query and INIT:CONT is ON, you will get a bad value returned, 9e+40. If you send SYST:ERR? error query, -213, Init ignored will be returned. READ#? contains the low level function INIT, since INIT:CONT is ON the INIT within READ#? generates an error. Set INIT:CONT to OFF when using READ#?.

There are no device-specific errors for the preset configuration or status reset commands except the high level -300, Device-specific error response.

Example programs for Reading Power Measurements are contained in Appendix A.

2.5.13 Instrument Triggering

***TRG**
TRIGger[:IMMediate]
TRIGger:SOURce
TRIGger:MODE
TRIGger:DELay
TRIGger:COUNt
INITiate:CONTinuous
INITiate[:IMMediate]
***WAI**

These SCPI commands trigger the measurement cycle. They do not configure or provide triggering for Peak Power Sensors. Those commands are defined in Section 2.5.21. For power meter operation, the TRIGger Subsystem is divided into two sections; Instrument Measurement Event Triggering, and Special Triggering Configuration commands for the fast reading buffered data, Burst, and Swift modes.

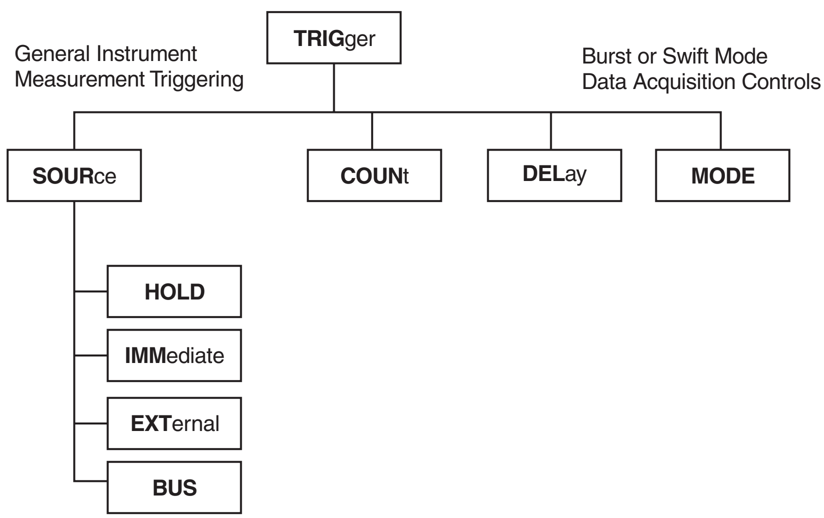


Figure 2-8: TRIGger Subsystem Command Tree

EXT triggering is performed with the rear panel BNC connector and functions only in the BURSt and SWIFt Modes. EXT triggering is not available in the NORMal Mode. Provision has been made for a hardware ready for new trigger type handshaking capability using the analog output.

BUS triggering is available for all operating modes, BURSt, SWIFt, and NORMal.

IMMediate triggering allows the 58542 to free run and perform continuous measurements. This is the default setting. You may also want to INITiate on IMMediate to increase measurement speed. During Normal Mode, with both of these controls set to IMM, power measurements can be read with MEAS, READ, or FETCh. IMMediate triggering is not compatible with the BURSt Mode. If you send the CALC#:MODE BURS command to enter the BURSt Mode, IMM instrument triggering for NORMal Mode will automatically be switched to TRIG:SOUR BUS. If you send the CALC#: MODE SWIF command to enter the SWIFt Mode, IMM instrument triggering will remain IMM but a device-specific error will be generated whenever you specify a TRIG:COUN# higher than 1.

2.5.13.1 *TRG

Syntax: *TRG

Example: OUTPUT @Pwr_mtr;*TRG ! BUS Trigger Event

Description: *TRG is an IEEE 488.2 compatible programming command to initiate BUS triggered measurements. The power meter interprets *TRG as a BUS source for instrument triggering events. This command will not trigger Peak Power Sensors. It is the same as the TRIG command.

2.5.13.2 TRIG

Syntax: TRIGger[:IMMediate]

Example: OUTPUT @Pwr_mtr;TRIG ! Trigger measurement cycle

Description: TRIG is the SCPI command form of *TRG. The power meter interprets TRIG as a BUS source for instrument triggering events. If INIT:CONT is OFF, measurements using FETCh#? will not proceed until triggering is armed and instrument triggering is actuated.

2.5.13.3 TRIG:SOUR

Syntax: TRIGger:SOURce*space*<instrument triggering source IMMediate, BUS, HOLD, or EXTernal>

Example: OUTPUT @Pwr_mtr;TRIG:SOUR IMM ! Set instrument measurement trigger
! source to IMMediate

Description: TRIG:SOUR IMM sets trigger control to the 58542. BUS triggering sets the triggering control to the controller software using TRIG or *TRG commands. HOLD halts triggering sequences. EXTernal sets triggering control to the front panel BNC connector. The following data shows triggering and operating mode compatibility.

Operating Mode	IMM	BUS	HOLD	EXT
NORMal	X	X	X	
SWIFt	X	X		X
BURSt		X		X

If the power meter has been configured for EXT triggering in the BURSt or SWIFt modes and the operating mode is switched to NORMal, the triggering configuration remains EXT. However, if the configuration is IMM in NORMal mode and the meter is switched to BURSt, it automatically switches to BUS. As a general rule, when a measurement subroutine switches operating modes, you should send the triggering configuration.

2.5.13.4 TRIGger:SOURce?

Example: OUTPUT @Pwr_mtr;TRIG:SOUR? ! Set instrument measurement
! trigger source

Response: Instrument triggering source IMM< BUS, HOLD, or EXT.

Description: This query responds with the current setting of the TRIG:SOUR command.

2.5.13.5 TRIG:MODE

Syntax: TRIGger:MODE*space*<burst mode data gathering POST trigger receipt or PRE trigger receipt>

Example: OUTPUT @Pwr_mtr;TRIG:MODE POST ! Set Burst or Swift mode data collection
! relative to Timing of event trigger post

Description: This command is only for BURSt Mode measurements. Send this command only immediately after CALC#:MODE BURS. TRIG:MODE controls the operation of the FIFO data buffer inside the power meter. When set to POST, the burst of data points is taken after the receipt of a valid instrument trigger. When set to PRE, the burst of data is assumed to be those data points that have arrived immediately preceding the valid instrument trigger event. Using PRE requires some caution: The single processor in the 58542 is constantly performing data acquisition. If you send any configuration commands while it is collecting the PRE trigger data, you may interrupt the timing of the data point collection.

2.5.13.6 TRIGger:MODE?

Example: OUTPUT @Pwr_mtr;TRIG:MODE? ! Set instrument measurement
! trigger mode.

Response: The response is PRE or POST

Description: This query responds with the current setting of the BURST mode pre or post-triggered data collection. The query can be used in any operating mode.

2.5.13.7 TRIG:DEL

Syntax: TRIGger:DELay*space*<delay time between buffered readings in sec, 0.000 to 5.000>

Example 1: OUTPUT @Pwr_mtr;TRIG:DEL 5e-3 ! Set Burst mode measurement time to
! 5 ms between measurements.

Example 2: OUTPUT @Pwr_mtr;TRIG:DEL 0 ! Max. measurement rate (resolution,
! 1 msec)

Description: This command applies only to the BURSt Mode. It sets the time interval between the measurements and is accurate to about 5%. This is the only way to control the data acquisition of individual data points in a burst measurement. Burst measurement data is taken following or just prior to a single instrument trigger event. The number of measurements taken is controlled by TRIG:COUNt. Timing can be set in 0.001 second increments from 0.000 to 5.000 seconds, inclusive. Setting this control to 0.000 activates the highest reading rate (5100 rdgs/sec) possible with the 58542.

Triggering of the Peak Power Sensor occurs independently from instrument triggering. Therefore, there is a moderate ability to control triggering at the Peak Power Sensors. Generally, if you require control of individual data point triggering, the SWIFt mode will provide the highest measurement rates, with or without SRQ or hardware handshaking.

2.5.13.8 TRIGger:DELay?

Example: OUTPUT @Pwr_mtr;TRIG:DEL? ! Query Burst mode measurement
! time returns delay value

Response: A single number indicating the number of seconds between readings. Resolution to 0.001.

Description: This query returns the current setting of the delay time interval between burst mode and data points.

2.5.13.9 TRIG:COUN

Syntax: TRIGger:COUNt*space*<number of data values to buffer in memory from 1 to 5000 for the standard 8650; from 1 to 128,000 on one channel with option 02; and 1 to 64000 for two channels with option 02>

Example: OUTPUT @Pwr_mtr;TRIG:COUN 100 ! Set Burst or Swift mode buffer reading ! number to 100

Description: The power meter has an internal and expandable data buffer for storing measurement data until the slot 0 controller/resource manager is ready to read the data. During the SWIFt or BURSt modes, this command controls the number of power readings that are stored in the 58542 data buffer.

The value specified here applies to each individual channel. That is, if two sensors are attached and calibrated, the 58542 will perform the specified measurement and buffering for both channels. In the example shown below for this command, that would be 100 points per channel. Option 02 adds additional memory for 128,000 data point buffering. When two sensors are attached and calibrated, the maximum is 64,000 readings per channel.

During BURSt mode, the COUNt# of readings will be taken with just one instrument trigger. During the SWIFt Mode however, you can control the triggering of individual data points. If you set triggering to EXT, the power meter will perform one measurement each time it detects a TTL level signal. The value is then stored in the data buffer. This continues until the meter has received and processed a COUNt# of triggers and also read in a COUNt# of data points to the buffer.

2.5.13.10 TRIGger:COUNt?

Example: OUTPUT @Pwr_mtr;TRIG:DEL? ! Query Burst mode measurement ! time returns delay value

Response: Single number indicating the number of seconds between readings. Resolution to 0.001.

Description: This query returns the current setting of the delay time interval between burst mode and data points.

2.5.13.11 *WAI

Example: OUTPUT @Pwr_mtr;*WAI ! Wait, following command completion

Description: The *WAI command causes the power meter to wait until all previous commands and queries are completed before continuing operation. It functions only when using the Normal mode of operation.

Example programs for Instrument Triggering are contained in Appendix A.

2.5.14 Arming the Triggering Cycle

INITiate:CONTinuous
INITiate[:IMMediate]

The Initiate commands enable the power meter to acquire measurement data at the next instrument trigger. In the absence of an instrument signal, the 58542 is placed in the waiting-for-trigger-state. The default configuration is continuous initiation, INIT:CONT ON.

INITiate[:IMMediate] causes the power meter to exit the idle state and causes the trigger system to initiate and complete one full trigger cycle, returning to the idle state upon completion. For example, INITiate[:IMMediate] can be used with Peak Power Sensors to measure the power level of transient or one-shot pulsed microwave signals. After execution of the triggering sequence, send the FETCh? query command to return the measurement data from the 58542.

Perform triggering configuration with the TRIGger Subsystem commands. When TRIGger:SOURce is IMMediate, the measurement will start as soon as INITiate is sent to the 58542 and executed (or INITiate:CONTinuous ON sent and executed).

2.5.14.1 INIT:CONT

Syntax: INITiate:CONTinuous*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;INIT:CONT ON ! Continuous Triggering

Description: This allows user control of measurement cycle arming. When INIT:CONT is set to ON, measurement cycle arming occurs automatically. The default setting is OFF. When you need to take control of arming the triggering of the measurement cycle, set this control to OFF and use MEAS#? or INIT with READ#? to perform the measurement. Another legal measurement control option would be INIT with TRIG: IMM and FETC#?

When INITiate:CONTinuous is ON, the instrument triggering cycle resets continuously. That is, upon instrument triggering, data is constantly being acquired and updated. Upon completion of a trigger cycle, the meter will immediately enter the wait-for-trigger state rather than the idle state, as is the case with INITiate:CONT OFF.

2.5.14.2 INIT:CONT?

Syntax: INITiate:CONTinuous?

Example: OUTPUT @Pwr_mtr;INIT:CONT? ! Query the INIT:CONT state

Response: Response is 1 for ON or or 0 for OFF.

Description: This query returns the current setting of the measurement cycles continuous initiation. The ON condition indicates that the 58542 is controlling the measurement initiation. The OFF condition indicates that measurement initiation is controlled by the controller/resource manager.

2.5.14.3 INIT:IMM

Syntax: INITiate[:IMMediate]

Example: OUTPUT @Pwr_mtr;INIT ! Initiate instrument triggering cycle

Description: This command enables the triggering cycle. Instrument triggering will not occur unless INIT has been enabled, even when you are sending BUS or EXT triggers to the 58542.

Either this command or INIT:CONT ON performs measurements with FETCh? and READ?. MEAS? is a high level function which contains the INITiation enabling function. Arming occurs automatically when using the MEAS#? command to automatically INITiate, TRIGger and FETCh the measurement data. However, if you are using READ#? or FETCh#? to read data, this command arms the 58542 for triggering upon the occurrence of the next valid trigger.

The INITiate[:IMMediate] command places the power meter in the wait-for-trigger-state. When used with EXT (TTL levels) instrument triggering, this command is useful for measuring signals that are asynchronous with programming (BUS triggered) sequences. When INITiate:CONTinuous is set to OFF, INITiate:[IMMediate] enables the instrument triggering cycle. Sending INITiate:[IMMediate] when INITiate:CONTinuous is set to ON will cause a -123 command error. The INITiate [:IMMediate] command is an event; no query form exists for this command.

2.5.15 Channel Configuration

CALCulate:POWer
CALCulate:RATio
CALCulate:DIFFerence
CALCulate?
CALCulate:UNIT
CALCulate:STATe
CALCulate:STATe?

2.5.15.1 CALC#:POW

Syntax: CALCulate<channel 1 or 2>[:CHANnel]:POWer*space* <sensor 1 or 2>

Example: OUTPUT @Pwr_mtr;CALC2:POW 1 ! Configures channel 2 to
! measure Sensor 1 power

Description: This command configures a sensor to an individual channel, and the channel measures the sensor power level. This command does not have a query format. For the example above, the response to a CALC2? query is POW 1. This command should be executed only after the designated sensor has been calibrated.

2.5.15.2 CALC#:RAT

Syntax: CALCulate<channel 1 or 2>[:CHANnel]:RATio*space* <sensor 1 or 2*comma*sensor 2 or 1>

Example: OUTPUT @Pwr_mtr;CALC1:RAT 2,1 ! Configures channel 1 to measure
! Sensor 2 over Sensor 1 power

Description: This command configures a channel as a ratio of the power levels present at the two sensors. When UNITs are dBm, RATio configuration produces measurements in dB. When UNITS are Watts, RATio configuration measurements are percentage. This command does not have a query format. For the above example, the response to a CALC1? query is RAT 2,1. This command should be executed only after the designated sensor has been calibrated.

2.5.15.3 CALC#:DIFF

Syntax: CALCulate<channel 1 or 2>[:CHANnel]:DIFFerence*space* <sensor 1 or 2*comma*sensor 2 or 1>

Example: OUTPUT @Pwr_mtr;CALC1:DIFF 2,1 ! Configures channel 1 to measure
! Sensor 2 minus Sensor 1 power

Description: This command configures a channel as the difference between the power levels present at the two sensors. UNITs are dBm or Watts. This command does not have a query format. For the above example, the response to a CALC1? query is DIFF 2,1. This command should be executed only after the designated sensor has been calibrated.

2.5.15.4 CALC#?

Syntax: CALCulate<channel 1 or 2>[:FUNCtion]?

Example: OUTPUT @Pwr_mtr;CALC2? ! Query channel 2 sensor Configuration

Response: POW #, RAT #,#, or DIFF #,#

Description: This command returns the current channel configuration. This command is a query format only.

2.5.15.5 CALC#:UNIT

Syntax: CALCulate<channel 1 or 2>:UNIT[:POWer]*space*<data units selection, dBm or Watt>

Example 1: OUTPUT @Pwr_mtr;CALC1:UNIT W ! Selects channel 1 Linear Units, Watts

Example 2: OUTPUT @Pwr_mtr;CALC1:UNIT dBm ! Selects channel 1 Log Units, dBm

Description: This command configures a channel to report power measurements in either linear Watts units or logarithmic dBm units. When UNIT is dBm, the RATio configuration produces measurements in dB, and POWer and DIFF measurements in dBm. When UNIT is Watts, the RATio configuration measurements are percentages, and POWer and DIFF measurements are in Watts.

2.5.15.6 CALC#:UNIT?

Syntax: CALCulate<channel 1 or 2>:UNIT[:POWER]?

Example: OUTPUT @Pwr_mtr;CALC1:UNIT? ! Query channel 1 unit of measure

Response: dBm or Watt.

Description: This command is the query format of the command above.

2.5.15.7 CALC#:STAT

Syntax: CALCulate<Channel 1 or 2>:STATe*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;CALC1:STAT OFF ! Disable channel 1 measurement

Description: This command turns a channel off or on. Default is ON. During Normal and Swift modes, measurement speed increases when only one sensor (Channel configuration is CALC#:POW) is performing measurements.

2.5.15.8 CALC#:STAT?

Syntax: CALCulate<channel 1 or 2>:STATe?

Example: OUTPUT @Pwr_mtr;CALC1:STAT? ! Query channel 1 measurement status

Response: 0 = off, 1 = on.

Description: This command is the query format of the command above.

Example programs for Channel Configuration are contained in Appendix A.

2.5.16 Cal Factor Correction

SENS#:CORR:FREQ
MEMory:FREQuency
MEMory:SELect
MEMory:SLOPe
MEMory:SENsor

Power Sensors have a measurable frequency response. During manufacture, this response is calibrated at 1 GHz intervals. Instead of printing the data on the sensor label, each Giga-tronics power sensor includes a built-in EEPROM which has been programmed with the frequency calibration factor data for that particular power sensor.

The following SCPI commands tell the 58542 the operating frequency of the measured microwave or RF signal. Thus, the meter automatically interpolates the correct Cal Factor and applies that value to the measurement data. By performing this automatically, you do not need to read Cal Factor data from the side of sensor housings and program the data into tables within your ATE programming. Except for advanced measurement techniques using Burst Mode, you are not required to write cal factor interpolation routines; this is all done by the power meter. As shown below, sensor Cal Factor data can be read into your computer. Also see the Sensor DIAG:EEPROM Commands in Chapter 5, Table 5-1 for Cal Factor programming.

2.5.16.1 SENS#:CORR:FREQ

Syntax: SENSe<sensor 1 or 2>:CORRection:FREQuency[:CW-FIXed] 5e7

Example: OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 5e7 ! Set sensor 2 frequency to 50 MHz

Description This command allows you to correct for sensor frequency response characteristics. The power meter uses this frequency to automatically interpolate and apply the Cal Factor for that sensor (computed from Cal Factor data points stored within each sensor’s EEPROM). Unless the EEPROM has been modified subsequent to shipment, the Cal Factor is always 0.0 dB at 50 MHz, 5e7. Be sure to correct for sensor cal factor by using this command, using automated V_{PROP}F correction, sending your own cal factor REFerence value, or correcting for Cal Factor in your ATE system by downloading Cal Factor points for the 58542.

2.5.16.2 SENS#:CORR:FREQ?

Syntax: SENSe<sensor 1 or 2>:CORRection:FREQuency?

Example: OUTPUT @Pwr_mtr;SENS1:CORR:FREQ ! Query sensor 1 current frequency
! value for 50 MHz

Response: +5.0000E+07

Description This command allows you to correct for sensor frequency response characteristics. The power meter uses this frequency to automatically interpolate and apply the Cal Factor for that sensor (computed from Cal Factor data points stored within each sensor’s EEPROM).

2.5.16.3 MEM:SEL

Syntax: MEMory[:TABLe]:SELect*space<ANALOGout,VPROpf1,VPROpf2>*

Example: OUTPUT @Pwr_mtr;MEM:SEL VPROpf1 ! Select memory table V_{PROP}F1
! for editing in following lines

Description: This command selects between one of three editable tables. Using the meter's automated V_{PROP}F capability to correct for Cal Factor requires that you configure the V_{PROP}F IN connector to match the signal that your source will output. This is performed by setting the frequency so that your source's V_{PROP}F output is 0.0 V. Then the slope relationship is entered in the form of a value representing the V/Hz relationship. If your source is a Giga-tronics GT 9000 Series Signal Generator, this relationship is 0.5 V/GHz or 5e-8 V/Hz.

V_{PROP}F Configuration

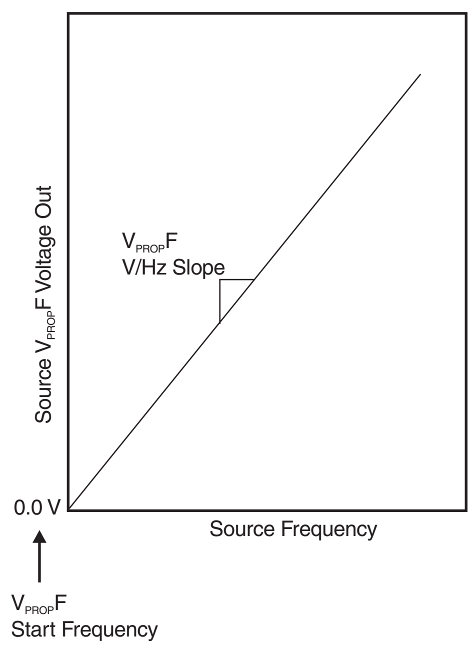


Figure 2-9: VPROpf Configuration

Syntax: MEMory[:TABLe]:SELect?

Example: OUTPUT @Pwr_mtr;MEM:SEL? ! Query the selected memory table

Response: The table is currently editable; ANALOG, VPROPF1 or VPROPF2.

Description: This command reports the currently active, editable table.

2.5.16.4 MEM:SLOP

Syntax: MEMory[:TABLe]:SLOPe*space<Volts per Hz>*

Example: OUTPUT @Pwr_mtr;MEM:SLOP 1e-9 ! Set $V_{PROP}F$ slope to 1 V/GHz

Description: This command sets the V/Hz relationship for automated Cal Factor correction.

2.5.16.5 MEM:SLOP?

Syntax: MEMory[:TABLe]:SLOPe?

Example: OUTPUT @Pwr_mtr;MEM:SLOP? ! Query $V_{PROP}F$ slope value

Response: The voltage to frequency value you set that corresponds to your source's $V_{PROP}F$ output.

Description: This command returns the V/Hz slope relationship for $V_{PROP}F$ frequency correction.

2.5.16.6 MEM:FREQ

Syntax: MEMory[:TABLe]:FREQuency*space<Start frequency>*

Example: OUTPUT @Pwr_mtr;MEM:FREQ 1e9 ! Set $V_{PROP}F$ frequency to 1 GHz

Description: This command sets the automated $V_{PROP}F$ correction start frequency. The frequency at which the voltage input into the 58542 is 0.0 V.

2.5.16.7 MEM:FREQ?

Syntax: MEMory[:TABLe]:FREQuency?

Example: OUTPUT @Pwr_mtr;MEM:FREQ? ! Query $V_{PROP}F$ start frequency

Response: The frequency value you set that corresponds to your source's $V_{PROP}F$ output at 0.0 V.

Description: This $V_{PROP}F$ command reports the start frequency.

2.5.16.8 SENS#:CORR:VPRO

Syntax: SENSe<sensor 1 or 2>:CORRection:VPROpf[:STATe]*space<ON-OFF>*

Example: OUTPUT @Pwr_mtr;SENS1:CORR:VPRO ON ! Enable sensor 1 $V_{PROP}F$ function

Description: Setting this control to ON activates $V_{PROP}F$ and deactivates SENS#:CORR:FREQ.

2.5.16.9 SENS#:CORR:VPRO?

Syntax: SENSe<sensor 1 or 2>:CORRection:VPROpf[:STATe]?

Example: OUTPUT @Pwr_mtr;SENS1:CORR:VPRO? ! Query $V_{PROP}F$ function

Response: 1 = On or 0 for OFF.

Description: This command reports ON or OFF status of $V_{PROP}F$ correction for the sensor.

Frequency response variations (which are reflected in Cal Factors in Giga-tronics sensors) do not change applicably over small frequency ranges. That is, the frequency that you send to the power meter

does not need to be exact. If the actual measurement frequency is within about 40 MHz of the value sent to the power meter, your measurement variation due to this discrepancy will typically be less than 0.02 dB, well below typical RSS measurement accuracy levels.

* ***NOTE:*** *Example programs for Cal Factor Corrections are contained in Appendix A of this manual.*

2.5.16.10 Cal Factor Error Control

See Error Messages in Section 2.5.38.

Selected basic SCPI syntax and execution errors apply to these commands.

Device-specific errors include the following and other -300 level errors.

A common device-specific error occurs when the frequency sent to the 58542 in the SENS#:CORR:FREQ ### command is outside the sensor operating frequency range. For example, sending SENS1:CORR:FREQ 18.4e9 when an 18 GHz (max) 80301A CW Power Sensor is attached will yield a device-specific error.

2.5.17 High Speed Measurements

CALCulate:DATA?
CALCulate:MODE
TRIGger:MODE
TRIGger:DELay
TRIGger:COUNt

Measurements in Normal mode are fastest with only one sensor attached. When two sensors are attached, the Normal mode measurement rate is reduced. This applies for all three major measurement commands, FETCh, READ, and MEASure. Both of the averaging types, MOVing and REPeat, slow down when in the Normal mode. MOVing provides faster averaging; the speed is equivalent to REPeat whenever the Averaging number is 1.

When performing measurements in either Swift or Burst modes, measurement rates are the same with two sensors as with one sensor. Approximate measuring speeds are listed in Table 2-3.

2.5.17.1 CALC:DATA?

Syntax: CALCulate<channel 1 or 2>:DATA?

Example: OUTPUT @Pwr_mtr;CALC1:DATA? ! Query channel 1 & 2 BURSt mode data

Description: This command interrupts the burst measurement and dumps the measurement buffer. All empty locations will be filled with -300 before the data dump. The Burst mode remains active after this command so that a trigger will start a new measurement reference.

The command returns measurement data (while the measurement is completing) during BURSt mode only. This command is not used in NORMal or SWIFt Modes. Operation is similar to using FETCh? Both are instrument (not channel) commands. That is, when both sensors are connected, calibrated, and the channel set to ON, this command will return power level data for both channels.

Operating Mode Control

The Operating mode is controlled through the CALC#:MODE command. The choices are NORMal, BURSt, and SWIFt. Sending CALC1:MODE will set the operating mode for the entire instrument, not just for channel 1. For example, you cannot send CALC1:MODE BURS, and then CALC2:MODE SWIF to get channel 1 in BURSt mode and channel 2 in SWIFt mode. After the CALC2:MODE SWIF command, both channels are in SWIFt mode.

2.5.17.2 CALC:MODE

Syntax: CALCulate<channel 1 or 2>:MODE*space<NORMal, BURSt, or SWIFt Mode Selection>*

Example: OUTPUT @Pwr_mtr;CALC2:MODE SWIF ! Set measurement mode to SWIFt

Description: This command sets the measurement mode. This is NOT a channel-specific command; it is an instrument configuration command. Selecting one channel changes both channels to the desired mode. Use this command to select the level of functionality and performance for your unique applications. NORMal Mode is full functioned. SWIFt Mode allows individual data point triggering. BURSt Mode allows the fastest measurement rates.

The following commands are duplicates of those given in the Instrument Triggering section of this chapter, and are shown here for convenience. These commands must be used with BURSt mode operation.

2.5.17.3 CALC<channel 1 or 2>:MODE?

Example: OUTPUT @Pwr_mtr;CALC!:MODE?, ! Query 58542 measurement mode

Response: NORM, SWIF, or BURS.

Description: This query returns the current operationg mode.

2.5.17.4 TRIG:MODE

Syntax: TRIGger:MODE*space<burst mode data gathering POST trigger receipt or PRE trigger receipt>*

Example: OUTPUT @Pwr_mtr;TRIG:MODE POST ! Set Burst or Swift mode data collection ! relative to Timing of event trigger post

Description This command is only for BURSt Mode measurements. Send this command only after CALC#:MODE BURS. TRIG:MODE controls the operation of the FIFO data buffer in the 58542. When set to POST, the burst of data points is taken after the receipt of a valid instrument trigger. When set to PRE, the burst of data is assumed to be those data points that have arrived immediately preceding the valid instrument trigger event. Using PRE requires some caution: the single processor inside the 58542 is performing data acquisition constantly. If you send any configuration commands while the 58542 is collecting the PRE trigger data, you may interrupt the timing of the data point collection.

2.5.17.4.1 TRIGger:MODE?

Example: OUTPUT @Pwr_mtr;TRIG:MODE? ! Query Burst or Swift mode returns

Response: PRE or POST.

Description: This query command responds with the current setting of the BURSt mode pre- or post-triggered data collection. Query can be used ian y operating mode.

2.5.17.5 TRIG:DEL

Syntax: TRIGger:DELay*space<delay time between buffered readings in sec, 0.000 to 5.000>*

Example 1: OUTPUT @Pwr_mtr;TRIG:DEL 5e-3 ! Set Burst mode measurement time to
! 5 ms between measurements.

Example 2: OUTPUT @Pwr_mtr;TRIG:DEL 0 ! Max. measurement rate, 5100/sec.
(resolution, one msec)

Description: This command applies only to BURSt Mode. It sets the time interval between the measurements; it is accurate to about 5%. This is the only way to control the meter's data acquisition of individual data points in a burst measurement. Burst measurement data is taken following or just prior to a single instrument trigger event. The number of measurements taken is controlled by TRIG:COUNt. Timing can be set in 0.001 second increments from 0.000 to 5.000, inclusive. Setting this control to 0.000 activates the highest reading rate possible with 8650; 5100 readings per second.

Triggering of the Peak Power Sensor occurs independently from instrument triggering. There is a moderate ability to control triggering at the Peak Power Sensors. Generally, if you require control of individual data point triggering, SWIFt mode will provide the highest measurement rates, with or without SRQ or hardware handshaking. The 58542 controls its gain ranges transparently by verifying internal range changes just before and just after individual measurement points. If the internal range does not remain the same, the data point will be discarded as invalid.

2.5.17.6 TRIGger:DELAY?

Example: OUTPUT @Pwr_mtr;TRIG:DEL? ! Query Burst or Swift mode returns

Response: A single number indicating the number of seconds between readings. Resolution to 0.001.

Description: This query returns the current setting of delay time interval between burst mode data points.

