Standard Commands for Programmable Instruments (SCPI)

Volume 4: Instrument Classes

VERSION 1999.0

May, 1999

Printed in U.S.A.

Table of Contents

Chapter 1 Introduction

Chapter 2 Chassis Dynamometers

Chapter 7 Power Supplies

Chapter 8 RF & Microwave Sources

Chapter 9 Signal Switchers

1 Introduction

This volume defines the SCPI commands and behavior needed to implement functionality sets associated with common classes of instruments. Each chapter covers a particular instrument class. A class could be, for example, a power supply, voltmeter, or switcher. Each chapter stands by itself so a designer may focus on only the chapters that apply to a particular design.

1.1 Scope

This volume covers the major classes of electronic instruments supported by the commands and behaviors as defined in SCPI Volumes 1 through 3. This volume does not extend SCPI with unsupported or contradictory commands or behaviors.

As new applications for SCPI are explored, additional chapters will be added.

1.2 Definition of Terms

Functionality - an aspect of instrument operation which is described by SCPI commands and their associated behavior.

- **Functionality set**
  A grouping of functionality which shall be implemented as a whole.
- **Instrument class**
  A generally recognized type of instrument, for example switcher, power supply, or voltmeter.
- **Base functionality**
  That functionality set required in an instrument of a given class.
- **Additional functionality**
  One or more functionality sets defined for an instrument class beyond the base functionality.

1.3 Purpose

This volume is designed to further the overall SCPI goal of reducing ATE program development time by:

- Reducing the ATE product development time by guiding designers in the usage of SCPI from the more familiar, instrument class point of view.
- Achieving a higher degree of consistency among implementations of the same instrument class.

1.3.1 Guiding Designers

The Command Reference Volume describes commands that cover a wide variety of instrument classes and technologies. Picking commands from this alphabetically sorted list appropriate for a particular instrument can be daunting.

This volume uses terms familiar to the designer to help relate SCPI to previous design experiences. Volume 4 provides a translation from the SCPI Command Reference vocabulary to the vocabulary common to an instrument class.

The Command Reference purposely presents certain subsystems in broad terms. For example, triggering is a function used in almost all instruments, but in slightly different ways. This volume provides further guidance to designers on how to apply certain subsystems, like TRIGger, to a particular instrument class.

1.3.2 Achieving Consistency

Because SCPI covers many instrument classes, it cannot require all instruments to implement all commands. It can, however, define an instrument base functionality by specifying a relatively small set of commands and behaviors. A programmer can depend on a level of consistency among SCPI instruments of the same class.

By staying within the base functionality, a programmer can avoid all Command Errors. Execution errors are still possible. For example, this volume cannot specify <numeric_value> ranges. The programmer, however, can use the required MIN/MAX queries to establish range limits. Specific required values are shown where the parameter type is <CHARACTER PROGRAM DATA>.

Additional functionality may be defined for some instrument classes. None of these functionality sets are required, but if any set is implemented it shall be implemented in its entirety.

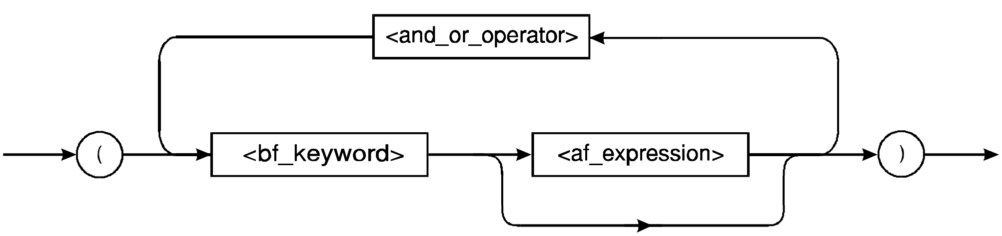
1.4 Instrument Classification

Each chapter defines base functionality for an instrument class. This base functionality is designated with a keyword using SCPI program mnemonic rules. Keywords are also defined for additional functionality. An ongoing goal of the this volume is to further classify additional functionality for each instrument class.

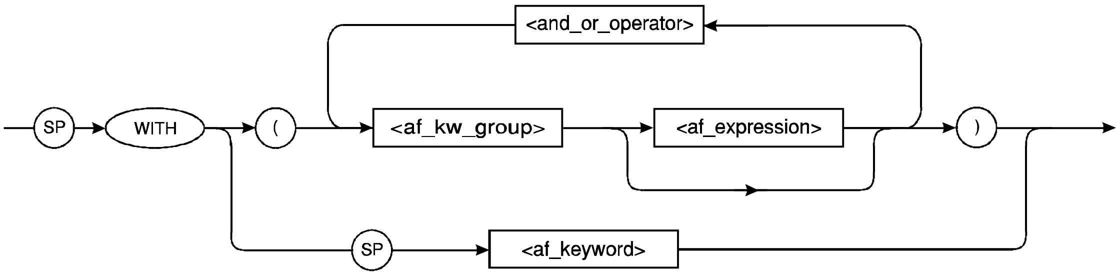
An instrument is classified using one or more base functionality keywords along with optional additional functionality keywords. This combination of keywords is an <instrument_specifier>. Some instruments implement the SYSTem:CAPability? query, whose response is its <instrument_specifier>.

1.4.1 Syntax

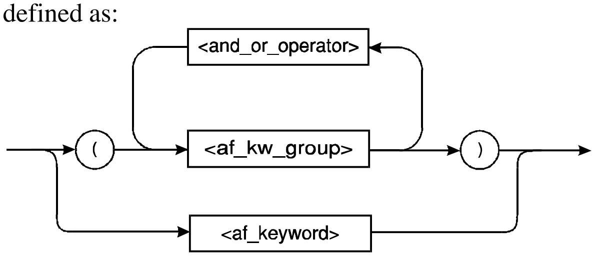
The syntax of <instrument_specifier> is defined as:



<bf_keyword> is a base functionality keyword as defined in one of the instrument chapters. <af_expression> is defined as:



<af_kw_group> is defined as:



<af_keyword> is an additional functionality keyword defined in the same instrument chapter which defines the associated <bf_keyword>.

Both <bf_keyword> and <af_keyword> use the mnemonic generation rules described in Syntax & Style, 6.2.1 when constructing the keywords. The keywords have no short form, but may have a numeric suffix. Only <bf_keyword> and <af_keyword> elements defined in this volume may appear in the response to SYSTem:CAPability?.

The <and_or_oper> is either ‘&’ (ASCII hexadecimal 26) or ‘|’ (ASCII hexadecimal 7C). The ‘&’ operator connects functionalities which may be active concurrently. The ‘|’ operator indicates that a choice of functionality must be made.

The operators used in the preceding syntax definitions follow the rules of the following the precedence rules in Table 1-1.

Operator	Associativity
()	left to right
WITH	right to left
&	left to right
|	left to right

1

Table 1-1
Precedence and Order of Evaluation
of <instrument_specifier> Operators

1.4.2 Examples

The simplest <instrument_specifier> describes an instrument with just one base functionality. A power supply with no additional functionality might have an <instrument_specifier> of:

```
(PSUPPLY)
```

A simple digital multi-meter might contain several base functionalites, so its <instrument_specifier> could be:

```
(DCVOLTMETER|ACVOLTMETER|OHMMETER)
```

The operator, |, indicates that only one of the base functionalities is available at a time.

A more sophisticated DMM could have an <instrument_specifier> like:

```
(DCVOLTMETER WITH(ETRIGGER&TERMINALS)|ACVOLTMETER
WITH(ETRIGGER&TERMINALS) |DCAMMETER
WITH(ETRIGGER&TERMINALS)|ACAMMETER WITH(ETRIGGER&TERMINALS)
|OHMMETER WITH(ETRIGGER&TERMINALS&OCOMPENSATED) |FOHMMETER
WITH(ETRIGGER&TERMINALS&OCOMPENSATED))
```

Within the <af_expression>, the operator, &, indicates all the additional functionalities are available simultaneously.

An instrument may contain multiple base functionalities that are used at the same time. The base functionalities may even be described in different chapters of this volume. A scanning voltmeter could be described with:

```
(DCVOLTMETER&SWITCHER WITH SCAN)
```

1.5 Compliance

An instrument that complies with SCPI even though it has less than base functionality for its instrument class is still a SCPI instrument, and may be claimed by its manufacturer as such. However, the manufacturer may not use <instrument_specifier> to describe its instrument unless it meets the requirements of Volume 4.

Any product which uses an <instrument_specifier> to describes its capabilities and whose response to SYSTem:VERSion? is greater than or equal to 1994.0 shall also implement SYSTem:CAPability?

Note that every instrument will likely have functionality not covered by this volume. The designer must still follow the requirements in other volumes when implementing SCPI for those functionalities.

1.6 Chapter Organization

Chapters are nominally organized into sections as follows, where x is the chapter number:

x.1	Base Functionality
x.1.1	Base Measurement Instructions (as applies)
x.1.2	Base Device-oriented Functions
x.1.3	Base Status Reporting
x.2	Additional Functionality (as applies)
x.3	Programming Examples
x.4	Instrument Class specific commands for CALibration
x.5	Instrument Class specific commands for DIAGnostics
x.6	MEMory subsystem additions

Some instrument classes naturally divide into major sub-classes. For example, meters fall into sub-classes of voltmeter, ammeter, ohmmeter, and others. In this situation, separate chapters or other variations on chapter organization may be used.

In some instances, it is desired to standardize the way an Instrument or Instrument Class uses commands in the CALibration and/or DIAGnostic subsystem beyond the commands that are included in Volume II of SCPI for all instruments. In these cases, Instrument Class specific commands can be defined in the Instrument Classes volume for SCPI, but only for CALibration and DIAGnostic commands. When this is the case, the commands will be listed as required commands in the Base Functionality or Extended functionality sections of the Instrument Class chapter, and then the commands will be defined in x.4 Instrument Class specific commands for CALibration and/or x.5 Instrument Class specific commands for DIAGnostics sections of Instrument Class chapter to which they apply.

2 Chassis Dynamometers

The Chassis Dynamometer is a basic sourcing instrument. It supplies a force to a vehicle to simulate actual road load driving conditions. The SOURce root node is optional.

All SCPI compliant products implement the commands listed in Syntax & Style section 4.2.

Figure 2-1 shows a simplified model for a chassis dynamometer. It is derived from Command Reference, Chapter 2. The shaded parts are not described here.

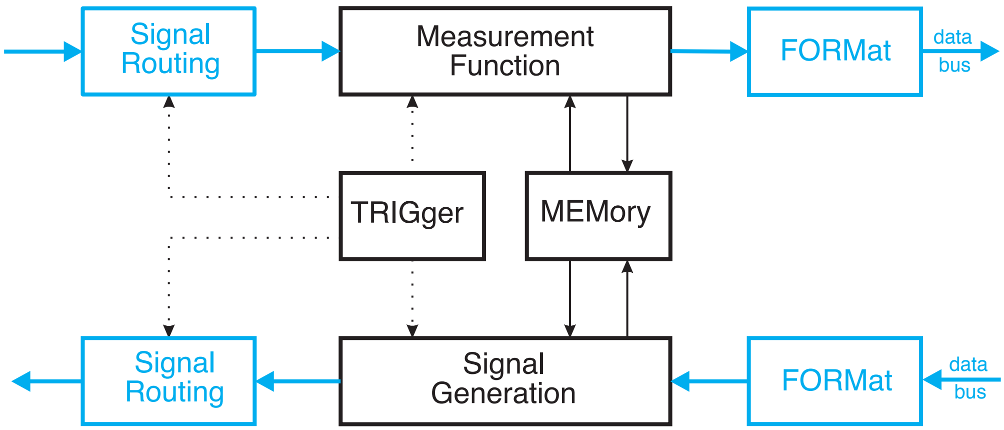


Figure 2-1 Simplified Model for a Chassis Dynamometer

Figure 2-2 shows how the "Measurement Function" block in Figure 2-1 is expanded in a chassis dynamometer to measure Force, Speed and Distance. The CALCulate block is dashed and its behavior is not described here.

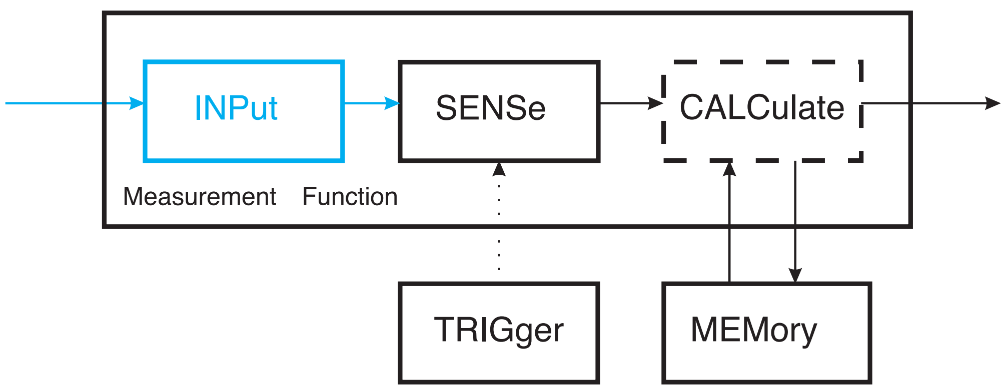


Figure 2-2 Expanded Measurement Function Model of a Chassis Dynamometer

Figure 2-3 shows how the "Signal Generation" block in Figure 2-1 is expanded in a chassis dynamometer to source Force, Speed and other parameters to the chassis dynamometer.