2.5.17.7 TRIG:COUN

Syntax: TRIGger:COUNt*space<number of data values to buffer in memory from 1 to 5000*

Example: OUTPUT @Pwr_mtr;TRIG:COUN 100 ! Set Burst or Swift mode buffer reading
! number to 100

Description: The 58542 has an internal data buffer for storing measurement data until the slot 0 controller/resource manager is ready to read the data. During SWIFt or BURSt modes, this command controls the number of power readings that are stored in the 58542 data buffer.

The value specified here applies to each channel. If two sensors are attached and calibrated, the 58542 will perform the specified measurement and buffering for both channels. In the example above, that would be 100 points per channel.

During BURSt mode, the COUNt # of readings will be taken with just one instrument trigger. During SWIFt Mode however, you can control the triggering of individual data points. If you set triggering to EXT, the 58542 will perform one measurement each time a TTL level signal is detected by the 58542. Then the value is stored in the data buffer. This continues until the 58542 has received and operated the COUNt # of triggers and also read in the COUNt # of data points to the buffer.

2.5.17.7.1 TRIGger:COUNt?

Example: OUTPUT @Pwr_mtr;TRIG:COUN? ! Query Burst or Swift mode buffer ! number

Response: Single number indicating the number of seconds between readings. Resolution to 0.001.

Description: This query returns the current setting of the delay time interval between BURSt or SWIFt Mode.

2.5.18 Approximate Measurement Speeds

Table 2-3 lists the 58542 approximate measurement speeds in readings per second.

Table 2-3: Approximate Measurement Speeds

Range (dBm)	22 to 4	6 to -10	-7 to -21	-18 to -30	-27 to -37	-35 to -48	-46 to -66	-64 to -99
Normal 1	53	53	53	53	53	53	53	53
Normal 2	23	23	23	23	24	6.9	6.6	5.8
Normal 3	22	22	22	16	17	0.5	0.06[2]	0.06[2]
Swift IMM	71	71	71	71	71	71	71	71
Swift Bus 1	39	39	39	39	40	25	25	25
Swift Bus 2	53	53	53	53	56	24	24	24
Swift Ext	150[1]	150[1]	150[1]	150[1]	150[1]	150[1]	150[1]	150[1]
Burst[3]	5100	5100	5100	5100	5100	5100	5100	5100

Notes:

1. HP 9000-300 Computer, HP BASIC Language, HP-E1505B Controller
2. Speed will be 0.24 for average number of 32, and 0.12 for average number of 64
3. Speed only includes measurement time

Normal 1: SENS1:AVER:TCON MOV
INIT:COUN ON
FETC1

Normal 2: SENS1:AVER:TCON MOV
INIT:COUN ON
MEAS1?

Normal 3: SENS1:AVER:TCON REP
MEAS1?

Swift Imm

Swift 1: (SRQ Handshake)
Swift 2: (without SRQ Handshake)
Swift 3: (without TTL Handshake)

Burst

2.5.19 Triggering Notes

Refer to Instrument Triggering in Section 2.5.13 for more information.

EXT triggering is performed with the front panel BNC connector and will work only in BURSt and SWIFt Modes. EXT triggering is not available in NORMal Mode. Provision has been made for a hardware ready type handshaking capability using the analog output. This feature can be used as a loop back test when performing troubleshooting procedures.

BUS triggering is available for all operating modes, BURSt, SWIFt and NORMal.

IMMediate triggering is not compatible with BURSt and SWIFt Modes. If you send the CALC#:MODE BURS command to enter the BURSt Mode, IMM instrument triggering for NORMal Mode will automatically be switched to TRIG:SOUR BUS. If you use the CALC#:MODE SWIF command to enter the SWIFt Mode, IMM instrument triggering will remain IMM, but a device-specific error will be generated whenever you specify a TRIG:COUN # higher than 1. Using the FETCh#? measurement query, you can take accurate measurements anyway; do not set TRIG:COUN.

Using FETCh#? when you have time-dependent measurement processes can be a little tricky unless you use SRQs. If the 58542 has not had enough time to process the measurement or has not received a trigger, it will return an abnormally large number — 9.e40 is common, but other obviously invalid readings can occur. Not using SRQs is fastest for measurement speed, but you will have to manage the measurement/triggering timing problem closely.

2.5.20 Measurement Level Notes

The SWIFt Mode should not be used for measuring power in the bottom 10 dB of the CW dynamic range. At low power levels, the NORMal mode reduces the measurement speed to account for the effects of noise.

Example programs for High Speed Measurements are contained in Appendix A.

2.5.21 Peak Power Sensor Triggering

2.5.21.1 SENSe:TRIGger

The Peak Power Sensors sample power levels almost instantaneously. Since there is a sampler built into the Peak Power sensor housing, there are several controls to configure the source of the sensor trigger signal. These include the delay time from triggering to when the sample is to be taken (Sample Delay), and the trigger level. All Peak Power Sensors will operate in the SWIFt and BURSt modes when in the CW measurement mode (SOURce) configuration (not when using INT or EXT trigger).

2.5.21.2 SENSe:TRIG:SOUR

Syntax: SENSe<sensor 1 or 2>:TRIGger:SOURce*space*<INTernal, EXTernal, or CW>

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:SOUR CW

! Set sensor 1 peak trigger mode to CW mode

Description: This command controls the trigger source for the sensors. Both the 80350A Series and 80340 Series of Peak Power Sensors respond to this command. INTernal triggering is performed automatically by the 80350A Sensor; however, the trigger level must be set to a value of power that is lower than the power level of the pulsed signal incident upon the sensor. EXTernal triggering is performed by inputting a baseband pulse into the rear connector on the sensor housing. BNC to SMB cables have been provided for this purpose. For high speed trigger pulses, SMB to SMA adapters are available from Giga-tronics. CW triggering sets the power sensors to measure CW power and triggering is essentially disabled. CW is automatically set by the 58542 during power sweep calibration; do not apply any triggering to the peak power sensor during power sweep calibration.

Be sure to set the trigger level, SENS:TRIG:LEV ##.# after the INTernal or EXTernal triggering is selected.

2.5.21.3 SENSe:TRIG:SOUR?

Syntax: SENSe<sensor 1 or 2>:TRIGger:SOURce?

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:SOUR? ! Query sensor 1 peak trigger mode

Response: INT, EXT, or CW.

Description: This command reports the current trigger source setting.

2.5.21.4 SENSe:TRIG:DEL

Syntax: SENSe<sensor 1 or 2>:TRIGger:DELay[:MAGnitude]:*space<delay in seconds from -20e-9 to 105e-3>*

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:DEL 1e-6 ! Set sensor 2 peak delay value to 1e-6 ! seconds

Description: The SENS:TRIG:DEL # is the time in seconds between the 80350A Series Peak Power Sensor's receipt of a sensor trigger (INT is auto, EXT is a trigger input) and the time that the sampler is fired. This delay is typically accurate to about 0.5%; repeatability is an order of magnitude better. There is a delay line built into the peak power sensor. Therefore, you can measure the power on the leading edge of the pulsed signal even if it is the same edge on which the sensor was triggered. This also allows limited look ahead capability as identified by the allowed negative time values for this command. The other reason negative time values are allowed is due to the sample delay offset capability. (The 80340 Series Peak Power Sensors do not support this command).

2.5.21.5 SENSe:TRIG:DEL?

Syntax: SENSe<sensor 1 or 2>:TRIGger:DELay:[MAGnitude]?

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:DEL? ! Query sensor 1 peak delay value

Response: Peak Sensor sample delay in seconds from -20e-9 to 105e-3.

Description: This query reports the current value of the peak power sensor sample delay.

2.5.21.6 SENSe:TRIG:DEL:STAT

Syntax: SENSe<sensor 1 or 2>:TRIGger:DELay:STATe*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:DEL:STAT ON! Enable peak sensor 1 delay function

Description: This command activates the sample delay in the 80350A Series Peak Power Sensors. The 80340 Peak Power Sensors also respond to this command, but with only a fixed time delay.

2.5.21.7 SENSe:TRIG:DEL:STAT?

Syntax: SENSe<sensor 1 or 2>:TRIGger:DELay:STATe?

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:DEL:STAT?! Query sensor 1 peak delay state

Response: Peak Sensor trigger delay state returns a 1 or a 0. 1 indicating the trigger delay is on. 0 indicating the trigger delay is off.

Description: This query reports the current value of the peak power sensor delay state.

2.5.21.8 SENSe:TRIG:LEV

Syntax: SENSe<sensor 1 or 2>:TRIGger:LEVel[:MAGnitude]*space*<R In EXT triggering use -0.1 to 5.000 V. In INT triggering use -30 to +20 dBm>

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:LEV 1.7! Set peak sensor 1 trigger level to1.7 volts

Description: This command sets the trigger level for the 80350A Series Peak Power Sensors. Under INTernal triggering the value is in dBm (-30 to +20 dBm, with default of -20 dBm). When EXTernal triggering is used the value is in Volts (-0.100 to 5.000 V, with a default of 1.700 V). The trigger level must be lower than the signal that is triggering the peak power sensor. Under Internal triggering, this means that a -16 dBm signal must be triggered with a setting that is less than -16 dBm. EXT responds similarly. The input impedance to the EXT trigger port on the back of the sensor housing is not 50 ohms: it is a high impedance to prevent damage from triggering inputs such as 5.0 V TTL signals. If you use a high speed 50 W trigger source, this may cause unwanted signal reflections and noise on the trigger line. An SMB to SMB attenuator is available from Giga-tronics to alleviate this condition. The attenuation reduces the reflected noise, and reduces/eliminates the noisy trigger characteristics. (80340 Series Peak Power Sensors do not support this command.)

Be sure to set the trigger level after the INTernal or EXTernal triggering is selected.

2.5.21.9 SENSe:TRIG:LEV?

Syntax: SENSe<sensor 1 or 2>:TRIGger:LEVel[:MAGnitude]?

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:LEV? ! Set peak sensor 1 trigger level time

Response: Current peak power sensor trigger level setting. A single value in Volts or dBm.

Description: This query reports the current trigger level setting. Under INTernal trigger the value is in dBm. When EXTernal triggering is used the value is in volts.

2.5.21.10 SENSe:TRIG:OFFS

Syntax: SENSe<sensor 1 or 2>:TRIGger:OFFSet[:MAGnitude]

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:OFFS 0 ! Set peak sensor 1 trigger offset time to 0 ! second

Description: This command sets the peak power sensor trigger offset time.
(The 8034XA Series Power Sensors do not support this command).

2.5.21.11 SENSe:TRIG:OFFS?

Syntax: SENSe<sensor 1 or 2>:TRIGger:OFFSet[:MAGnitude]?

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:OFFS? ! Set peak sensor 1 trigger offset time to 0 ! second

Response: Response is the current peak power sensor trigger offset setting; a single value in seconds.

Description: This command sets the peak power sensor trigger offset time.
(The 8034XA Series Power Sensors do not support this command).

2.5.21.12 SENSe:TRIG:TOTA?

Syntax: SENSe<sensor 1 or 2>:TRIGger:TOTAl[:MAGnitude]?

Example: OUTPUT @Pwr_mtr;SENS1:TRIG:TOTA? ! Query peak sensor 1 total trigger delay ! time

Response: Current peak power sensor total trigger delay setting; a single value in seconds

Description: This query reports the peak power sensor total trigger delay time

2.5.22 Averaging

2.5.22.1 SENSe:AVERage

2.5.22.2 SENSe:AVER:COUN

Averaging is applied during normal operating mode. In the normal mode, Avg $_n$ results in $_n$ x 192 samples taken. In the Swift or Burst modes, Avg $_n$ results in 2_n samples taken. Auto averaging takes 4 samples all the time. For auto averaging in the normal mode, the averaging number is level dependent with very low averaging at +20 dBm, and many samples taken at very low levels.

Syntax: SENSe<sensor 1 or 2>:AVERage:COUNt*space<1,2,4,8,16,32,64,128,256,or 512>*

Example: OUTPUT @Pwr_mtr;SENS1:AVER:COUN 16 ! Set sensor 1 average number to 16

Description: This command averages successive sensor readings in a digital averaging filter. As the measured signal level approaches the noise floor of the power sensor, measurement data values begin to fluctuate. To measure very low signal levels (within a few dB of the sensor noise floor), set a large value for averaging such as 128, 256, or 512. Be sure to zero the sensor, CALibration:ZERO, prior to critical measurements.

2.5.22.3 SENSe:AVER:COUN?

Syntax: SENSe<sensor 1 or 2>:AVERage:COUNt?

Example: OUTPUT @Pwr_mtr;SENS1:AVER:COUN? ! Query sensor 1 current averaging value

Description: This query returns the current averaging value of the meter's internal digital averaging filter.

2.5.22.4 SENSe:AVER:COUN:AUTO

Syntax: SENSe<sensor 1 or 2>:AVERage:COUNt:AUTO*space<ON or OFF>*

Example: OUTPUT @Pwr_mtr;SENS2:AVER:COUN:AUTO ON
! Select sensor 2 Auto-Averaging mode

Description: Auto-Averaging optimizes the measurement averaging value for reading stability and update rate when the measurement data is being repeatedly updated. At high power levels (measured), minimum averaging is applied to optimize measurement speed. As the power level drops, additional averaging is applied to steady the measured data reading.

2.5.22.5 SENSe:AVER:COUN:AUTO?

Syntax: SENSe<sensor 1 or 2>:AVERage:COUNt:AUTO?

Example: OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO? ! Query sensor 1 current averaging value

Description: This query returns the current averaging value of the meter's internal digital averaging filter.

2.5.22.6 SENSe:AVER:TCON

Syntax: SENSe<sensor 1 or 2>:AVERage:TCONtrol*space<data acquisition averaging method, MOVing or REPeat>*

Example: OUTPUT @Pwr_mtr;SENS1:AVER:TCON REP ! Set sensor 1 to acquire fresh ! measurement data before averaging

Description: The SENSe:AVERage:TCONtrol commands are instrument level commands that control the internal measurement data averaging of the power meter. Two modes are selectable, MOVing and REPeat.

The MOVing control applies the current measurement to the previously measured data. That is, even if the averaging number is 16, you will get a measurement data point every time a sample is taken, not just after every 16 samples. There is some limited built in level sensing capability. For example, when Auto-Averaging is used and the power is suddenly changed from a low value to a high, MOVing averaging throws out the very low level previous data as the gain range changes and begins averaging fresh data. Thus, wide abrupt changes in power level are not masked by the MOVing averaging. The 58542 can fully range in about 4 ms. MOVing averaging is similar to TR1 mode in typical benchtop power meters.

REPeat averaging performs full averaging every time the 58542 is triggering. Unless the averaging number is set to 1, REPeat is slower than MOVing averaging. REPeat control is similar to TR2 mode in typical benchtop power meters. Fresh data is measured and averaged with each complete measurement cycle.

2.5.22.7 SENSe:AVER:TCON?

Syntax: SENSe<sensor 1 or 2>:AVERage:TCONtrol?

Example: OUTPUT @Pwr_mtr;SENS1:AVER:TCON? ! Query sensor 1 average method

Response: REP or MOV or NOR.

Description: This query returns the average method chosen by the command form or the default.

2.5.23 Relative or Referenced Measurements

2.5.23.1 CALCulate:REFerence

Relative and referenced measurements are used when one measured value needs to be compared to another measured value on the same channel. For example, this function is used when it is desired to monitor the power level variation around an initial turn on or reference set value.

Referenced measurements are performed when finding the 1 dB compression power of an amplifier or using a return loss bridge. When a stable source is used, a relative measurement is useful for measuring the loss through an attenuator with a single sensor or channel.

Due to the ability to set a specific value of reference using the CALC:REF[:MAG] # command, this reference measurement function can also be used to set your own calibration factors. Thus the channel and sensor offset functions can remain dedicated to other setup dependent controls in your programming.

2.5.23.2 CALC:REF:STAT

Syntax: CALCulate<channel 1 or 2>:REFerence:STATe*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;CALC2:REF:STAT ON ! Activate level reference for relative ! measurements

Description: This command activates the meter's level reference to give it the ability to perform relative measurements.

2.5.23.3 CALC:REF:STAT?

Syntax: CALCulate<channel 1 or 2>:REFerence:STATe?

Example: OUTPUT @Pwr_mtr;CALC2:REF:STAT? ! Query ON or OFF status

Response: 0 = off; 1 = on

Description: This query returns the ON or OFF status of the reference function.

2.5.23.4 CALC:REF

Syntax: CALCulate<channel 1 or 2>:REFerence [:MAGnitude] <dB Offset value from -299.999 to 299.999>

Example: OUTPUT @Pwr_mtr;CAL2:REF -30.11 ! Set channel 2 reference offset value in dB

Description: This command sets the value to subtract from the channel measurement value. This command applies to the channel (1, or 2, or 1/2, ...), not the sensor (1 or 2, only).

2.5.23.5 CALC:REF?

Syntax: CALCulate<channel 1 or 2>:REFerence [:MAGnitude]?

Example: OUTPUT @Pwr_mtr;CAL1:REF? ! Query channel reference value

Response: A single number reporting the reference value.

Description: This query reports the current value being subtracted from channel measurements.

2.5.23.6 CALC:REF:COLL

Syntax: CALCulate<channel 1 or 2>:REFerence:COLLect

Example: OUTPUT @Pwr_mtr;CALC2:REF:COLL ! Current Rdg. ==> Ref level.
! Take channel 2 reading as reference value

Description: The CALCulate:REFerence:COLLect command will cause the level of the current power measurement to become the reference level. There is no query form of this command. After you send this command, the channel measurement will be zero, assuming the power levels at the sensor(s) did not change and REF:STAT is ON.

2.5.24 Error Control

Selected basic SCPI syntax and execution error apply to these commands.

Device-specific errors include the following and other -300 level errors.

Use CALC:REF commands only in the NORMal Mode. Attempts to use these commands in the SWIFt and BURSt modes will be ignored and the error, -300, Device-specific error; Normal mode is off. will be generated.

2.5.25 Offsets

2.5.25.1 SENSe:CORRection

Sensor power offsets apply to the individual sensor. Use sensor offsets to account for the loss of attenuators, which are commonly used during measurement to reduce standing waves to improve measurement accuracy. If you are measuring high power signals (>100 mW), but are not using Giga-tronics high power sensors, you will need to use a power attenuator between the high power output and the sensor input to prevent damage. Enter the value of attenuation as a sensor offset, and the 58542 will automatically respond with the actual power level output in its measurement data.

2.5.25.2 SENSe:CORRection:OFFset:COLLect

Syntax: SENSe<sensor 1 or 2>:CORRection:OFFSet:COLLect

Example: OUTPUT @Pwr_mtr;SENS1:CORR:OFFS:COLL
! Take sensor 1 current reading as offset
! value

Description: This command enters the current power level reading as the offset value.

2.5.25.3 SENSe:CORRection:OFFset

Syntax: SENSe<sensor 1 or 2>:CORRection:OFFSet[:MAGnitude]*space*
<offset value in dB, -99.99 to 99.99>

Example: OUTPUT @Pwr_mtr;SENS2:CORR:OFFS 10.2 ! Compensate sensor 2 for 10.2 dB
! attenuation

Description: This command enters a specific sensor offset value.

2.5.25.4 SENSe:CORRection:OFFset?

Syntax: SENSe<sensor 1 or 2>:CORRection:OFFSet[:MAGnitude]?

Example: OUTPUT @Pwr_mtr;SENS2:CORR:OFFS? ! Query sensor 2 offset value

Response: 0.0000E+00 for 0 dB.

Description: This query reports the current dB value of the sensor offset which is being subtracted from measurements.

2.5.25.5 SENSe:CORRection:OFFset:STATe

Syntax: SENSe<sensor 1 or 2>:CORRection:OFFSet:STATe*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;SENS2:CORR:STAT ON ! Enable sensor 2 offset correction

Description: This command activates and deactivates the sensor offset function.

2.5.25.6 SENSe:CORRection:OFFset:STATe?

Syntax: SENSe<sensor 1 or 2>:CORRection:OFFSet:STATe?

Example: OUTPUT @Pwr_mtr;SENS2:CORR:STAT? ! Query sensor 2 offset function status

Description: This query reports the ON or OFF status of the sensor offset function.

2.5.26 SRQ & Status Monitoring

***CLS**
***ESE**
***ESR?**
***OPC and *OPC?**
***SRE**
STATus:OPERation
STATus:PREset
***STB?**

2.5.26.1 *CLS

Example: OUTPUT @Pwr_mtr;*CLS ! Clear SRQ and status byte registers

Description: *CLS is the clear status command defined by IEEE 488.2. This command clears all of the status bytes to the value 0. After a service request interrupt is transmitted from the 58542 to the controller, use the *STB command to read the status byte from the 58542. Then reset the SRQ and use *CLS clear status command to reset the numeric status indication of the status byte/registers to 0 (all bits will be 0).

*CLS does not affect the enabled or disabled status of the status byte/register masks. For example, if you have bit 8 of the operation register enabled before sending *CLS, it will remain enabled afterward. *CLS also clears the output queue (takes 0.3 sec to complete this function).

2.5.26.2 *ESE

Syntax: ESE*space*<event status register value, 0 to 255>

Example: OUTPUT @Pwr_mtr;*ESE 16 ! Enable bit 4 of event status register mask

Description: This command sets/clears the event status register enable mask.

2.5.26.3 *ESE?

Syntax: *ESE?

Example: OUTPUT @Pwr_mtr;*ESE? ! Query currently enabled bits of event status
! register mask

Response: Sum of the bits set in the event status enable register.

Description: This query returns the mask of the event status register.

2.5.26.4 *ESR?

Example: OUTPUT @Pwr_mtr;*ESR? ! Return event status register value

Description: This query returns the current status register value.

2.5.26.5 *OPC

Syntax: *OPC

Example 1: OUTPUT @Pwr_mtr;*OPC ! *OPC allows one time SRQ enable

Description: *OPC determines when an operation is completed. This command is generally used to monitor the completion of long measurement sequences. It sets the Operation Complete bit in the event status register upon completion of operation.

2.5.26.6 *OPC?

Syntax: *OPC?

Example 2: OUTPUT @Pwr_mtr;*OPC? ! send a 1 upon operation complete

Response: A 1 is returned when all operations are complete.

Description: This command automatically returns a 1 all pending operations are complete. This command is typically used for a frequency measurement where large averaging is used or 58542/Sensor calibration.

2.5.26.7 *SRE

Syntax: *SRE*space*<event status register value, 128, 64, 32, 16, 8, 4, 2, 1>

Example: OUTPUT @Pwr_mtr;*SRE 32 ! Enable bit 5 of Status Byte register

Description: This command sets the mask of the Status Byte register.

2.5.26.8 *SRE?

Syntax: *SRE*?*

Example: OUTPUT @Pwr_mtr;*SRE? ! Query Bits Enable of Status Byte register

Response: The status register mask value.

Description: This query returns the mask of the Status Byte register.

2.5.26.9 STAT:OPER

Syntax: STATus:OPERation[:EVENt]?

Example: OUTPUT @Pwr_mtr;STAT:OPER? ! Query operation event register result

Description: This query returns the operation event register result.

Syntax: STATus:OPERation:ENABle*space*<event register value 0 through 65535>

Example: OUTPUT @PWR_mtr;STAT:OPER:ENAB 1 ! Sets the Event Register enable mask

Description: This command sets the enable mask, which allows true conditions in the event register to be reported in the summary bit.

2.5.26.10 STAT:PRES

Syntax: STATus:PRESet

Example: OUTPUT @Pwr_mtr;STAT:PRES ! Clears all the status register value

Description: This command clears all status registers.

2.5.26.11 *STB?

Example: OUTPUT @Pwr_mtr;*STB? ! Query status byte register

Description: This query returns status byte register result.

The SCPI status reporting structure includes the IEEE 488.2 Status Registers. Please note that bit 5 of the OPERation Status Register is only used during BURSt and SWIFt modes. NORMal mode does not use this function.

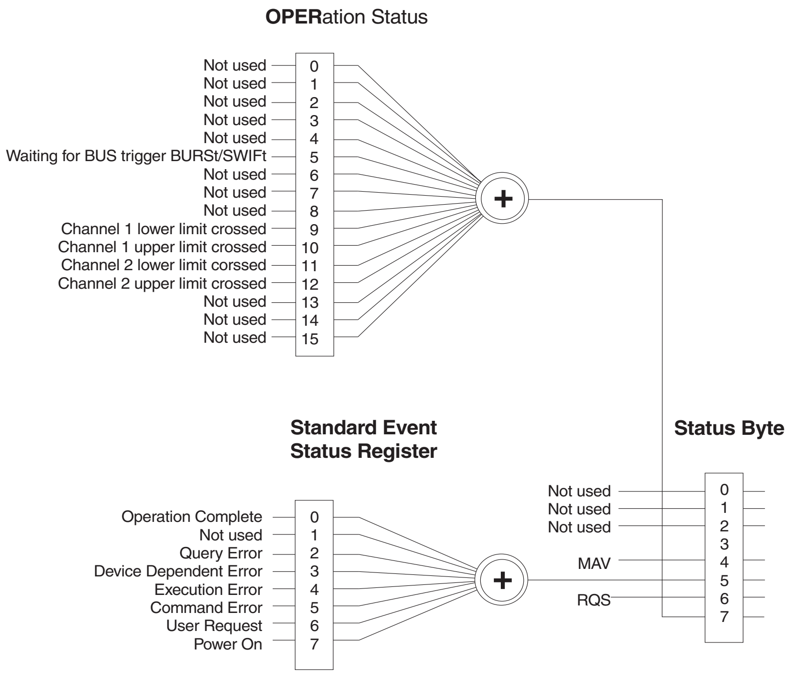


Figure 2-10: The SCPI Status Structure Registers

2.5.27 Event Status Register

The *ESE command is used in combination with group commands that form the meter's service request system. These commands and their responses are almost identical to the IEEE 488 (GPIB) SRQ service request command structures. The *ESE command is used to enable bits of the event status register mask.

The *ESE command is one of the commands you can use to monitor the status of the power meter. Together with the status byte (also see commands *STB? or *SRE?) and the operation status register (STATus:OPERation), the event status register provides information on several critical 58542 functions and error conditions.

7	6	5	4	3	2	1	0
Power On Value = 124	0 Value = 64	Command Error Value = 32	Execution Error Value =16	Device-Dependent Error Value = 8	Query Error Value = 4	0 Value = 2	Operation Complete Value = 1

*

NOTE: *The logical value in the register is used both to enable a bit's event function and to enable that bit of the event status register mask to report the event status (following the *ESR? command).*

The event status register is 8 bits long and is structured as follows:

Bits 1 and 6 are not used. When bit 3 is a 1, a device-dependent error has occurred. When bit 4 is a 1, an execution error has occurred. When bit 5 is a 1, a command error has occurred. Bit 7 is a 1 when the 58542 is turned ON. The 58542 does not have a standby mode; therefore, it is probably not useful to enable bit 7 of the event status register mask (by sending the command *ESE 128).

In the default state, the meter's event status register is masked, or set to off. None of the bits are enabled. If an execution error occurs, a service request will not be sent to the controller and the event status register will remain set to 0. The *ESE command is used to enable individual bits. These bits must be enabled individually. They can be cleared or turned off as a group by sending the *CLS command.

To use the event status register, bit 5 of the status byte must also be enabled by sending the command *SRE 32.

*ESR? reports the enabled settings of the event status register mask. The *ESR? query returns the current value of the event status register.

*

NOTE: *A Functional Description of the service request system can be found under *SRE in the IEEE Standard Codes, Formats, Protocols, and Common Commands (ANSI/IEEEE STD 488.2-1987) publication.*

*

*The logical value in the register is used both to enable a bit's event function and to enable that bit of the event status register mask to report the event status (following the *ESR? command).*

2.5.28 Status Byte Register

7	6	5	4	3	2	1	0
Operational Status	Require Service (RQS)	Event Status	Message Available	0	0	0	0

Notes:

1. The condition indicated in bits 1 through 5 must be enabled by the Service Request Mask to cause a Service Request Condition. The mask is set with the *SRE command followed by ASCII characters. The value of the byte is determined by summing the weight of each bit to be checked.
2. The RQS (bit 6) is true when any of the conditions of bits 4, 5, and 7 are enabled and occur.
3. Bits remain set until the Status Byte is cleared.

2.5.29 Min/Max Configuration & Monitoring

CALCulate:MAXimum
CALCulate:MINimum

MIN and MAX channel monitoring records the minimum and maximum variation of channel amplitude over time. Time zero is set by setting the MIN or MAX STATe to ON. The default Preset and turn on configuration is ON. After setting to ON, the minimum and maximum value of the measured value on that channel will be recorded. To reset the MIN and MAX value back to the current value, you only need to set the STATe to ON again. The STAT OFF does not need to be sent during reset of MIN and MAX values. Individual sensor minimum and maximum power level variations can be monitored only if that sensor is defined directly to a channel with POW 1 or POW 2. MIN and MAX channel monitoring is only available in Normal Mode. BURSt and SWIFt modes do not perform MIN and MAX monitoring.

2.5.29.1 CALC:MAX

Syntax: CALCulate<channel 1 or 2>:MAXimum:STATe*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;CALC2:MAX:STAT ON ! Enable channel 2 maximum tracking

Description: ON activates the MAX monitoring function. The maximum value of the channel (not sensor) is monitored until the value is reset by CALC#:MAX:STAT ON or CALC#:MAX:STAT OFF is sent.

Syntax: CALCulate<channel 1 or 2>:MAXimum:STATe?

Example: OUTPUT @Pwr_mtr;CALC2:MAX:STAT? ! Query channel 2 maximum mode

Response: 1 for ON or 0 for OFF.

Description: Reports ON or OFF status of MAX monitoring.

Syntax: CALCulate<channel 1 or 2>:MAXimum[:MAGnitude]?

Example: OUTPUT @Pwr_mtr;CALC2:MAX? ! Query channel 2 maximum value in dBm

Response: Highest power reading since CALC2:MAX:STAT ON was sent.

Description: This command reports the value maintained as the maximum in the 58542 max monitor.

2.5.29.2 CALC:MIN

Syntax: CALCulate<channel 1 or 2>:MINimum:STATe*space*ON or OFF

Example: OUTPUT @Pwr_mtr;CALC1:MIN:STAT ON ! Enable channel 1 minimum tracking

Description: This command activates the minimum channel value monitor.

Syntax: CALCulate<channel 1 or 2>:MINimum:STATe?

Example: OUTPUT @Pwr_mtr;CALC1:MIN:STAT? ! Query channel 1 minimum mode

Response: 1 for ON or 0 for OFF

Description: This queries the activation or deactivation of the minimum channel value monitor.

Syntax: CALCulate<channel 1 or 2>:MINimum [:MAGnitude]?

Example: OUTPUT @Pwr_mtr;CALC1:MIN? ! Query channel minimum value in dBm

Response: Lowest power reading since CALC2:MIN:STAT ON was sent.

Description: This command reports the value maintained as the minimum in the 58542 min monitor.

2.5.30 Limit Line Configuration & Monitoring

2.5.30.1 CALCulate:LIMit

The CALCulate:LIMit commands specify and query the status of power measurement limit values and limit line pass/fail checking. This allows the 58542 to monitor measured values and determine if the values are outside certain limits or above/below a single limit. The upper limit cannot be specified any lower than the lower limit; meaning that an exclusion zone of values cannot be specified.

Limit values can be specified separately for either of the two software calculation channels - 1 and 2. These channels can be specified to correspond to power sensors 1 and 2 (default) or as a combination of the power sensors. For example, set limit lines to monitor overload or under range conditions on an amplifier’s output power and gain by specifying channel 1 as sensor 1 limits checking and channel 2 as Sensor 2/Sensor 1. Power sensor assignments to channel measurement definition is also part of the CALCulate Subsystem. Limit monitoring can not be performed for individual sensors unless a channel is configured for single sensor, POW 1 or POW 2 operation.

Upper and Lower limits can be active simultaneously.

Unless serial requests are enabled, the CALCulate:LIMit commands can not be configured to automatically alert the controller directly during a limit violation; the 58542 must be queried to receive information regarding pass/fail status of limit line violations.

Automatic notification of limit line violations is accomplished using the status byte and the operation register. A controller can be notified of a limit line violation via the request service, *SRE command. After the controller receives the request service, query the event status register. Check the status register for a limit line violation or send CALCulate:LIMit:FAIL? to check if a limit line is being or has been violated. 1 indicates a limit line violation. 0 indicates the channel measurement is within the limit lines.

2.5.30.2 CALCulate:LIMit:CLE

Syntax: CALCulate<channel 1 or 2>:LIMit:CLEar[:IMMediate]

Example: OUTPUT @Pwr_mtr;CALC1:LIM:CLE ! Reset channel 1 limit violation indicator to 0

Description: This command resets the limit line pass/fail indicator to 0.

2.5.30.3 CALC:LIM:FCO?

Syntax: CALCulate<channel 1 or 2>:LIMit:FCOunt?

Example: OUTPUT @Pwr_mtr;CALC1:LIM:FCO? ! Query number of channel 1 limit failures

Response: Single number; the number of times the limit lines were exceeded.

Description: This command reports the number of times a limit line has been exceeded since the limit checking function was set to ON, or the limit line monitor was cleared with CALC#:LIM:CLE.

2.5.30.4 CALC:LIM:FAIL?

Syntax: CALCulate<channel 1 or 2>:LIMit:FAIL?

Example: OUTPUT @Pwr_mtr;CALC1:LIM:FAIL? ! Check for channel 1 limit line violation

Response: 0 = OK; 1 = fail.

Description: This command reports whether or not a limit line has been exceeded since the limit checking function was set to ON, or the limit line monitor was cleared with CALC#:LIM:CLE.

2.5.30.5 CALC:LIM:LOW

Syntax: CALCulate<channel 1 or 2>:LIMit:LOWer*space*<numeric value in dB from -299.99 to 299.99>

Example: OUTPUT @Pwr_mtr;CALC1:LIM:LOW -50.0 ! Set channel 1 lower limit line to -50 dBm

Description: This command specifies the lower limit line power level. The value should allow for any offset values currently in use. The default value is -299.99 dB or dBm.

2.5.30.6 CALC:LIM:LOW?

Syntax: CALCulate<channel 1 or 2>:LIMit:LOWer?

Example: OUTPUT @Pwr_mtr;CALC1:LIM:LOW? ! Query channel 1lower limit checking

Response: 0 = off; 1 = on

Description: This command queries the value of the lower limit line.

2.5.30.7 CALC:LIM:STAT

Syntax: CALCulate<channel 1 or 2>:LIMit:STATe*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;CALC1:LIM:STAT ON ! Enable channel 1 upper and lower limit ! checking

Response: 0 = off; 1 = on

Description: This command activates and deactivates the limit line checking. The default condition is OFF.

2.5.30.8 CALC:LIM:STAT?

Syntax: CALCulate<channel 1 or 2>:LIMit:STATe?

Example: OUTPUT @Pwr_mtr;CALC1:LIM:STAT? ! Query channel 1 ON or OFF status

Response: 1 for ON or 0 for OFF

Description: This command reports the activated or deactivated status of limit line checking.

2.5.30.9 CALC:LIM:UPP

Syntax: CALCulate<channel 1 or 2>:LIMit:UPPer*space*<numeric value in dBm or dB from -299.99 to 299.99>

Example: OUTPUT @Pwr_mtr;CALC1:LIM:UPP 17.0 ! Set channel 1 upper limit line to 17 dBm

Description: This command specifies the upper limit line power level. The value should allow for any offset values currently in use. The default value is 299.99 dB or dBm.

2.5.30.10 CALC:LIM:UPP?

Syntax: CALCulate<channel 1 or 2>:LIMit:UPPer?

Example: OUTPUT @Pwr_mtr;CALC1:LIM:UPP? ! Query channel 1upper limit line setting

Description: This query returns the value of the upper limit line.

2.5.31 Analog Output

MEMory:CHANNel
MEMory:UNIT
MEMory:POWer
MEMory:VOLTage
OUTPut:ANAlog

The ANALOG OUT BNC connector on the front panel can be configured to output a voltage that corresponds to the power levels on one of the channels. This is useful for applications such as source leveling or printing to a chart recorder.

The Analog Output operates only in NORMal Mode. It is automatically deactivated during SWIFt or BURSt modes, and comes back when operation is returned to the NORMal Mode.

2.5.31.1 MEM:CHANnel

Syntax: MEMory[:TABLe]:CHANNel*space*<channel 1 or 2>

Example: OUTPUT @Pwr_mtr;MEM:CHAN 2 ! Set analog out to channel 2

Description: This command selects which channel will be present on the analog output connector.

2.5.31.2 MEM:CHAN?

Syntax: MEMory[:TABLe]:CHANNel?

Example: OUTPUT @Pwr_mtr;MEM:CHAN? ! Query analog out channel number

Description: This query report which channel will be present on the analog output connector.

2.5.31.3 MEM:SEL

Syntax: MEMory[:TABLe]:SELect*space*<ANALOGout,VPROPF1,VPROPF2>

Example: OUTPUT @Pwr_mtr;MEM:SEL VpROPF1 ! Select memory table VpROPF1
! for editing in following lines

Description: This command selects between one of three editable tables. Using the meter's Analog Output capability requires you to configure the numerically linear power measurement to voltage output relationship by defining the corresponding end points of power and voltage.

2.5.31.4 MEM:SEL?

Syntax: MEMory[:TABLe]:SELect?

Example: OUTPUT @Pwr_mtr;MEM:SEL? ! Query selected memory table

Response: The table currently editable: ANALOG, $V_{PROPF}1$, or $V_{PROP}F2$.

Description: This query reports the currently editable table in the 58542.

2.5.31.5 MEM:UNIT

Syntax: MEMory[:TABLe]:UNIT*space*<choice of units dBm or Watt>

Example: OUTPUT @Pwr_mtr;MEM:UNIT dBm ! Set analog out power unit

Description: This command selects between one of two configurations, log units or linear units. In either case the voltage output will be numerically linear. Be sure to set the units control properly (dBm or W) before trying to set the numeric values.

2.5.31.6 MEM:UNIT?

Syntax: MEMory[:TABLe]:UNIT?

Example: OUTPUT @Pwr_mtr;MEM:TABL:UNIT? ! Query analog out power unit

Description: The units for a channel analog output configuration.

2.5.31.7 MEM:POWer

Syntax: MEMory[:TABLe]:POWer*space*<start value>*comma*<stop value>

Example 1: OUTPUT @Pwr_mtr;MEM:POW -70,20 ! Set analog out power range (in dBm)
! from -70 to 20 dBm

Example 2: OUTPUT @Pwr_mtr;MEM:POW 0,1e2 ! Set analog out power range (in Watts)
! from 0 to 0.01 Watts

Description: If log units are selected for analog output control, this command selects the beginning and end points of the power levels which will be assigned corresponding voltages at the analog output. If linear units are selected for analog output control, this command selects the beginning and end points of the power levels which will be assigned corresponding voltages at the analog output. Also see MEM:VOLT

2.5.31.8 MEM:POWer?

Syntax: MEMory[:TABLe]:POWer?

Example: OUTPUT @Pwr_mtr;MEM:POW? ! Query analog out power range

Response: Two dBm or Watt values separated by a comma indicating analog output range.

Description: This query reports the beginning and end points of the power levels of the analog output configuration. When log units are selected, the units are dB or dBm. When linear units are selected the units are Watts.

2.5.31.9 MEM:VOLTage

Syntax: MEMory[:TABLe:VOLTage*space*<start value from -10 to +10 V>*comma*<stop value from -10 to +10 V>

Example: OUTPUT @Pwr_mtr;MEM:VOLT -7,2 ! Set analog out voltage range from ! -8 to 2 volts corresponding to power

Description: This command is not units dependent. It selects the beginning and end points of the voltage levels which you desire to be present at the analog output. These voltages correspond to the power end points that are defined with either MEM:VOLT or MEM:POW.

2.5.31.10 MEM:VOLTage?

Syntax: MEMory[:TABLe:VOLTage?

Example: OUTPUT @Pwr_mtr;MEM:VOLT? ! Query analog out voltage range

Response: Two voltage values separated by a comma indicating analog output range.

Description: This command is not units dependent. It selects the beginning and end points of the voltage levels which you desire to be present at the analog output. These voltages correspond to the power end points that are defined with either MEM:VOLT or MEM:POW.

2.5.31.11 OUTP:ANAlog

Syntax: OUTPut[:BNC]:ANAlog[:STATe]*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;OUTP:ANA ON ! Enable analog out function

Description: This command activates and deactivates the analog output. The analog output will not operate unless this control is set to ON.

2.5.31.12 OUTP:ANAlog?

Syntax: OUTPut[:BNC]:ANAlog[:STATe]?

Example: OUTPUT @Pwr_mtr;OUTP:ANA? ! Query analog out function status

Response: 1 for ON or 0 for OFF.

Description: This command activates and deactivates the analog output. The analog output will not operate unless this control is set to ON.

2.5.32 Saving & Recalling Configurations

*RCL
*SAV

The 58542 has 21 instrument state memory registers. Registers 1 through 20 are available for store and recall. Register 0 contains the previous state of the instrument and can be used to toggle between two different instrument configuration states.

Instrument configuration can be saved to registers 1 through 20.

CAUTION

Any configuration items which are not listed under the *RST or PRESet conditions are not savable. Make sure all aspects of your configuration are savable. For example, sensor power sweep calibration curves can not be saved in the configuration memory registers. Sensors must be calibrated to the 58542 power meter each time a new sensor is attached.

2.5.32.1 *RCL

Syntax: *RCL*space*<memory location number 0 to 20>

Example: OUTPUT @Pwr_mtr;*RCL 19 ! Recall 58542 register 19

Description: Recalls instrument configuration. 0 is the PRESet configuration.

2.5.32.2 *SAV

Syntax: *SAVe*space*<memory location number 1 to 20>

Example: OUTPUT @Pwr_mtr;*SAV 20 ! Save at 58542 register 20

Description: Saves current configuration to memory. You cannot save a configuration to memory position 0.

2.5.33 Halting Operation

ABORt

Example: OUTPUT @Pwr_mtr;ABOR ! Halts measurement & triggering

Description: This command stops operation, but it does not interrupt the completion of the current action. For example, sensor calibration is not interrupted. Burst mode data collection is not interrupted.

*

NOTE: *When using the 8035XA Peak Sensor, if a Time-Out occurs due to the sensor not triggerng (i.e., Level too Low) then send 'Abort' to clear the meter.*

2.5.34 Preset Configuration

***RST**
STATus:PRESet
SYSTem:PRESet

2.5.34.1 *RST

Example: OUTPUT @Pwr_mtr;*RST ! Reset 58542 configuration

Description: This command resets the 58542 configuration to a known condition (see Table 2-4). These are not the power ON conditions. The 58542 has an internal battery which powers a non-volatile memory chip to retain configuration information. The only configuration that will change between power OFF and power ON is noted at the end of the table.

2.5.34.2 STATus:PRESet

Example: OUTPUT @Pwr_mtr;STAT:PRES ! Clears all the status register value

Description: This command resets the 58542 Status information buffers. See Table 2-4. Note that these are not the power ON conditions.

2.5.34.3 SYSTem:PRESet

Example: OUTPUT @Pwr_mtr;SYST:PRES ! Reset 58542 configuration

Description: This command resets the 58542 configuration to a known condition. SYST:PRES is identical in function to *RST. (Command Default, Minimum, Maximum)

Table 2-4: Reset & Power On Default Commands

Command	Default	Minimum	Maximum
CALCulate1[:CHANnel]: CALCulate2[:CHANnel]:	POWer 1 POWer 2		
CALCulate1-2: LIMit STATe UPPer LOWer FAIL? FCOunt? MAXimum: STATe [MAGnitude] MINimum: STATe [MAGnitude] MODE REFerence: STATe [MAGnitude] STATe UNIT	 OFF 299.999 -299.999 0 0 OFF 299.999 OFF -299.99 NORmal OFF 0 ON dBm	 -299.999 -299.999 -299.999 -29.999 -299.99	 299.999 299.999 299.999 29.99 299.99
INITiate:CONTinuous	OFF	,	
MEASure1-2[:SCALar:POWer]	With Auto average on	,	
MEMory[:TABLe]: CHANnel FREQuency POWer SELect ANALOGout SLOPe VOLTage UNIT	 1 0 -80,20 0 -10,10 dBm	 1 0 -80,-80 0 -10,-10	 2 40e9 20,20 10e9 10,10
OUTput: ROScillator[:STATe] [BNC:]ANAlog[:STATe]	 OFF OFF	,	
SENSe1-2: AVERage: COUNt: AUTO TCONtrol CORRection: FREQ OFFset: STATe [MAGnitude] VPROpf[:STATe] TRIGger: DELay: STATe [MAGnitude] LEVel (if INT trig) LEVel (if EXT trig) SOURce	 1 ON MOVing 5e7 OFF 0 OFF OFF 1e-6 -20.0 1.7 IMMediate	 1 0 -99.999 -5e-8 -30.0 0.1	 512 4e10 99.999 104e-3 20.0 5.000
TRIGger: COUNt DELay MODE SOURce	 1 0.000 POST IMMediate	 0 0.000	 128000 5.000

Table 2-4: Reset & Power On Default Commands (Continued)

Command	Default	Minimum	Maximum
UNIT1-2[:POWer]	dBm		
POWER ON will override the internal battery back up memory and reset only the following configuration items as shown below. All other configuration items will remain the same as was true before power OFF.			
CALCulate1-2:MODE	NORMal		
INITiate:CONTinuous	OFF		
MEMory[:TABLe]:SELect	ANALOGout		
OUTput:ROSCillator[:STATe]	OFF		
TRIGger: COUNt DELay MODE SOURce	 1 0.000 POST IMMediate	 0 0.000	 128000 5.000

2.5.35 Identification Commands

*IDN
SENSe:CORRection:EEPROM:TYPE?
SYSTem:VERSion

Identification commands ensure that the power meter and sensor are appropriate for your test program. The sensor identification commands allow you to monitor what model of sensor is attached and ensure that the sensor has been properly calibrated to the power meter.

*

NOTE: *Example programs for Identification Commands are contained in Appendix A of this manual.*

2.5.35.1 *IDN?

Example: OUTPUT @Pwr_mtr;*IDN? ! Query inst. mfgr & model #

Response: Giga-tronics, 58542,0,1.09 where 1.09 is software version

Description: The *IDN? is the identify query command defined by IEEE 488.2. Upon receipt of this command, the power meter will output a string that identifies itself as the Giga-tronics 58542 and indicates the firmware version number.

2.5.35.2 SENS:CORR:EEPROM:TYPE?

Syntax: SENSe<sensor 1 or 2>:CORRection:EEPROM:TYPE?

Example: OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:TYPE? ! Query sensor 1 EEPROM type

Response: 80301,1818436 which is model number and serial number respectively

Description: This command returns sensor model and serial number information.

2.5.35.3 SYST:VERSion

Syntax: SYSTem:VERSion?

Example: OUTPUT @Pwr_mtr;SYST:VERS? ! Query SCPI version

Response: 1990.0 (Is this correct?)

Description: This query returns the compiled SCPI version.

2.5.36 Calibrator Controls

2.5.36.1 OUTP:ROSC

Syntax: OUTPut:ROSCillator[:STATe]*space*<ON or OFF>

Example: OUTPUT @Pwr_mtr;OUTP:ROSC ON ! Turn ON Calibrator Oscillator

Description: The reference oscillator (Calibrator port) connection is on the front panel of the power meter. The OUTPut:ROSCillator ON command turns on this 0 dBm, 50 MHz output. The reference oscillator power level should be calibrated annually. Please refer to the Calibration Procedures in Chapter 4 for additional details.

2.5.36.2 OUTP:ROSC?

Syntax: OUTPut:ROSCillator[:STATe]?

Example: OUTPUT @Pwr_mtr;OUTP:ROSC? ! Query ON or OFF status

Response: 1 or 0 (1 = ON)

Description: This query command returns the OFF or OFF status of the meter's built-in reference oscillator.

2.5.37 Self-Test

2.5.37.1 *TST?

Syntax: *TST?

Example: OUTPUT @Pwr_mtr;*TST? ! Query self-test result

Response: 0 if all is OK, otherwise, it returns a value of 1.

Description: If an error occurred during Self-Test, the amber TRIG LED will remain on until cleared by sending the SYST:ERR? command which will return -300 Device-Specific Error; Self Test Failed.
Table 2-5 lists the indications and limitations that may occur when the Self-Test command is applied to the 58542 instrument. The Result, Minimum and Maximum indications are in millvolts.

Table 2-5: Self-Test Error Indications & Limitations

Error Number	Result	Minimum	Maximum	Test Description
100	0	0	50	**OUTPUT DAC** This test steps the analog output DAC from 0 volts to 10 volts in 1 volt steps, and measures the resultant output voltage with the system ADC. It then compares the results with the limits shown to the left and, if it is outside any limit, it will set the error number corresponding to that voltage level.
101	1000	950	1050	
102	2000	1900	2100	
103	3000	2850	3150	
104	4000	3800	4200	
105	5000	4750	5250	
106	6000	5700	6300	
107	7000	6650	7350	
108	8000	7600	8400	
109	9000	8550	9450	
110	10000	9500	10500	
200	0	0	50	**OFFSET DAC** This test steps the sensor input amplifier offset DAC from 0 volts to 5 volts in 1 volt steps, and measures the resultant output voltage with the system ADC. It then compares the results with the limits stated to the left and, if outside any limit, it will set the error number corresponding to that voltage level.
201	1000	950	1050	
202	2000	1900	2100	
203	3000	2850	3150	
204	4000	3800	4200	
205	5000	4750	5250	
400	0	0	50	**CALIBRATOR DAC** This test steps the calibrator control DAC from 0 to 10 volts in 1 volt steps and measures the resultant output voltage with the system ADC. It then compares the results with the limits shown to the left and if outside any limit it will set the error number corresponding to that voltage level.
401	1000	950	1050	
402	2000	1900	2100	
403	3000	2850	3150	
404	4000	3800	4200	
405	5000	4750	5250	
406	6000	5700	6300	
407	7000	6650	7350	
408	8000	7600	8400	
409	9000	8550	9450	
410	10000	9500	10500	
600	0	50	50	**OFFSET AMP 1** This test steps the sensor input amplifier offset DAC from 0 to 5 volts in 1 volt steps and measures the resultant output voltage from the first amplifier stage with the system ADC. It then compares the results with the limits shown to the left and if outside any limit it will set the error number corresponding to that voltage level.
601	1000	950	1050	
602	2000	1900	2100	
603	3000	2850	3150	
604	4000	3800	4200	
605	5000	4750	5250	
800	0	0	50	**GAIN AMP 2** This test steps the sensor input amplifier offset DAC from 0 volts to 5 volts in 1 volt steps, sets the second stage gain to 1 and measures the second stage output voltage with the system ADC. It then compares the results with the limits shown to the left and if outside any limit it will set the error number corresponding to that voltage level.
801	1000	950	1050	
802	2000	1900	2100	
803	3000	2850	3150	
804	4000	3800	4200	
805	5000	4750	5250	

Table 2-5: Self-Test Error Indications & Limitations (Continued)

Error Number	Result	Minimum	Maximum	Test Description
1000	0	0	50	**THRU FILTER** This test steps the sensor input amplifier offset DAC from 0 to 5 volts in 1 volt steps, and measures the filter amplifier output voltage with the system ADC. It then compares the results with the limits shown to the left and if outside any limit it will set the error number corresponding to the voltage level.
1001	1000	950	1050	
1002	2000	1900	2100	
1003	3000	2850	3150	
1004	4000	3800	4200	
1005	5000	4750	5250	
Gain = 1, Amp 2: 1200	1000	950	1050	**GAIN = 1, AMP 2** This test sets the sensor input amplifier offset DAC to 1 volt and measures the output of the second stage amplifier with its gain set to 1. It then compares the results with the limits shown to the left and if outside any limit it will set the error number corresponding to that voltage level. Then it sets the gain of the second stage to 8 and measures the output of that stage. Any out of tolerance voltage will be indicated by a corresponding error number. This checks the gain of this stage at a gain of 8.
Gain = 8, Amp 2: 1201	8000	7200	8800	
Gain = 1, Amp 2: 1400	100	90	110	**GAIN = 1, AMP 2** This test sets the sensor input amplifier offset DAC to 0.1 volt and measures the output of the second stage amplifier with its gain set to 1. It then compares the results with the limits shown to the left and if outside any limit it will set the error number corresponding to that voltage level. Then it sets the gain of the second stage to 64 and measures the output of that stage. Any out of tolerance voltage will be indicated by a corresponding error number. This checks the gain of the second stage at a gain of 64.
Gain = 64, Amp 2: 1401	6400	5760	7040	
1600	9000	8000	10000	**TOO HIGH** This test checks the upper limit for the auto-range system by applying a voltage from the offset DAC. The output of the second stage is monitored while also monitoring the ! too-high comparator output. If the comparator output comes true outside of the voltage limits specified to the left the corresponding error number is set.
1800	1000	700	1100	**TOO LOW** This test checks the lower limit for the auto-range system by applying a voltage from the offset DAC. The output of the second stage is monitored while also monitoring the ! too-low comparator output. If the comparator output comes true outside of the voltage limits specified to the left the corresponding error number is set.
2000	7000	6300	7700	**CAL HEATER** This measures the calibrator thermistor heater voltage. This is half of the actual voltage on the heater transistor collector which will be about 0.5 volts below the level of the 15 volt supply after the heater is stable. Since this takes about a minute after power on the POST does not check this. However the command *TST? does.
2100	0	0	500	**SW +5 OFF** This test measures the switched 5 volt logic supply applied to the sensor when it is switched off. Whenever a peak power sensor (8034X or 8035XA) is connected, this test will be bypassed.

Table 2-5: Self-Test Error Indications & Limitations (Continued)

Error Number	Result	Minimum	Maximum	Test Description
2101	5000	4500	5500	**SW +5 ON** This test measures the switched 5 volt logic supply applied to the sensor when it is switched on. Whenever a peak power sensor (8034X or 8035XA) is connected, this test will be bypassed.
3000		0x3c0000	0x3FFFFF	**RAM BANK 1** This does a byte by byte check of the optional extra 128K RAM at addresses 3c0000 to 3fffff hex. An error at any address will result in an error number of 3000. If the optional RAM is not present (which is tested by checking the first byte for write/read cycle ability), then this test is aborted without an error flag.
3001		0x400000	0x43FFFF	**RAM BANK 2** This does a byte by byte check of the standard RAM at addresses 400000 to 43ffff hex. This is done in a non-destructive manner by saving the data that is present and then writing a pattern, reading it back, checking it for validity, and then restoring the original. An error at any address will result in an error number of 3001.
3100		0x300000	0x37FFFF	**ROM** This does a complete checksum of the program ROM at addresses 300000 to 37ffff hex and compares it with the correct value stored in the ROM. Any error will result in an error number of 3100.

2.5.38 Error Messages

2.5.38.1 SYSTem:ERRor?

Syntax: SYSTem:ERRor?

Example: OUTPUT @Pwr_mtr;SYST:ERR? ! Query system error message

Description: This command reads error messages from the error buffer. Use *CLS and CLEAR @PM_address to clear the SYST:ERR buffer just prior to entering measurement configurations and measurement routines. A listing of compatible standard SCPI error and device-specific errors follows.

Table 2-6: SCPI Standard Error Messages

Error Number	Description
0	No Error
-7	Invalid Error Number
-108	Parameter Not Allowed
-111	Header Separator Error
-113	Undefined Header;- - - -
-120	Numeric Data Error
-200	Execution Error
-211	Trigger Ignored
-213	Init Ignored
-214	Trigger Deadlock
-230	Data Corrupt or Stale
-300	Device-specific errors (see Table 2-7)

Table 2-7: Device Specific Error Messages

Command	Error Message	Example of Problem
ABORt		
CALCulate1:DATA?	Burst mode is off	Not burst mode; Command requires BURSt mode
CALCulate1:DIFFerence 1,1 CALCulate1:RATio 1,1	Conflict in channel configuration	Same sensor
CALCulate1:LIMit:LOWer :UPPer	Conflict between upper and lower limits	Upper is smaller than lower or lower is greater than upper
CALCulate1:MAXimum:STATe	Channel is not valid	No sensor or sensor not calibrated
CALCulate1:MINimum:STATe	Channel is not valid	No sensor or sensor not calibrated
CALCulate1:MODE	No valid sensor Channel is not valid	No sensor No sensor or sensor not calibrated
CALCulate1:REFerence:COLLect	Normal mode is off Channel is not valid	Burst mode setup No sensor or sensor not calibrated
CALibrate1	No sensor Sensor not connected to calibrator Sensor calibration error	No sensor Calibration error
CALibrate1:ZERO	No sensor Sensor zeroing error	 Zeroing error
DIAGnostic:SENSe1:EEPROM:CORRection	No valid sensor sensor eeprom data:correction factor number does not match	No sensor
DIAGnostic:SENSe1:EEPROM:READ	No valid sensor	No sensor
DIAGnostic:SENSe1:EEPROM:CALFSPecial DIAGnostic:SENSe1:EEPROM:CALFSTanda rd	No valid sensor Sensor eeprom data is over space Sensor eeprom data:cal factor number does not match	No sensor Too much data
DIAGnostic:SENSe:EEPROM:FREQSTandar d?	No valid sensor Sensor eeprom data is over space	No sensor Too much data
DIAGnostic:SENSe1:EEPROM:CALFRange DIAGnostic:SENSe1:EEPROM:TYPE	 No valid sensor	 No sensor
DIAGnostic:SENSe1:EEPROM:FREQSPecia l	No valid sensor Sensor eeprom data is over space Sensor eeprom data:freq is above upper freq limit Sensor eeprom data:freq is below lower freq limit Sensor eeprom data:freq number do not match	No sensor Too much data
DIAGnostic:SENSe1:EEPROM:TYPE	No valid sensor Sensor eeprom data:upper freq should be larger than lower freq	No sensor
DIAGnosticSENSe1:EEPROM:WRITe	No valid sensor Sensor eeprom data:write error Password is incorrect	No sensor Sensor eeprom error Password is activated
FETCh1?	Data corrupt or stale TTL trigger mode not applicable	No valid sensor Normal mode
INITiate:IMMediate	Init ignored	Continuous is on

Table 2-7: Device Specific Error Messages (Continued)

Command	Error Message	Example of Problem
MEASure1?	Data corrupt or stale Normal mode is off	No valid sensor Burst or swift mode
MEMory:TABLe:FREQuency	No valid sensor V_{PROP}F Table is not selected V_{PROP}F Table freq data error	No sensor Select others Start freq is greater than sensor maximum
MEMory:TABLe:POWer	Analog out Table is not selected Analog out Table data input error	Other table Lower power is greater than upper power
MEMory:TABLe:SLOPe	No valid sensor V_{PROP}F Table is not selected	Select others No sensor
MEMory:TABLe:UNIT	Analog out Table is not selected	Other table
MEMory:TABLe:VOLTage	Analog out Table is not selected Analog out Table data input error	Other table Lower voltage is greater than upper voltage
READ1?	Data corrupt or stale Normal mode is off Trigger deadlock	No valid sensor Burst or swift mode is on TRIG:SOUR is not in IMM mode
SENSe1:EEPROM:CALFactor? SENSe1:EEPROM:FREQuency?	No valid sensor Sensor eeprom data is over space	No sensor Too much data
SENSe1:CORRection:VPROpf:STATe	No valid sensor	No sensor
SENSe1:TRIGger:DELay SENSe1:TRIGger:LEVel	No valid sensor Sensor is in CW source mode	No pulse sensor
SENSe1:TRIGger:DELay:STATe ON	No valid sensor	No pulse sensor
TRIGger:COUNter	Normal mode is on Counter has to be one in Swift immediate source No valid sensor Channel is not valid	Not in burst or swift mode Swift mode, imm source No sensor Ratio mode setup
TRIGger:IMMediate	Init ignored	Source is in immediate mode
TRIGger:MODE PRE	No PRE mode for trigger immediate source,	
TRIGger:SOURce:IMMediate	Counter has to be one in Swift immediate source No valid sensor Channel is not valid	Swift mode is on No sensor Ratio mode setup

3

Theory of Operation

3.1 Introduction

This chapter contains a functional description of the electrical circuits contained on the PC board assemblies of the 58542 VXI Universal Power Meter. Table 3-1 lists the circuit assemblies by their reference designations, and includes the assembly part number and schematic drawing number for each board.