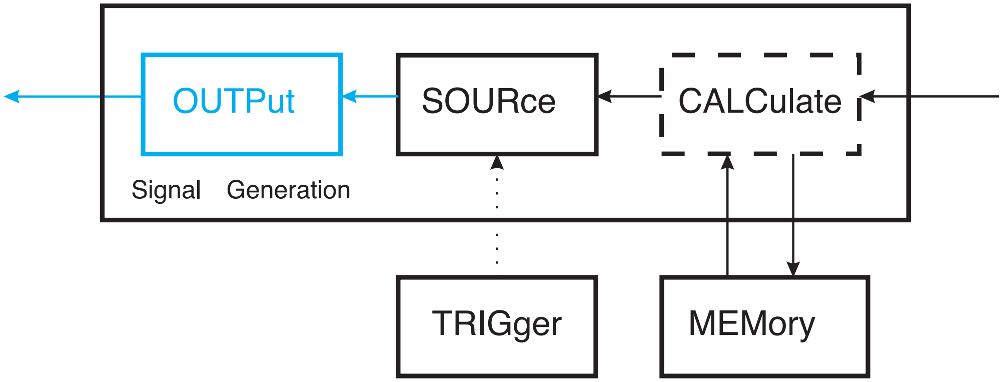


Figure 2-3 Expanded Signal Generation Model of a Chassis Dynamometer

Figure 2-4 shows an instrument model for a Chassis Dynamometer. It indicates how the "Measurement Function" and "Signal Generation" blocks of the standard SCPI instrument model, Figure 2-1 of Command Reference, are modified for the dynamometer. Additionally, it shows how this measurement system uses system, status, calibration and control functions to set-up for performance of functional tasks. Calculations are part of the functional tasks of the dynamometer and thus are shown as part of the internal function block as are dynamometer specific tasks, e.g., coast downs, road load simulations/derivations and warm-ups. SCPI commands or responses to these chassis dynamometer internal functions are communicated through the I/O block shown.

The dynamometer has eleven possible states.

1. Idle
2. Warmup
3. Force Sensor Zero
4. Drive Line Loss
5. Coast Down
6. Parasitic Losses
7. Base Inertia
8. Go to Speed
9. Go to Force
10. Road Load Determination
11. Road Load Simulation.

2.1 Base Functionality

This section describes the base functionality for Chassis Dynamometer.

The <bf_keyword> for a Chassis Dynamometer is CDYNO. In addition to the functions shown below, all Chassis Dynamometers for use in emissions test cells shall implement the

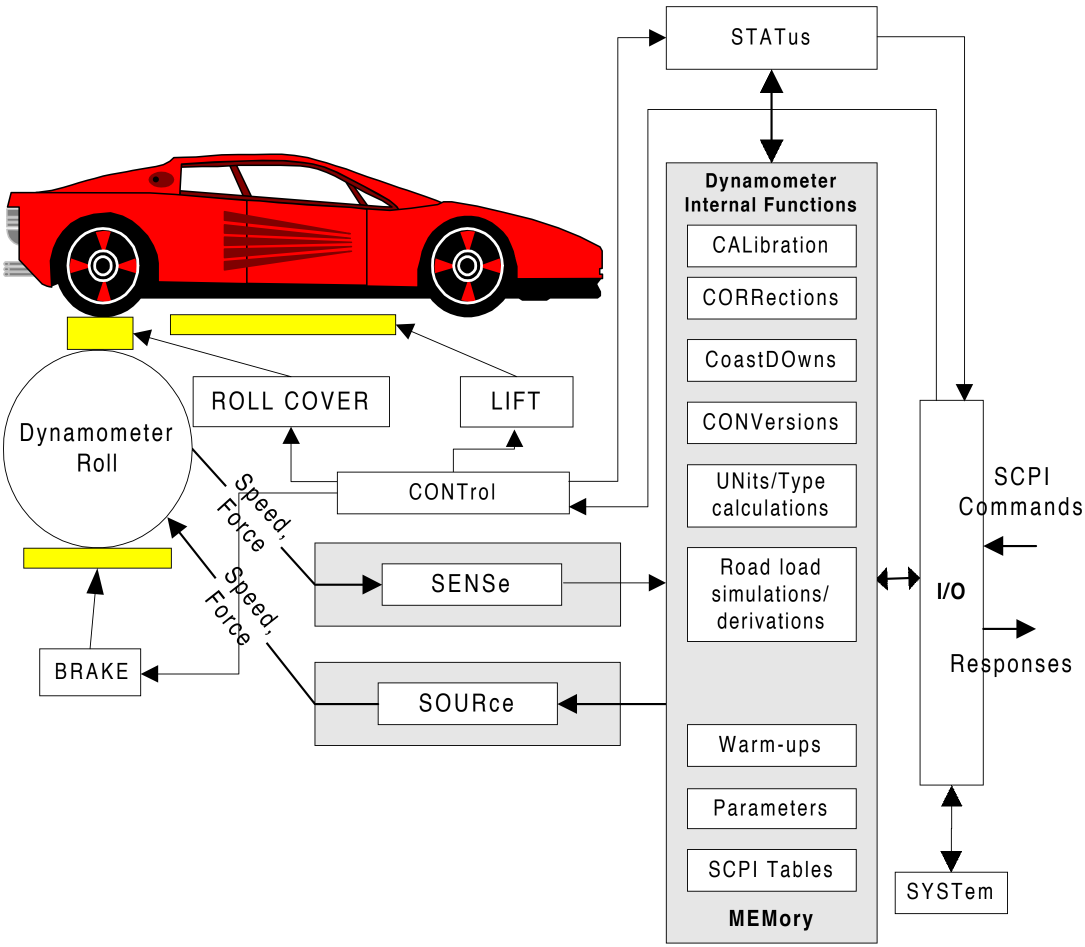


Figure 2-4 SCPI Chassis Dynamometer Model

functions shown in the base functionality for Emission Test Cells, <bf_keyword> ETCELL, shown in Chapter 6 of SCPI Instrument Classes.

2.1.1 Base Measurement Instructions

The base functionality for a chassis dynamometer contains no MEASurement instructions, (commands from the MEASurement subsystem), but does include SENSe commands, which set up and cause measurements such as speed and force to be made.

2.1.2 Base Device-oriented Functions

The base functionality for a chassis dynamometer include :SOURce, :SENSe, :MEMory, CONTrol, and :CALibration subsystems. It also defines device dependent STATus bits and instructions for chassis dynamometers. A SCPI-compliant chassis dynamometer shall implement all of the commands listed in this section.

2.1.2.1 CALibration subsystem

The Chassis Dynamometer CALibration subsystem shall provide for calibrating or verifying various components that affect a chassis dynamometer measurement capability by implementing the following commands. The INITiate command causes the calibration function to be run. The CALibration subsystem commands for a chassis dynamometer are as follows.

KEYWORD	PARAMETER FORM	SCPI Reference
:CALibration		Vol 2-5
:BINertia		Vol 4-2.4.1.1
| :AVERage?	<numeric_value>	Vol 4-2.4.1.1.1
| :HSPeed	<numeric_value>	Vol 4-2.4.1.1.2
| :INITiate		Vol 4-2.4.1.1.3
| :LSPeed	<numeric_value>	Vol 4-2.4.1.1.4
| :NRUNs	<numeric_value>	Vol 4-2.4.1.1.5
| :SDEViation?	<numeric_value>	Vol 4-2.4.1.1.6
| :UPDate		Vol 4-2.4.1.1.7
:PLOSs		Vol 4-2.4.1.2
| :APCoefficients	<numeric_value>,<numeric_value>,...	Vol 4-2.4.1.2.1
| :INITiate		Vol 4-2.4.1.2.2
| :LATime	<numeric_value>	Vol 4-2.4.1.2.3
| :STIMe	<numeric_value>	Vol 4-2.4.1.2.4
| :UPDate		Vol 4-2.4.1.2.5
:WARMup		Vol 4-2.4.1.3
| :INITiate		Vol 4-2.4.1.3.1
| :SPEed	<numeric_value>	Vol 4-2.4.1.3.2
| :TIMeout	<numeric_value>	Vol 4-2.4.1.3.3
:ZERO		Vol 4-2.4.1.4
| :FSENsor		Vol 4-2.4.1.4.1
| | :INITiate		Vol 4-2.4.1.4.1.1
| | :LEVel?		Vol 4-2.4.1.4.1.3
| | :LATime	<numeric_value>	Vol 4-2.4.1.4.1.2
| | :SPEed	<numeric_value>	Vol 4-2.4.1.4.1.4
| | :STIMe	<numeric_value>	Vol 4-2.4.1.4.1.5
| | :UPDate		Vol 4-2.4.1.4.1.6

2.1.2.2 CONTrol subsystem

The CONTrol subsystem commands for a chassis dynamometer are as follows.

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:CONTrol</td><td></td><td>Vol 2-6</td></tr>
<tr><td>:BRAKe</td><td></td><td>Vol 2-6.3</td></tr>
<tr><td>| [:STATe]</td><td><Boolean></td><td>Vol 2-6.3.1</td></tr>
<tr><td>:COVer</td><td></td><td>Vol 2-6.5</td></tr>
<tr><td>| [:ADJust]</td><td><OPEN|CLOSe|SCLOse|SOPEn></td><td>Vol 2-6.5.1</td></tr>
<tr><td>| :POSition?</td><td></td><td>Vol 2-6.5.2</td></tr>
<tr><td>:IDLE</td><td></td><td>Vol 2-6.7</td></tr>
<tr><td>| :INITiate</td><td></td><td>Vol 2-6.7.1</td></tr>
<tr><td>:LIFT</td><td></td><td>Vol 2-6.8</td></tr>
<tr><td>| [:ADJust]</td><td><UP|DOWN></td><td>Vol 2-6.8.1</td></tr>
<tr><td>| :POSition?</td><td></td><td>Vol 2-6.8.2</td></tr>
<tr><td>:MCONtrol</td><td></td><td>Vol 2-6.9</td></tr>
<tr><td>| [:STATe]</td><td><Boolean></td><td>Vol 2-6.9.1</td></tr>
<tr><td>:ROTation</td><td></td><td>Vol 2-6.10</td></tr>
<tr><td>| [:DIRection]</td><td><FORWard|REVerse></td><td>Vol 2-6.10.1</td></tr>
<tr><td>:VCDevice</td><td></td><td>Vol 2-6.11</td></tr>
<tr><td>| [:STATe]</td><td><Boolean></td><td>Vol 2-6.11.1</td></tr>
<tr><td>| :TDIameter</td><td><numeric_value></td><td>Vol 2-6.11.2</td></tr>
</table>

2.1.2.3 MEMory subsystem

The MEMory subsystem of the chassis dynamometer shall include the following commands:

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:MEMory</td><td></td><td>Vol 2-13</td></tr>
<tr><td>:CLEar</td><td></td><td>Vol 2-13.2</td></tr>
<tr><td>| [:NAME]</td><td><name></td><td>Vol 2-13.2.1</td></tr>
<tr><td>| :TABLe</td><td><name></td><td>Vol 2-13.2.2</td></tr>
<tr><td>:DATA?</td><td><data></td><td>Vol 2-13.4</td></tr>
<tr><td>:DELete</td><td></td><td>Vol 2-13.5</td></tr>
<tr><td>| :ALL</td><td><name></td><td>Vol 2-13.5.1</td></tr>
<tr><td>| [:NAME]</td><td><name></td><td>Vol 2-13.5.2</td></tr>
<tr><td>:TABLe</td><td></td><td>Vol 2-13.11</td></tr>
<tr><td>| :ARATe</td><td></td><td>Vol 4-2.4.3.1</td></tr>
<tr><td>| | [:MAGNitude]</td><td><numeric_value> {,<numeric_value>}</td><td>Vol 4-2.4.3.1.1</td></tr>
<tr><td>| | | :POINts?</td><td></td><td>Vol 4-2.4.3.1.1.1</td></tr>
<tr><td>| :CINertia</td><td></td><td>Vol 4-2.4.3.2</td></tr>
<tr><td>| | [:MAGNitude]</td><td><numeric_value> {,<numeric_value>}</td><td>Vol 4-2.4.3.2.1</td></tr>
<tr><td>| | | :POINts?</td><td></td><td>Vol 4-2.4.3.2.1.1</td></tr>
<tr><td>| :DEFine</td><td><structure_string>[,<numeric_value>]</td><td>Vol 2-13.11.11</td></tr>
<tr><td>| :DRATe</td><td></td><td>Vol 4-2.4.3.3</td></tr>
<tr><td>| | [:MAGNitude]</td><td><numeric_value> {,<numeric_value>}</td><td>Vol 4-2.4.3.3.1</td></tr>
<tr><td>| | | :POINts?</td><td></td><td>Vol 4-2.4.3.3.1.1</td></tr>
</table>

2

KEYWORD	PARAMETER FORM	SCPI Reference
	:FORCe		Vol 2-13.11.12
		:AACCeleration		Vol 4-2.4.3.4.1
			[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.4.1.1
				:POINts?		Vol 4-2.4.3.4.1.1.1
		:ADECeleration		Vol 4-2.4.3.4.2
			[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.4.2.1
				:POINts?		Vol 4-2.4.3.4.2.1.1
		[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 2-13.11.12.1
			:POINts?		Vol 2-13.11.12.1.1
		:ZOFFset		Vol 4-2.4.3.4.4
			[:MAGNitude]	<numeric_value>	Vol 4-2.4.3.4.4.1
	:DLOSs		Vol 4-2.4.3.5
		[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.5.1
			:POINts?		Vol 4-2.4.3.5.1.1
	:PCOefficients<n>		Vol 4-2.4.3.6
		[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.6.1
			:POINts?		Vol 4-2.4.3.6.1.1
	:SELect	<table_name>	Vol 2-13.11.22
	:SPEed		Vol 2-13.11.23
		[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 2-13.11.23.1
			:POINts?		Vol 2-13.11.23.1.1
		:STARt		Vol 4-2.4.3.7.2
			[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.7.2.1
				:POINts?		Vol 4-2.4.3.7.2.1.1
		:STOP		Vol 4-2.4.3.7.3
			[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.7.3.1
				:POINts?		Vol 4-2.4.3.7.3.1.1
	:TIME		Vol 2-13.11.24
		:ACCeleration		Vol 4-2.4.3.8.1
			[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.8.1.1
				:POINts?		Vol 4-2.4.3.8.1.1.1
		:DECeleration		Vol 4-2.4.3.8.2
			[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 4-2.4.3.8.2.1
				:POINts?		Vol 4-2.4.3.8.2.1.1
		[:MAGNitude]	<numeric_value> {,<numeric_value>}	Vol 2-13.11.24.1
			:POINts?		Vol 2-13.11.24.1.1

The Chassis Dynamometer shall have the following tables.

Table Name	Elements	Description
CDownSpeed	SPEed:STARt, SPEed:STOP	Coastdown speed interval definitions
CDownResult	TIME, FORCe, POWer	Results of the coastdown runs. Each coastdown has ‘n’ intervals as defined in the CDownSpeed table
CDownMCoefficients	PCO0, PCO1, PCO2, PCO3	Coastdown measured Polynomial Coefficients for a coastdown or road load derivation procedure
CDownDCoefficients	PCO0, PCO1, PCO2, PCO3	Coastdown calculated dynamometer setting coefficients during coastdown or road load derivation
ParasiticLOSs	SPEed, DLOSs	parasitic loss computation results
PCOefficients	PCO0, PCO1, PCO2, PCO3	Parasitic data curve fit results from the Parasitic Loss procedure
DriveLineLoss	SPEed, DLOSs	Steady state drive line loss measurement
DriveLineCoefficients	PCO0, PCO1, PCO2, PCO3	Results of a curve fit of the DriveLineLoss table data, after the Drive Line Loss procedure
BaseINertia	CINertia, TIME:ACCel, TIME:DECel, ARATe, DRATe, FORCe:AACCel, FORCe:ADECel	base inertia procedure data
FsensorZero	ZOFFset	Single value table whose value represents the zero offset used to compensate for shift. CALibration:ZERO:FSENsor :LEVel value will replace this when CAL:ZERO:FSENsor:UPDate is sent.

For definition of the new :MEMory:TABLe Commands defined above, see Section 2.4.3 in this chapter of Volume 4, Instrument Classes.

2.1.2.4 SENSe subsystem

The SENSe subsystem of a chassis dynamometer shall provide access to function values by implementing the following commands. The <sensor_function> options shall include: FORCe|TPLoss|SPEed[:REAR]|SPEed:FRONt|FERRor|ACCeleration|DISTance|| DISTance:RESet|TIME

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:SENSe</td><td></td><td>Vol 2-18</td></tr>
<tr><td>:DATA?</td><td>[<data_handle>]</td><td>Vol 2-18.13.1</td></tr>
<tr><td>:DISTance</td><td></td><td>Vol 2-18.9</td></tr>
<tr><td>| :RESet</td><td></td><td>Vol 2-18.9.1</td></tr>
<tr><td>:FUNCtion</td><td></td><td>Vol 2-18.13.2</td></tr>
<tr><td>| :CONCurrent</td><td><Boolean></td><td>Vol 2-18.13.2.1</td></tr>
<tr><td>| :OFF</td><td><sensor_function>{,<sensor_function>}</td><td>Vol 2-18.13.2.2</td></tr>
<tr><td>| |</td><td>where the sensor functions to be supported shall include:</td><td></td></tr>
<tr><td>| |</td><td>FORCe|TPLoss|SPEed[:REAR]|SPEed:FRONt</td><td></td></tr>
<tr><td>| |</td><td>|FERRor|ACCeleration|DISTance</td><td></td></tr>
<tr><td>| |</td><td>|DISTance:RESet|TIME</td><td></td></tr>
<tr><td>| | :ALL</td><td></td><td>Vol 2-18.13.2.2.1</td></tr>
<tr><td>| | :COUNt?</td><td></td><td>Vol 2-18.13.2.2.2</td></tr>
<tr><td>| [:ON]</td><td><sensor_function>{,<sensor_function>}</td><td>Vol 2-18.13.2.3</td></tr>
<tr><td>| |</td><td>where the sensor functions supported</td><td></td></tr>
<tr><td>| |</td><td>are same as :OFF</td><td></td></tr>
<tr><td>| | :ALL</td><td></td><td>Vol 2-18.13.2.3.1</td></tr>
<tr><td>| | :COUNt?</td><td></td><td>Vol 2-18.13.2.3.2</td></tr>
<tr><td>| :STATe?</td><td><sensor_function></td><td>Vol 2-18.13.2.4</td></tr>
</table>

2.1.2.5 SOURce subsystem

The SOURce subsystem of the chassis dynamometer shall provide the following commands

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:SOURce</td><td></td><td>Vol 2-19</td></tr>
<tr><td>:ACCeleration</td><td></td><td>Vol 2-19.1</td></tr>
<tr><td>| [:LEVel]</td><td><numeric_value></td><td>Vol 2-19.1.1</td></tr>
<tr><td>:FORCe</td><td></td><td>Vol 2-19.8</td></tr>
<tr><td>| :CDOWn</td><td></td><td>Vol 2-19.8.1</td></tr>
<tr><td>| | :INITiate</td><td></td><td>Vol 2-19.8.1.1</td></tr>
<tr><td>| | :NRUNs</td><td><numeric_value></td><td>Vol 2-19.8.1.3</td></tr>
<tr><td>| | :RLDerivation</td><td></td><td>Vol 2-19.8.1.4</td></tr>
<tr><td>| | | :FACCeptance</td><td><numeric_value></td><td>Vol 2-19.8.1.4.1</td></tr>
<tr><td>| | | :INITiate</td><td></td><td>Vol 2-19.8.1.4.2</td></tr>
<tr><td>| | | :RMAXimum</td><td><numeric_value></td><td>Vol 2-19.8.1.4.3</td></tr>
<tr><td>| | | :RVERify</td><td><numeric_value></td><td>Vol 2-19.8.1.4.4</td></tr>
<tr><td>| | :SOFFset</td><td><numeric_value></td><td>Vol 2-19.8.1.2</td></tr>
<tr><td>| :CONFigure</td><td></td><td>Vol 2-19.8.2</td></tr>
</table>

KEYWORD	PARAMETER FORM	SCPI Reference
| | :ABRake		Vol 2-19.8.2.1
| | | :GAIN	<numeric_value>	Vol 2-19.8.2.1.1
| | | [:STATe]	<Boolean>	Vol 2-19.8.2.1.2
| | | :THReshold	<numeric_value>	Vol 2-19.8.2.1.3
| | :GRADe		Vol 2-19.8.2.2
| | | :LEVel	<numeric_value>	Vol 2-19.8.2.2.1
| | | [:STATe]	<Boolean>	Vol 2-19.8.2.2.3
| | | :SOURce	INTernal | EXTernal	Vol 2-19.8.2.2.2
| | [:VEHicle]		Vol 2-19.8.2.3
| | | :DCOefficient	<numeric_value_list>,<numeric_value>	Vol 2-19.8.2.3.1
| | | :DINertia	<numeric_value>	Vol 2-19.8.2.3.2
| | | [:STATe]	<Boolean>	Vol 2-19.8.2.3.3
| | | :TCOefficient	<numeric_value_list>,<numeric_value>	Vol 2-19.8.2.3.4
| | | :TINertia	<numeric_value>	Vol 2-19.8.2.3.5
| | | :WEIGht	<numeric_value>	Vol 2-19.8.2.3.6
| :INITiate		Vol 2-19.8.3
| [:LEVel]	<numeric_value>	Vol 2-19.8.4
| :RLSimulation		Vol 2-19.8.5
| | :INITiate		Vol 2-19.8.5.1
:SPEed		Vol 2-19.20
| :INITiate		Vol 2-19.20.1
| [:LEVel]	<numeric_value>	Vol 2-19.20.2
| :SSDLoss		Vol 2-19.20.3
| | :INITiate		Vol 2-19.20.3.1
| | :LATime	<numeric_value>	Vol 2-19.20.3.2
| | :STIMe	<numeric_value>	Vol 2-19.20.3.3

2

2.1.3 STATus Subsystem

Chassis Dynamometer shall implement the status reporting structure described in SCPI Syntax & Style Volume 1, Chapter 9, Status Reporting. The SCPI Command Reference Volume 2, Chapter 20, STATus Subsystem defines the commands that shall be used to control the status reporting structure.

KEYWORD	PARAMETER FORM	SCPI Reference
:STATus		Vol 2-20
:OPERation		Vol 2-20.1
| :BIT<n>	<n=8 -12>	Vol 2-20.1.1
| | :CONDition?		Vol 2-20.1.2
| | :ENABle	<NRf>|<non-decimal numeric>	Vol 2-20.1.3
| | [:EVENt]?		Vol 2-20.1.4
| | :NTRansition	<NRf>|<non-decimal numeric>	Vol 2-20.1.6
| | :PTRansition	<NRf>|<non-decimal numeric>	Vol 2-20.1.7
| :CONDition?		Vol 2-20.1.2
:QUEStionable?		Vol 2-20.3
| :BIT<n>	<n=9 -12>	Vol 2-20.3.1
| | :CONDition?		Vol 2-20.3.2
| | :ENABle	<NRf>|<non-decimal numeric>	Vol 2-20.3.3
| | [:EVENt]?		Vol 2-20.3.4
| | :NTRansition		Vol 2-20.3.6
| | :PTRansition		Vol 2-20.3.7
| :CONDition?		Vol 2-20.3.2
| [:EVENt]		Vol 2-20.3.4

2.1.3.1 OPERation

:STATus:OPERation

For a chassis dynamometer, the bit of interest in the OPERation status structure is the CDYNO summary bit. While the chassis dynamometer is performing its function, bit 9 (CDYNO summary) shall be used for the chassis dynamometer status. The fanned out register is shown in Figure 2-5.

2.1.3.1.1 CDYNo:CONDition?<16 bit integer word>

:STATus:OPERation:CDYNo:CONDition?

The Dynamometer Program Status Register contains 16 bits that indicate what operation is being performed on the dynamometer. The most significant bit (bit 15) is always 0. The dynamometer will be in one of the following processes at all times. All processes are mutually exclusive and are overlapped, (i.e. with overlapped processes, a command may be sent while a process is executing) but will not be executed until that procedure is completed.