Table 3-1: VXI Power Meter Circuit Assemblies

Ref #	Title	Assembly P/N	DWG
A1	Analog PC Assembly	21359	21360
A2	Digital PC Assembly	21356	21357
A3	VXI Processor PC Assembly	This PCA is an OEM item	

3.2 Circuit Description

The 58542 Power Meters electrical circuitry resides mainly on three PC boards; the Analog PC Board (A1), the Digital PC Board (A2), and the VXI Processor PC Board (A3). The Processor contains the microprocessor, some ROM and RAM and the VXI interface chip. The remainder of the required ROM and RAM is contained on the Digital board.

Two cables interface to the meter through the VXI Bus Interface connections on the rear panel. The Bus Interface goes to the Processor board through connectors P1 and P2. Three front panel BNCs connect to the Analog board. See DWG 21406 in Chapter 7 for specific interconnect information.

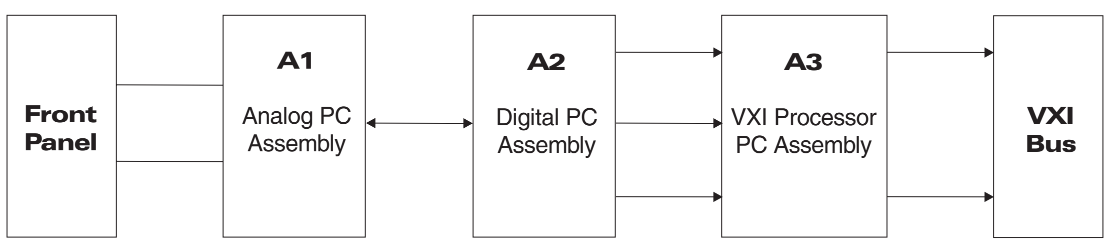


Figure 3-1: VXI Power Meter System Block Diagram

3.3 Analog PC Board (A1)

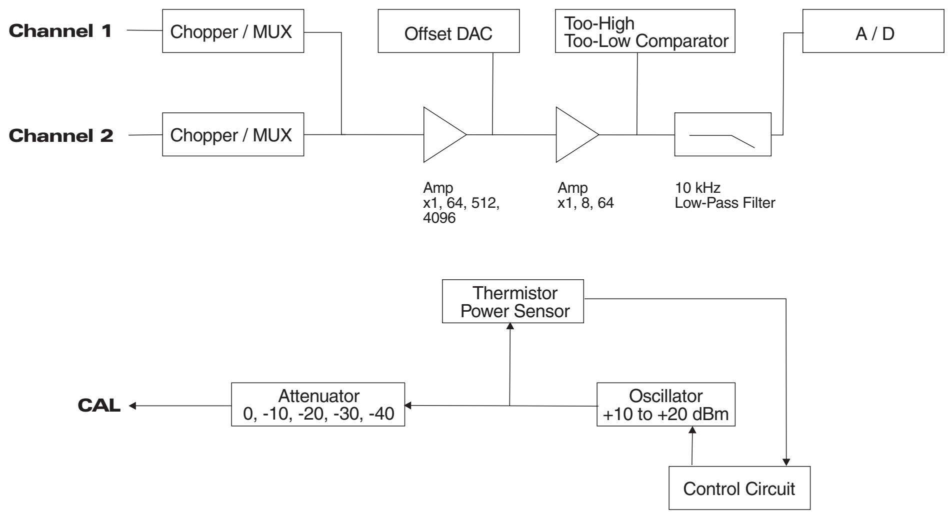


Figure 3-2: Analog PC Assembly Block Diagram

The sensors are connected from the front panel through W3 and W2 which are connected to the Analog Board through W2P1 and W2P2 for sensor 1 and W3P1 and W3P2 for sensor 2. The detected DC voltage from the sensor is a differential voltage applied to pins 3 and 4 of J7 or J9. This differential voltage goes to the U28 or U29 FET chopping circuit. The outputs are pins 8 and 9 of the chopper. An incoming signal can either be fed straight through or inverted. The signal is fed straight through when A1 and A0 are both low. It is fed through in the inverted mode when A1 is low and A0 is high. When A0 is low and A1 is high the signal is shorted together and grounded, and when A0 and A1 are both high, the input is open (floating). This provides chopper stabilized amplification when low power signals are being received by switching the FET switch from the inverting to non-inverting mode and back again at a rate of 300 times per second.

U31 is the 1st stage amplifier, and has programmable gain. The programmable gain is provided by U30 and the resistor ladder composed of R6 through 12 which will program gains of 1, 64, 512 and 4096. U32A is the 2nd stage amplifier that provides programmed gains of 1, 8 and 64 using FET switch U33 and resistor ladder R16, R17 and R18. C167 (though shown on the schematic) is not loaded on the board.

U32B/C/D and associated components provide a 6-pole bessel filter in the low pass configuration -10 kHz with a 120 ms settling time. In normal operation, the signal will always go through this filter. Other paths such as unfiltered, TP8, and 1st amp signals are provided for testing only and not used in normal operation.

The 1st amplifier, U31, has an offset voltage injected into it via pin 7. This offset voltage comes from the 12-bit DAC, U15. Offset voltages range from +5 V to -5 V which can effectively remove about a –1.2 mV offset at the detector. U34A&B are comparators which monitor the input signal for a too high or too low condition. The too low comparator, U34B, will fire if the voltage is below 1 V, and the too high comparator, U32A will fire if the voltage is above 9 V. This provides an A/D conversion range of 1 to 9 V. The too high and too low signals are used by the software to determine whether or not a range change should be made. There are 7 ranges. Four ranges are processed by U31 (1st Amp), and three ranges by U32A. Appropriate gains are set to keep voltages at the A/D conversion point between 1 V and 9 V. R45 and R46, divide the 12 V regulated voltage to 1 V such that the too low comparator will fire below 1 V to assert the too low signal. R36 and R37, divide the 12 V signal to 3 V, and R40 and R41 divide the incoming signal by 3 so that the too high comparator will always fire at 9 V.

Each detector has a thermistor included in its housing so that the power meter can read the temperature of the sensor. The voltage from that thermistor is applied through J7-6 (for channel 1) and amplified by a gain of 2 by U35C (shown on Sheet 2). Channel 2 sensor's thermistor voltage comes in through J9-6 and is amplified by 2 at U35A. The thermistor voltages are also routed to U35D and U36A&B. This circuit is used to detect whether a sensor has been attached or disconnected. The connection or disconnection of a sensor causes a transient voltage which is passed through C10 (for channel 2) or C11 (for channel 1). This is detected by a window comparator consisting of U35D and U36A&B. Whenever the voltage is outside the normal DC bias range, the detector change will cause either a rise or a fall in the voltage that will be detected by the comparator. The comparator will output an interrupt to the processor, IRQ, which will be the output of U36A&B.

U16A and U9A provide a latch for the interrupt so that the processor will have time to respond. Pin 2 of U9A is used to enable or disable interrupts. U36C buffers the interrupt signal. The interrupt 1 signal goes to the CPU board.

The sensors also have EEPROMs that connect to the system through a serial interface. Channel 1 uses J8-6 for the clock and J8-5 for the data. This is a bi-directional device wherein data needs to be written to the sensor and also read from the sensor. U24E buffers incoming data, and U24F buffers outgoing data. Q3 provides the necessary open collector interface. U26 supplies the clock during a read or write action. The clock signal is buffered by U24D. Channel 2 has a duplicate circuit consisting of U24A/B/C and Q2. DC supplies of +5 V and –15 V are also routed to the sensors. The 15 V supplies are routed through solid state fuses to provide protection in case of any shorts. These are resettable solid state fuses which do not need to be replaced. The +5 V can be switched on and off. This is controlled by U26, and buffered by Q8 and Q1.

To recap the preliminary actions of the incoming signals; they first go through the chopper, the 1st stage amplifier, the 2nd stage amplifier, and then the filter. The signals are then sent to U23, a 16:1 multiplexer. U23 can select one signal to route to U25, the A/D converter. Most of the other signals are used for testing purposes only. During normal operation, the signal path will be through the bessel filter.

Thermistor voltages enter on pins 24 and 25 for channels 1 and 2. The only other signal that is used for normal operation is the $V_{PROP}F$ input that enters through J3 on the front panel. CR15 provides input protection, and U21A is a x1 amplifier/buffer. The $V_{PROP}F$ signal can be read at pin 26 of the multiplexer. U25 is a 14-bit A/D converter which operates at an 11 ms conversion time. The input is via pin 6 on TP13. This device can measure between 0 and 10 V. U25 operating power supplies are limited to –12 V (pins 11 and 5). Those voltages are derived from U39 and U43 which regulate the –15 V to –12 V.

R105, R104, and C31 configure U33 to measure from 0 to 10 V. Pin 4 of U33 provides the start convert signal, and the processor asserts this signal to start an A/D conversion. After the conversion has completed, pin 3 can be asserted to output the data to the bus. These two functions are supplied by chip select ACS7 and ACS1. ACS1 is the output enable and ACS7 is the start convert. EOC is end of convert, which occurs at pin 2 of U33, and is routed to data byte 15 so that the process can interrogate if the data conversion is complete. D15 will be asserted if the A/D has completed its conversion.

U15 is the offset 12-bit D/A converter which provides the 1st stage amplifier with –5 V offset voltage. U17 is similar but configured for 0 to 10 V. The output of U17 is buffered by U21B, current limited by R108, and available at J4 for analog output.

U23 has a number of signals available for testing purposes. Offset voltage is available on pin 5. Output DAC voltage is available on pin 4, and switched 5 V signals on pin 7.

Sheet 5 of DWG 21360 details the interconnections for the Calibrator circuitry and will be discussed in Section X, but U27 also provides three 8-bit ports for control of the multiplexers in the Analog section, and an input port for the Too High and Too Low comparators.

3.4 Calibrator Module

The Calibrator Module is located on the Analog PC Board. It is the heart of the 58542 Power Meter in that it is a patented system that allows the power sensors to be calibrated against an internal thermistor power standard (see Figure 3-3). In contrast to the conventional fixed-level calibrators, the 58542 calibrator produces a range of power levels over a 50 dB dynamic range to an accuracy of a few thousandths of a dB.

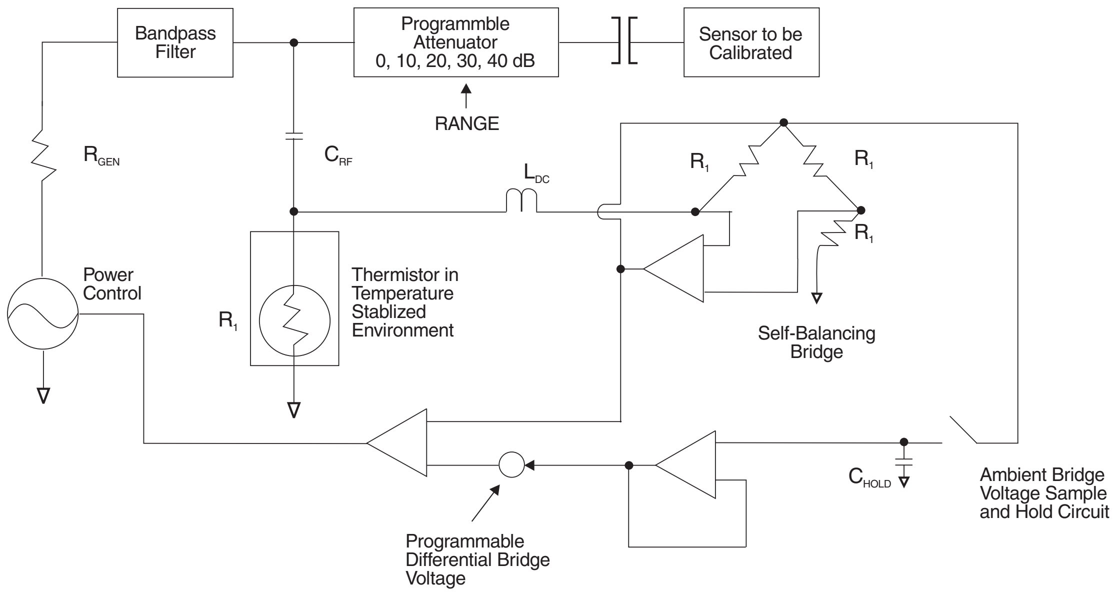


Figure 3-3: Calibrator Internal Power Standard Configuration

The thermistor is mounted in a self-balancing bridge configuration using DC substitution in the bridge. The thermistor is maintained at a fixed operating point and the DC power (P_{DC}) in the thermistor is related to the RF power (P_{RF}) by the simple relationship:

$$P_{DC} + P_{RF} = P_{AMBIENT} = \text{constant}$$

The constant, $P_{AMBIENT}$, is found by turning the RF power off and measuring the ambient voltage, $V_{AMBIENT}$ to which the self-balancing bridge settles. The advantage of this approach is that the linearity of the thermistor-leveled oscillator is limited only by the accuracy with which DC voltages can be measured and the stability of the RF calibrator. To ensure exceptional stability, the thermistor assembly is enclosed in a temperature-stabilized environment and a low drift sampling circuit is used to hold the ambient bridge voltage. The RF power can then be programmed by controlling a difference voltage, δV, at the summing node. The power is related to the voltage by:

$$P_{RF} = \frac{V_{AMBIENT}\delta V - \delta V^2}{R_1} = \frac{\delta V^2}{2R_1}$$

This permits the RF power to be precisely controlled over a dynamic range of about 12 to 15 dB. The dynamic range is extended using a switched attenuator, the properties of which are determined using the thermistor-leveled oscillator itself. The effective attenuation (including all mismatch effects) of each attenuator relative to the next is measured by finding a pair of powers, one for each attenuator, that produces identical signals from the sensor under test. Because the sensor under test is used at a fixed operating point, no knowledge of its detection law is required.

The operation of the various functions of the Calibrator Module can be understood more easily if the circuits are discussed individually. The functional sections of this module include the following:

1. The 50 MHz oscillator (Q4) and its current control circuit consisting of U4D, Q5 and U2C.
2. The RF output circuit consisting of the low pass filter, the stepped attenuator, and the connector and cable to the front panel of the power meter.
3. The oven to maintain the control thermistor at a constant 60°C. It is located on the small board attached to the bottom of the Q1 heater transistor. The board contains the RT1 and RT2 thermistors and the Q7 control transistor.
4. The thermistor bridge to measure the RF power by DC substitution. It consists of RT1, U1 and Q6.
5. The track and hold circuit that remembers the ambient bridge voltage, using U2B, U8D and U3A.
6. The 14-bit DAC and reference supply to measure the ambient bridge voltage and control the RF output level, made up of U11, U7, U8C&D, U4, U14A, U2A&B, U3A, U5, U10, U16B&C, U13B&C and U9B.
7. The correction circuit to measure the temperature of the PIN diode attenuator so that a correction for the temperature dependent loss of the diodes can be corrected, consisting of RT2 and U14C.
8. Calibrator NVRAM control circuit, U26 and U18.

3.4.1 50 MHz Oscillator

The first section of the Calibrator Module consists of a colpits oscillator circuit with a controllable power output. The output power is measured by the thermistor bridge and set by varying the DC current through Q4. This current is supplied by a voltage to current converter consisting of U14D, Q5 and U4. The power generated by Q4 is linearly related to the current through it. Thus, the voltage from U4 that is converted to current by U14D and Q5 is linearly related to the RF power generated. When the calibrator is set for 0 dBm, the voltage at U4-6 is near 0 volts.

3.4.2 RF Output

The 50 MHz oscillator output is capacity coupled to the low pass filter, L13, L14, L15, and associated capacitors. The resultant harmonic-free RF is applied to the switched PIN attenuator, CR8 - 14 and associated resistors and control amplifiers U19 and U14B. The first section is 10 dB, the output section is 20 dB, and a resistor between sections adds another 10 dB. Thus, the output power can be programmed from +20 to -30 dBm.

3.4.3 Oven

The measuring thermistor is maintained at a constant 60°C by being mounted on the Q1 heater transistor, which is driven from the sensing thermistor RT2 by way of the Q7 current amplifier. RT2 is mounted very close to RT1 so that both are maintained at the same temperature. When RT2 reaches 60°C, the voltage across it is just enough to maintain drive to the heater. This condition will be maintained regardless of the ambient temperature.

3.4.4 Thermistor Bridge

RT1 is connected in a self-balancing bridge circuit which delivers just enough power to the thermistor to keep it at 500 W. Thus, if part of the power delivered to it is from the RF generated by the oscillator and the rest is from the DC current of the bridge, then by reducing the amount of DC power, the circuit will increase the drive to the oscillator as needed to keep the total power in RT1 constant. It is necessary to measure only the amount of DC power reduction to know the amount of RF power present. In this way, a precisely known RF output level can be established.

3.4.5 Track & Hold and DAC

In order to know how much power is being added by the oscillator, it is necessary to measure the power delivered to the thermistor with no RF present. This is done by turning off the oscillator power (closing switch U2C), and then measuring the voltage out of the control bridge. This is known as the ambient bridge voltage. To make this measurement, the following conditions are established: U8D and U2B are switched open, and U8A & C are switch closed. By using the U13 DAC, a successive approximation measurement of the voltage is made. The output of the DAC is connected to one input of U4 and the bridge is connected to the other. Thus, it becomes a comparator that makes it possible for the computer to tell when the output voltage of the DAC is greater than the bridge voltage, and so complete the successive approximation. Once this is done, the DAC is set for 0 V output, U8A is opened, U8B, U8D and U2B are closed, and the track and hold capacitor, C39 will charge up to the voltage which represents the zero RF power condition of the bridge. When the oscillator is turned on by U9C, then the sampling switch, U2B, will open and allow C39 to supply this “RF Off” condition to the measuring circuit. Any voltage from the DAC will now reduce the amount of DC power being delivered to the thermistor bridge, and the control circuit will add just enough current to the oscillator to cause its output to add back that much RF power into the bridge.

3.4.6 Correction Thermistor Circuit

The compensation thermistor is mounted near CR13 to sense the temperature of the 20 dB attenuator section that produces the 0 dBm output. This is the only absolute power specified. All other power levels are measured by the software relative to 0 dBm.

3.4.7 Calibrator NVRAM Control Circuit

The calibrator serial number and the correction constant for the 0 dBm output level, as well as the date of calibration and password for rewrite access, is contained in a Non-Volatile RAM. The read and write for it is provided by the parallel peripheral interface (PPI) U26. Before allowing access to the NVRAM, the software looks for a logic 1 on port A, bit zero of the PPI and, if that is present, it asks the operator for the password. If the correct password is supplied, the collected data will be written into U18. If the jumper W1 is set to supply a logic 0 to the PPI, the operator will have write-access to U18 without needing a password.

3.4.8 Sensor NVRAM

Each sensor has a NVRAM to store all of the calibration constants, the date of calibration, place of calibration, etc. This NVRAM is also password protected but has no hardware switch to defeat it. The read/write control for it is furnished by U24A/B/C, U24D/E/F, Q2 and Q3. Q1 and Q8 control the 5 V supply to reduce the amount of heat in the sensor, as well as reducing the noise from the supply.

3.4.9 Sensor Interrupt

Each time a sensor is connected or disconnected from the 58542, a CPU interrupt is generated by causing the thermistor voltage change to set a latch, which signals the CPU that it needs to check for a sensor change. The latch is driven from a “window” comparator, U35D and U36A & B. This comparator is driven from capacitors which are connected to each of the thermistor lines from the sensors. The latch is enabled or cleared by a signal from the PPI, U26.

3.4.10 Digital Control Circuit

The digital control circuit is the interface between the CPU and the preceding functions.

3.5 Digital PC Board (A2)

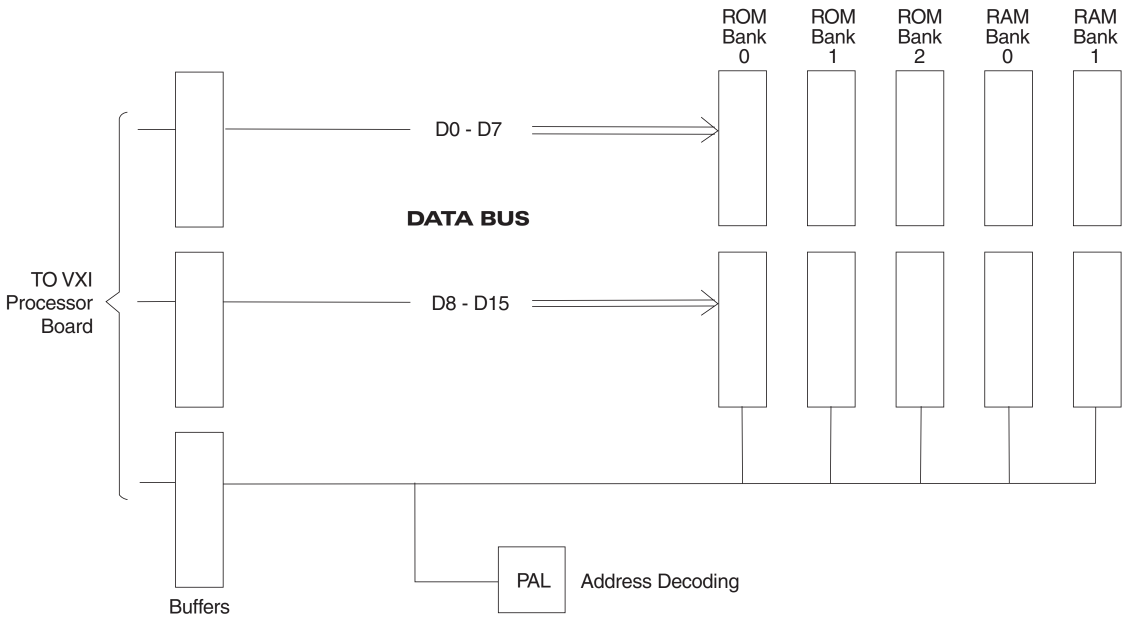


Figure 3-4: Digital PC Assembly (A2) Block Diagram

I/O address decoding is done first in the PAL (U10), and secondarily in U2. During valid I/O addressing, wait states are generated. The PAL monitors a clock cycle count (from U7A) to hold off DTACK on the CPU. W1 is jumpered to four wait states. During ROM and RAM accesses, the PAL asserts DTACK immediately with no wait states.

DTACK is passed back to the VXI Processor board. U12A provides an open collector signal.

RAM bank 1 is connected to the battery. When power is lost, U6 de-selects RAM bank 1 and connects Vcc to the battery. This bank is then non-volatile and can be used for data storage. C24 is in parallel with the battery and allows it to be replaced without losing data. Battery current can be measured at R1, TP1 and TP2. At 2 mA, the voltage across the test points should measure about 2 mV.

U8 acts as an 8-bit input port addressed at CSTRG to monitor the eight TTL trigger lines on the VXI bus. The 58542 can trigger measurements either from these eight lines or from the TTL trigger BNC input on the front panel.

The Digital board interfaces with the Analog board at J1. All digital lines are buffered (U3, U4 and U5), and additional analog chip select decoding is done with U1.

U9A and U9B generate the upper data byte and lower data byte write signals. U11 buffers the R/*W and Reset signals.

W2 jumpers the three interrupt signals from the Analog board to the VXI Processor board. These jumpers should only be removed for troubleshooting.

3.6 VXI Processor PC Board (A3)

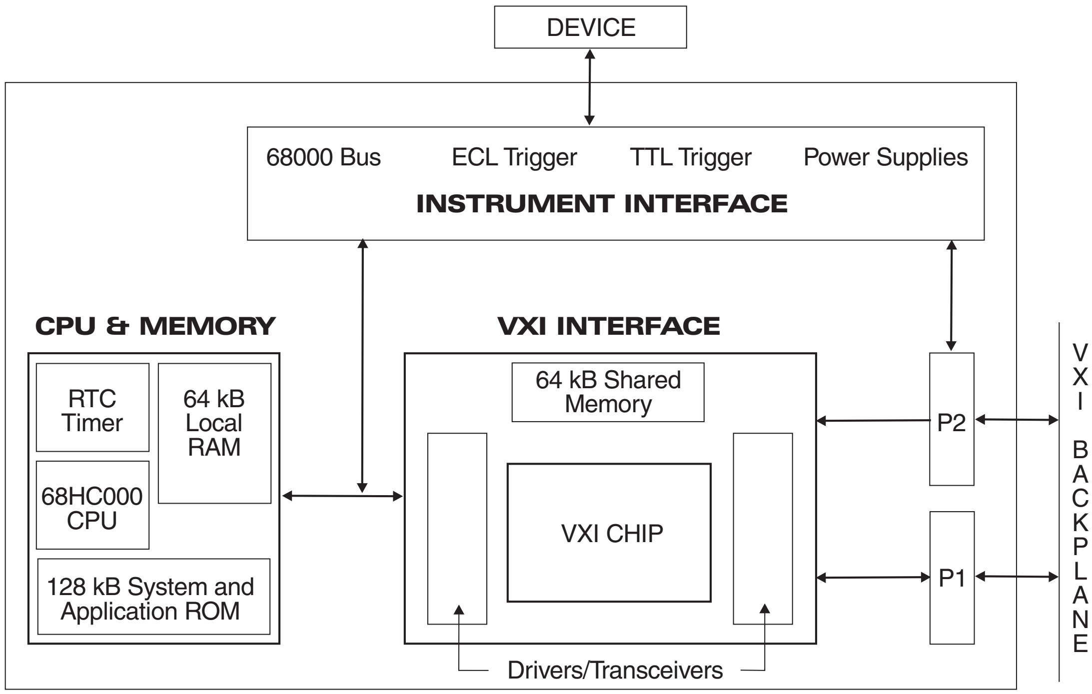


Figure 3-5: VXI Processor (A3) Block Diagram

The following circuit description is given for information only. The VXI Processor and Memory PC boards are OEM assemblies. If these boards are not functioning properly, the problem will usually be indicated by the instrument not responding to an Identification query (*IDN? - see Section 2.5.35.1). See Section 5.3 for replacement instructions.

The VXI Processor PC Board circuit functions are divided into three main sections:

- VXI Interface
- CPU and Memory
- LEDs and Drivers

3.6.1 VXI Interface

The VXI interface contains 64 kB of Shared Memory, a VXI interface gate array, and the drivers and transceivers to enable the VME and CPU to access the Shared Bus.

3.6.1.1 VXI Gate Array

U28 is a 120-pin gate array, packaged in a 13x13-pin grid array. The gate array generates the necessary signals that control the flow of data from the processor section through the shared bus, and to the VXI bus, and vice versa.

The gate array also controls the LEDs that indicate whether the VXI is accessing the VXI A16 or the A24/A32 registers, and the FAIL LED, which indicates whether the VXI A16 registers have been initialized.

3.6.1.2 Shared Memory

U29 and U30 are 32 k x 8 static RAM chips located on the Shared Bus for the development of VXI Shared Memory Protocols.

3.6.1.3 Shared Memory Decoders

U33 generates the necessary strobes and control signals to the Shared Memory static RAMs.

3.6.1.4 Drivers and Transceivers

U36 and U37 transceivers enable the data lines from the VXI bus onto the shared bus, and vice versa.

The A1 through A15 address lines are latched by U38 and U39 latching transceivers from the VXI bus onto the shared bus to implement address pipe-lining. The address lines are not latched going the other way.

The U40 transceiver buffers the address modifier lines.

The VXI chip controls the direction, and enables the transceivers.

U23 and U24 latch the processor data lines D00 to D15 to drive the upper 16 address lines of the VXI A32 space to implement A32 bus mastership.

The A24 through A31 address lines are buffered by U45 from the VXI bus P2 connector to the gate array.

U35 is a GAL which controls the direction of the data strobes, data transfer acknowledge, and the bus error from the VXI bus to the gate array and vice versa.

3.6.1.5 Logical Address Switch

U19 buffers the outputs of the address switch SW1 to enable the processor to read the logical address from the switch.

3.6.1.6 TTL Triggers and Local Bus

The TTL triggers and local bus are not used by the Processor board, but are made available to be used on the Digital board through the P7 pin connector.

3.6.2 CPU and Memory

3.6.2.1 Processor

The Interface circuitry of the Processor board uses an 8 MHz 68HC000 CMOS processor (U7). See the latest Motorola data sheet for further information on this chip.

3.6.2.2 Real Time Clock/Timer

U1 generates the system tick for a pSOS kernel operating system. It also adds time and event capabilities to the application code. For further information, see the latest Hitachi 146818 data sheet.

3.6.2.3 Interrupt Controller

The processor uses seven levels of auto vectored interrupts. U6 encodes the priority levels for the processor. Three levels of auto vectored interrupts are available to the user. These are signals AVINT4, AVINT3, and AVINT2. The higher numbered interrupt signal has the higher priority. The user accesses these signals from the P5 pin connector on the Processor board.

3.6.2.4 Bus Error Timer

U2A divides down the VXI 16 MHz clock for the 8 MHz processor. U2B is used to generate a bus error signal if timeout occurs before data transfer acknowledgement. The timer generates this signal if a processor access cycle exceeds 16 microseconds.

3.6.2.5 Local Address Decoders

U17 and U18 are address decoding GALs. These circuits generate the chip selects and the control lines for access to the RAM, ROM, and other functions.

3.6.2.6 Drivers and Transceivers

U26 and U27 transceivers drive the local data from the processor bus to the shared bus, and vice versa.

The processor address lines are driven by U21 and U22 to the shared bus.

The VXI gate array, U28, enables these buffers.

3.6.2.7 Local Memory

U9 and U11 are two 32K X 8 static RAM chips on the processor bus.

3.6.2.8 LED Drivers

The actual LEDs are located on the Analog PC Board. U32 is a one-shot used to widen the pulse of the VME A16 and A24/A32 signals. U34 is used to drive the RUN, HALT, MODID, A16 and A24/A32 LEDs. The FAIL LED is driven directly from the VXI chip. The TRIGGER LED is driven under software control.

A

Program Examples

A.1 Introduction

This appendix contains examples of the various operating programs in the Model 58542 VXIbus Universal Power Meter.

The examples shown in this chapter are written in HTBasic™ format. Different languages will use different commands, but the string sent or received will always be the same. In HTBasic, the OUTPUT command sends a string to the GPIB bus. The number or variable after the word OUTPUT is the GPIB address of the instrument.

A.2 Sensor Calibration Examples

Sensors must be calibrated to the meter before performing measurements. Only one sensor at a time can be calibrated to a given 58542 VXI Power Meter input. That is, each time a sensor is calibrated to the meter using the power sweep calibration, previous calibrations for that meter input are voided automatically. Thus, you are assured that your measurement system is always performing under a valid sensor calibration.

A.2.1 Sensor Calibration Example 1

The following example performs power sweep calibration of the power sensor connected to input number 2 and sends back a pass/fail flag upon completion using CAL2?

```
410     ASSIGN @Pwr_mtr to 70101
420     CLEAR @Pwr_mtr
430     WAIT 1
440     PRINT Connect the sensor to the power sensor cable.
450     INPUT Then connect the sensor to the calibrator and hit ENTER.,Dmy
460     PRINT Calibrating Sensor 2...
470     OUTPUT@Pwr_mtr,CAL2?
480     ENTER@Pwr_mtr,Pass_cal
490     IF Pass_cal=0 THEN PRINT Calibration Passed
500     IF Pass_cal=1 THEN
510     PRINT Calibration FAILED, Sensor 2 on calibrator?
560     ELSE
570     IF Pass_cal<>0 THEN PRINT Strange Response to CAL2?. Clear output Queue?
580     END IF
590     !
```

A.2.2 Sensor Calibration Example 2

The power meter measurement functions are operated to tight tolerances during sensor calibration. If the sensor passes calibration, it is a good health check of the 58542 VXI Power Meter. In general, the power meter and sensors are operating properly if they pass the calibration process. For instance, the calibration process will fail if the sensor detector elements have been damaged. Also see the self test functions, which are initiated using the *TST command or the DIAG: commands (see Table 5-).

The following program uses the CAL? query format to perform power sweep calibration of a power sensor and sends back a pass/fail flag upon completion. Completion of the calibration function is monitored via service requests using the *OPC, operation complete command.

```
330   !
340   !################################
350!
350   !    CALIBRATE THE SENSORS
370   !
380   !################################
390   ASSIGN@Pwr_mtr to 70101
400   CLEAR@Pwr_mtr
410   WAIT 1
420   OUTPUT@Pwr_mtr;*CLS! Clear status registers and error queue buffer
425   WAIT 0.3
430   OUTPUT@Pwr_mtr;*ESE 1! Standard Event Status Enable bit 0, Operation Complete
440   OUTPUT@Pwr_mtr;*SRE 32! Service Request Enable Status Byte bit 5, Standar d
      ! Event Status Register
450   !
460   ON INTR 7 GOSUB Spoll_intr
470   ENABLE INTR 7;2
480   !
490   ! Calibration Routine
500   !
510   INPUT Connect the sensor to the calibrator. Then press ENTER.,Dmy
520   PRINT Calibrating Sensor 1...
525   Srq_flag=0
530   OUTPUT@Pwr_mtr;CAL1?;*OPC! For zeroing substitute CAL1:ZERO? for CAL1?
540   !
550   !
560   ! Now wait for srq
570   !
580   !
590   !
600   WHILE Srq_flag=0
610   END WHILE
620   !
630   ! Enter Pass/Fail Flag, This comes with query format, CAL1?, rather than command format, CAL1.
640   !
650   ENTER@Pwr_mtr;Pass_cal
660   IF Pass_cal=0 THEN PRINT Calibration Passed
670   IF Pass_cal=1 THEN
680   OUTPUT@Pwr_mtr;SYST:ERR?
690   ENTER@Pwr_mtr;Err_msg$! Reading an error message clears it.
700   PRINT The power meter reports the following error.,ERR_msg$
710   PRINT Calibration FAILED, sensor on calibrator?
720   ELSE
```

```
730  IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Other Event Status Register Bits Too?
740  END IF
750  !
760  STOP
770  !
780  Spoll_intr: !
790  !
800  State=SPOLL(@Pwr_mtr)
810  PRINT SPOLL INTR:;State
820  Srq_flag=1
830  ENABLE INTR 7;2
840  !
850  RETURN
860  !
870  !
880  !    END OF SENSOR CAL
890  !####################################
900  !
910  END
```

A.2.3 Sensor Calibration Example 3

The following program uses the CAL format to perform power sweep calibration of a power sensor. Completion of the calibration function is monitored via service requests using the *OPC, operation complete command. The Standard Event Status Register is used to report pass/fail by asserting bit 3, value 8, the Device Dependent Error bit.

```
320  !
330  ! ###################################
340  !
330  !    CALIBRATE THE SENSORS
360  !
370  !####################################
380  !
390  ASSIGN@Pwr_mtr to 70101
400  CLEAR@PWR_mtr
410  WAIT 1
420  OUTPUT@Pwr_mtr;*CLS! Clear status registers and error queue buffer
425  WAIT 0.3
430  OUTPUT@Pwr_mtr;*ESE 1! Standard Event Status Enable bit 0, Operation complete
440  OUTPUT@Pwr_mtr;*SRE 32! Service Request Enable Status Byte bit 5,
     ! Standard Event Status Register
450  !
460  ON INTR 7 GOSUB Spoll_intr
470  ENABLE INTR 7;2
480  !
490  ! Calibration Routine
500  !
510  INPUT Connect the sensor to the calibrator. Then press ENTER.,Dmy
520  PRINT Calibrating Sensor 1
525  Srq_flag=0
530  OUTPUT@Pwr_mtr;CAL1;*OPC! For zeroing substitute CAL1:ZERO for CAL1
540  OUTPUT@Pwr_mtr;*OPC! If not sent with previous line, send *OPC here
550  !
```

```
560   ! Now wait for srq
570   !
580   !
590   !
600   WHILE Srq_flag=0
610   END WHILE
620   !
630   ! Enter Error Reporting From Standard Event Status Register
640   !
650   OUTPUT@Pwr_mtr;*ESR?
660   ENTER@Pwr_mtr;Esr
670   IF Esr=0 THEN PRINT Operation was not completed. Also, no errors.
680   IF Esr=1 THEN
690   PRINT Operation Completed. No errors reported during CAL execution.
700   ELSE
710   IF Pass_cal<>0 THEN PRINT Operation completed. An error has occurred or power
      was just turned on.
720   OUTPUT@Pwr_mtr;SYST:ERR?
730   ENTER@Pwr_mtr;Err_msg$! Reading an error message clears it
740   PRINT The power meter reports the following error.,Err_msg$
750   PRINT Calibration FAILED, sensor on calibrator?
760   PRINT Esr
770   !
780   END IF
790   !
800   !
810   STOP
820   !
830 Spoll_intr:!
840   !
850   State=SPOLL(@Pwr_mtr)
860   PRINT SPOLL INTR:;State
870   Srq_flag=1
880   ENABLE INTR 7;2
890   !
900   RETURN
910   !
920   !
930   !  END OF SENSOR CAL
940   ! ############################
950   !
960   END
```

A.2.4 Sensor Calibration Example 4

The following program prompts you to connect a return loss bridge for calibration and attach an open or short to the bridge test port. It then performs a power sweep calibration and sends back a pass/fail flag upon completion.

```
410    ASSIGN @Pwr_mtr to 70101
420    CLEAR @Pwr_mtr
430    WAIT 1
```

```
440    PRINT Connect the 80503 Precision Return Loss Bridge to the input #2 power sensor cable.
445    PRINT Connect the open or short calibration connector to the bridge test port.
450    INPUT Then connect the bridge input port to the calibrator and hit ENTER.Dmy
460    PRINT Calibrating Sensor 2...
470    OUTPUT @Pwr_mtr;CAL2?
480    ENTER @Pwr_mtr;Pass_cal
490    IF Pass_cal=0 THEN PRINT Calibration Passed
500    IF Pass_cal=1 THEN
510    PRINT Calibration FAILED, Bridge connected to calibrator?
560    ELSE
570    IF Pass_cal<>0 THEN PRINT Strange Response to CAL2?. Clear output Queue?
580    END IF
590    !
```

A.2.5 Sensor Calibration Example 5

The following program prompts you to remove the high power attenuator from a high power sensor main housing and connect the main housing to the power sweep calibrator port. It then performs a power sweep calibration and sends back a pass/fail flag upon completion.

```
410    ASSIGN @Pwr_mtr to 70101
420    CLEAR @Pwr_mtr
430    WAIT 1
440    PRINT Connect the 80325 50W Power Sensor to the input #2 power sensor cable.
445    PRINT Remove the high power attenuator or leave it attached to the measurement port on the
DUT
450    INPUT Then connect the high power sensor main housing to the calibrator and hit ENTER.,Dmy
460    PRINT Calibrating Sensor 2...
470    OUTPUT @Pwr_mtr;CAL2?
480    ENTER @Pwr_mtr;Pass_cal
490    IF Pass_cal=0 THEN PRINT Calibration Passed. Re-attach the high power attenuator!
500    IF Pass_cal=1 THEN
510    PRINT Calibration FAILED, Main sensor housing connected to calibrator?
560    ELSE
570    IF Pass_cal<>0 THEN PRINT Strange Response to CAL2?. Clear output Queue?
580    END IF
590    !
```

A.3 Sensor Zeroing Examples

A.3.1 Sensor Zeroing Example 1

The following example requests that the operator turn off the signal source or disconnect the sensor from the source prior to zeroing. It is preferable to disable the source and leave the sensor attached to the measurement port for proper zeroing. If the source cannot be disabled, use a switch or attach a connector to the grounded metal near your DUT measurement port. DO NOT attach the sensor to the power meter's calibrator port. You want the zeroing offset process to account for the DUT's ground plane noise and thermal emf, not the power meter's.

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
430   WAIT 1
440   INPUT Disable the source's RF output or attach it to a grounded connector. Then hit
ENTER.,Dmy
450   PRINT Zeroing Sensor 1
460   OUTPUT @Pwr_mtr;CAL1:ZERO?
470   ENTER @Pwr_mtr;Pass_zer
480   IF Pass_zer=0 THEN PRINT Zeroing Passed
490   IF Pass_zer=1 THEN
500   PRINT Zeroing FAILED, Source turned off?
550   ELSE
560   IF Pass_zer<>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
570   END IF
580   !
```

A.3.2 Sensor Zeroing Example 2

The following program turns off the signal source, zeros the power sensor, reports a pass/fail indicator upon completion, and turns the source back on.

```
400   ASSIGN @Pwr_mtr to 70101
410   ASSIGN @Source to 70102
420   CLEAR @Pwr_mtr
430   WAIT 1
440   OUTPUT @Source RF0! Turns source output power off (<-90 dBm).
450   PRINT Zeroing Sensor 1
460   OUTPUT @Pwr_mtr;CAL1:ZERO?
470   ENTER @Pwr_mtr;Pass_zer
480   IF Pass_zer=0 THEN PRINT Zeroing Passed
490   IF Pass_zer=1 THEN
500   Ques$=Zeroing FAILED, Source turned off?
510   OUTPUT @Pwr_mtr;SYST:ERR?
520   ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530   PRINT Err_msg$
540   PRINT Ques$
550   ELSE
560   IF Pass_zer <>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
570   END IF
580   !
590   OUTPUT @Source RF1! Turns source output power back on.
```

Zeroing is recommended whenever you are performing critical final power measurements in the bottom 15 to 20 dB of a power sensors dynamic range. Zeroing removes zero drift error from your measurement. At higher power levels, zero drift is typically insignificant when compared to the other sources of error such as source/sensor mismatch uncertainty.

A.3.3 Error Control Examples

Selected basic SCPI syntax and execution errors apply to these commands.

The following command will operate properly.

```
460     OUTPUT @Pwr_mtr;CAL1?! Calibrate channel 1
```

The following command is not a legal command. The 58542 only has two channels.

```
460     OUTPUT @Pwr_mtr;CAL3?! Calibrate channel 3
```

If you ask for error information, the error reporting is as follows:

```
510     OUTPUT @Pwr_mtr;SYST:ERR?
520     ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530     PRINT Err_msg$
```

Program Output:

-7,Invalid Error Number

The following command is not a legal command. The command is mis-typed.

```
460     OUTPUT @Pwr_mtr;CALS?! Calibrate channel 3
```

If you ask for error information, the error reporting is as follows.

```
510     OUTPUT @Pwr_mtr;SYST:ERR?
520     ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530     PRINT Err_msg$
```

Program Output:

-113,Undefined Header;CALS

Device-specific errors include the following.

-300,Device-specific error;No Sensor

The following examples demonstrate the conditions for various errors.

```
380     ASSIGN @Pwr_mtr to 70101
390     CLEAR @Pwr_mtr
400     WAIT 1
401     INPUT Disconnect Sensor From Cable for Error Demonstration. Then hit ENTER.,Dmy
410     PRINT Calibrating Sensor 1
420     OUTPUT @Pwr_mtr;CAL1?
430     ENTER @Pwr_mtr;Pass_cal
440     IF Pass_cal=0 THEN PRINT Calibration Passed
450     IF Pass_cal=1 THEN
460     Ques$=Calibration FAILED, sensor on calibrator?
470     OUTPUT @Pwr_mtr;SYST:ERR?
480     ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
490     PRINT Err_msg$
500     PRINT Ques$
520     ELSE
```

```
521    IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
530    END IF
531    !
532    !
533    STOP
```

Program Output:

Calibrating Sensor 1
-300, Device-specific error;No Sensor
Calibration FAILED, sensor on calibrator?

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
440   WAIT 1
450   INPUT Disconnect the sensor from the cable 1 second after hitting ENTER. Hit ENTER.,Dmy
460   PRINT Calibrating Sensor 1
470   OUTPUT @Pwr_mtr;CAL1?
480   ENTER @Pwr_mtr;Pass_cal
490   IF Pass_cal=0 THEN PRINT Calibration Passed
500   IF Pass_cal=1 THEN
510   Ques$=Calibration FAILED, sensor on calibrator?
520   OUTPUT @Pwr_mtr;SYST:ERR?
530   ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
540   PRINT Err_msg$
550   PRINT Ques$
560   ELSE
570   IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
580   END IF
590!
600   !
610   STOP
```

Program Output:

Calibrating Sensor 1
-300, Device-specific error;Sensor not connected to calibrator
Calibration FAILED, sensor on calibrator?

```
410    ASSIGN @Pwr_mtr to 70101
420    CLEAR @Pwr_mtr
440    WAIT 1
450    INPUT Do not connect the sensor to the calibrator. Hit ENTER.,Dmy
460    PRINT Calibrating Sensor 1
470    OUTPUT @Pwr_mtr;CAL1?
480    ENTER @Pwr_mtr;Pass_cal
490    IF Pass_cal=0 THEN PRINT Calibration Passed
500    IF Pass_cal=1 THEN
510    Ques$=Calibration FAILED, sensor on calibrator?
520    OUTPUT @Pwr_mtr;SYST:ERR?
530    ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
540    PRINT Err_msg$
550    PRINT Ques$
560    ELSE
570    IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
```

```
580 END IF
590 !
600 !
610 STOP
```

Program Output:

```
Calibrating Sensor 1
-300, Device-specific error;Sensor not connected to calibrator
Calibration FAILED, sensor on calibrator?
```

```
410 ASSIGN @Pwr_mtr to 70101
420 CLEAR @Pwr_mtr
440 WAIT 1
449 PRINT Connect the sensor to the calibrator.
450 INPUT Remove the sensor 10 seconds after hitting ENTER. Hit ENTER.,Dmy
460 PRINT Calibrating Sensor 1
470 OUTPUT @Pwr_mtr;CAL1?
480 ENTER @Pwr_mtr;Pass_cal
490 IF Pass_cal=0 THEN PRINT Calibration Passed
500 IF Pass_cal=1 THEN
510 Ques$=Calibration FAILED, sensor on calibrator?
520 OUTPUT @Pwr_mtr;SYST:ERR?
530 ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
540 PRINT Err_msg$
550 PRINT Ques$
560 ELSE
570 IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
580 END IF
590 !
600 !
610 STOP
```

Program Output:

```
Calibrating Sensor 1
-300,Device-specific error;Sensor calibration error
Calibration FAILED, sensor on calibrator?
```

```
410 ASSIGN @Pwr_mtr to 70101
420 CLEAR @Pwr_mtr
430 WAIT 1
440 INPUT Connect the sensor to a source set to about 0dBm. Hit ENTER.,Dmy
450 PRINT Zeroing Sensor 1
460 OUTPUT @Pwr_mtr;CAL1:ZERO?
470 ENTER @Pwr_mtr;Pass_zer
480 IF Pass_zer=0 THEN PRINT Zeroing Passed
490 IF Pass_zer=1 THEN
500 Ques$=Zeroing FAILED, Source turned off?
510 OUTPUT @Pwr_mtr;SYST:ERR?
520 ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530 PRINT Err_msg$
540 PRINT Ques$
550 ELSE
560 IF Pass_zer<>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
```

```
570   END IF
580   !
590   !
600   STOP
```

Program Output:

```
Zeroing Sensor 1
-300,Device-specific error;Sensor zeroing error
Zeroing FAILED, Source turned off?
```

```
410  ASSIGN @Pwr_mtr to 70101
420  CLEAR @Pwr_mtr
430  WAIT 1
440  INPUT Disconnect the sensor from the sensor cable. Hit ENTER.,Dmy
450  PRINT Zeroing Sensor 1
460  OUTPUT @Pwr_mtr;CAL1:ZERO?
470  ENTER @Pwr_mtr;Pass_zer
480  IF Pass_zer=0 THEN PRINT Zeroing Passed
490  IF Pass_zer=1 THEN
500  Ques$=Zeroing FAILED, Source turned off?
510  OUTPUT @Pwr_mtr;SYST:ERR?
520  ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530  PRINT Err_msg$
540  PRINT Ques$
550  ELSE
560  IF Pass_zer<>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
570  END IF
580  !
590  !
600  STOP
```

Program Output:

```
Zeroing Sensor 1
-300,Device-specific error;No Sensor
Zeroing FAILED, Source turned off?
```

A.4 Reading Power Measurement Examples

This section gives you a quick start in performing measurements. Therefore, the easiest measurement commands are in the first example, MEAS#? in NORMal Mode. This may not be the optimum 58542 configuration for your application. There are three example groups, NORMal Mode, SWIFt Mode, and BURSt Mode. The use of the three 58542 measurement data commands, MEAS#?, READ#?, and FETCh#? are explained in detail under the NORMal Mode, which is the meter's power-on default mode.

SWIFt and BURSt Mode measurements use only FETCh#?, not READ#? or MEAS#?. In this getting started section, high speed SWIFt and BURSt Mode examples are few in number and brief in description; detailed information on high speed power measurements can be found in Section (High Speed Measurements).

A.4.1 Reading Power Measurements Example 1

The following program configures the power meter for single channel operation, and uses the MEAS#? command to perform limited Auto-Configuration and return power measurement data in NORMal Mode. Prior to executing the measurement cycle, the MEAS#? command performs low level configuration functions for averaging and trigger sequence arming automatically. Use MEAS#? to get started performing power measurements quickly. Typically, MEAS#? will satisfy most of your power measurement needs; however, more advanced power meter users of NORMal Mode may prefer lower level controls of the READ#? command and the higher measurement rates of the FETCh#? command. High speed measurements are performed using the SWIFt and BURSt Modes, not NORMal Mode.

Note that the MEAS#? command's measurement response speed is slower at very low power levels and faster at high power levels. This is due to a chopper stabilization system used in the NORMal Mode (also used in SWIFt Mode, but not BURSt mode) and Auto-Averaging when using SENS#:AVER:TCON REP. At the most sensitive gain ranges, chopper stabilization allows the very small voltages from the power sensor to be accurately measured. Also, at lower power meter gain ranges, Auto-Averaging automatically adds additional averaging to counter the effects of noise in the measurement signal. The extra time required to perform these Auto-Averaged measurements will not be apparent unless you switch the default averaging control, SENS#:AVER:TCON MOV, to SENS#:AVER:TCON REP.

When you use the MEAS#? command, Auto-Averaging is automatically set to ON. If another part of your programming was using manual averaging, be sure to turn Auto-Averaging back off (SENS#:AVER:COUN:AUTO OFF) before you exit the section of your program code using MEAS? When operating in NORMal Mode you can always use the SENS#:AVER:COUN? query to find out the current averaging value for a particular sensor, and the SENS#:AVER:COUN ### command to set a higher or lower averaging value.

```
260     ASSIGN @Pwr_mtr to 70101
270     CLEAR @Pwr_mtr
280     WAIT 1
290     OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in
Normal Mode!!
300     OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
310     !
320     OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
        ! Applies Cal Factor for 1.44 GHz to measurement data.
330     !
340     FOR I=1 TO 10
350     OUTPUT @Pwr_mtr;MEAS1?! Measures Power at sensor.
360     ENTER @Pwr_mtr;Rdg
370     PRINT Rdg
380     NEXT I
390     !
510     END
```

A.4.2 Reading Power Measurements Example 2

If you send the MEAS#? command, two configuration changes occur. These are listed as shown below. If you prefer to perform measurements without changing the configuration, use the READ#? command.

Command/Configuration ItemMEAS#? Auto-Setting

SENS#:A VER:COUN:AUTOON
INIT:CONTOFF

No other items than the above will change configuration when you send the MEAS#? command.

The following example illustrates that Auto-Averaging changes the active averaging number. Since Auto-Averaging was activated through use of the MEAS#? command, the averaging value currently in use when Auto-Averaging is set to OFF is taken as the new averaging number. This is likely to differ from the averaging number in effect before the MEAS#? command was sent. If this is an undesirable effect, remember to read the averaging number using SENS#:AVER:COUN? and store the value for post Auto-Averaging output at the conclusion of the MEAS#? power meter reading measurement routine.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
290   WAIT 1
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM
310   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO OFF
320   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO?
330   ENTER @Pwr_mtr;A$
340   PRINT A$,1 is AUTO ON
350   OUTPUT @Pwr_mtr;SENS1:AVER:COUN 16
360   OUTPUT @Pwr_mtr;SENS1:AVER:COUN?
370   ENTER @Pwr_mtr;A$
380   PRINT A$,16 MEANS AVERAGING #=16 ACCEPTED.
390   OUTPUT @Pwr_mtr;INIT:CONT ON
400   OUTPUT @Pwr_mtr;INIT:CONT?
410   ENTER @Pwr_mtr;A$
420   PRINT A$,1 is INIT:CONT ON
430   OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
440   !
450   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
      ! Applies Cal Factor for 1.44 GHz.
460   !
470   FOR I=1 TO 10
480   OUTPUT @Pwr_mtr;MEAS1?! Measures Power at sensor.
490   ENTER @Pwr_mtr;Rdg
500   PRINT Rdg
510   NEXT I
520   !
530   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO?
540   ENTER @Pwr_mtr;A$
550   PRINT A$,1 is AUTO ON
560   OUTPUT @Pwr_mtr;INIT:CONT?
570   ENTER @Pwr_mtr;A$
580   PRINT A$,1 is INIT:CONT ON
590   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO OFF
600   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO?
610   ENTER @Pwr_mtr;A$
620   PRINT A$,1 is AUTO ON
630   OUTPUT @Pwr_mtr;SENS1:AVER:COUN?
640   ENTER @Pwr_mtr;A$
650   PRINT A$,Value not equal to 16 means reset by Auto-Averaging during MEAS#?
```

```
760    !
770    END
```

A.4.3 Reading Power Measurements Example 3

The following program configures the power meter for dual channel operation and uses the MEAS#? command to return power measurement data in NORMal Mode. Please note that measurement speed per channel decreases slightly in NORMal mode when two channels, versus only a single channel, are connected and calibrated. In NORMal Mode, single channel measurement rates are slightly faster than two channel measurement rates. If you are using CW power sensors, also please note that power measurements are not performed at specifically the exact, simultaneous instant in time. In NORMal Mode there will be a short time delay of about 3 ms between the two channels' sample points, about two and a half orders of magnitude faster than traditional CW power meters. In BURSt and SWIFT mode this time is about 1 ms. If you need to guarantee that power sampling occurs on both channels within a smaller interval of time, this can be accomplished to within about 2% of the Sample Delay time plus 2 ns using the 80350A Series Peak Power Sensors and triggering both sensors at the same time. The example below operates with either Peak or CW power sensors.

```
260    ASSIGN @Pwr_mtr to 70101
270    CLEAR @Pwr_mtr
280    WAIT 1
290    OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in
Normal Mode!!
300    OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
310    OUTPUT @Pwr_mtr;CALC1:UNIT W ! Transmitter output power in Watts.
320    OUTPUT @Pwr_mtr;CALC2:RAT 1,2! Transmitter gain stays in default, dBm.
330    !
340    OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
       ! Applies Cal Factor for 1.44 GHz to sensor 1 data.
350    OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 0.96E9
       ! Applies Cal Factor for 960 MHz to sensor 2 data.
360    !
370    FOR I=1 TO 10
380    OUTPUT @Pwr_mtr;MEAS1?;MEAS2?! Measures Power at sensors.
390    ENTER @Pwr_mtr;Chn_pow 1
400    ENTER @Pwr_mtr;Chn_1rat2
410    PRINT Chn_pow1,Chn_1rat2
420    NEXT I
430    !
550    END
```

A.4.4 Reading Power Measurements Example 4

The following program configures the power meter for single channel operation and uses the READ#? command to return power measurement data in NORMal Mode. Using the READ#? command will not change power meter configuration items under any circumstances. Note that INIT:CONT ON is illegal when using READ#?. When continuous trigger arming is desired, use INIT:CONT ON with the FETCh#? command if desired. Also, use TRIG:SOUR IMM when using the READ#? command.

```
280ASSIGN @Pwr_mtr to 70101
290CLEAR @Pwr_mtr
```

```
300WAIT 1
310OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in
Normal Mode!!
320OUTPUT @Pwr_mtr;TRIG:SOUR IMM! TRIG:SOUR must be IMM when using READ#?
330OUTPUT @Pwr_mtr;INIT:CONT OFF! INIT:CONT arming must be OFF when using READ#?
340!
350OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1 ! Sets averaging to manual and one sample per
reading.
360OUTPUT @Pwr_mtr;CALC1:POW 2! Channel 1 configured to measure sensor 2 power
370!
380OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
! Applies Cal Factor for 17.4 GHz to measurement data.
390!
400FOR I=1 TO 10
410OUTPUT @Pwr_mtr;READ2?! Measures Power at sensor.
420ENTER @Pwr_mtr;Rdg
430PRINT Rdg
440NEXT I
450!
650END
```

A.4.5 Reading Power Measurements Example 5

READ#? measurement response speed is slower at very low power levels and faster at high power levels. This is due to a chopper stabilization system used in the NORMal Mode which is also used in the SWIFt Mode, but not the BURSt mode. At the most sensitive gain ranges, chopper stabilization allows the very small voltages from the power sensor to be accurately measured.

The following program configures the power meter for dual channel operation and uses the READ#? command to return power measurement data in NORMal Mode.

```
260    ASSIGN @Pwr_mtr to 70101
270    CLEAR @Pwr_mtr
280    WAIT 1
290    OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in
Normal Mode!!
300    OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
310    OUTPUT @Pwr_mtr;CALC1:UNIT W! Transmitter output power in Watts.
320    OUTPUT @Pwr_mtr;CALC2:RAT 1,2! Transmitter gain stays in default, dBm.
330    !
340    OUTPUT @Pwr_mtr;TRIG:SOUR IMM! TRIG:SOUR must be IMM when using READ#?
350    OUTPUT @Pwr_mtr;INIT:CONT OFF! INIT:CONT arming must be OFF when using READ#?
360    !
370    OUTPUT @Pwr_mtr;SENS1:AVER:COUN 1
       ! Sets averaging to manual and one sample per reading.
380    OUTPUT @Pwr_mtr;SENS2:AVER:COUN 16
       ! Sets averaging to manual and 16 samples per reading.
390    !
400    OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
       ! Applies Cal Factor for 1.44 GHz to sensor 1 data.
410    OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 0.96E9
       ! Applies Cal Factor for 960 MHz to sensor 2 data.
420    !
430    FOR I= 1 TO 10
```

```
440    OUTPUT @Pwr_mtr;READ1?;READ2?! Measures Power at sensors.
450    ENTER @Pwr_mtr;Chn_pow1
460    ENTER @Pwr_mtr;Chn_1rat2
470    PRINT Chn_pow1,Chn_1rat2
480    NEXT I
490    !
610    END
```

A.4.6 Reading Power Measurements Example 6

FETCh? allows finer control of the meter's measurement sequences. The low level control function of FETCh#? is to first, process the measurement channel information based upon sensor data and configuration settings and then, place the result in the meter data output buffer to be read by the slot 0 controller/resource manager.

The following program uses the INIT command to control acceptance of measurement values in conjunction with the FETCh? command. When using FETCh#?, both trigger sequence arming and triggering/data acquisition must be controlled in your program. This is juxtaposed with MEAS#? and READ#? which, being higher level commands, include these functions.

This program will allow the fastest measurement speed performance in NORMal mode.

```
280    ASSIGN @Pwr_mtr to 70101
290    CLEAR @Pwr_mtr
300    WAIT 1
310    OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in
Normal Mode!!
320    OUTPUT @Pwr_mtr;TRIG:SOUR IMM! Power Meter controls triggering with TRIG or *TRG
330    OUTPUT @Pwr_mtr;INIT:CONT ON! Power Meter controls instrument trigger arming
340    !
350    OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1! Sets averaging to manual and one sample per
reading.
360    OUTPUT @Pwr_mtr;CALC1:POW 2! Channel 1 configured to measure sensor 2 power
370    !
380    OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
       ! Applies Cal Factor for 17.4 GHz to data.
390    !
400    FOR I=1 TO 10
430    OUTPUT @Pwr_mtr;FETC1?! Measures Power at sensor.
440    ENTER @Pwr_mtr;Rdg
450    PRINT Rdg
460    NEXT I
470    !
670    END
```

A.4.7 Reading Power Measurements Example 7

The following example is similar to the example above; however, it now uses TRIG:SOUR BUS instead of IMMediate so that triggering is controlled by the TRIG command. EXT or TTLT triggering can not be used in NORMal Mode. Additionally, INIT:CONT is set to OFF, allowing the INIT (or INIT:IMM) command to control arming of the triggering cycle.