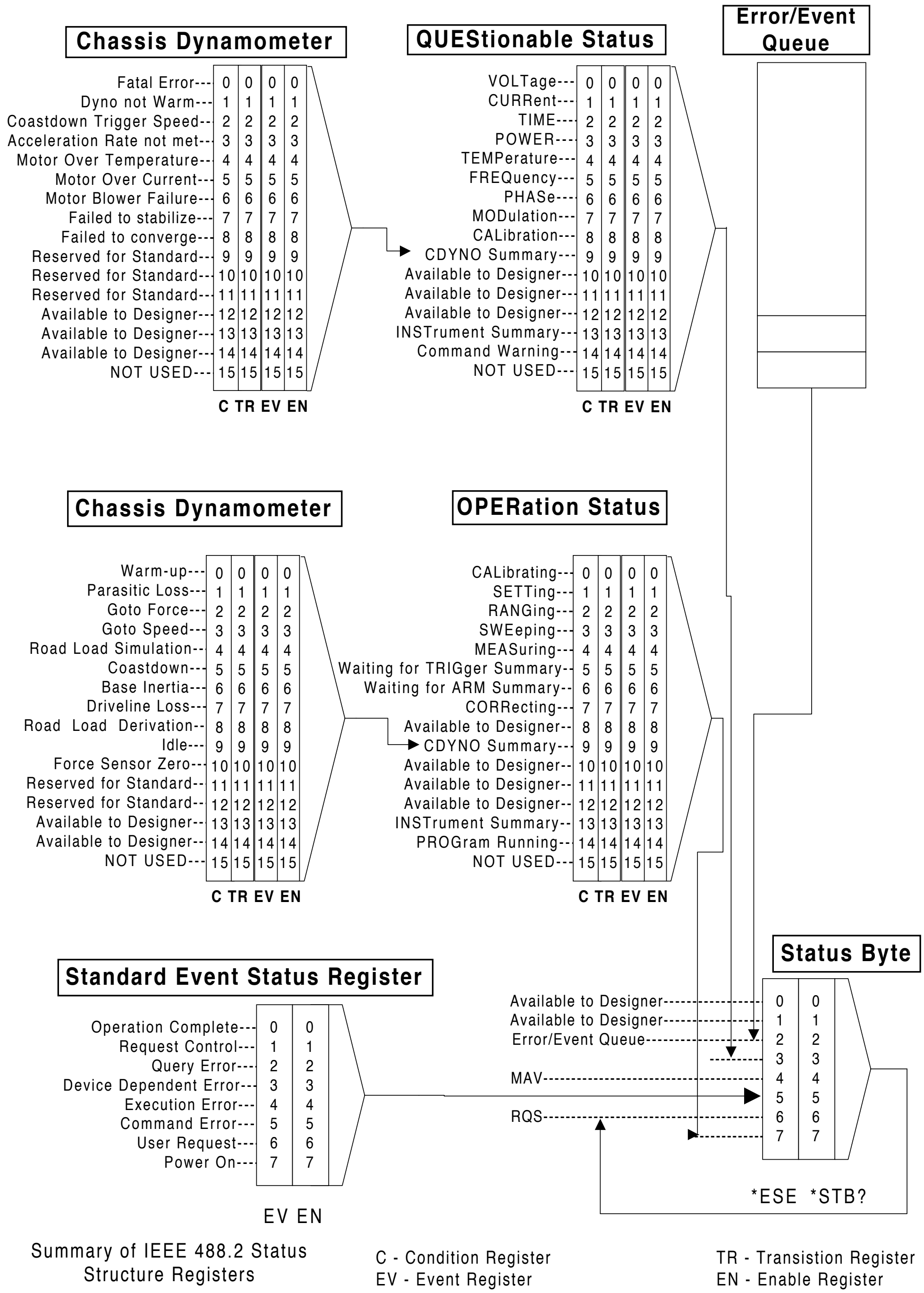


Figure 2-5 SCPI Status Registers

CDYNO OPERation CONDition Status Register

Bit #	Event	
0	Warm Up	Set if Dynamometer is in Warm Up Process
1	Parasitic Loss	Set if Dynamometer is in Parasitic Loss Process.
2	Go To Force	Set if Dynamometer is in Go To Force Process.
3	Go To Speed	Set if Dynamometer is in Go To Speed Process.
4	Road Load Simulation	Set if Dynamometer is in Road Load Simulation Process.
5	Coast Down	Set if Dynamometer is in Coast Down Process.
6	Base Inertia	Set if Dynamometer is in Base Inertia Process.
7	Driveline Loss	Set if Dynamometer is in Driveline Loss Process.
8	Road Load Derivation	Set if Dynamometer is in Road Load Derivation Process.
9	Idle	Set if Dynamometer is in Idle state.
10	Force Sensor Zero	Set if Dynamometer is in Force Sensor Zero state
11	Reserved for standard	
12	Reserved for standard	
13	Available to Designer	
14	Available to Designer	
15	Not Used	Will always read as 0

2.1.3.2 QUEStionable

:STATus:QUEStionable

For the chassis dynamometer, the bit of interest in the QUEStionable status structure is the CDYNO summary bit. When the chassis dynamometer suspects the dynamometer specification is violated, bit 9 (CDYNO summary) shall be set. The fanned out register is as follows:

2.1.3.2.1 CDYNo:CONDition?<16 bit integer word>

:STATus:QUEStionable:CDYNo:CONDition?

The Dynamometer Questionable Status Register contains 16 bits that indicate whether the data being acquired is of questionable quality. The most significant bit (bit 15) is always 0.

The dynamometer updates this register immediately when the dynamometer encounters an error that may generate questionable data. Bits 0-9 are the minimum subset of errors that may generate faulty data. Bits 10-14 are available for a dynamometer manufacturer for other questionable data or for other summary bits.

CDYNO QUEStionable CONDition Status Register

Bit #	Questionable Event	
0	Fatal Error	Set when the dynamometer experiences a fatal error that causes the motor contactor to drop out, such as Torque Safety Limit, Speed Safety Limit or other dynamometer specific fatal error. An appropriate message must be placed on the Error/Event message queue when set.
1	Dynamometer not Warm	This bit is set if the dynamometer is not warm or has timed out after being warm but not used within the TIMeout period.
2	Coast down Trigger Speed	For a coast down, the dynamometer determines an acceptable trigger speed based on coast down inputs. If the dynamometer does not agree with the set trigger speed, this bit shall be set.
3	Acceleration Rate not met	The dynamometer will set this bit if the requested acceleration rate for a Process is out of tolerance with the actual acceleration. The tolerance is based on the physical characteristic of the dynamometer.
4	Motor Over Temperature	For dynamometers that monitor motor temperature
5	Motor Over Current	For dynamometers that monitor motor current
6	Motor Blower Failure	For dynamometers that monitor the blower
7	Failed To Stabilize	The dynamometer failed to stabilize
8	Failed to Converge	The dynamometer failed to converge.
9	Reserved for Standard	
10	Reserved for Standard	
11	Reserved for Standard	
12	Available to Designer	
13	Available to Designer	

14	Available to Designer	
15	Not Used	Will always read as 0

2.2 Additional Functionality

Not applicable.

2.3 Programming Examples

2.3.1 Coastdown

Note that this example does not start with *RST or *CLS. These commands would not be appropriate when carryover settings or status are important.

Brake off	:CONTrol:BRAKe OFF	
Contactor On -	:CONTrol:MCONtrol ON	
See if Dyno is warm	STATus:QUEStionable:CDYNo :CONDition?	Controller reads response, looks at bit 1
Let Dyno know if vehicle is on or off	:SOURce:FORCe:CONFigure :VEHicle ON	OFF or ON for vehicle
Define table of speed points for input into coastdown	:MEMory:TABLe:SELect CdownSpeed	This table represents the coastdown speed interval that data will be measured and collected.
	:MEMory:TABLe:DEFine "SPEed:STARt,SPEed:STOP",5	For this example, 5 elements have been allocated (30 maximum).
	:MEMory:TABLe:SPEed :STARt 24.6,20.2,15.7,11.2,6.7	Meters/sec values corresponding to 55, 45, 35, 25, 15 MPH start, and 45, 35, 25, 15, 5 MPH stop speeds
	:MEMory:TABLe:SPEed :STOP 20.2,15.7,11.2,6.7,2.2	
Define the number of coastdown runs	:SOURce:FORce:CDOWn :NRUNs 5	Repeat coastdown 5 times
Define table for coastdown outputs	:MEMory:TABLe:SELect CDownResult	

	:MEMory:TABLe:DEFine "TIME, FORCe, POWer", 6	Theoretical and measured values from coastdown. The table is NRUNs + 1 big.
Set Inertia	:SOURce:FORCe:CONFigure :VEHicle:DINERtia 1350	Dyno Inertia setting in kilograms
Set Road load coefficients	:SOURce:FORCe:CONFigure :VEHicle:DCOefficient 26.7, -0.98, 0.65	Numeric list of polynomial coefficients. These corresponds to 6, -0.1 and 0.3 in english units (pounds force (lbf) and miles per hour).
	:SOURce:FORCe:CONFigure :VEHicle:TCOefficient 26.7, -0.98, 0.65	Target settings
Set coastdown input parameters	:SOURce:ACCelerate 1	How fast to accelerate the dyno to top coastdown point (meters/sec^2)
	:SOURce:FORCe:CONFigure :VEHicle:TINertia 1350	Target inertia.
	:SOURce:FORCe:CDOWn :SOFFset 2.2	Speed above top set point (meters/sec)
Setup to sense data	:SENSe:FUNCtion:CONCurrent ON :SENSe:FUNCtion:ON "FORCe", "SPEed", "ACCeleration", "DISTance"	Allow multiple sensor functions to be sensed. Turn on desired sensor functions.
Setup trigger system for sensing data	:TRIGger:SOURce TIMer :TRIGger:TIMer 0.100	Select internal timer on dynamometer Take samples every 100 ms, samples are averaged over 100 ms
Initiate Coastdown	:SOURce:FORCe:CDOWn :INITiate	Start coastdown (this command clears output)
Coastdown Complete?	:STATus:OPERation:CDYNo :CONDition?	Wait until bit 5 is false indicating coastdown complete.
Checkout completion	:STATus:QUEStionable :CONDition?	If bit 9 is true then :STATus:QUEStionable :CDYNo:CONDition? can be used to check for specific errors.

Iteratively review realtime data	:INITiate:CONTinous ON:SENSe:DATA?	Start acquiring data Returns FORCe, SPEed, ACCeleration, and DISTance.
Retrieve Coastdown results	:MEMory:DATa? CDownResult	Retrieve Theoretical and Measured data
	:MEMory:DATa? CDownMCoefficients	For definition of CDownMCoefficient see SAE J2264. This is a curve fit of the data
Idle the dynamometer	:CONTrol:IDLE:STATe ON	

2.3.2 Road Load Simulation

Note that this example does not start with *RST or *CLS. These commands would not be appropriate when carryover settings or status are important.

Brake off	:CONTrol:BRAKe OFF	
Contactor On	:CONTrol:MCONtrol ON	
See if dyno is warm	STATus:QUEStionable:DYNO :CONDition?	Controller reads response, looks at bit 1
Set Inertia	:SOURce:FORCe:CONFiure :VEHicle:DINERtia 1350	
Set Road load coefficients	:SOURce:FORCe:CONFigure :VEHicle:RLOad:DCOefficient 26.7, -0.98, 0.65	Numeric list of polynomial coefficients. These corresponds to 6, -0.1 and 0.3 in english units (pounds force (lbf) and miles per hour).
Set Vehicle Weight	:SOURce:FORCe:CONFigure :VEHicle:WEIGht 1300	Vehicle weight in kilograms
Set the augmented braking	:SOURce:FORCe:CONFigure :ABRake ON	Augmented braking
Set the gain for augmented braking	:SOURce:FORCe:CONFigure :ABRake:GAIN 0.00	
Set the threshold for augmented braking	:SOURce:FORCe:CONFigure :ABRake:THReshold 500	Sets the threshold to 500 Newtons

Set up the grade simulation	:SOURce:FORCe:CONFigure :GRADe ON	
	:SOURce:FORCe:CONFigure :GRADe:LEVel 5	%in terms in grade - 100% is equal to sin(45deg)
Setup to sense data	:SENSe:FUNCtion:CONCurrent ON	Allow multiple sensor functions to be sensed.
	:SENSe:FUNCtion:ON “FORCe”, “SPEed”, “ACCeleration”, “DISTance”, “FERRor”	Turn on desired sensor functions.
Setup trigger system for sensing data	:TRIGger:SOURce TIMer :TRIGger:TIMer 0.100	Select internal timer on dynamometer. Take samples every 100 ms, samples are averaged over 100 ms
Initiate Road Load Simulation	:SOURce:FORCe:RLSimulate :INITiate	Reset the distance to zero.
RLS Complete?	:STATus:OPERation:CDYNo :CONDition?	Make sure bit 4 is on indicating dyno is in simulation mode.
Check for errors	:STATus:QUEStionable :CONDition?	If bit 9 is true then :STATus:QUEStionable:CDYNo:CONDition? can be used to check for specific errors.
Iteratively review realtime data	:INITiate:CONTinous ON :SENSe:DATA?	Start acquiring data. Returns FORCe, SPEed, ACCeleration, DISTance and FERRor.
Brake on Brake off	:CONTrol:BRAKe ON :CONTrol:BRAKe OFF	Optional dynamic control of the mechanical brake. Used in conjunction with an automatic driver to hold zero speed during idle periods.
Optionally reset the distance	:SENSe:DISTance:RESet	
Idle the Dynamometer	:CONTrol:IDLE:STATe ON	If idle is complete i.e., dyno at zero speed the Operation complete flag is true (same as above wrt. Checkout completion)

2.4 New Commands for this Instrument Class

2.4.1 Dynamometer CALibration Subsystem

The Dynamometer CALibration subsystem contains the commands needed for calibrating or verifying various components that affect a dynamometers' measurement capabilities. This includes load sensor zero, parasitic loss determination, base inertia determination, and dynamometer warm-up.

```
KEYWORD              PARAMETER FORM                      NOTES
:CALibration
  :BINertia
  |  :AVERage?       <numeric_value>                     [query only]
  |  :HSPeed         <numeric_value>
  |  :INITiate                                           [event; no query]
  |  :LSPeed         <numeric_value>
  |  :NRUNs          <numeric_value>
  |  :SDEviation?    <numeric_value>                     [query only]
  |  :UPDate                                             [event; no query]
  :PLOSs
  |  :APCoefficients <numeric_value>,<numeric_value>,...
  |  :INITiate                                           [event; no query]
  |  :LATime         <numeric_value>
  |  :STIMe          <numeric_value>
  |  :UPDate                                             [event; no query]
  :WARMup
  |  :INITiate                                           [event; no query]
  |  :SPEed          <numeric_value>
  |  :TIMeout        <numeric_value>
  :ZERO
  |  :FSENsor
  |  |  :INITiate                                        [event; no query]
  |  |  :LATime      <numeric_value>
  |  |  :LEVel?                                          [query only]
  |  |  :SPEed       <numeric_value>
  |  |  :STIMe       <numeric_value>
  |  |  :UPDate                                          [event; no query]
```

2.4.1.1 :BINertia

CALibration:BINertia

Base Inertia

For chassis dynamometers, this node describes the procedure to determine the base mechanical inertia of the dynamometer roll system (all rotating mechanical components outside of the control loop). This procedure consists of an acceleration and deceleration of the dynamometer rolls, from which a mechanical inertia calculation is made. The dynamometer is motored from an initial speed to a final speed at a given constant acceleration, and this interval is timed. Immediately, the dynamometer is motored back from the final speed to the initial speed at the same constant acceleration (inverted), and this interval is timed. For this pair of intervals, the force transducer output is continuously measured, and the average interval force calculated. The average acceleration is calculated using the interval initial and final speed values, and the interval time. This data is used to calculate the mechanical inertia (Inertia=Force/acceleration). The results are stored in the BaseINertia predefined table.

2.4.1.1.1 :AVERage?

CALibration:BINertia:AVERage?