```
280    ASSIGN Pwr_mtr to 70101
290    CLEAR @Pwr_mtr
300    WAIT 1
310    OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in
Normal Mode!!
320    OUTPUT @Pwr_mtr;TRIG:SOUR BUS! Program controls triggering with TRIG or *TRG
330    OUTPUT @Pwr_mtr;INIT:CONT OFF! Program controls instrument trigger arming
340    !
350    OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1! Sets averaging to manual and one sample per
reading.
360    OUTPUT @Pwr_mtr;CALC1:POW 2! Channel 1 configured to measure sensor 2 power
370    !
380    OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
       ! Applies Cal Factor for 17.4 GHz to data.
390    !
400    FOR I=1 TO 10
410    OUTPUT @Pwr_mtr;INIT! INIT arms the triggering and measurement cycle
420    OUTPUT @Pwr_mtr;TRIG! BUS trigger
430    OUTPUT @Pwr_mtr;FETC1?! Measures Power at sensor.
440    ENTER @Pwr_mtr;Rdg
450    PRINT Rdg
460    NEXT I
470    !
670    END
```

A.4.8 Reading Power Measurements Example 8

The following program shows fast BUS triggering in the SWIFt Mode. TRIG (or *TRG) is used to acquire data, and FETCh#? processes and outputs the data to the slot 0 controller/resource manager. This program does not use the meter's data buffering capability.

```
170    ASSIGN @Pwr_mtr to 70101
180    CLEAR @Pwr_mtr
190    WAIT 1
200    OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
205 WAIT 0.3
210    PRINT Running...
220!
230    OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
240    OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
250    ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
260    !
270    !#############################################
280    !
290    ! Entering SWIFt Mode
300    !
310    OUTPUT @Pwr_mtr;CALC1:MODE SWIF! Enters SWIFt Mode for fastest individual data point
       ! triggered measurements.
320    !
330    OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
       ! Applies Cal Factor in SWIFt mode
340    ! Can be sent before or after CALC#:MODE SWIF
350    !
360    OUTPUT @Pwr_mtr;TRIG:SOUR BUS! BUS or EXT triggering is slower than IMM
```

```
370    ! Can be sent before or after CALC#:MODE SWIF
380    !
390    FOR I=1 TO 10
400    FOR K=1 TO 20
410    OUTPUT @Pwr_mtr;TRIG! TRIG is the SCPI Bus trigger. Can also use *TRG
420    OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
430    ENTER @Pwr_mtr;Chan1sens_1(K)
440    !ENTER @Pwr_mtr;Chan1sens_1(K),Chan2sens_2(K)
       ! Use this line when two sensors are attached.
450    NEXT K
460    !
470    PRINT Chan1sens_1(*)
480    !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
490    PRINT “”
500    NEXT I
510    !
520    END
```

A.4.9 Reading Power Measurements Example 9

The following program uses BURSt Mode for the fastest measurement rates possible. The maximum measurement speed is performed when TRIG:DEL is set to 0. TRIG:DEL controls the speed of sampling and data buffering. When TRIG:DEL is set to 0.004, samples will be taken every 4 milliseconds, on both channels if connected, and stored in internal data buffer. This speed does not control or account for the meter’s internal data processing time after data acquisition or the speed of data transfer to your controller. This second component of time, the time to get data from the 58542, is proportional to the number of data points measured. Therefore, the example below uses only one channel and keeps the number of points buffered to a minimum.

Both channels’ data will be taken at the same time during BURSt Mode. Power meter measurement speed does not change when two sensors are connected versus only one sensor. However, the meter’s processing time and the time to transmit the data over the VXI and GPIB interfaces takes longer due to the additional sensor data. If two sensors are connected, calibrated, and their respective channels are set to ON, then you must read both arrays of data. Only read one array of data when one sensor is attached, calibrated, and set to ON.

As shown in the examples, send the CALC#:MODE BURS command prior to sending the BURSt Mode configuration commands TRIG:SOUR, TRIG:MODE, TRIG:COUN, and TRIG:DEL.

```
10     ALPHA ON
20     CLEAR SCREEN
30     !
40     OPTION BASE 1
50     DIM Id$[50],Err_msg$[70]
60     DIM Ques$[200],A$[80],Chan1sens_1(500)
70     DIM Chan2sens_2(500)
80     !
90     !
100    !####################################
110    !      Instrument ADDRESS ALLOCATION
120    !
130    ASSIGN @Slot0 TO 70100! 70100 is Logical Address of the Slot 0 controller
```

```
140    ASSIGN @Pwr_mtr TO 70101! 70101 is Power Meter in 1st position right of slot 0
150    !####################################
160    !
170    ! Identify Attached Instruments
180    !
190    OUTPUT @Slot0;*Slot0;*IDN?
200    ENTER @Slot0;Id$
210    PRINT SLOT 0 is ;Id$
220    !
230    OUTPUT @Pwr_mtr;*IDN?
240    ENTER @Pwr_mtr;Id$
250    PRINT SLOT 1 is ;Id$
260    WAIT 1
270    !
280    !
290    CLEAR @Pwr_mtr
300    WAIT 1
310    OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
320    WAIT 0.3
330    OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
340    OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
350    ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
360    OUTPUT @Pwr_mtr;TRIG:SOUR IMM! IMM set here to highlight conflict with BURSt
operation.
370    !
380    !####################################
390    !
400    ! Entering BURSt Mode
410    !
420    OUTPUT @Pwr_mtr;CALC1:MODE BURS! Enters BURSt Mode for fastest measurement
speeds.
430    !
440    OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
       ! Applies Cal Factor in BURSt mode
450    ! Can be sent before or after CALC#:MODE BURS
460    !
470    OUTPUT @Pwr_mtr;TRIG:SOUR BUS! IMM triggering is illegal in BURSt Mode, Use BUS or
EXT.
480    ! Can be sent before or after CALC#:MODE BURS
490    !
500    OUTPUT @Pwr_mtr;TRIG:MODE POST! Data acquired after trigger, not before as with PRE.
510    ! Send only after CALC#:MODE BURS
520    !
530    OUTPUT @Pwr_mtr;TRIG:COUN 500! 500 readings acquired and stored with each trigger.
540    ! Send only after CALC#:MODE BURS, following TRIG:MODE.
550    ! Be sure COUN# matches ENTER variable dimension.
560    !REDIM Chan1sens_1(500),Chan2sens_2(500)! REDIM to smaller array size only if necessary.
570!
580    OUTPUT @Pwr_mtr;TRIG:DEL .000! 0 millisecond between rdgs setting is 5100 rdgs/sec.
590    ! Send only after CALC#:MODE BURS, following TRIG:COUN.
600    !
610    !
620    FOR I=1 TO 10
630    Time1=TIMEDATE
```

```
640!
650   OUTPUT @Pwr_mtr;TRIG! TRIG is the SCPI Bus trigger. Can also use *TRG
660   OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
670   ENTER @Pwr_mtr;Chan1sens_1(*)
680   !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
      ! Use this line when two sensors are attached.
690   !
700   Time2=TIMEDATE
710   Time=Time2-Time1
720   Speed=500/Time
730   PRINT Chan1sens_1(*)
740   !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
750   PRINT Speed; readings per second, round trip.
760   PRINT “”
770   NEXT I
780   !
790   END
```

A.5 Instrument Triggering Examples

A.5.1 TRIGgering Example 1

The following two programs illustrate the operation of TRIG:SOUR HOLD with the MEAS#? and FETCh#? measurement data queries. This first program shows TRIG:SOUR HOLD used with MEAS#?. Since MEAS#? is a high level command containing it’s own trigger sequence arming, triggering, and measurement data acquisition functions, the program returns valid measurement data. The output from the SYST:ERR query is -0, No error.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
275   WAIT 1
280   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
290   WAIT 0.3
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode.
310   OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
      ! Applies Cal Factor for 1.44 GHz to data.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR HOLD! Halts triggering when used with FETCh#?
350   !
360   FOR I=1 TO 10
370   OUTPUT @Pwr_mtr;MEAS1?! MEAS#? returns valid data with TRIG:SOUR set to HOLD.
380   ENTER @Pwr_mtr;Rdg
390   PRINT Rdg
400   NEXT I
410   !
420   OUTPUT @Pwr_mtr;SYST:ERR?! Query error buffer
430   ENTER @Pwr_mtr;A$
440   PRINT A$
530   END
```

A.5.2 TRIGgering Example 2

When FETCh#? is used while TRIG:SOUR is HOLD, invalid data, 9.e+40, is returned and the SYST:ERR? query returns -230, Data corrupt or stale.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
275 WAIT 1
280   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
290   WAIT 0.3
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode.
310   OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
      ! Applies Cal Factor for 1.44 GHz to data.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR HOLD! Halts triggering when used with FETCh#?
350   !
360   FOR I=1 TO 10
370   OUTPUT @Pwr_mtr;FETC1?! FETC#? will not acquire valid data when TRIG:SOUR
      ! is set to HOLD.
380   ENTER @Pwr_mtr;Rdg
390   PRINT Rdg
400   NEXT I
410   !
420   OUTPUT @Pwr_mtr;SYST:ERR?! Query error buffer
430   ENTER @Pwr_mtr;A$
440   PRINT A$
530   END
```

A.5.3 TRIGgering Example 3

Please note, the READ#? measurement data query requires TRIG:SOUR IMM for proper operation. While TRIG:SOUR is HOLD, data output is also invalid, 9.e+40, but the SYST:ERR? query response is different, -214, Trigger deadlock.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
275   WAIT 1
280   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
290   WAIT 0.3
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode.
310   OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
      ! Applies Cal Factor for 1.44 GHz to data.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR HOLD! Halts triggering when used with FETCh#?
350   !
360   FOR I=1 TO 10
370   OUTPUT @Pwr_mtr;READ1?! READ#? requires TRIG:SOUR IMM not, HOLD.
380   ENTER @Pwr_mtr;Rdg
390   PRINT Rdg
400   NEXT I
410   !
420   OUTPUT @Pwr_mtr;SYST:ERR?! Query error buffer
```

```
430    ENTER @Pwr_mtr;A$
440    PRINT A$
530    END
```

A.5.4 TRIGgering Example 4

The following examples show the use of BUS triggering with FETCh#? in NORMal and BURSt Modes.

```
270    ASSIGN @Pwr_mtr to 70101
280    CLEAR @Pwr_mtr
290    WAIT 1
300    OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode.
310    OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
320    OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
       ! Applies Cal Factor for 1.44 GHz to data.
330    !
340    OUTPUT @Pwr_mtr;TRIG:SOUR BUS
350    !
360    FOR I=1 TO 10
362 OUTPUT @Pwr_mtr;TRIG! TRIG is the SCPI Bus trigger. Can also use *TRG
370    OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
380    ENTER @Pwr_mtr;Rdg
390    PRINT Rdg
400    NEXT I
410    !
530    END
```

A.5.5 TRIGgering Example 5

BURSt Mode BUS triggering with FETCh?

```
10     ALPHA ON
20     CLEAR SCREEN
30     !
40     OPTION BASE 1
50     DIM Id$[50],Err_msg$[70]
60     DIM Ques$[200],A$[80],Chan1sens_1(50)
70     DIM Chan2sens_2(50)
80     !
90     !
100    !#####################################
110    ! Instrument ADDRESS ALLOCATION
120    !
130    ASSIGN @Slot0 TO 70100! 70100 is Logical Address of the Slot 0 controller
140    ASSIGN @Pwr_mtr TO 70101! 70101 is Power Meter, in 1st position right of slot 0
150    !#####################################
160    !
170    ! Identify Attached Instruments
180    !
190    OUTPUT @Slot0;*IDN?
200    ENTER @Slot0;Id$
```

```
210   PRINT SLOT 0 is ;Id$
220   !
230   OUTPUT @Pwr_mtr;*IDN?
240   ENTER @Pwr_mtr;Id$
250   PRINT SLOT 1 is ;Id$
260   WAIT 1
270   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr; *CLS! Clears old messages from SYST:ERR buffer
320   !
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
350   ! RAT and DIFF are illegal in SWIFt and BURSt Modes
360   OUTPUT @Pwr_mtr;TRIG:SOUR IMM! IMM set here to highlight conflict with BURSt
operation
370   !
380   !################################
390   !
400   !Entering BURSt Mode
410   !
420   OUTPUT @Pwr_mtr;CALC1:MODE BURS! Enters BURSt Mode for fastest measurement
speeds
430   !
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
      ! Applies Cal Factor in burst mode
450   ! Can be sent before or after CALC#:MODE BURS
460   !
470   OUTPUT @Pwr_mtr;TRIG:SOUR BUS! IMM triggering is illegal in BURSt Mode, Use BUS or
EXT.
480   ! Can be sent before or after CALC#:MODE BURS
490   !
500   OUTPUT @Pwr_mtr;TRIG:MODE POST! Data acquired after trigger, not before as with PRE.
510   ! Send only after CALC#:MODE BURS
520   !
530   OUTPUT @Pwr_mtr;TRIG:COUN 50! 50 readings acquired and stored with each trigger
540   ! Send only after CALC#:MODE BURS
550   ! Be sure COUN# matches ENTER variable dimension.
560   !REDIM Chan1sens_1(50),Chan2sens_2(50)! REDIM to smaller array size only if necessary
570   !
580   OUTPUT @Pwr_mtr;TRIG:DEL .001! 1 millisecond between rdgs, 0 ms is 5100 rdgs/sec
590   ! Send only after CALC#:MODE BURS
600   !
610   !
620   FOR I=1 TO 10
630   WAIT .01! For handshaking compensation. If necessary, use wait
640
650
660   OUTPUT @Pwr_mtr;TRIG! TRIG is the SCPI Bus trigger. Can also use *TRG
670   WAIT .01! For handshaking compensation. If necessary, use wait
680   OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
690   ENTER @Pwr_mtr;Chan1sens_1(*)
700   !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
```

```
        ! Use this line when two sensors are attached.
710     PRINT Chan1sens_1(*)
720     !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
730     PRINT “”
740     NEXT I
750     !
760     END
```

A.5.6 TRIGgering Example 6

Be careful when you use BURSt Mode. For example, you must use INIT:CONT ON; you don’t have manual control of trigger sequence arming using INIT, as you do using the FETCh#? command with the NORMal Mode. This is of particular concern using TTL level triggering on either the front panel EXT connector or the VXI backplane TTLT trigger functions. You must control the triggering through control of the trigger source. That is, you must use INIT:CONT ON in BURSt Mode; you can’t use INIT control arming of the trigger sequence.

The following example uses EXTernal TTL level triggering using the external trigger input on the front panel of the 58542 VXI Universal Power Meter. Twenty readings are stored in the measurement buffer. Then FETCh#? is used during SWIFt Mode. Please note there is a TTL level hardware handshake capability using the ANALOG OUT BNC connector which is also on the front panel. The Analog BNC will output a high (5V) when the instrument is ready for triggering. After a trigger is received, the Analog BNC output goes low (0V).

TRIG:SOUR HOLD halts the SWIFt Mode triggering sequence.

```
10      ALPHA ON
20      CLEAR SCREEN
30      !
40      OPTION BASE 1
50      DIM Id$[50],Err_msg$[70]
60      DIM Ques$[200],A$[80],Chan1sens_1(20)
70      DIM Chan2sens_2(20)
80      !
90      !
100     !###########################################
110     !       Instrument ADDRESS ALLOCATION
120     !
130     ASSIGN @Slot0 TO 70100! 70100 is Logical Address of the Slot 0 controller
140     ASSIGN @Pwr_mtr TO 70101! 70101 is Power Meter, in 1st position right of slot 0
150     !############################################
160     !
170     !
180     CLEAR @Pwr_mtr
190     WAIT 1
200     OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
210     WAIT 0.3
220     OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
230     OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
240     ! RATio and DIFFerence configurations are illegal in
        ! SWIFt and BURSt Modes.
250     !
```

```
260   !#############################################
270   !
280   ! Entering SWIFt Mode
290   !
300   OUTPUT @Pwr_mtr;CALC1:MODE SWIF! Enters SWIFt Mode for fastest individual data point
      ! triggered measurements.
310   !
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.22E8
      ! Applies Cal Factor in SWIFt mode
330   ! Can be sent before or after CALC#:MODE SWIF
340   !
350   OUTPUT @Pwr_mtr;TRIG:SOUR EXT! TTL level triggering, Each trigger acquires one data
      ! point in SWIFt Mode.
360   ! Can be sent before or after CALC#:MODE SWIF
370   !
380   OUTPUT @Pwr_mtr;TRIG:COUN 20! Stores 20 points for simultaneous output using
      ! FETCh#?.
390   ! Must be sent after CALC#:MODE SWIF
400   !
410   !
420   FOR I=1 TO 10
430   !WAIT 2! If necessary, use wait
440
450   OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
460   ENTER @Pwr_mtr;Chan1sens_1(*)
470   !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
      ! Use this line when two sensors are attached.
480   !
490   PRINT Chan1sens_1(*)
500   !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
510   PRINT “”
520   NEXT I
530   !
540   END
```

The fastest SWIFt Mode measurement speeds are achieved with TRIG:SOUR:IMM and TRIG:COUN set to values larger than about 25. See High Speed Measurements in Section for additional information and examples.

The fastest BURSt Mode measurement speeds are achieved with TRIG:COUN at about 80 and TRIG:DEL 0. See High Speed Measurements in Section for additional information and examples.

A.6 Channel Configuration Examples

A.6.1 Single Sensor Power Measurement

Default definition of software calculation channels 1 and 2 are for sensors 1 and 2, respectively. This configuration allows measurement of the power level incident upon sensor 1 on software configuration channel 1, and measurement of the power level incident upon sensor 2 on software configuration channel

2. Normal Mode and Swift Mode measurements are faster with only one sensor set to STATe ON. Since this is the default configuration, you might want to turn one channel off occasionally. This will only be necessary when two sensors are attached and both are calibrated. Channel configuration can be changed by sending the CALCulate:RATio or CALCulate:DIFFerence commands.

```
OUTPUT @Pwr_mtr;*RST! Configure 58542 to Default Setup
OUTPUT @Pwr_mtr;CALC1?;CALC2?! Query channel configuration
ENTER @Pwr_mtr;Chn1_config,Chn2_config
PRINT Chn1_config,Chn2_config
```

Program Output: POW 1 POW 2

The following program reverses the default sensor-to-channel assignments.

```
OUTPUT @Pwr_mtr;CALC2:POW 1! Configures channel 2 to measure Sensor 1 power
OUTPUT @Pwr_mtr;CALC1:POW 2! Configures channel 1 to measure Sensor 2 power
OUTPUT @Pwr_mtr;CALC1?;CALC2?! Query channel configuration
ENTER @Pwr_mtr;Chn1_config,Chn2_config
PRINT Chn1_config,Chn2_config
```

Program Output:POW 2 POW 1

A.6.2 Ratio Measurement

The CALCulate:RATio command is used to automatically measure the ratio of the power levels incident on the two sensors. Permissible settings are Sensor 1/Sensor 2 & Sensor 2/Sensor 1.

```
OUTPUT @Pwr_mtr;CALC1:RAT 2,1! Configures channel 1 to measure Sensor 2 over
! Sensor 1 power
OUTPUT @Pwr_mtr;CALC2:RAT 1,2! Configures channel 2 to measure Sensor 1 over
! Sensor 2 power
OUTPUT @Pwr_mtr;CALC1?;CALC2?! Query channel configuration
ENTER @Pwr_mtr;Chn1_config,Chn2_config
PRINT Chn1_config,Chn2_config
```

Program Output:RAT2,1 RAT1,2

A.6.3 Difference Measurement

The CALCulate:DIFFerence command is used to automatically measure the difference of the power levels incident upon the two sensors.

```
OUTPUT @Pwr_mtr;CALC1:DIFF 2,1! Configures channel 1 to measure Sensor 2 minus
        ! Sensor 1 power
OUTPUT @Pwr_mtr;CALC1?! Query Channel 1 configuration
ENTER @Pwr_mtr;Chn1_config
PRINT Chn1_config
```

Program Output:DIFF2,1

A.7 Cal Factor Examples

Entering a frequency causes the power meter to use frequency calibration factors which are stored in the power sensor's internal EEPROM. Generally, frequency calibration factors are stored in one gigahertz steps.

Two methods are available for frequency entry. Use SENSe:CORRection:FREQuency to enter a specific carrier frequency, or use SENSe:CORRection:VPROpf to enable the meter's voltage proportional to frequency input BNC. (See front panel of the meter.)

A.7.1 Cal Factor Example 1

The following program automatically applies the correct cal factor for an 8.23 GHz measurement frequency to the measured data value.

```
310   ASSIGN @Pwr_mtr to 70101
320   CLEAR @Pwr_mtr
330   WAIT 1
350   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
      ! Can ONLY send Cal Factor Correction Commands in
      ! Normal Mode!!
360   !
370   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 8.23E9
      ! Applies Cal Factor for 8.23 GHz to measurement data.
380   !
400   OUTPUT @Pwr_mtr;MEAS1?! Measures Power of 8.23 GHz signal.
410   ENTER @Pwr_mtr;Rdg
420   PRINT Rdg
460   !
470   END
```

A.7.2 Cal Factor Example 2

The following program steps from 1.8 GHz to 2.2 GHz in 10 MHz intervals. The measurement at each step is automatically corrected for cal factor.

```
340   ASSIGN @Pwr_mtr to 70101
350   CLEAR @Pwr_mtr
360   WAIT 1
370   REAL Rdg(41)
380   REAL Frq(41)
390   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL! Can ONLY send Cal Factor Correction
Commands in
      ! Normal Mode!!
400   !
410   FOR I=1 TO 41
420   !
430   Freq=1.8E+9+(I-1)*1.E+7
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ;Freq
      ! Applies Cal Factor to measurement data.
450   !
460   OUTPUT @Pwr_mtr;MEAS1?
470   ENTER @Pwr_mtr;Rdg(I)
```

```
480    OUTPUT @Pwr_mtr;SENS1:CORR:FREQ?! Power meter outputs freq setting to confirm
command
       ! reception.
490    ENTER @Pwr_mtr;Frq(I)
500    NEXT I
510    !
520    PRINT Rdg(*)
530    PRINT Frq(*)
540    !
550    END
```

A.7.3 Cal Factor Example 3

The following program also steps from 1.8 GHz to 2.2 GHz in 10 MHz intervals. The measurement at each step is automatically corrected for cal factor using the VpropF connector on the front panel. In the first part of the program the power meter's VpropF input is configured for compatibility with the Giga-tronics Source.

```
340    ASSIGN @Pwr_mtr to 70101
350    CLEAR @Pwr_mtr
360    WAIT 1
370    REAL Rdg(41)
380    OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
       ! Can ONLY send VpropF Config. Commands in Normal
       ! Mode!!
390    !
400    ! Configure VpropF input for Cal Factor correction
410    !
420    OUTPUT @Pwr_mtr;MEM:SEL VPROPF1! Selects memory table for sensor on for editing
430    OUTPUT @Pwr_mtr;MEM:FREQ 0.0! Sets freq that corresponds to 0.0V out from source.
440    OUTPUT @Pwr_mtr;MEM:SLOP 0.5E-9! Sets the voltage to freq (must be linear) relationship of
       ! source's output (V/Hz).
450    !
460    OUTPUT @Pwr_mtr;SENS1:CORR:VPRO ON
       ! Turns VpropF on and all other sources of Cal Factor
       ! correction OFF
470    !
480    FOR I=1 TO 41
490    !
500    Freq=1.8E+9+(I-1)*1.E+7
510    OUTPUT @Source;CW;Freq;HZ
520    !
530    OUTPUT @Pwr_mtr;MEAS1?
540    ENTER @Pwr_mtr;Rdg(I)
550    NEXT I
560    !
570    PRINT Rdg(*)
580    !
590    END
```

A.7.4 Cal Factor Example 4

The following program shows you how to input your own specific User Calibration Factor during measurement. This is not the same as the special Frequency Cal Factors that can be programmed into the EEPROM. (See Sensor EEPROM Commands in Section for more information). This is a technique for you to apply a specific known value during your measurements. It is useful when you have performed a sensor calibration with other devices attached to the sensor input, such as a power splitter or coupler, but you do not want to change or reprogram the CAL factor information inside the power sensor's EEPROM. First the program sets the 58542 to 50 MHz where the Cal Factors of Giga-tronics power sensors are always 0.0 dBm.

```
310   ASSIGN @Pwr_mtr to 70101
320   CLEAR @Pwr_mtr
330   WAIT 1
340   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
      ! Can ONLY send Sensor Offset Commands in
      ! Normal Mode!!
350   !
360   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 5E7
      ! Cal Factors always 0.00dB at 50 MHz.
370   OUTPUT @Pwr_mtr;SENS1:CORR:OFFS:STAT ON
      ! Enables sensor offset control.
380   OUTPUT @Pwr_mtr;SENS1:CORR:OFFS 6.024
      ! Enters 6.024 dB as a Sensor Offset
390   !
400   OUTPUT @Pwr_mtr;MEAS1?! Measures Power
410   ENTER @Pwr_mtr;Rdg
420   PRINT Rdg
430   !
440   END
```

A.7.5 Cal Factor Example 5

In Burst and Swift Mode, the meter's functionality is restricted to allow the microprocessor to devote most of its operation to performing measurement operations. If you are measuring a single frequency, you will not need this technique for your Burst or Swift Mode data. In Burst or Swift Mode, measurement correction for Cal Factor is always performed with the SENS:CORR:FREQ ### command. By sending this command before you enter the Burst or Swift Mode from the Normal mode, all subsequent Burst or Swift Mode measurements will be corrected for that frequency.

The above operation creates a problem for you if you change frequency during Burst or Swift Mode measurement. The following program is used to apply correct cal factors to swept or multi frequency measurements that have been performed during Burst or Swift Modes. In these modes, swept measurement correction functions are performed in your computer, thus increasing the measurement speed more than would otherwise be possible. First the Cal Factors are read from the sensor's EEPROM. Then the Cal Factors are put through an interpolation algorithm into a correction data array that matches the start/stop frequencies of the test source and the number of measurement points to be collected during measurement. Then measurement is performed, and the correction factor array is added to the measurement data array before being output to a file or swept onto a screen display.

```
10     ALPHA ON
20     CLEAR SCREEN
```

```
30    OPTION BASE 1
40    !
50    DIM Freqs(80),Clfcs(80),Corr_clf(5000),Rdgs(5000),Rdgs_corr(5000)
      ! These are in REDIMs later in program
60    DIM Id$[90],Err_msg$[70]
70    DIM Ques$[200],Calf$[200]
80    !
90    Tim_per_pnt = 5 ! milliseconds per sample point. Set 1 to 999.
100   !
110   !############################################
120   !     Instrument ADDRESS ALLOCATION
130   !
140   ASSIGN @Slot0 TO 70100! 70100 is Logical Address of the Slot 0
      ! controller (Resource Manager)
150   ASSIGN @Sweeper TO 720! Use normal address for non-VXI instruments
160   ASSIGN @Pwr_mtr TO 70101! Power Meter is next to the slot 0 Resource Manager
170   !############################################
180   !
190   ! Identify Attached Instruments
200   !
210   OUTPUT @Slot0;*IDN?
220   ENTER @Slot0;Id$
230   PRINT SLOT 0 is ;Id$
240   !
250   OUTPUT @Pwr_mtr;*IDN?
260   ENTER @Pwr_mtr;Id$
270   PRINT SLOT 1 is ;Id$
280   WAIT 1
290   !
300   !###########################################
310   !
320   ! CALIBRATE THE SENSORS
330   !
340   !###########################################
350   !
360   CLEAR @Pwr_mtr
370   WAIT 1
380   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
      ! Can ONLY send Sensor Offset
      ! Commands in Normal Mode!!
390   !
400   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 5E7
      ! Cal Factors always 0.00dB at 50MHz
410   !
420   ! Find out the number of Cal Factors in EEPROM
430   ! Include Std Cal Factors at 1 GHz intervals and any special Cal Factors.
440   !
450   OUTPUT @Pwr_mtr;DIAG:SENS1:EEPROM:CALFR?
460   !ENTER @Pwr_mtr;Frs_std_freq,Std_freq_step,No_std_freqs,No_spl_freqs
470   ENTER @Pwr_mtr;Frst_std_freq,Std_freq_step,No_std_freqs
480   No_spl_freqs=1! Added to correct bug in DIAG:SENS#:EEPROM:CALFR?
490   PRINT Frst_std_freq,Std_freq_step,No_std_freqs,No_spl_freqs
500   !
510   !
```

```
520   ! When you query the 58542 for frequencies and Cal Factors there will be
530   ! No_std_freqs + No_spl_freqs = number of items you need to read
540   !
550   No_cal_pnts=No_std_freqs+No_spl_freqs
560   REDIM Freqs(No_cal_pnts),Clfcs(No_cal_pnt)
      ! Dimension according to number of Cal Factors to be read in.
570   PRINT There are ;No_cal_pnts; of Cal Factors in this sensor.
580   !
590   OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:FREQ?
      ! Asks the 58542 for the Frequency array from sensor 1
600   ENTER @Pwr_mtr;Freqs(*)
610   PRINT Freqs(*)
620   !
630   OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:CALF?
      ! Asks the 58542 for the Frequency array from sensor 1
640   ENTER @Pwr_mtr;Clfcs(*)
650   PRINT Clfcs(*)
660   !
670
!###############################################################################
680   ! Now that all the Cal Factors are loaded with their corresponding frequencies,
690   ! we need to create a table of interpolated Cal Factor points based upon the frequencies used
700   ! and number of measurement points in the test program.
710   !
720   ! First get a couple pieces of necessary information
730   !
740   INPUT Input sweep START frequency in GHz.,Strt_freq
750   INPUT Input sweep STOP frequency in GHz.,Stop_freq
760   INPUT Number of points per sweep. 100 to 400 suggested.,Swep_pnts
770   IF Strt_freq>Stop_freq THEN
780   PRINT Make STOP freq. > START freq.
790   WAIT 1
800   GOTO 740
810   ELSE
820   IF Strt_freq<.051 THEN GOTO 740
830   IF Swep_pnts<1 THEN GOTO 740! You can put additional requirements in this section.
840   END IF
850   Strt_freq=1.E+9*Strt_freq
860   Stop_freq=1.E+9*Stop_fr! Set units to Hz
870   REDIM Corr_clf(Swep_pnts),Rdgs(Swep_pnts),Rdgs_corr(Swep_pnts)
      ! Re-sized to match number of measuremen
      ! points in sweep
880   !
890   ! Interpolation routine creates Cal Factor Correction Table in Clf_corr(*)
900   ! Values are in dB!!!! NOT W linear units
910   !
920   FOR I=1 TO Swep_pnts! For each point in Clf_corr(*)
930   Rdgs(I)=1
940   Corr_freq=Strt_freq+(I-1)*(Stop_freq-Strt_freq)/(Swep_pnts-1)
      ! Corr_freq is freq corresponding to Clf_corr(I)
950   FOR K=1 TO No_cal_pnts! Find next highest and lowest Cal Factor
      ! frequency from Corr_freq
960   IF Freqs(K)>=Corr_freq THEN
970   Next_higher_f=Freqs(K)
```

```
980   Next_higher_clf=Clfcs(K)
990   Next_lower_f=Freqs(K-1)
1000  Next_lower_clf=Clfcs(K-1)
1010  K=No_cal_pnts
1020  !
1030  END IF
1040  NEXT K
1050  F_delta_multpl=(Corr_freq-Next_lower_f)/(Next_higher_f-Next_lower_f)
      ! Multiplier for next formula
1060  Corr_clf(I)=Next_lower_clf+F_delta_multpl*(Next_higher_clf-Next_lower_clf)
      ! Interpolates the Cal Factor Value
1070  PRINT Corr_freq,Corr_clf(I)
1080  NEXT I
1090  !
1100  ! Set the source for swept operation
1110  !
1120  PRINT Configuring Gigatronics 7200 Microwave Sweeper.
1130  OUTPUT @Sweeper,HP! Set Giga-tronics 7200 to HP emulation mode
1140  OUTPUT @Sweeper;OPFA;Strt_freq
1150  OUTPUT @Sweeper;OPFB;Stop_freq
1160  OUTPUT @Sweeper;FC;Tim_per_pnt*Swep_pnts;MS
1170  !
1180  ! Perform Measurement in Burst Mode or Swift Mode for Fast measurements.
1190  ! Shared Variables may include Sweep_pnts and Tim_per_pnt
1200  !
1210  GOSUB Burst_mode_swp! Goes to the swept frequency, Burst/Swift Mode Data
      ! acquisition subroutine
1220  !
1230  ! Data comes back in from Burst Mode-or Swift Mode-as rdgs(*). Always in log units, dBm.
1240  !
1250  MAT Rdgs_corr=Rdgs+Corr_clf! Array math function that adds the measurement
      ! data to the cal factor array
1260  !
1270  GOSUB Output_data! Output to screen graph, printout, file, etc.....
1280  !
1290  END
```

A.8 High Speed Measurement Examples

A.8.1 High Speed Measurement Example 1

The following program shows the fastest SWIFt Mode measurement speed possible. IMMediate triggering is used to allow the 58542 to trigger a measurement automatically. Be sure to only use one channel defined to one sensor for the fastest speeds.

```
170  ASSIGN @Pwr_mtr to 70101
180  CLEAR @Pwr_mtr
190  WAIT 1
200  OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
210  WAIT 0.3
220  OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
```

```
230   OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
240   ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
250   !
260   !################################
270   !
280   ! Entering SWIFt Mode
290   !
300   OUTPUT @Pwr_mtr;CALC1:MODE SWIF! Enters SWIFt Mode for fastest continuous
      ! measurements with IMM.
310   !
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 16.97E9
      ! Applies Cal Factor in SWIFt mode
330   ! Can be sent before or after CALC#:MODE SWIF
340   !
350   OUTPUT @Pwr_mtr;TRIG:SOUR IMM
360
370   !
380   !
390   Loopcount=100
400   Time1=TIMEDATE
410   FOR I=1 TO Loopcount
420   OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
430   ENTER @Pwr_mtr;Chan1sens_1(I)
440   NEXT I
450   Time2=TIMEDATE
460   Time=Time2-Time1
470   Speed=Loopcount/Time! Units are readings per second.
480   !
490   PRINT Chan1sens_1(*)
500   PRINT Speed; readings per second.
510   !
520   END
```

A.8.2 High Speed Measurement Example 2

Using one of the slowest system configurations available, external PC controller with GPIB slot 0 resource manager and programming through a very slow Basic program, the previous program achieved a speed of 24 readings per second. With faster systems using embedded PCs and faster software, measurement speeds have been recorded as high as 71 readings per second.

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(200)
70    DIM Chan2sens_2(200)
80    Counter=1
90    !
100   !###########################################
110   !    Instrument ADDRESS ALLOCATION
120   !
```

```
130   ASSIGN @Slot0 TO 70100! 70100 is Logical Address of the Slot 0controller
140   ASSIGN @Pwr_mtr TO 70101! 70101 is Power Meter in 1st position right of slot 0
150   !###########################################
160   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
320   WAIT 0.3
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
350   ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
370   !
380   !###########################################
390   !
400   ! Entering SWIFt Mode
410   !
420   OUTPUT @Pwr_mtr;CALC1:MODE SWIF! Enters SWIFt Mode for fastest continuous
      ! measurements with IMM.
430   !
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 16.97E9
      ! Applies Cal Factor in SWIFt mode
450   ! Can be sent before or after CALC#:MODE SWIF
460   !
470   OUTPUT @Pwr_mtr;TRIG:SOUR IMM! IMM triggering is illegal in SWIFt Mode, Use BUS or
EXT.
480   ! Can be sent before or after CALC#:MODE SWIF
490   !
610   !
611   Loopcount=50
613   Time1=TIMEDATE
620   FOR I=1 TO Loopcount
670   !OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
671   OUTPUT @Pwr_mtr;FETC1?;FETC2?! Use this line when two sensors are attached.
680   !ENTER @Pwr_mtr;Chan1sens_1(I)
681   ENTER @Pwr_mtr;Chan1sens_1(I),Chan2sens_2(I)
      ! Use this line when two sensors are attached.
730   NEXT I
731   Time2=TIMEDATE
733   Time=Time2-Time1
743   Speed=Loopcount/Time! Units are readings per second.
753   !
763   !PRINT Chan1sens_1(*)
773   PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
783   !PRINT Speed; readings per second.
784   PRINT Speed; readings per second per channel.
793   !
803    END
```

A.8.3 High Speed Measurement Example 3

Using the same system configuration, the previous program achieved a speed of 25 readings per second per channel. With faster systems using embedded PCs and faster software, measurement speeds have been recorded as high as 71 readings per second.

The following program shows fast BUS triggering in the SWIFt Mode. TRIG (or *TRG) is used to acquire data, and FETCh#? processes and outputs the data to the slot 0 controller/resource manager. This program does not use the meter's data buffering capability.

```
170ASSIGN @Pwr_mtr to 70101
180   CLEAR @Pwr_mtr
190   WAIT 1
200   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
205   WAIT 0.3
210   PRINT Running...
220   !
230   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
240   OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
250   ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
260   !
270   !###############################################
280   !
290   ! Entering SWIFt Mode
300   !
310   OUTPUT @Pwr_mtr;CALC1:MODE SWIF! Enters SWIFt Mode for fastest individual data point
      ! triggered measurements.
320   !
330   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
      ! Applies Cal Factor in SWIFt mode
340   ! Can be sent before or after CALC#:MODE SWIF
350   !
360   OUTPUT @Pwr_mtr;TRIG:SOUR BUS! BUS or EXT triggering is slower than IMM
370   ! Can be sent before or after CALC#:MODE SWIF
380   !
390   FOR I=1 TO 10
400   FOR K=1 TO 20
410   OUTPUT @Pwr_mtr;TRIG! TRIG is the SCPI Bus trigger. Can also use *TRG
420   OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
430   ENTER @Pwr_mtr;Chan1sens_1(K)
440   !ENTER @Pwr_mtr;Chan1sens_1(K),Chan2sens_2(K)
      ! Use this line when two sensors are attached.
450   NEXT K
460   !
470   PRINT Chan1sens_1(*)
480   !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
490   PRINT “”
500   NEXT I
510   !
520   END
```

A.8.4 High Speed Measurement Example 4

The following program shows SWIFt Mode measurements using EXT TTL triggering and Buffered data. INIT:CONT must be set to ON; you can not use INIT with INIT:CONT set to OFF. This program buffers 30 measurements in the 58542 before group download to the controller using the TRIG:COUN command in line 390. In the SWIFT mode, this command must be sent after the TRIG:SOUR command.

```
170   ASSIGN @Pwr_mtr to 70101
180   CLEAR @Pwr_mtr
190   WAIT 1
200   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
210   WAIT 0.3
220   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
230   OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
240   ! RATio and DIFFerence configurations are illegal in
      ! SWIFt and BURSt Modes.
250   !
260   !###########################################
270   !
280   ! Entering SWIFt Mode
290   !
300   OUTPUT @Pwr_mtr;CALC1:MODE SWIF! Enters SWIFt Mode for fastest individual data point
      ! triggered measurements.
310   !
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.22E8
      ! Applies Cal Factor in SWIFt mode
330   ! Can be sent before or after CALC#:MODE SWIF
340   !
350   OUTPUT @Pwr_mtr;TRIG:SOUR EXT! TTL level triggering, Each trigger acquires one
      ! data point in SWIFt Mode.
360   ! Can be sent before or after CALC#:MODE SWIF
370   !
380   Num_pts=30
390   OUTPUT @Pwr_mtr;TRIG:COUN;Num_pts! Stores 20 points for simultaneous output using
FETCh#?.
400   ! Must be sent after CALC#:MODE SWIF
410   !
420   REDIM Chan1sens_1(Num_pts),Chan2sens_2(Num_pts)
430   !
440   Time1=TIMEDATE
450   WAIT 5! If necessary, use wait statements
460   ! For example, use it if TRIG:COUN # is high or TTL
      ! trigger source is slow.
470   OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
480   !OUTPUT @Pwr_mtr;FETC1?;FETC2?
490   ENTER @Pwr_mtr;Chan1sens_1(Num_pts)
500   !ENTER @Pwr_mtr;Chan1sens_1(Num_pts),Chan2sens_2(Num_pts)
      ! Use this line when two sensors are attached.
510   Time2=TIMEDATE
520   Time=Time2-Time1
530   Speed=Num_pts/Time
540!
550   PRINT Chan1sens_1(*)
560   !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
570   PRINT Speed; readings per second.
```

```
580  PRINT “”
590  !
600  END
```

A.8.5 High Speed Measurement Example 5

The following program uses BURSt Mode for the fastest measurement rates possible. The maximum measurement speed is performed when TRIG:DEL is set to 0. TRIG:DEL controls the speed of sampling and data buffering. When TRIG:DEL is set to 0.004, samples will be taken every 4 milliseconds, on both channels if connected, and stored in the meter’s internal data buffer. This speed does not control or account for the meter’s internal data processing time after data acquisition or the speed of data transfer to your controller. This second component of time, the time to get data from the 58542, is proportional to the number of data points measured. Therefore, the example below uses only one channel and keeps the number of points buffered to a minimum.

Both channels’ data will be taken at the same time during BURSt Mode. Power meter measurement speed does not change when two sensors are connected versus only one sensor. However, the 58542 processing time and the time to transmit the data over the VXI and GPIB interfaces takes longer due to the additional sensor data. If two sensors are connected, calibrated, and their respective channels are set to ON, then you must read both arrays of data. Only read one array of data when one sensor is attached, calibrated, and set to ON.

As shown in the examples, send the CALC#:MODE BURS command prior to sending the BURSt Mode configuration commands TRIG:SOUR, TRIG:MODE, TRIG:COUN, and TRIG:DEL.

```
10   ALPHA ON
20   CLEAR SCREEN
30   !
40   OPTION BASE 1
50   DIM Id$[50],Err_msg$[70]
60   DIM Ques$[200],A$[80],Chan1sens_1(500)
70   DIM Chan2sens_2(500)
80   !
90   !
100  !########################################
110  ! Instrument ADDRESS ALLOCATION
120  !
130  ASSIGN @Slot 0 TO 70100! 70100 is Logical Address of the Slot 0 controller
140  ASSIGN @Pwr_mtr TO 70101! 70101 is Power Meter in 1st position of slot 0
150  !############################################
160  !
170  ! Identify Attached Instruments
180  !
190  OUTPUT @Slot0;*IDN?
200  ENTER @Slot0;Id$
210  PRINT SLOT 0 is ;Id$
220  !
230  OUTPUT @Pwr_mtr;*IDN?
240  ENTER @Pwr_mtr;Id$
250  PRINT SLOT 1 is ;Id$
260  WAIT 1
270  !
280  !
290  CLEAR @Pwr_mtr
```

```
300   WAIT 1
310   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
320   WAIT 0.3
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
350   ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
360   OUTPUT @Pwr_mtr;TRIG:SOUR IMM! IMM set here to highlight conflict with BURSt
operation
370   !
380   !##########################################
390   !
400   ! Entering BURSt Mode
410   !
420   OUTPUT @Pwr_mtr;CALC1:MODE BURS! Enters BURSt Mode for fastest measurement
speeds.
430   !
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
      ! Applies Cal Factor in BURSt mode
450   ! Can be sent before or after CALC#:MODE BURS
460   !
470   OUTPUT @Pwr_mtr;TRIG:SOUR BUS! IMM triggering is illegal in BURSt Mode, Use BUS or
      ! EXT.
480   ! Can be sent before or after CALC#:MODE BURS
490   !
500   OUTPUT @Pwr_mtr;TRIG:MODE POST! Data acquired after trigger, not before as with PRE.
510   ! Send only after CALC#:MODE BURS
520   !
530   OUTPUT @Pwr_mtr;TRIG:COUN 500! 500 readings acquired and stored with each trigger.
540   ! Send only after CALC#:MODE BURS, following
      ! TRIG:MODE.
550   ! Be sure COUN# matches ENTER variable dimension.
560   !REDIM Chan1sens_1(500),Chan2sens_2(500)! REDIM to smaller array size only if necessary.
570   !
580   OUTPUT @Pwr_mtr;TRIG:DEL .000! 0 millisecond between rdgs setting is 5100rdgs/sec.
590   ! Send only after CALC#:MODE BURS, following TRIG:COUN.
600   !
610   !
620   FOR I=1 TO 10
630   Time1=TIMEDATE
640   !
650   OUTPUT @Pwr_mtr;TRIG! TRIG is the SCPI Bus trigger. Can also use *TRG
660   OUTPUT @Pwr_mtr;FETC1?!FETC#? acquires data
670   ENTER @Pwr_mtr;Chan1sens_1(*)
680   !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
      ! Use this line when two sensors are attached.
690   !
700   Time2=TIMEDATE
710   Time=Time2-Time1
720   Speed=500/Time
730   PRINT Chan1sens_1(*)
740   !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
750   PRINT Speed; readings per second, round trip.
760   PRINT “”
770   NEXT I
```

```
780   !
790   END
```

A.8.6 High Speed Measurement Example 6

The preceding program performed 500 measurements at a rate of 5100 per second then processed and output the data to the controller. Round trip speed was between 120 and 140 per second. Using 100 measurements per BURSt instead of 500, the round trip speed was 100 to 115 per second.

The following example is similar to the previous program except for the use of EXT triggering to initiate the BURSt measurement.

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(500)
70    DIM Chan2sens_2(500)
80    !
90    !
100   !###############################################
110   !    Instrument ADDRESS ALLOCATION
120   !
130   ASSIGN @Slot0 TO 70100! 70100 is Logical Address of the Slot 0 controller
140   ASSIGN @Pwr_mtr TO 70101! 70101 is Power Meter in 1st position right of slot 0
150   !###############################################
160   !
170   ! Identify Attached Instruments
180   !
190   OUTPUT @Slot0;*IDN?
200   ENTER @Slot0;Id$
210   PRINT SLOT 0 is ;Id$
220   !
230   OUTPUT @Pwr_mtr;*IDN?
240   ENTER @Pwr_mtr;Id$
250   PRINT SLOT 1 is ;Id$
260   WAIT 1
270   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;*CLS! Clears old messages from SYST:ERR buffer
320   WAIT 0.3
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM! Normal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1! Channels 1 or 2 can be either POW 1 or POW 2
350   ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
360   !
370   !############################################
380   !
390   ! Entering BURSt Mode
400   !
410   OUTPUT @Pwr_mtr;CALC1:MODE BURS! Enters BURSt Mode for fastest measurement
speeds.
```

```
420  !
430  OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
420  ! Applies Cal Factor in burst mode
440  ! Can be sent before or after CALC#:MODE BURS
450  !
460  OUTPUT @Pwr_mtr;TRIG:SOUR EXT! EXT TTL Level input begins BURSt Mode
measurement.
470  ! Can be sent before or after CALC#:MODE BURS
480  !
490  OUTPUT @Pwr_mtr;TRIG:MODE POST! Data acquired after trigger, not before as with PRE.
500  ! Send only after CALC#:MODE BURS
510  !
520  OUTPUT @Pwr_mtr;TRIG:COUN 500! 500 readings acquired and stored with each trigger.
530  ! Send only after CALC#:MODE BURS, following TRIG:MODE
540  ! Be sure COUN# matches ENTER variable dimension.
550  !REDIM Chan1sens_1(500),Chan2sens_2(500)! REDIM to smaller array size only if necessary.
560  !
570  OUTPUT @Pwr_mtr;TRIG:DEL .000! 0 millisecond between rdgs setting is 5100 rdgs/sec.
580  ! Send only after CALC#:MODE BURS, following TRIG:COUN.
590  !
600  !
610  FOR I=1 TO 10
620  !WAIT .01! Wait for trigger compensation. If necessary, use wait statements
630
640
650  Time1=TIMEDATE
660  !
670  OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
680  WAIT 2! for POST triggering
690  ENTER @Pwr_mtr;Chan1sens_1(*)
700  !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
     ! Use this line when two sensors are attached.
710  !
720  Time2=TIMEDATE
730  Time=Time2-Time1
740  Speed=500/Time
750  PRINT Chan1sens_1(*)
760  !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
770  PRINT Speed; readings per second, round trip.
780  PRINT “”
790  NEXT I
800  !
810  END
```

The preceding program performs measurement at approximately 5100 measurements per second. Assuming pretriggering during the PRINT statements, a round trip speed of 180 to 195 measurements per second was achieved.

A.9 Relative or Referenced Measurement Examples

A.9.1 Relative or Referenced Measurements Example 1

The following program automatically sets a power level reference when the computer's ENTER key is actuated. From that point forward the power level - relative to the power level at the time the ENTER key was actuated - is monitored and displayed with minimum and maximum values since the reference was set.

```
290  ASSIGN @Pwr_mtr to 70101
300  CLEAR @Pwr_mtr
310  WAIT 1
320  !
330  OUTPUT @Pwr_mtr;CALC1:MODE NORMAL! Can ONLY use REFerence commands in
NORMal Mode!!
340  OUTPUT @Pwr_mtr;CALC2:POW 2! Channel 2 configured to measure sensor 2 power
350  !
360  OUTPUT @Pwr_mtr;TRIG:SOUR IMM! TRIG:SOUR must be IMM when using READ#?
370  OUTPUT @Pwr_mtr;INIT:CONT OFF! INITiate:CONTinuous arming must be OFF when using
     ! READ#?
380  !
390  OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1! Sets averaging to manual and one sample per
     ! measurement reading.
400  !
410  OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
     ! Applies Cal Factor for 17.4 GHz to measurement data.
420  !
430  !###########################################
440  !
450  ! Relative measurement setup using CALC#:REF:COLL.
460  !
470  OUTPUT @Pwr_mtr;CALC2:REF:MAG 0.00! Resets REFerence level from value set with
CALC2:REF:COLL
480  OUTPUT @Pwr_mtr;CALC2:REF:STAT ON! Enables REFerence operation.
490  !
500  INPUT Press ENTER key to set to 0.0dB reference operation,Dmy
510  OUTPUT @Pwr_mtr;CALC2:REF:COLL! Takes current measurement and adds inverse to further
     ! measurements
520  !
530  ! Setup power meters' MIN and MAX monitors
540  !
550  OUTPUT @Pwr_mtr;CALC2:MIN:STAT ON
560  OUTPUT @Pwr_mtr;CALC2:MAX:STAT ON
570  !
580  PRINT Power Variation, Largest Min, Largest Max.
590  WHILE Cont_meas=1
600  OUTPUT @Pwr_mtr;MEAS2?! Measures Power at sensor.
610  ENTER @Pwr_mtr;Rdg
620  !
630  OUTPUT @Pwr_mtr;CALC2:MIN?! Grabs minimum since CALC2:MIN:STAT ON
640  ENTER @Pwr_mtr;Min
650  OUTPUT @Pwr_mtr;CALC2:MAX?! Grabs maximum since CALC2:MAX:STAT ON
660  ENTER @Pwr_mtr;Max
670  PRINT Rdg,"",Min,"",Max
```

```
680 END WHILE
690 !
890 END
```

A.9.2 Relative or Referenced Measurements Example 2

The following program finds the 1 dB compression of an amplifier/transmitter at a single frequency.

```
300 ASSIGN @Pwr_mtr to 70101
310 CLEAR @Pwr_mtr
320 WAIT 1
330 !
340 OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
    ! Can ONLY use REFerence commands in NORMal Mode!!
350 OUTPUT @Pwr_mtr;CALC1:POW 1! Channel 1 configured to measure sensor 1 power
360 OUTPUT @Pwr_mtr;CALC1:UNIT W! Transmitter output power in Watts.
370 OUTPUT @Pwr_mtr;CALC2:RAT 1,2! Transmitter gain stays in default, dBm.
380 OUTPUT @Pwr_mtr;CALC2:UNIT DBM! Transmitter output power in Watts. Units are channel,
    ! not sensor, specific
390 !
400 !
410 OUTPUT @Pwr_mtr;TRIG:SOUR IMM! TRIG:SOUR must be IMM when using READ#?
420 OUTPUT @Pwr_mtr;INIT:CONT OFF! INITiate:CONTinuous arming must be OFF when using READ#?
430 !
440 OUTPUT @Pwr_mtr;SENS1:AVER:COUN 1! Sets averaging to manual and one sample per measurement reading.
450 OUTPUT @Pwr_mtr;SENS2:AVER:COUN 4! Sets averaging to manual and 4 samples per measurement reading.
460 !
470 OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.94E9
    ! Applies Cal Factor for 1.94 GHz, Sensor 1.
480 OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 7.0E7
    ! Applies Cal Factor for 70 MHz IF input, Sensor 2.
490 !
500 !##################################################
510 !
520 ! System Setup
530 !   Power Splitter attached to source.
540 !   DUT input and sensor 2 on the splitter outputs.
550 !   Sensor 1 at DUT output.
560 !      Check max power rating on sensor 1 and use an appropriate attenuator if necessary.
570 !      Use offset to account for any necessary attenuation when not using Giga-tronics high power sensors.
580 !
590 !##################################################
600 !
610 ! Relative measurement setup using CALC#:REF:COLL.
620 !
630 OUTPUT @Pwr_mtr;CALC2:REF:MAG 0.00! Resets REFerence level from value set with CALC2:REF:COLL
640 OUTPUT @Pwr_mtr;CALC2:REF:STAT ON! Enables REFerence operation.
650 !
660 Smal_sig_pwr=-30
```

```
670  OUTPUT @Source;OPPL;Smal_sig_pwr! Set power to small signal region of amp, -30dBm.
680  OUTPUT @Pwr_mtr;CALC2:REF:COLL! Takes current measurement and adds inverse to further
     ! measurements
690  !
700  !   STEP POWER FROM SMALL SIGNAL REGION TO ABOVE 1 dB COMPRESSION
LEVEL
710  !
720  FOR I=Smal_sig_pwr TO Smal_sig_pwr+40 STEP .05
730  OUTPUT @Source;OPPL;I! Increments power level into amplifier's input. Add wait statement
     ! at 735 if source settles too slow.
740  OUTPUT @Pwr_mtr;READ2?! Measures Sensor 1 over sensor 2, dB relationship
750  ENTER @Pwr_mtr;Rdg
760  PRINT Rdg
770  !
780  Comp_lvl=-1
790  IF Rdg<Comp_lvl THEN! Finds 1 dB compression level
800  OUTPUT @Pwr_mtr;READ1?! Measures Output Power at sensor 1.
810  ENTER @Pwr_mtr;Rdg
820  PRINT 1 dB Compression level is ;Rdg; Watts, Output Power.
830  GOSUB End
840  END IF
850  NEXT I
860  !
1060 End: !
1070 END
```

A.10 SRQ Interrupt Example

A.10.1 Enable General Operation Interrupts

```
10   ASSIGN @Pwr_mtr to 70101! Set GPIB address
20   DIM Buf$[100]! Define data buffer
30   ON INTR 7 GOSUB Srq_isp! Assign interrupt service function
40   ENABLE INTR 7;2! enable interrupt
50   !
60   CLEAR @Pwr_mtr! Reset instrument
70   WAIT 1
80   OUTPUT @Pwr_mtr;*CLS! Clear interface and output queue
90   WAIT 0.3
100  !
110  OUTPUT @Pwr_mtr;*ESE 1! Enable OPC bit
120  OUTPUT @Pwr_mtr;STAT:OPER:ENAB 7712! Enable channel 1 & 2 limit and trigger wait
mask
130  !
140  Srq_flag=0! Reset task done flag
150  OUTPUT @Pwr_mtr;CAL1:ZERO?;*OPC! Zero sensor 1
160  WHILE Srq_flag = 0! Wait until it is done
170  END WHILE
180  ENTER @Pwr_mtr;Result! Read zero result
190  IF Result = 0 THEN PRINT Zero is good
```

```
200 ELSE
210 PRINT Zero is no good
220 END IF
230 ENTER @Pwr_mtr;Esr_val! Read ESR status
240 !
250 Srq_flag = 0! Reset task done flag
260 Meas_flag = 0! Reset measurement flag
270 OUTPUT @Pwr_mtr;MEAS1?;*OPC! Measure channel 1 data
280 WHILE Srq_flag = 0! Wait until it is done
290 END WHILE
300 IF Meas_flag = 0 THEN! If data is not read
310 ENTER @Pwr_mtr;Result! Read measurement data
320 END IF
330 ENTER @Pwr_mtr;Esr_val! Read ESR status
340 !
350 STOP
360 !
370 Srq_isp:!
380 State = SPOLL(@Pwr_mtr)! Serial poll
390 !
400 IF BIT(State,3) THEN! Error bit
410  OUTPUT @Pwr_mtr;SYST:ERR?! Query error message
420  ENTER @Pwr_mtr;Buf$! Read in
430  PRINT Error:;Buf$! Print out message
440  END IF
450 !
460 IF BIT(State,7) THEN! OSB bit
470 !
480  IF BIT(State,4) THEN! MAV bit
490  ENTER @Pwr_mtr;Result! Query measurement data
500  Meas_flag = 1! Set data read flag
510  END IF
520 !
530  OUTPUT @Pwr_mtr;STAT:OPER?! Query operation status register
540  ENTER @Pwr_mtr;Status
550  PRINT OSB;Status
560  IF BIT(Status,5) THEN PRINT Swift/Burst trigger ready
570  IF BIT(Status,8) THEN PRINT Channel 1 lower limit
580  IF BIT(Status,9) THEN PRINT Channel 1 upper limit
590  IF BIT(Status,10) THEN PRINT Channel 2 lower limit
600  IF BIT(Status,11) THEN PRINT Channel 2 upper limit
610  END IF
620 !
630 IF BIT(State,5) THEN! ESR bit
640  OUTPUT @Pwr_mtr;*ESR?! Reset event status register
650  END IF
660 !
670 Srq_flag = 1! Set task done flag
680 ENABLE INTR 7;2
690 RETURN
860 END
```

A.11 Instrument & Sensor Identification Examples

A.11.1 Instrument Identification

The following example reads manufacturer identification, model number and software version number. Software version number is important for troubleshooting and factory technical support. Make sure you or your system users can identify the software version when requesting technical support.

```
10   ALPHA ON
20   DIM ID$[60],Mfgr$[12]
     ! FOR INSTRUMENT ID VIEWING AT START UP
30   !
40   CLEAR SCREEN
50   !
60   !###################################################
70   ! Instrument ADDRESS ALLOCATION
80   !
90   ASSIGN @Slot0 TO 70100! 70200 is Logical Address of the Slot 0 controller
100  ASSIGN @Pwr_mtr TO 70101! Default Power Meter Address of 255 allows VXI dynamic
110  ! address configuration
120  !
130  !###################################################
140  !
150  OUTPUT @Slot0;*IDN?
160  ENTER @Slot0;Id$
170  PRINT SLOT 0 is ;Id$
180  !
190  OUTPUT @Pwr_mtr;*IDN?
200  ENTER @Pwr_mtr;Mfgr$,Model,No,Software_ver
     ! DIM Mfgr$ to 12 characters
210  PRINT A;Mfgr$; model;Model;with software version;Software_ver;is installed in slot 1.
211  !
212  ! Main Measurement Loop Start.
```

Instrument identification can be read as a single string as is done in lines 150 to 170 above. Sometimes, however, it is more convenient to read these four items separately for use in comment statements or as identifiers for selecting special command libraries or subroutines.

A.11.2 Sensor Model Identification

A simple method of separating the manufacturer sub-string from the three following numeric items requires you to read the manufacturer alphabetical letters into a string variable of limited size. Use a DIM or a REDIM statement to limit size to 12 characters. Thus Model and Software Version can easily be read into numeric variables. No is always 0; it is placed in the *IDN? command output as called for by the IEEE 488.2 Standard. The 58542 can not report it's serial number - which is printed on a label positioned on the side of the instrument - over the bus. If preferred, there is space on the 58542's front panel for the serial number label. The factory will place it there upon request.

The following example reads the model number and serial number of the power sensor currently in use. What does currently in use mean? It means that this was the last sensor connected to this meter input. It does not mean that the sensor is currently attached or that the sensor is calibrated. For in-program checking for sensor attachment or completed power sweep calibration, see the Error Control Examples in Section or the Reading Power Measurements in Section . By operating in this manner, you can use a simple WHILE/UNTIL loop in your program to detect when the operator connects the correct power sensor.