This query returns the running average of all base inertia values acquired since the last CALibration:BINertia:INITiate was invoked.

2.4.1.1.2 :HSPeed <numeric_value>

CALibration:BINertia:HSPeed

High Speed

The higher magnitude speed (m/s) of the speed interval.

At *RST, this value is set to 16.

2.4.1.1.3 :INITiate

CALibration:BINertia:INITiate

Begin motoring the dynamometer through the ACCeleration/deceleration pairs, using :SOURce:ACCeleration as the acceleration/deceleration rate, from LSPeed to HSPeed and back, NRUNs times. All query commands must be accepted and responded to while this command is executing. This command describes an event and therefore has no associated *RST condition.

When this command is initiated for dynamometers, the dynamometer must:

1. Check critical ancillaries, if any, for proper conditions: Brakes Off, Motor Contactor On, parameters within dynamometer safety limits. For conditions that prohibit operation of this procedure, issue appropriate warning messages, and execute the Idle Procedure.

2. Set the DYNO operation condition register Base Inertia Procedure bit indicating the Base Inertia is executing. Note: The DYNO operation condition register will be further defined in Volume 4.

3. Accelerate/decelerate the dynamometer through the speed intervals, recording time, average acceleration, average force for the specified number of runs. The data is placed in the BaseInertia table after each acceleration/deceleration is run. Update the :CALibration:BINertia:SDEViation value based on this run. End the procedure when the specified runs are complete.

4. Clear the DYNO operation condition register Base Insrtia Procedure bit indicating the Base Inertia is not executing.

5. Issue the process complete message.

6. Execute the Idle Procedure.

7. Update the on-line mechanical inertia value, if desired.

2.4.1.1.4 :LSPeed <numeric_value>

CALibration:BINertia:LSPeed

Low Speed

The lower magnitude speed (m/s) of the speed interval.

At *RST, this value is set to 6.

2.4.1.1.5 :NRUNs <numeric_value>

CALibration:BINertia:NRUNs

Number of Runs - NRUNs

The number of ACCeleration/deceleration pairs to perform. NRUNs will equal the number of base inertia values acquired.

At *RST, this value is set to 10.

2.4.1.1.6 :SDEViation?

CALibration:BINertia:SDEViation?

Standard Deviation

This query returns the running standard deviation of all base inertia values acquired since the last CALibration:BINertia:INITiate was invoked.

2.4.1.1.7 :UPDate

CALibration:BINertia:UPDate

Place on-line the last determined average base mechanical inertia value. This command describes an event and therefore has no associated *RST condition.

2.4.1.2 :PLOSs

:CALibration:PLOSs

Parasitic Loss

For chassis dynamometers this node describes the commands associated with the parasitic loss procedure. This procedure measures and, if requested, updates the parasitic loss corrections for the dynamometer. The parasitic losses are measured by operating the dynamometer at various selected speeds and measuring the force required to maintain the constant speed. This procedure uses the :MEMory table 'ParasiticLOSs' that defines the speed points at which parasitic loss measurements are taken. The parasitic loss coefficients are calculated at the end of the procedure and reside in the memory table PCOefficient.

2.4.1.2.1 :APCoeff <numeric_value >,numeric_value>,<numeric_value>...

:CALibration:PLOSs:APCoeff

Active Parasitic Coefficients - APCoefficients

This is the presently active parasitic loss curve used by the dynamometer. The dynamometer will maintain coefficients for forward and reverse operations. The appropriate values must be placed in the APCoeff memory, based on roll rotation. The Active Parasitic Coefficients set of retrieved will be based on the currently selected direction.

2.4.1.2.2 :INITiate

:CALibration:PLOSs:INITiate

This command starts the parasitic loss procedure. This is an overlapped command. This command describes an event and therefore has no associated *RST condition.

When this command is initiated for dynamometers, the dynamometer must:

1. Check critical ancillaries, if any, for proper conditions: Brakes Off, Motor Contactor On, parameters within dynamometer safety limits. For conditions that prohibit operation of this procedure, issue appropriate warning messages, and execute the Idle Procedure.

2. Set the DYNO operation condition register Parasitic Loss Procedure bit indicating the Parasitic Loss procedure is executing. Note: The DYNO operation condition register will be further defined in Volume 4.

3. Accelerate dynamometer to the first speed point defined in the ParasiticLOSs memory table. Stabilize at speed point for defined stabilization time.

4. Average the force at the roll surface required to maintain the speed set point over the latency time. The average force value is placed in the ParasiticLOSs table for the specified speed point. Continue until all speed points have been executed in the ParasiticLOSs table. The first zero speed in the ParasiticLOSs table ends the procedure.

5. Perform a curve fit on the ParasiticLOSs table data and place the coefficients in the memory table PCoeff.

6. Clear the DYNO operation condition register Parasitic Loss Procedure bit indicating the parasitic loss procedure is not executing.

7. Issue the process complete message.

8. Execute the Idle Procedure.

2.4.1.2.3 :LATime

:CALibration:PLOSs:LATime

Loss Averaging Time - sec

The length of time in seconds to average the losses at a speed point.

At *RST, this value is set to 5.

2.4.1.2.4 :STIMe <numeric_value>

CALibration:PLOSs:STIMe

Stabilization Time - sec

Time for stabilization prior to LATime.

At *RST, this value is set to 5.

2.4.1.2.5 :UPDate

:CALibration:PLOSs:UPDate

Update the dynamometer actual current parasitic loss coefficients (APCoefficients node) with the coefficients contained in memory table PCOefficients.

This command is an event and has no associated *RST condition.

2.4.1.3 :WARMup

CALibration:WARMup

This node controls parameters relating to the warmup of the device.

For chassis dynamometers, this node describes the procedure to set up the conditions required to ready the dynamometer for testing. The warm-up consists of running the dynamometer at a fixed speed over a period of time. Warm-up completion is defined by either running the dynamometer for a fixed period of time, or by monitoring parasitic losses and comparing them to a target value tolerance.

2.4.1.3.1 :INITiate

CALibration:WARMup:INITiate

Start the warmup procedure.

For chassis dynamometers, motor the dynamometer roll to SPEed, and hold this speed for a TIMeout period of time. All query commands must be accepted and responded to while this command is executing. This command describes an event and therefore has no associated *RST condition.

When this command is initiated for dynamometers, the dynamometer must:

1. Check critical ancillaries, if any, for proper conditions: Brakes Off, Motor Contactor On, parameters within dynamometer safety limits. For conditions that prohibit operation of this procedure, issue appropriate warning messages, and execute the Idle Procedure.

2. Set the DYNO operation condition register Warm up Procedure bit indicating the Warm-up is executing. Note: The DYNO operation condition register will be further defined in Volume 4.

3. Accelerate dynamometer to the speed point defined by :CALibration:WARMup:SPEed.

4. Average the force at the roll surface required to maintain the speed set point over :CALIbrate:WARMup:LATime. The average force can be monitored by :SENSe:FORCe?

5. The procedure will end when the elapsed time is greater than :CALibration:WARMup:TIMeout.

6. Clear the DYNO operation condition register Warm up Procedure bit indicating the Warm-up is not executing.

7. Issue the process complete message.

8. Execute the Idle Procedure.

2.4.1.3.2 :SPEed <numeric_value>

CALibration:WARMup:SPEed

For chassis dynamometers this is the fixed roll speed (m/s) at which the warm-up will be run.

At *RST, this value is set to 22.

2.4.1.3.3 :TIMeout <numeric_value>

CALibration:WARMup:TIMeout

For chassis dynamometers this is the length of time in seconds the warm-up will run before automatically returning to zero speed.

At *RST, this value is device dependent.

2.4.1.4 :ZERO

:CALibration:ZERO

For chassis dynamometers, this node describes the procedure to set up the conditions required and perform a mathematical zero correction of the roll force measurement system calibration. The correction is determined from a measurement of the roll force sensor at a given roll speed. The results are stored in the FSensorZero pre-defined table.

2.4.1.4.1 :FSENsor

CALibration:ZERO:FSENsor

Force Sensor

For chassis dynamometers this node sets up the conditions required and performs a mathematical zero correction of the roll force measurement system calibration. The correction is determined from a measurement of the roll force sensor at a given roll speed.

2.4.1.4.1.1 :INITiate

CALibration:ZERO:FSENsor:INITiate

This is an overlapped command. Perform the measurement of the roll force, acquiring for the length of time specified by :LATime. This command describes an event and therefore has no associated *RST condition.

When this command is initiated for dynamometers, the dynamometer must:

1. Check critical ancillaries, if any, for proper conditions: Brakes Off, Motor Contactor On, parameters within dynamometer safety limits. For conditions that prohibit operation of this procedure, issue appropriate warning messages, and execute the Idle Procedure.

2. Set the DYNO operation condition register Force Sensor Zero Procedure bit indicating the Zero is executing. Note: The DYNO operation condition register will be further defined in Volume 4.

3. Accelerate dynamometer to the speed specified in :CALibration:ZERO:FSENsor:SPEed. Stabilize for :STIMe.

4. Average the force at the roll surface required to maintain the speed set point over the :LATime. The average force value is placed in :LEVel.

5. Clear the DYNO operation condition register Force Sensor Zero Procedure bit indicating the Zero is not executing.

6. Issue the process complete message.

7. Execute the Idle Procedure.

2.4.1.4.1.2 :LATime

CALibration:ZERO:FSENsor:LATime

Loss Averaging Time - sec

The length of time to average the measured zero reading.

At *RST, this value is set to 0.5.

2.4.1.4.1.3 :LEVel?

CALibration:ZERO:FSENsor:LEVel?

Returns the measured force value used to determine the mathematical zero correction. This value is in units of force at the roll surface. This is query only.

2.4.1.4.1.4 :SPEed <numeric_value>

CALibration:ZERO:FSENsor:SPEed

The roll speed set point for the force measurement period in m/s.

At *RST, this value is set to 0.

2.4.1.4.1.5 :STIMe <numeric_value>

CALibration:ZERO:FSENsor:STIMe

Stabilization Time - sec

Time for stabilization prior to LATime.

At *RST, this value is set to 5.

2.4.1.4.1.6 :UPDate

CALibration:ZERO:FSENsor:UPDate

Once the mathematical zero correction has been determined, apply the results to the on-line calibration. This is not a queryable command and therefore has no *RST associated with it.

2.4.2 DIAGnostic Subsystem

There are no Instrument Class specific commands for DIAGnostics.

2.4.3 MEMory Subsystem

The MEMory subsystem of the chassis dynamometer shall include the following new commands:

```
KEYWORD                 PARAMETER FORM                              NOTES
:MEMory
  :TABLe
  |  :ARATe
  |  |  [:MAGNitude]    <numeric_value> {,<numeric_value>}
  |  |  |  :POINts?                                                  [query only]
  |  :CINertia
  |  |  [:MAGNitude]    <numeric_value> {,<numeric_value>}
  |  |  |  :POINts?                                                  [query only]
  |  :DEFine            <structure_string>[,<numeric_value>]
  |  :DRATe
  |  |  [:MAGNitude]    <numeric_value> {,<numeric_value>}
  |  |  |  :POINts?                                                  [query only]
  |  :FORCe
  |  |  :AACCeleration
  |  |  |  [:MAGNitude] <numeric_value> {,<numeric_value>}
  |  |  |  |  :POINts?                                               [query only]
  |  |  :ADECeleration
  |  |  |  [:MAGNitude] <numeric_value> {,<numeric_value>}
  |  |  |  |  :POINts?                                               [query only]
  |  |  [:MAGNitude]    <numeric_value> {,<numeric_value>}
  |  |  |  :POINts?                                                  [query only]
  |  |  :ZOFFset
  |  |  |  [:MAGNitude] <numeric_value>
  |  :DLOSs
  |  |  [:MAGNitude]    <numeric_value> {,<numeric_value>}
  |  |  |  :POINts?                                                  [query only]
  |  :PCOefficients<n>
  |  |  [:MAGNitude]    <numeric_value> {,<numeric_value>}
  |  |  |  :POINts?                                                  [query only]
  |  :SELect            <table_name>
  |  :SPEed
  |  |  [:MAGNitude]    <numeric_value> {,<numeric_value>}
  |  |  |  :POINts?                                                  [query only]
  |  |  :STARt          <numeric_value>{,<numeric_value>...}
  |  |  |  :POINts?                                                  [query only]
  |  |  :STOP           <numeric_value>{,<numeric_value>...}
  |  |  |  :POINts?                                                  [query only]
  |  :TIME
  |  |  :ACCeleration
  |  |  |  [:MAGNitude] <numeric_value> {,<numeric_value>}
  |  |  |  |  :POINts?                                               [query only]
  |  |  :DECeleration
  |  |  |  [:MAGNitude] <numeric_value> {,<numeric_value>}
  |  |  |  |  :POINts?                                               [query only]
```

KEYWORD	PARAMETER FORM	NOTES
		[:MAGNitude]	<numeric_value> {,<numeric_value>}	
			:POINts?		[query only]

New commands for the MEMory:TABLe subsystem are defined below.

2.4.3.1 :ARATe

MEMory:TABLe:ARATe

Specifies the AccelerationRATe points of the TABLe.

2.4.3.1.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:ARATe:MAGNitude

Specifies the AccelerationRATe[:MAGNitude] points of the TABLe. The default units of this command are meters per second per second.

At *RST, the contents of this list item is device dependent.

2.4.3.1.1.1 :POINts?

MEMory:TABLe:ARATe:MAGNitude:POINts?