Tracking the model number allows you to be sure that a sensor is being used which is appropriate for your measurement application. For example, some of your measurement routines may require the use of Peak Power Sensors rather than CW Power Sensor or use of high power sensors rather than standard sensors. Your program can automatically check for the appropriate sensor and prompt to Please connect Giga-tronics Sensor model #####, if necessary.

```
240  !
250  GOSUB Pwr_amp_test
260  !
270  GOSUB Store_results
280  !
290  END LOOP! Return to beginning of main program.
300  !
310  Pwr_amp_test:! Output is 30 to 40 Watts, TOO HIGH for Standard Sensors
320  !
330  Reqd_snsr1_mdl=80355! Actual 50W Peak Power Sensor model is 80355A.
340  ! The A is ignored in Gigatronics software to simplify programs like
     ! this one.
350  Reqd_snsr2_mdl=80350! Standard 80350A Peak Power Sensor on input
360  !
370  !
380  !###############################################
390  !
400  ! Read and compare sensors for correct model number
410  !
420  OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:TYPE?
     ! responds with model and serial numbers
430  ENTER @Pwr_mtr;New_snsr1_model,New_snsr1_serno
440  OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:TYPE?
     ! responds with model and serial numbers
450  ENTER @Pwr_mtr;New_snsr1_model,New_snsr1_serno
460  !
470  IF Reqd_snsr1_mdl=New_snsr1_model THEN
480  ELSE
490  PRINT The amplifier’s high power output must be attenuated for this test.,
500  PRINT Connect a Gigatronics model;Reqd_snsr1_mdl;to the channel 1 power sensor cable.,
510  INPUT Then press ENTER,Dmy
520  PRINT""
530  GOSUB Pwr_swp_call
540  END IF
550  !
560  IF Reqd_snsr2_mdl=New_snsr2_model THEN
570  ELSE
580  PRINT Connect a Gigatronics model;Reqd_snsr2_mdl;to the channel 2 power sensor cable.,
590  INPUT Then press ENTER,Dmy
600  GOSUB Pwr_swp_cal2
610  END IF
620  !
630  GOSUB Compare_serno! Calibrates sensors, if necessary. SEE next example
640  !
```

```
650  GOSUB Gain_tst! Configures and performs amplifier gain test.
660  GOSUB Onedb_comp_tst! Configures and performs amplifier 1 dB compression test.
670  !
680  RETURN
690  !
700  Pwr_swp_call:! Power Sweep Calibration calibrates the sensor to the meter.
710  !
```

A.11.3 Sensor Serial Number Identification

Tracking sensor serial numbers is important. Measurements cannot be performed unless the sensor has been calibrated to the meter using the built-in power sweep calibrator: this ensures that measurements will never be performed with an uncalibrated sensor. The Giga-tronics 58542 tracks this requirement by reading the sensor's serial number. Thus, by reading in the sensor's serial number at the beginning of measurement subroutines, you can automatically determine whether or not power sweep calibration is required. This is illustrated in the following example.

```
990  Compare_serno: !
995  !
1000 !##############################################
1010 !
1020 ! This program identifies sensors by their serial number
1030 ! and jumps to a calibration subroutine if it is a new sensor.
1040 !
1050 !##############################################
1060 !
1070 ! Sensor Identification routine at beginning of measurement sequence.
1080 ! New sensors will not operate unless they pass calibration first.
1090 !
1100 OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:TYPE?
1110 ENTER @Pwr_mtr;New_snsr1_model,New_snsr1_serno
1120 OUTPUT @Pwr_mtr;SENS2:CORR:EEPROM:TYPE?
1130 ENTER @Pwr_mtr;New_snsr2_model,New_snsr2_serno
1140 !
1150 IF Old_snsr1_serno=New_snsr1_serno THEN
1160 !
1170 ELSE
1180 GOSUB Pwr_swp_cal1! New Sensor 1, Must Calibrate To the Meter
1190 END IF
1200 !
1210 IF Old_snsr2_serno=New_snsr2_serno THEN
1220 !
1230 ELSE
1240 GOSUB Pwr_swp_cal2! New Sensor 2, Must Calibrate To the Meter
1250 END IF
1260 !
1270 ! Concludes Sensor Serial Number Comparison
1280 ! Use a similar routine when a particular sensor model is required.
1290 ! To do both: Nest an IF-THEN for model # ahead of GOSUB
     ! Pwr_swp_cal#.
1300 !
1310 !########################################################
1320 !
1330 ! Main loop
```

```
1340  !
1350  OUTPUT @Pwr_mtr;SENS1:AVER:TCON:MOV
      ! MOV, Average new sample with previous samples
1360  OUTPUT @Pwr_mtr;SENS2:AVER:TCON:MOV
      ! MOV, Average new sample with previous samples
1370  !
1380  LOOP
1390  OUTPUT @Pwr_mtr;MEAS1?
      ! TRIGGER AND READ CHANNEL 1
1400  ENTER @Pwr_mtr;MEAS1
1410  PRINT CHANNEL 1 IS ;Meas1;dBm.
1420  !
1430  OUTPUT @Pwr_mtr;MEAS2?! TRIGGER AND READ CHANNEL 2
1440  ENTER @Pwr_mtr;Meas2
1450  PRINT CHANNEL 2 IS ;Meas2;dBm.
1460  END LOOP
1470  !
1480  !########################################################
1490  !
1500  ! SUBROUTINES
1510  !
1520  !
1530  Pwr_swp_cal1:! Calibrates Sensor 1 Power Sweep Calibration calibrates the
      ! sensor to the meter.
1540  !
1541  PRINT Calibrating.........
1550  OUTPUT @Pwr_mtr;CAL1?! Starts Calibration and delivers a pass/fail flag
1560  ENTER @Pwr_mtr;Pass_cal
1570  IF Pass_cal=0 THEN
1580  PRINT Calibration Passed
1590  Old_snsr1_model=New_snsr1_model
1600  Old_snsr1_serno=New_snsr1_serno
1610  ELSE
1620  PRINT Calibration FAILED, Sensor 1 Attached To Calibrator?
1630  WAIT 10! Wait for Operator to attach sensor
1640  GOTO Pwr_swp_cal1
1650  END IF
1660  RETURN
1670  !
1680  Pwr_swp_cal2:! Calibrates Sensor 2
1690  !
1691 PRINT Calibrating........
1700  OUTPUT @Pwr_mtr;CAL2?! Starts Calibration and delivers a pass/fail flag
1710  ENTER @Pwr_mtr;Pass_cal
1720  IF Pass_cal=0 THEN
1730  PRINT Calibration Passed
1740  Old_snsr2_model=New_snsr2_model
1750  Old_snsr2_serno=New_snsr2_serno
1760  ELSE
1770  PRINT Calibration FAILED, Sensor 2 Attached To Calibrator?
1780  WAIT 10! Wait for Operator to attach sensor
1790  GOTO Pwr_swp_cal2
1800  END IF
```

```
1810 RETURN
1940 END
```

B

Power Sensors

B.1 Introduction

This appendix contains the selection, specifications and calibration data for power sensors used with Giga-tronics power meters. This appendix is divided into the following sections:

- Power Sensor Selection
- Peak Power Sensor Selection
- Remote Calibration Factors

B.2 Power Sensor Selection

The Standard 80300A Series Sensors measure CW signals from -70 to +20 dBm; the 58542 Universal Power Meters also use the 8035XA Series Peak Power Sensors for measuring radar and digital modulation signals.

Giga-tronics True RMS sensors are recommended for applications such as measuring quadrature modulated signals, multi-tone receiver intermodulation distortion power, noise power, or the compression power of an amplifier. These sensors include a pad to attenuate the signal to the RMS region of the diode's response. This corresponds to the -70 dBm to -20 dBm linear operating region of Standard CW Sensors. The pad improves the input VSWR to ≤1.15 at 18 GHz.

High Power (1, 5, 25 and 50 Watt) and Low VSWR sensors are also available for use with the power meter.

Table B-1 lists the Giga-tronics power sensors used with the power meters.

B.2.1 CW Power Sensors

Table B-1: Power Sensor Selection Guide

Model	Freq. Range/ Power Range	Max. Power	Power Linearity4 (Freq >8 GHz)	RF Conn	Length	Dia.	Wgt	VSWR
Standard CW Sensors								
80301A	10 MHz to 18 GHz -70 to +20 dBm	+23 dBm (200 mW)	-70 to -20 dBm -0.00 dB -20 to +20 dBm -0.05 dB/10 dB	Type N(m) 50W	114.5 mm (4.5 in)	32 mm (1.25 in)	0.18 kg (0.4 lb)	1.12:0.01 - 2 GHz 1.22:2 - 12.4 GHz 1.29:12.4 - 18 GHz
80302A				APC-7 50W				
80303A	10 MHz to 26.5 GHz -70 to +20 dBm		-70 to +20 dBm -0.00 dB -20 to +20 dBm -0.1 dB/10dB	Type K(m)[1] 50W				1.12:0.01 - 2 GHz 1.22:2 - 12.4 GHz 1.38:12.4 - 18 GHz 1.43:18 - 26.5 GHz 1.92:26.5 - 40 GHz
80304A	10 MHz to 40 GHz -70 to 0 dBm		-70 to -20 dBm -0.00 dB -20 to 0 dBm -0.2 dB/10 dB					
Low VSWR CW Sensors								
80310A	10 MHz to 18 GHz -64 to +26 dBm	+29 dBm (800 mW)	-64 to -14 dBm -0.00 dB -14 to + 26 dBm -0.05 dB/10 dB	Type K(m)[1] 50W	127mm (5.0 in)	32 mm (1.25 in)	0.23 kg (0.5lb)	1.13:0.01 - 2 GHz 1.15:2 - 12 GHz 1.23:12 - 18 GHz 1.29:18 - 26.5 GHz 1.50:26.5 - 40 GHz
80313A	10 MHz to 26.5 GHz -64 to +26 dBm		-64 to -14 dBm -0.00 dB -14 to + 26 dBm -0.1 dB/10 dB					
80314A	10 MHz to 40 GHz -64 to +6 dBm		-64 to -14 dBm -0.00 dB -14 to + 6 dBm -0.2 dB/10 dB					
1W CW Sensors								
80320A	10 MHz to 18 GHz -60 to +30 dBm	+30 dBm (1 W)	-60 to -10 dBm -0.00 dB -10 to +30 dBm -0.05 dB/10 dB	Type K(m)[1] 50W	127 mm (5.0 in)	32 mm (1.25 in)	0.23 kg (0.5 lb)	1.11:0.01 - 2 GHz 1.12:2 - 12 GHz 1.18:12 - 18 GHz 1.22:18 - 26.5 GHz 1.36:26.5 - 40 GHz
80323A	10 MHz to 26.5 GHz -60 to +30 dBm		-60 to -10 dBm -0.00 dB -10 to +30 dBm -0.1 dB/10 dB					
80324A	10 MHz to 40 GHz -60 to +10 dBm		-60 to -10 dBm -0.00 dB -10 to +10 dBm -0.2 dB/10 dB					
5W CW Sensor2								
80321A	10 MHz to 18 GHz -50 to +37 dBm	+37 dBm (5 W)	-50 to +0 dBm -0.00 dB 0 to +37 dBm -0.05 dB/10 dB	Type N(m) 50W	150 mm (5.9 in)	32 mm (1.25 in)	0.23 kg (0.5 lb)	1.20:0.01 - 2 GHz 1.25:6 - 12.4 GHz 1.35:12.4 - 18 GHz
25W CW Sensor[3]								
80322A	10 MHz to 18 GHz -40 to +44 dBm	+44 dBm (25 W)	-40 to +10 dBm -0.00 dB +10 to +44 dBm -0.05 dB/10 dB	Type N(m) 50W	230 mm (9.0 in)	104 mm (4.1 in)	0.3 kg (0.6 lb)	1.20:0.01 - 2 GHz 1.30:6 - 12.4 GHz 1.40:12.4 - 18 GHz
50W CW Sensor[3]								
80325A	10 MHz to 18 GHz -40 to +47 dBm	+47 dBm (50 W)	-40 to +10 dBm -0.00 dB +10 to +47 dBm -0.05 dB/10 dB	Type N(m) 50W	230mm (9.0 in)	104 mm (4.1 in)	0.3 kg (0.6 lb)	1.25:0.01 - 2 GHz 1.35:6 - 12.4 GHz 1.45:12.4 - 18 GHz
True RMS Sensors (-30 to +20 dBm)								
80330A 80333A 80334A	10 MHz to 18 GHz 10 MHZ to 26.5 GHz 10 MHz to 40 GHz	+33 dBm (2 W)	-30 to +20 dBm -0.00 dB	Type K(m)[1] 50W	152.5 mm (6.0 in)	32 mm 1.25 in)	0.27 kg (0.6 lb)	1.12:0.01 - 12 GHz 1.15:12 - 18 GHz 1.18:18 - 26.5 GHz 1.29:26.5 - 40 GHz
80340 Series Peak Power Sensors (-30 to +20 dBm)								

Table B-1: Power Sensor Selection Guide (Continued)

Model	Freq. Range/ Power Range	Max. Power	Power Linearity4 (Freq >8 GHz)	RF Conn	Length	Dia.	Wgt	VSWR
80340A	50 MHz to 18 GHz	+23 dBm (200 mW)	-30 to -20 dBm -0.13 dB 0 to +20 dBm	Type N(m)[1] 50W	146 mm (5.75 in)	37 mm (1.44 in)	0.3 kg (0.6lb)	1.12:0.01 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 - 18 GHz
80343A 80344A	50 MHz to 26.5 GHz 50 MHz to 40 GHz		-0.13 dB -0.01 dB/dB	Type K(m)[1] 50W				1.50:18 - 26.5 GHz 1.92:26.5 - 40 GHz

Notes:

1. The K connector is electrically and mechanically compatible with the APC-3.5 and SMA connectors.
2. Power coefficient equals <0.01 dB/Watt.
3. Power coefficient equals <0.015 dB/Watt.
4. For frequencies above 8 GHz, add power linearity to system linearity.
5. Peak operating range above CW maximum range is limited to <10% duty cycle.
6. Includes uncertainty of reference standard and transfer uncertainty. Directly traceable to NIST.
7. Square root of sum of the individual uncertainties squared (RSS).
8. Cal Factor numbers allow for 3% repeatability when connecting attenuator to sensor, and 3% for attenuator measurement uncertainty and mismatch of sensor/pad combination. Attentuator frequency response is added to the Sensor Cal Factors which are stored in he sensor's EEPROM.

Table B-2: Power Sensor Cal Factor Uncertainties

Freq. (GHz)		Sum of Uncertainties (%)[6]						Probable Uncertainties (%)[7]					
Lower	Upper	80301A 80302A 80303A 80304A 80340A 80343A 80344A	80310A 80313A 80314A	80320A 80323A 80324A	80321A[8] 80322A[8] 80325A[8]	80330A 80333A 80334A	80301A 80302A 80340 80401A 80402A 80303A 80304A 80343 80344	80310A 80313A 80314A	80320A 80323A 80324A	80321A[8] 80322A[8] 80325A[8]	80330A 80333A 80334A	80321A[8] 80322A[8] 80325A[8]	80330A[8] 80333A[8] 80334A[8]
0.1	1	1.61	3.06	2.98	2.96	7.61	2.95	1.04	1.64	1.58	1.58	4.54	1.58
1	2	1.95	3.51	3.58	3.57	7.95	3.55	1.20	1.73	1.73	1.73	4.67	1.73
2	4	2.44	4.42	4.33	4.29	8.44	4.27	1.33	1.93	1.91	1.91	4.89	1.90
4	6	2.67	4.74	4.67	4.63	8.67	4.60	1.41	2.03	2.02	2.01	5.01	2.01
6	8	2.86	4.94	4.87	4.82	8.86	4.80	1.52	2.08	2.07	2.06	5.12	2.06
8	12.4	3.59	6.04	5.95	5.90	9.59	5.87	1.92	2.55	2.54	2.53	5.56	2.53
12.4	18	4.09	6.86	6.76	6.69	10.09	6.64	2.11	2.83	2.80	2.79	5.89	2.78
18	26.5	——	9.27	9.43	9.28	——	9.21	——	3.63	3.68	3.62	——	3.59
26.5	40	——	15.19	14.20	13.86	——	13.66	——	6.05	5.54	5.39	——	5.30

Notes:

1. The K connector is electrically and mechanically compatible with the APC-3.5 and SMA connectors.
2. Power coefficient equals <0.01 dB/Watt.
3. Power coefficient equals <0.015 dB/Watt.
4. For frequencies above 8 GHz, add power linearity to system linearity.
5. Peak operating range above CW maximum range is limited to <10% duty cycle.
6. Includes uncertainty of reference standard and transfer uncertainty. Directly traceable to NIST.
7. Square root of sum of the individual uncertainties squared (RSS).
8. Cal Factor numbers allow for 3% repeatability when connecting attenuator to sensor, and 3% for attenuator measurement uncertainty and mismatch of sensor/pad combination. Attentuator frequency response is added to the Sensor Cal Factors which are stored in he sensor’s EEPROM.

B.2.2 Peak Power Sensors

Table B-3: 8035XA Peak Power Sensor Selection Guide

Peak Power Sensors								
Model	Freq. Range/ Power Range	Max. Power	Power Linearity[4]	RF Conn	Dimensions		Wgt	VSWR
					Length	Dia.		
Standard Peak Power Sensors								
80350A	45 MHz to 18 GHz -20 to +20 dBm, Peak -30 to +20 dBm, CW	+23 dBm (200 mW) CW or Peak	-30 to -20 dBm -0.00 dB -20 to +20 dBm -0.05 dB/ 10 dB	Type N(m) 50W	165 mm (6.5 in)	37 mm 1.25 in)	0.3 kg (0.7 lb)	1.12:0.045 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 -18 GHz
80353A	45 MHz to 26.5 GHz -20 to +20 dBm, Peak -30 to +20 dBm, CW		-30 to -20 dBm -0.00 dB -20 to +20 dBm -0.1 dB/ 10 dB	Type K(m)[1] 50W				1.12:0.045 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 -18 GHz 1.50:18 - 26.5 GHz
80354A	45 MHz to 40 GHz -20 to +0.0 dBm, Peak -30 to +0.0 dBm, CW		-30 to -20 dBm -0.00 dB -20 to 0.0 dBm -0.2 dB/ 10dB					1.12:0.045 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 -18 GHz 1.50:18 - 26.5 GHz 1.92:26.5 - 40 GHz
5W Peak Power Sensor [2,5]								
80351A	45 MHz to 18 GHz 0.0 to +40 dBm, Peak -10 to +37 dBm, CW	CW: +37 dBm (5 W Avg.) Peak: +43 dBm	-10 to +0 dBm -0.00 dB +0 to +40 dBm -0.05 dB/ 10 dB	Type N(m) 50W	200 mm (7.9 in)	37 mm (1.25 in)	0.3 kg (0.7 lb)	1.15:0.045 - 4 GHz 1.25:4 - 12.4 GHz 1.35:12.4 -18 GHz
25W Peak Power Sensor [3,5]								
80352A	45 MHz to 18 GHz +10 to +50 dBm, Peak 0.0 to +44 dBm, CW	CW: +44 dBm (25 W Avg.) Peak: +53 dBm	0.0 to +10 dBm -0.00 dB +10 to +50 dBm -0.05 dB/ 10 dB	Type N(m) 50W	280 mm (11.0 in)	104 mm (4.1 in)	0.3 kg (0.7 lb)	1.20:0.045 - 6 GHz 1.30:6 - 12.4 GHz 1.40:12.4 -18 GHz
50W Peak Power Sensor[3,5]								
80355A	45 MHz to 18 GHz +10 to +50 dBm, Peak 0.0 to +47 dBm, CW	CW: +47 dBm (50 W Avg.) Peak: +53 dBm	0.0 to +10 dBm -0.00 dB +10 to +50 dBm -0.05 dB/ 10 dB	Type N(m) 50W	280 mm (11.0 in)	104 mm (4.1 in)	0.3 kg (0.7 lb)	1.25:0.045 - 6 GHz 1.35:6 - 12.4 GHz 1.45:12.4 -18 GHz

Notes:

1. The K connector is electrically and mechanically compatible with the APC-3.5 and SMA connectors.
2. Power coefficient equals <0.01 dB/Watt (AVG).
3. Power coefficient equals <0.015 dB/Watt (AVG).
4. For frequencies above 8 GHz, add power linearity to system linearity.
5. Peak operating range above CW maximum range is limited to <10% duty cycle.

Table B-4: Peak Power Sensor Cal Factor Uncertainties

Freq. (GHz)		Sum of Uncertainties (%)[1]					Probable Uncertainties (%)[2]		
Lower	Upper	80350A	80353A 80354A	80351A[3]	80352A[3]	80355A[3]	80350A	80353A 80354A	80351A[3] 80352A[3] 80355A[3]
0.1	1	1.61	3.06	9.09	9.51	10.16	1.04	1.64	4.92
1	2	1.95	3.51	9.43	9.85	10.50	1.20	1.73	5.04
2	4	2.44	4.42	13.10	13.57	14.52	1.33	1.93	7.09
4	6	2.67	4.74	13.33	13.80	14.75	1.41	2.03	7.17
6	8	2.86	4.94	13.52	13.99	14.94	1.52	2.08	7.25
8	12.4	3.59	6.04	14.25	14.72	15.67	1.92	2.55	7.56
12.4	18	4.09	6.86	19.52	20.97	21.94	2.11	2.83	12.37
18	26.5	——	9.27	——	——	——	——	3.63	——
26.5	40	——	15.19	——	——	——	——	6.05	——

Notes:

1. Includes uncertainty of reference standard and transfer uncertainty. Directly traceable to NIST.
2. Square root of sum of the individual uncertainties squared (RSS).
3. Cal Factor numbers allow for 3% repeatability when connecting attenuator to sensor, and 3% for attenuator measurement uncertainty and mismatch of sensor/pad combination. Attenuator frequency response is added to the Sensor Cal Factors which are stored in the sensor's EEPROM.

For additional specifications, see the Series 80350A manual and data sheet.

B.2.3 Directional Bridges

The 80500 CW Directional Bridges are designed specifically for use with Giga-tronics power meters to measure the Return Loss/SWR of a test device. Each bridge includes an EEPROM which has been programmed with Identification Data for that bridge.

Table B-5: Directional Bridge Selection Guide

Bridge Selection Guide								
Model	Freq. Range/ Power Range	Max. Power	Power Linearity[4]	Input	Test Port	Direct-tivity	Wgt	**VSWR**
80501	10 MHz to 18 GHz -35 to +20 dBm	+27 dBm (0.5W)	-35 to +10 dBm -0.1 dB +10 to +20 dBm -0.1 dB -0.005 dB/dB	Type N(f) 50 W	Type N(f) 50 W	38 dB	0.340 kg	<1.17:0.01 - 8 GHz <1.27:8 - 18 GHz
80502					APC-7(f) 50 W	40 dB		<1.13:0.01 - 8 GHz <1.22:8 - 18 GHz
80503	10 MHz to 26.5 GHz -35 to +20 dBm			SMA(f) 50 W	SMA(f) 50 W	35 dB		<1.22:0.01 - 8 GHz <1.22:8 - 18 GHz <1.27:18 - 26.5 GHz
80504	10 MHz to 40 GHz -35 to +20 dBm			Type K(f) 50 W	Type K(f) 50 W	30 dB	0.198 kg	<1.35:0.01 - 8 GHz <1.35:8 - 18 GHz <1.35:18 - 26.5 GHz <1.44:26.5 - 40 GHz

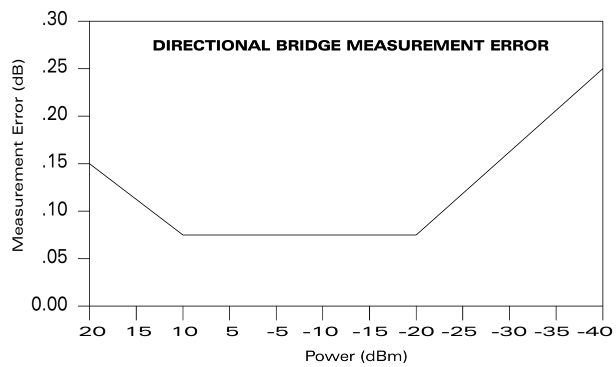
The Selection Guide in Table B-5 shows primary specifications. Additional specifications are:

Bridge Frequency Response: Return loss measurements using the 8541/2 power meter can be frequency compensated using the standard Open/Short supplied with the bridge.

Insertion Loss: 6.5 dB, nominal, from input port to test port

Maximum Input Power: +27 dBm (0.5 W)

Directional Bridge Linearity Plus Zero Set & Noise vs. Input Power (50 MHz, 25 C ±5 C):



Dimensions:
80501: 76 x 50 x 28 mm (3 x 2 x 1 1/8 in)
80502: 76 x 50 x 28 mm (3 x 2 x 1 1/8 in)
80503: 19 x 38 x 29 mm (3/4 x 1 1/2 x 2 1/8 in)
80504: 19 x 38 x 29 mm (3/4 x 1 1/2 x 2 1/8 in)

Weight:
80501: 340 g (12 oz)
80502: 340 g(12 oz)
80503: 198 g (7 oz)
80504: 198 g (7 oz)

Directional Bridge Accessories: An Open/Short is included for establishing the 0 dB return loss reference during path calibration.

B.3 Power Sensor Calibration

This procedure is for calibrating a power sensor by remote control with a Model 58542 Universal Power Meter over the IEEE 488 interface bus. This procedure writes the cal factors to the sensor EEPROM.

Power sensors have built-in EEPROM data that manage the cal factors by a set of frequencies entered during calibration of the sensor at the factory. You can program additional cal factors with special data for user-specific frequencies.

A cal factor expressed in dB is programmed for each factory-calibrated frequency. The calibration process compares the measurement to the frequency standard and applies the cal factor to offset frequency deviations.

Equipment Required

58542 Universal Power Meter • GPIB Controller • Power Sensor

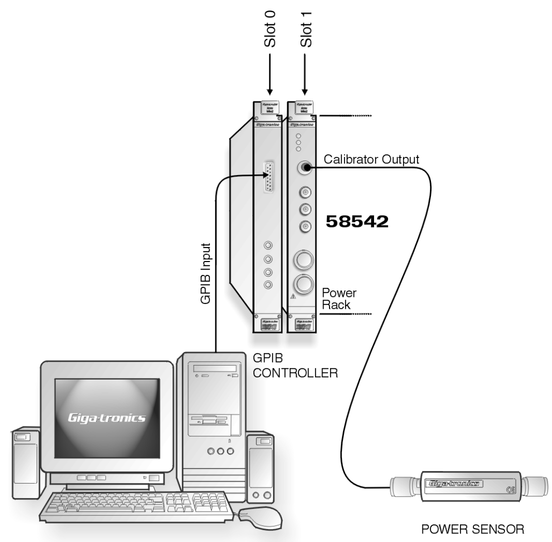


Figure B-1: Power Sensor Calibration Setup

Procedure

Connect the power sensor to Channel A or B on the Series 58542 front panel and perform the following steps. In this procedure, bold letters are commands; the query form of a command has a question mark (?) at the end of the command. This form returns the data in the EEPROM.

1. DIAG:SENS1 (or 2):EEPROM:READ

   Read sensor 1 (or 2) EEPROM data into the 58542 editor buffer.

   Example: OUTPUT@ Pwr_mtr; DIAG:SENSl:EEPROM:READ

2. (Optional) DIAG:SENSl (or 2):EEPROM:CALFREQST?

   a.) Query the sensor 1 (or 2) standard cal factor frequency table.

   b.) Read the standard cal from the input buffer and parse the return string for the start frequency and number of standard frequencies.

   c.) Calculate and set the frequencies of the cal factor table.

3. DIAG:SENSl (or 2):EEPROM:CALFST?
   a.) Query sensor 1 (or 2) standard cat factor table.
   b.) Read the standard cal from the input buffer and parse the return string for the standard cal factors.
   c.) Modify the standard cal factor string to new cal factor table.
   d.) Write new cat factor table into 58542 editor buffer.

4. DIAG:SENS1 (or 2):EEPROM:WRIT 0

   Commit 58542 to write new cal factor table from editor buffer to sensor 1 (or 2) EEPROM. The sensor EEPROM routine will complete momentarily. Trigger LED will activate during sensor write routine. 0 (zero) at end of command string indicates no password is set for sensor write routine. If a password (numerical code) has been previously set, the 0 is replaced with password. The password is used to prevent unauthorized entry of cal factors. Password factory default is 0.

   i.e. DIAG:SENSl:EEPROM:WRIT 4369 Send new cal factors to EEPROM

Short program example

```
ASSIGN @ Pwr_mtr to 70101 'set Pwr_mtr to primary address 1 and secondary address 1
CLEAR @ Pwr_mtr
OUTPUT @ Pwr_mtr; DIAG:SENS1:EEPROM:READ
OUTPUT @ Pwr_mtr; DIAG:SENS1 :CALFST "0.20,0.30,0.40,0.50,0.60,0.70,0.80,
0.90,0.10,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18"
OUTPUT @ Pwr_mtr; DIAG:SENS1:EEPROM:WRIT 0
```

*

NOTE: *Because the DIAG:SENS1:CALFST requires quotation marks within the string, some programming languages (i.e. Visual BASIC) will report compile errors. In this case, the command string can be generated as a string variable using chr$(34) as a replacement for the quotation mark. See example below.*

```
Example:  wrt$ = "DIAG:SENS1 :CALFST " + chr$(34) + "0.20,0.30,0.40,0.50,0.60,0.70,0.80,
0.90,0.10,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18" + chr$(34)
```

where wrt$ is the string variable.

C

Options

C.1 Introduction

This appendix describes all options that are currently available for use with the Model 58542 VXIbus Universal Power Meter.

C.2 Option 02: 256K Buffer

Option 02, part number 21335, adds 256K Buffer for Burst Mode readings. The following parts will be installed at the factory when the 58542 is ordered with this option.

21335 EXTRA MEMORY, 128K, 8540/02, REV. A						
	Item	P/N	Qty	Cage	Mfr’s P/N	Description
U	29	21165	1	61802	TC551001BPL-85	TC551001BPL-10 1M RAM
U	30	21165	1	61802	TC551001BPL-85	TC551001BPL-10 1M RAM

Index

58542 VXIbus Universal Power Meter

T

Z