Returns the number of points in the ARATe[:MAGNitude] list of the selected TABLe. If no ARATe[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.2 :CINertia

MEMory:TABLe:CINertia

Specifies the CalculatedINertia points of the TABLe.

2.4.3.2.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:CINertia:MAGNitude

Specifies the CalculatedINertia[:MAGNitude] points of the TABLe. The default units of this command are kilograms.

At *RST, the contents of this list item is device dependent.

2.4.3.2.1.1 :POINts?

MEMory:TABLe:CINertia:MAGNitude:POINts?

Returns the number of points in the CINertia[:MAGNitude] list of the selected TABLe. If no CINertia[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.3 :DRATe

MEMory:TABLe:DRATe

Specifies the DecelerationRATe points of the TABLe.

2.4.3.3.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:DRATe:MAGNitude

Specifies the DecelerationRATe[:MAGNitude] points of the TABLe. The default units of this command are meters per second per second.

At *RST, the contents of this list item is device dependent.

2.4.3.3.1.1 :POINts?

MEMory:TABLe:DRATe:MAGNitude:POINts?

Returns the number of points in the DRATe[:MAGNitude] list of the selected TABLe. If no DRATe[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.4 :FORCe

MEMory:TABLe:FORCe

See Volume 2, Section 13.11.12

2.4.3.4.1 :AACCeleration

MEMory:TABLe:FORCe:AACCeleration

Specifies the AverageACCeleration points of the TABLe.

2.4.3.4.1.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:FORCe:AACCeleration:MAGNitude

Specifies the AverageACCeleration[:MAGNitude] points of the TABLe. The default units of this command are meters per second per second.

At *RST, the contents of this list item is device dependent.

2.4.3.4.1.1.1 :POINts?

MEMory:TABLe:FORCe:AACCeleration:MAGNitude:POINts?

Returns the number of points in the AACCeleration[:MAGNitude] list of the selected TABLe. If no AACCeleration[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.4.2 :ADECeleration

MEMory:TABLe:FORCe:ADECeleration

Specifies the AverageDECeleration points of the TABLe.

2.4.3.4.2.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:FORCe:ADECeleration:MAGNitude

Specifies the AverageDECeleration[:MAGNitude] points of the TABLe. The default units of this command are meters per second per second.

At *RST, the contents of this list item is device dependent.

2.4.3.4.2.1.1 :POINts?

MEMory:TABLe:FORCe:ADECeleration:MAGNitude:POINts?

Returns the number of points in the ADECeleration[:MAGNitude] list of the selected TABLe. If no ADECeleration[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.4.3 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:FORCe:MAGNitude

See Volume 2, Section 13.11.12.1

2.4.3.4.3.1 :POINts?

MEMory:TABLe:FORCe:MAGNitude:POINts?

See Volume 2, Section 13.11.12.1.1

2.4.3.4.4 ZOFFset

MEMory:TABLe:FORCe:ZOFFset

Defines the ZeroOFFset value.

2.4.3.4.4.1 [:MAGNitude]

MEMory:TABLe:FORCe:ZOFFset:MAGNitude

Defines the ZeroOFFset value. For chassis dynamometers, this is a single value. This is a force, and the default units are newtons.

At *RST, the contents of this list item is device dependent.

2.4.3.5 :DLOSs

MEMory:TABLe:DLOSs

Specifies the DLOSs points of the TABLe.

2.4.3.5.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:DLOSs:MAGNitude

Specifies the DLOSs[:MAGNitude] points of the TABLe. The default units of this command are newtons.

At *RST, the contents of this list item is device dependent.

2.4.3.5.1.1 :POINts?

MEMory:TABLe:DLOSs:MAGNitude:POINts?

Returns the number of points in the DLOSs[:MAGNitude] list of the selected TABLe. If no DLOSs[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.6 :PCOefficients<n>

MEMory:TABLe:PCOefficients

Specifies the PolynomialCOefficients points of the TABLe.

2.4.3.6.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:PCOefficients:MAGNitude

Specifies the PolynomialCOefficients[:MAGNitude] points of the TABLe. The default units of this command are as follows: Units of the polynomial: N, N/(m/s), N/m/s)^2, N/(m/s)^3, ...

At *RST, the contents of this list item is device dependent.

2.4.3.6.1.1 :POINts?

MEMory:TABLe:PCOefficients:MAGNitude:POINts?

Returns the number of points in the PCOefficients[:MAGNitude] list of the selected TABLe. If no PCOefficients[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.7 :SPEed

MEMory:TABLe:SPEed

See Volume 2, Section 13.11.23

2.4.3.7.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:SPEed:MAGNitude

See Volume 2, Section 13.11.23.1

2.4.3.7.1.1 :POINts?

MEMory:TABLe:SPEed:MAGNitude:POINts?

See Volume 2, Section 13.11.23.1.1

2.4.3.7.2 :STARt

MEMory:TABLe:SPEed:STARt

For chassis dynamometers, this specifies the start speed used in Coastdown

2.4.3.7.2.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:SPEed:STARt:MAGNitude

Specifies the STARt[:MAGNitude] points of the TABLe. The default units of this start speed command are meters per second.

At *RST, the contents of this list item is device dependent.

2.4.3.7.2.1.1 :POINts?

MEMory:TABLe:SPEed:STARt:MAGNitude:POINts?

Returns the number of points in the STARt[:MAGNitude] list of the selected TABLe. If no STARt[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.7.3 :STOP

MEMory:TABLe:SPEed:STOP

For chassis dynamometers, this specifies the stop speed used in Coastdown

2.4.3.7.3.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:SPEed:STOP:MAGNitude

Specifies the STOP[:MAGNitude] points of the TABLe. The default units of this start speed command are meters per second.

At *RST, the contents of this list item is device dependent.

2.4.3.7.3.1.1 :POINts?

MEMory:TABLe:SPEed:STOP:MAGNitude:POINts?

Returns the number of points in the STOP[:MAGNitude] list of the selected TABLe. If no STOP[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.8 :TIME

MEMory:TABLe:TIME

See Volume 2, Section 13.11.24

2.4.3.8.1 :ACCeleration

MEMory:TABLe:TIME:ACCeleration

For chassis dynamometers, this command defines the elapsed time over an acceleration interval.

2.4.3.8.1.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:TIME:ACCeleration:MAGNitude

Specifies the TIME:ACCeleration[:MAGNitude] points of the TABLe. The default units of this command are seconds.

At *RST, the contents of this list item is device dependent.

2.4.3.8.1.1.1 :POINts?

MEMory:TABLe:TIME:ACCeleration:MAGNitude:POINts?

Returns the number of points in the ACCeleration[:MAGNitude] list of the selected TABLe. If no ACCeleration[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.8.2 :DECeleration

MEMory:TABLe:TIME:DECeleration

For chassis dynamometers, this command defines the elapsed time over a deceleration interval.

2.4.3.8.2.1 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:TIME:DECeleration:MAGNitude

Specifies the TIME:DECeleration[:MAGNitude] points of the TABLe. The default units of this command are seconds.

At *RST, the contents of this list item is device dependent.

2.4.3.8.2.1.1 :POINts?

MEMory:TABLe:TIME:DECeleration:MAGNitude:POINts?

Returns the number of points in the DECeleration[:MAGNitude] list of the selected TABLe. If no DECeleration[:MAGNitude] values have been sent POINts? returns 0. If no TABLe has been selected, POINts? returns NAN.

2.4.3.8.3 [:MAGNitude]<numeric_value>{,<numeric_value>}

MEMory:TABLe:TIME:MAGNitude

See Volume 2, Section 13.11.24.1

2.4.3.8.3.1 :POINts?

MEMory:TABLe:TIME:MAGNitude:POINts?

See Volume 2, Section 13.11.24.1.1

3 Digital Meters

A digital meter is a basic measuring instrument. It typically measures a single value at one point in time. This type of instrument is so general that it is first described generically. Attributes specific to a particular type of meter are described in their own subsections.

All SCPI compliant products implement the commands listed in Syntax & Style, 4.2.

Figure 3-1 shows an instrument model for a meter. It is derived from Command Reference, chapter 2. It is essentially the same as Figure 2-1 in Command Reference with various parts gray or dashed. The bold boxes and lines, solid and dashed, are described in this chapter. The FORMat and ROUTe blocks are shown with dashed lines because many meters use their default settings. *RST sets FORMat[:READings][:DATA] to ASCii and ROUTe:TERMinals to FRONt. The gray boxes are not described here. While a meter may have MEMory, its functionaily is optional and thus not described.

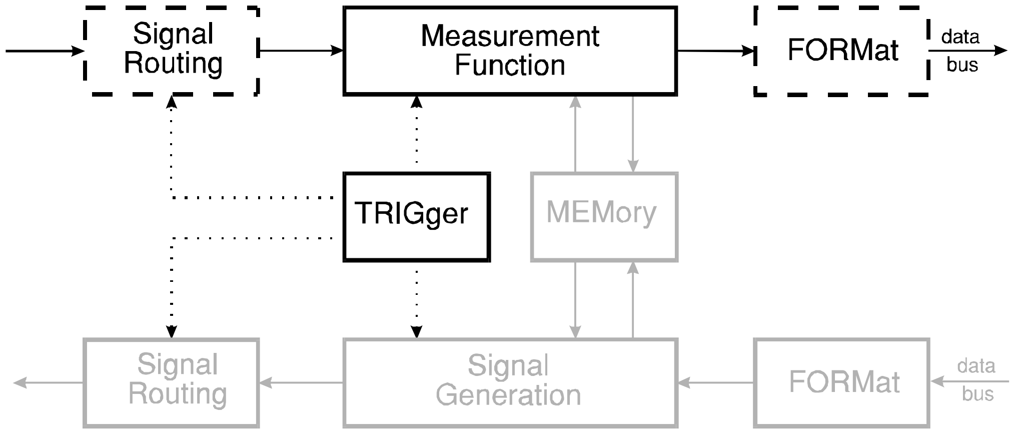


Figure 3-1 Simplified Model for a Programmable Meter

Figure 3-2 shows how the "Measurement Function" block in Figure 3-1 is expanded in a meter. It is the same as Figure 2-4 in Command Reference with some dashed and gray boxes. The CALCulate block is dashed because after *RST this block has no effect on the data. This chapter does not describe any CALCulate functions. The MEMory block is gray because while a meter may contain this function, this chapter does not address it.

A designer would never implement a generic meter, but would rather build one of the specific types. Table 1 lists the various classes of meters considered in this chapter along with the quantity they measure. A <meter_fn> is associated with each class of meter. The contents of the <meter_fn> shown in the table are used in the actual commands.

Some meters contain several instrument classes. A Digital Multi-Meter might contain a DC voltmeter, an AC voltmeter, a DC ammeter, an AC ammeter, an ohmmeter, and a 4-wire

3

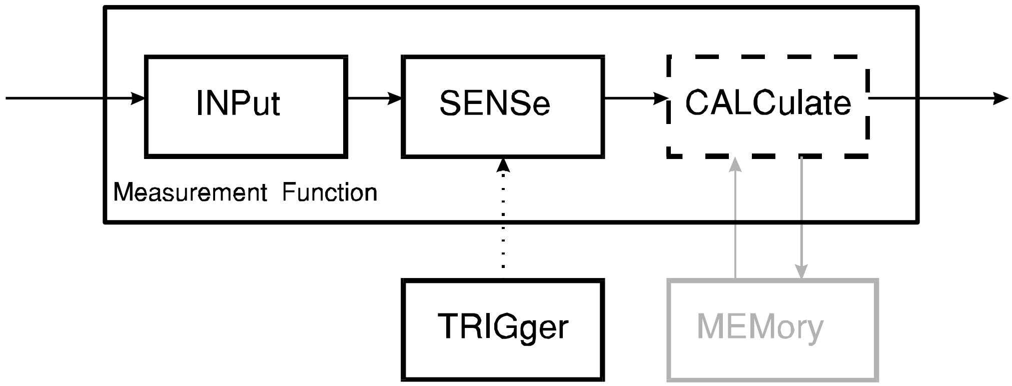


Figure 3-2 Expanded Measurement Function Model

ohmmeter. The SENSe:FUNCtion commands switch the instrument among its implemented classes. The TRIGger subsystem commands only appear once in the instrument and are shared by all the meters. Such a meter would not have TRIG1 for the DC voltmeter, TRIG2 for the AC voltmeter, etc.

Base Functionality Keyword <bf_keyword>	Common Usage Name	Measured Quantity	<meter_fn>
DCVOLTMETER	DC Voltmeter	DC Volts	VOLTage[:DC]
ACVOLTMETER	AC RMS Voltmeter	RMS Volts	VOLTage:AC
DCAMMETER	DC Ammeter	DC Amperes	CURRent[:DC]
ACAMMETER	AC RMS Ammeter	RMS Amperes	CURRent:AC
OHMMETER	Ohmmeter	Ohms	RESistance
FOHMMETER	4-wire Ohmmeter	Ohms	FRESistance

TABLE 1. Instrument Classes for Meters

3.1 Base Functionality

This section describes the functionality all meters implement.

3.1.1 Base Measurement Instructions

Since the basic function of a meter is to measure some quantity, it understands the measurement instructions in Command Reference, 3. A meter shall implement these commands with the appropriate substitution for <meter_fn>.

KEYWORD	PARAMETERS
CONFigure	
[:SCALar]	
:<meter_fn>	[<expected_value>[,<resolution>]]
CONFigure?	
FETCh	
[:SCALar]	
[:<meter_fn>]?	[<expected_value>[,<resolution>]]
READ	
[:SCALar]	
[:<meter_fn>]?	[<expected_value>[,<resolution>]]
MEASure	
[:SCALar]	
:<meter_fn>?	[<expected_value>[,<resolution>]]

The syntax of these commands is found in chapter 3 of the Command Reference. Meters may accept parentheses and <source_list> as described by the full syntax.

The response from *RST;MEAS:<meter_fn>? is a single <NR1>, <NR2>, or <NR3> because *RST sets FORMat to ASCII, TRIGger:COUNt to 1, and ARM:COUNt to 1.

READ? and FETCh? shall return a <RESPONSE MESSAGE UNIT> containing TRIGger:COUNt times ARM:COUNt numbers corresponding to that many measurements. If ARM:COUNt is not implemented, its value is one. The timing of measurements is controlled by the other settings in the TRIGger subsystem.

The command sequence MEAS:<meter_fn>?;FETCH? shall return two identical <RESPONSE MESSAGE UNIT> elements separated by a semicolon.

3.1.1.1 Effects of MEASure Query and CONFigure

MEASure query and CONFigure shall set:

SENSe:FUNCtion[:ON]	to	"[XNONe:]<meter_fn>"
INITiate:CONTinuous	to	OFF
TRIGger[:SEQuence]:SOURce	to	IMMediate
TRIGger[:SEQuence]:COUNt	to	1
TRIGger[:SEQuence]:DELay	to	MINimum

For meters that implement additional functionality ETRIGGER (see 3.2.1), MEASure query and CONFigure shall set:

ARM[:SEQuence][:LAYer]:SOURce to IMMediate
ARM[:SEQuence][:LAYer]:COUNt to 1

For meters that implement additional functionality TERMINALS (see 3.2.2), MEASure query and CONFigure shall set:

ROUTe:TERMinals unaffected

Note: Additional instrument settings may need to be changed to support the execution of the high-level measurement instruction. These settings are under the control of commands not described in this chapter and thus are not defined here.

3.1.1.2 The <expected_value> parameter

If <expected_value> is present and its value is not DEFault, SENSe:<meter_fn>:RANGe[:UPPer] shall be set to the legal value equal to or just larger than <expected_value>. SENSe:<meter_fn>:RANGe:AUTO shall be set OFF.

If <expected_value> is not present or its value is DEFault, SENSe:<meter_fn>:RANGe:AUTO shall be set ON and the instrument shall autorange to the best possible range for the measurement.

3.1.1.3 The <resolution> parameter

If <resolution> is present and its value is not DEFault, SENSe:<meter_fn>:RESolution shall be set to the legal value equal to or just less than <resolution>.

If <resolution> is not present or its value is DEFault, SENSe:<meter_fn>:RESolution shall be set to a legal value that gives a reasonably accurate result without taking excessively long.

3.1.2 Base Device-oriented Functions

3.1.2.1 FORMat Subsystem

This subsystem is not required. A meter shall implement the FORMat[:READings][:DATA] ASCii and FORMat[:READings]:DINTerchange OFF settings. Note: These are the *RST values, so the commands are not required.

3.1.2.2 SENSe subsystem

Any meter shall sense a selected function and control the range and resolution of the sensor. The ability to autorange is required. The ability to turn on multiple <sensor_functions> simultaneously is not required.

A meter shall implement these SENSe commands with the appropriate substitution for <meter_fn>.

KEYWORD	PARAMETER FORM	SCPI Reference
SENSe		Vol 2-18
:FUNCtion[:ON]	“[XNONe:]<meter_fn>”	Vol 2-18.13.2
:<meter_fn>		
:RANGe		
[:UPPer]	<numeric_value>	
:AUTO	<Boolean>	
:RESolution	<numeric_value>	

3.1.2.3 TRIGger subsystem

The primary purpose of the TRIGger subsystem is to control when measurements are made. The default timing is generally to take a measurement as soon as possible.

A meter shall implement these TRIGger commands.

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
[:IMMediate]		Vol 2-24.7.2
[:ALL]		Vol 2-24.7.2.1
ABORt		Vol 2-24.5
TRIGger		Vol 2-24.8
[:SEQuence]		Vol 2-24.8.1
:COUNt	<numeric_value>	Vol 2-24.8.1.2
:DELay	<numeric_value>	Vol 2-24.8.1.5
:SOURce	BUS | IMMediate | EXTernal | ...	Vol 2-24.8.1.17

Requiring BUS as a legal parameter to TRIGger:SOURce has the side effect of requiring DT1 capability in an IEEE 488.2 device. It also means the meter implements the *TRG common command.

A VXIbus device with a trigger function is required to implement the *Trigger* command. When TRIGger:SOURce is set BUS, a VXIbus meter shall use the *Trigger* command as the TRIGger source.

3.1.3 Base Status Reporting

Syntax & Style, chapter 9 describes the status reporting structure used in all SCPI instruments. Command Reference, chapter 20 defines the commands which control the status reporting structure.

3.1.3.1 QUEStionable Status

When a meter is measuring DC volts or AC volts, the meter shall set bit 0, VOLTage, in the QUEStionable status structure when the meter suspects the voltage is of questionable accuracy. When a meter is measuring DC current or AC current, the meter shall set bit 1, CURRent, in the QUEStionable status structure when the meter suspects the current is of questionable accuracy. When a meter is measuring ohms, the meter shall set either bit 0, VOLTage, or bit 1, CURRent when the meter suspects either is of questionable accuracy.

3.1.3.2 OPERation Status

While the meter is performing one of the operations listed in the OPERation status structure, that bit shall be set in the condition register. Every meter shall report at least RANGing, bit 2, MEASuring, bit 4, and Waiting for TRIG, bit 5.

3.2 Additional functionality

3.2.1 Extended Trigger

The <af_keyword> for extended trigger is ETRIGGER.

Meters with extended trigger functionality provide two layer triggering where a separate event must occur before the TRIGger layer is entered. The commands under ARM are used to provide this functionality rather than another SEQuence.

3.2.1.1 ETRIGGER Measurement Instructions

ETRIGGER adds no Measurement Instructions.

3.2.1.2 ETRIGGER Device-oriented Functions

3.2.1.2.1 TRIGger Subsystem

KEYWORD	PARAMETER FORM	SCPI Reference
ARM		Vol 2-24.6
[:SEQuence]		Vol 2-24.6.1
[:LAYer]		Vol 2-24.6.1.2
:SOURce	{BUS | IMMediate | ... }	Vol 2-24.6.1.2.14

3.2.1.3 ETRIGGER Status Reporting

Bit 6 (Waiting for ARM) in the OPERation Status Register is used to indicate the status of the trigger model.

3.2.2 Multiple Terminals

The <af_keyword> for multiple terminals is TERMINALS.

A meter generally has only a single input, though some meters may be able switch the input between front and rear terminals. These meters have the TERMINALS additional functionality.

3.2.2.1 TERMINALS Measurement Instructions

TERMINALS adds no Measurement Instructions.

3.2.2.2 TERMINALS Device-oriented Functions

3.2.2.2.1 ROUTe Subsystem

A meter which can select between front and rear terminals shall implement

KEYWORD	PARAMETER FORM	SCPI Reference
ROUTe		Vol 2-17
:TERMinals	FRONt | REAR	Vol 2-17.7

3.2.2.3 TERMINALS Status Reporting

TERMINALS adds no Status Reporting requirements.

3.2.3 Offset Compensation

The <af_keyword> for offset compensation is OCOMPENSATED.

An ohmmeter or 4-wire ohmmeter may be unable to measure resistance when residual voltages are present. An ohmmeter or 4-wire ohmmeter which can compensate for residual voltages when measuring resistance has the OCOMPENSATED additional functionality.

3.2.3.1 OCOMPENSATED Measurement Instructions

OCOMPENSATED adds no measurement instructions.

MEASure:RESistance? and CONFigure:RESistance shall set SENse:RESistance:OCOMpensated to OFF. MEASure:FRESistance? and CONFigure:FRESistance shall set SENse:FRESistance:OCOMpensated to OFF.

3.2.3.2 OCOMPENSATED Device-oriented Functions

3.2.3.2.1 SENSe Subsystem

An ohmmeter or 4-wire ohmmeter which can compensate for residual voltages shall implement:

KEYWORD	PARAMETER FORM	SCPI Reference
SENSe		Vol 2-18
:RESistance		Vol 2-18.18
| :FRESistance		
:OCOMpensated	<Boolean>	Vol 2-18.18.3

3.2.3.3 OCOMPENSATED Status Reporting

OCOMPENSATED adds no Status Reporting requirements.

3

3.3 Various Meter Classes

This section provides additional information and rules of behavior for specific meter classes.

3.3.1 DC Voltmeter

DC voltmeters measure the average voltage level of the signal across their inputs.

3.3.1.1 Range

DC voltmeters shall measure negative as well as positive voltages symmetrically about zero. SENSe:RANGe[:UPPer] determines how far from zero measurements can be made. The MINimum and MAXimum values for SENSe:RANGe[:UPPer] shall be positive. Since the range is symmetric about zero, SENSe:RANGe:LOWer is not needed, but if implemented its value shall be coupled to UPPer by the equation:

```
LOWer = -UPPer.
```

3.3.1.2 SENSe Commands

SENSe:VOLTage[:DC] commands are found in Command Reference, 18.20.

3.3.2 AC RMS Voltmeter

AC RMS voltmeters measure the root mean square of the voltage level across their inputs.

3.3.2.1 Range

Since the RMS of any signal is non-negative, the MINimum and MAXimum values for SENSe:RANGe[:UPPer] shall be positive. SENSe:RANGe:LOWer is not needed.

3.3.2.2 SENSe Commands

SENSe:VOLTage:AC commands are found in Command Reference 1994, 18.21.

3.3.3 Ohmmeter

Ohmmeters measure the resistance across their input terminals.

Typically, a DC current is sourced and the DC voltage is measured. Resistance is calculated by dividing the current into the voltage.

3.3.3.1 SENSe Commands

SENSe:RESistance commands are found in Command Reference 1994, 18.16.

3.3.3.2 4-wire Ohmmeter

4-wire ohmmeters measure the resistance across two of their input terminals with the assistance of two additional wires.

Typically, a DC current is sourced through one set of wires and the DC voltage is measured across the other set. As with a 2-wire ohms measurement, resistance is calculated by dividing the current into the voltage.

3.3.3.3 SENSe Commands

SENSe:FRESistance commands are found in Command Reference 1994, 18.16.

3.3.4 DC Ammeter

DC ammeters measure the average value of the current through their inputs.

3.3.4.1 Range

DC ammeters shall follow the same constraints on RANGe as DC voltmeters.

3.3.4.2 SENSe Commands

SENSe:CURRent[:DC] commands are found in Command Reference 1994, 18.6.

3.3.5 AC RMS Ammeter

AC RMS ammeters measure the root mean square of the current through their inputs.

3.3.5.1 Range

AC RMS ammeters shall follow the same constraints on RANGe as AC voltmeters.

3.3.5.2 SENSe Commands

SENSe:CURRent:AC commands are found in Command Reference 1994, 18.6.

3.4 Programming Examples

This section discusses how a user might program a meter for certain applications. The examples assume the meter is a DC voltmeter, but a meter of any type could have been used.

3.4.1 Simple Measurement

In an application where the user wants to make a simple measurement with no special consideration for signal conditioning or timing, the MEASure query is an excellent choice. An example might be checking if a power supply voltage is at a proper level. The instrument might receive the following query to make a voltage measurement:

```
MEASure:VOLTage:DC?
```

The instrument autoranges to the best range and makes the measurement with a reasonable resolution. A single value is returned as soon as possible. The instrument does not wait for any other triggers.

Another user might send:

```
MEASure:VOLTage:DC? 5,.05
```

A voltmeter that has 1, 10, and 100 volt ranges would be in the 10 volt range. Another voltmeter with 3, 30, and 300 volt ranges would use the 30 volt range. Resolution would probably get set to 10 mv corresponding to 3 1/2 digits. Again the measurement is made as soon as possible.

This same measurement could be made using low level commands.

```
*RST
SENSe:FUNCtion "VOLTage:DC";VOLTage:RANGe 5V;RESolution .05V
INITiate;FETCh?
```

Note that *RST sets all TRIGger subsystem SOURce to IMMediate and COUNt to 1.

3.4.2 Time Critical Measurement

To use an external triggering signal, the user cannot use MEASure query because MEASure query sets TRIGger:SOURce to IMMediate. Instead, CONFigure and READ? with TRIGger commands between them is the proper choice. The sequence:

```
CONFigure:VOLTage:DC 5V,.05V
TRIGger:SOURce EXTernal
READ?
```

sets up the SENSe functions easily. The single TRIGger command modifies the triggering subsystem just enough to do the job. The READ query initiates a measurement which waits for the external trigger before returning the result.

3.4.3 Multiple Measurements

The overhead of sending commands may prevent taking measurements made close together in time using a separate MEASure query or READ? for each measurement. The instrument could, however, buffer the readings. The sequence:

```
CONFigure:VOLTage:DC 5V,.05V
TRIGger:SOURce EXTernal;COUNt 10
READ?
```

returns ten separate measurements taken when ten external triggers occur. A defensive programmer would then send SYSTem:ERRor? and check for error -210, "Trigger error" or -211, "Trigger ignored" to see if there were any problems with the triggering.

4 Digitizers

A digitizer is an instrument designed primarily to measure time-swept voltage waveforms. Because a digitizer is primarily a sensing device, the SENSe root node is optional.

All SCPI compliant products implement the commands listed in Syntax & Style, 4.2.

Figure 4-1 shows an instrument model for a digitizer. It is derived from Command Reference, chapter 2. It is essentially the same as Figure 2-1 in Command Reference with various parts gray or dashed. The bold boxes and lines, solid and dashed, are described in this chapter. The ROUTe block are shown with dashed lines because many digitizers use their default settings.

The gray boxes are not described here. While a digitizer may have MEMory, its functionaily is optional and thus not described.

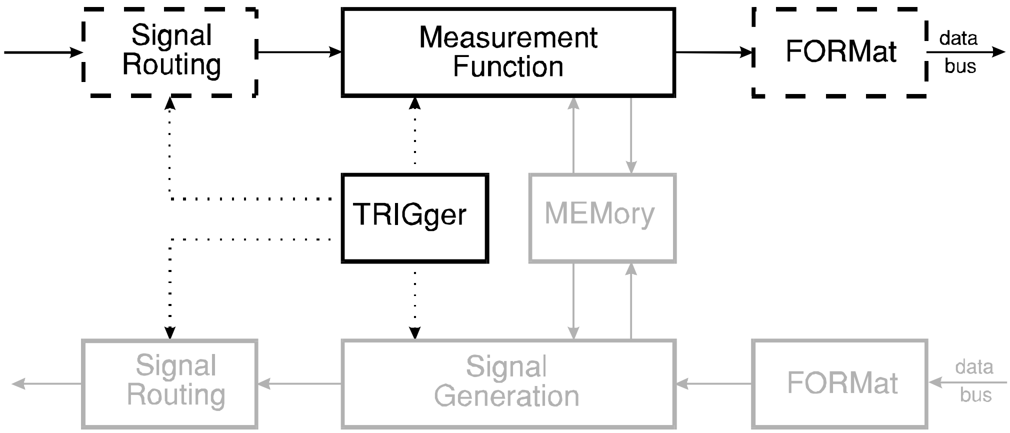


Figure 4-1 Simplified Model of a Digitizer

Figure 4-2 shows how the “Measurement Function” block in Figure 4-1 is expanded in a digitizer. It is the same as Figure 2-4 in Command Reference with some dashed and gray boxes. The CALCulate block is dashed because after *RST this block has no effect on the data. This chapter does not describe any CALCulate functions. The MEMory block is gray because while a digitizer may contain this function, this chapter does not address it.

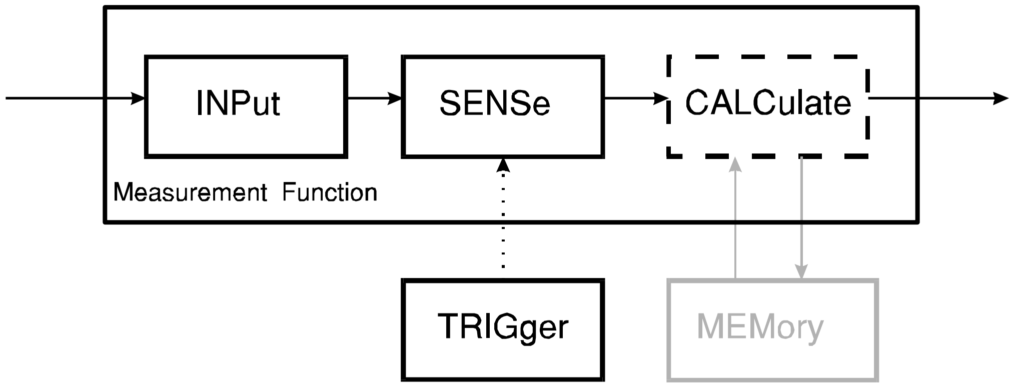


Figure 4-2 Expanded Measurement Function Model

4.1 Base Functionality

This section describes the functionality all digitizers implement.

The <bf_keyword> for a digitizer is DIGITIZER.

Digitizer inputs are typically called channels. These are generally numbered 1, 2, 3, ... or lettered A, B, C, ... Several SCPI commands in this section use numeric suffixes on header keywords or parameters. The intention is that these suffixes unify operations that apply to the associated channel.

Two-channel capability is shown in the examples to illustrate this concept. A single channel digitizer may be implemented by omitting the command and or parameter options with the numeric suffix 2. More than two channels may be implemented by extending the numeric suffix.

4.1.1 Base Measurement Instruction

The base functionality for a digitizer requires no measurement instructions (MEASure, CONFigure, READ, FETCh).

4.1.2 Base Device-oriented Functions

The base functionality for digitizers shall contain the commands described in this section.

4.1.2.1 Input functions

KEYWORD	PARAMETER FORM	SCPI Reference
INPut[1]|INPut2		Vol 2-11
:COUPling	AC	DC	GND	Vol 2-11.3

This command controls the coupling characteristics of the input to the amplifier.

At *RST, the coupling for digitizers shall be DC.

As previously mentioned, INPut1 is associated with the first input channel, and so on.

4.1.2.2 SENSe amplitude functions

KEYWORD	PARAMETER FORM	SCPI Reference
[SENSe]		Vol 2-18
:VOLTage[1]|:VOLTage2		Vol 2-18.24
[:DC]		Vol 2-18.24.1
:RANGe		Vol 2-18.24.1.5
:LOWer	<numeric_value>	Vol 2-18.24.1.5.2
:OFFSet	<numeric_value>	Vol 2-18.24.1.5.4
:PTPeak	<numeric_value>	Vol 2-18.24.1.5.5
[:UPPer]	<numeric_value>	Vol 2-18.24.1.5.1

VOLTage1 is associated with INPut1, VOLTage2 is associated with INPut2, and so on.

The PTPeak command sets the amplifier gain of the specified channel.

The OFFSet command sets the voltage offset of the amplifier, the midpoint of the PTPeak range. Measurement results remain constant when offset is added.

UPPer and LOWer, also expressed in volts, are coupled to OFFSet and PTPeak as described in the introduction to Command Reference, chapter 18.

4.1.2.3 SENSE timebase functions

KEYWORD	PARAMETER FORM	SCPI Reference
[SENSe]		Vol 2-18
:SWEep		Vol 2-18.23
:POINts	<numeric_value>	Vol 2-18.23.8
:TIME	<numeric_value>	Vol 2-18.23.12
:TINTerval	<numeric_value>	Vol 2-18.23.13

POINts controls the record length of the acquisition.

TIME is the full time span of the acquisition.

TINTerval controls the time interval between points.

NOTE that if the number of sweep points and and the time interval are changed then the sweep time will change as well.

At *RST, a digitizer shall set its SWEep parameters to a known, device-specified value.

4.1.2.4 SENSe One-of-N Function Control

Digitizers shall implement the following commands that allow the selection of the active channel using [SENSe:]FUNCtion.

KEYWORD	PARAMETER FORM	SCPI Reference
[SENSe]		Vol 2-18
:DATA?	[<data_handle>]	Vol 2-18.13.1
:FUNCtion		
:CONCurrent	<Boolean>	Vol 2-18.13.2.1
:OFF	<sensor_function>	Vol 2-18.13.2.2
[:ON]	<sensor_function>	Vol 2-18.13.2.3
:STATe?	<sensor_function>	Vol 2-18.13.2.4

Where <sensor_function> and <data_handle> for channel 1 shall be specified as "XTIMe:VOLTage[:DC] 1" and so on.

[SENSe]:DATA? shall return data for the channel specified in the query parameter or shall return data for the active channel(s).

4.1.2.5 Formatting

KEYWORD	PARAMETER FORM	SCPI Reference
:FORMat		
[:DATA]	<type> [, <numeric_value>]	Vol 2-

The parameter ASCii and the <numeric_value> zero shall be supported.

At *RST, the :FORMat type shall be ASCii,0.

4.1.2.6 Trigger functions

KEYWORD	PARAMETER FORM	SCPI Reference
INITitiate		
[:IMMediate]		Vol 2-24.7.2
[:ALL]		Vol 2-24.7.2.1
ABORt		Vol 2-24.5
TRIGger		Vol 2-24.8
[:SEQuence[1]]		
:COUPling	AC | DC	Vol 2-24.8.1.3
:LEVel	<numeric_value>	Vol 2-24.8.1.12
:SLOPe	POSitive | NEGative | EITher	Vol 2-24.8.1.16
:SOURce	INTernal[1] | ...	Vol 2-24.8.1.17

Triggering sourced from channel 1 shall be specified as INTernal1 and so on.

At *RST, SOURce shall be set to INTernal1, COUPling shall be set to DC, SLOPe shall be set to POSitive, and LEVel shall be set to a known, device-specified value.

4.1.3 Base Status Reporting

All SCPI digitizers shall implement the status reporting structure described in Syntax & Style, Status Reporting. Command Reference, STATus Subsystem defines the commands which shall be used to control the status reporting structure.

4.1.3.1 QUEStionable

The base functionality of a digitizer adds no requirements for this register.

4.1.3.2 OPERation

At a minimum, the digitizer shall report Waiting for TRIG, bit 5.

4.2 Programming Example

The following set of commands sets up the digitizer to acquire a waveform on channel 1 for AC coupling, 10 peak to peak signal with a sample recorded every microsecond. The trigger source is channel 1 and is DC coupled, looking for a positive slope transition of 100 millivolts. After setting up the digitizer the trigger system is initiated and the resultant captured waveform is returned to the controller in ASCII format.

Longform version:

```
*RST
INPut:COUPling AC
SENSe:VOLTage:DC:RANGE:PTPeak 10 V
SENSe:SWEep:TINTerval 1E-6 S
SENSe:FUNCtion:CONCurrent OFF
SENSe:FUNCtion:ON “XTIMe:VOLTage:DC 1"
TRIGger:SEQuence1:LEVel 0.1V
INITiate:IMMediate:ALL
SENSe:DATA?
```

Shortform version:

```
*RST
INP:COUP AC
VOLT:RANG:PTP 10
SWE:TINT 1E-6 S
FUNC:CONC 0;ON “XTIM:VOLT 1"
TRIG:LEV 0.1
INIT
DATA?
```

5 Emissions Benches

An emissions bench is a complex measuring device. It measures the constituent concentration in a sample of gas introduced into the bench. It typically consists of several instruments, at least one of which is a gas analyzer, but also can include a mini-diluter, a gas divider, as well as temperature, pressure, and flow measurement instruments.

All SCPI compliant products implement the commands listed in Syntax and Style, 4.2.

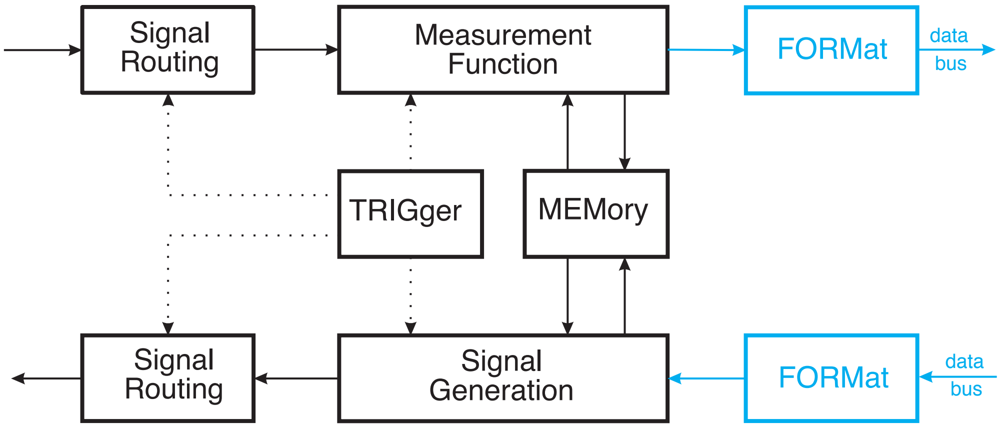


Figure 5-1 Simplified Model for an Emissions Bench

As indicated in Figure 5-1 above, emissions benches make use of signal (gas) routing commands, the TRIGger subsystem, extensive use of measurement function commands (using the :SENSe block), and extensive use of the :MEMory subsystem. In addition, the :CALibrate and :SYSTem subsystems are utilized in both standard and emissions bench-specific ways.

5.1 Base Functionality

This section describes the base functionality for emissions benches.

The <bf_keyword> for an emissions bench is EBENCH.

5.1.1 Base Measurement Instructions

The base functionality for an emissions bench contains no commands from the :MEASurement subsystem.

5.1.2 Base Device-Oriented Functions

The base functionality for an emissions bench include :SENSe, :MEMory, :TRIGger, :CALibration, :CONTrol, :SYSTem, :DIAGnostic, :INSTrument, and :ROUTe subsystems. It also defines device dependent :STATus bits and instructions for emissions benches. A SCPI-compliant emissions bench shall implement all of the commands listed in this section.

5.1.2.1 CALibration subsystem

CALibration commands for emissions benches, detailed in section 5.4 of this volume, are as follows:

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:CALibration</td><td></td><td>Vol 4-5.4</td></tr>
<tr><td>:LINearize</td><td></td><td>Vol 4-5.4.1</td></tr>
<tr><td>| :ACCept</td><td></td><td>Vol 4-5.4.1.1</td></tr>
<tr><td>| :ACQuire</td><td></td><td>Vol 4-5.4.2</td></tr>
<tr><td>| :AUTO</td><td>ONCE</td><td>Vol 4-5.4.2.1</td></tr>
<tr><td>| :CALCulate</td><td>AUTO | POLYnomial<n> |</td><td>Vol 4-5.4.2.2</td></tr>
<tr><td>| |</td><td>SRATional<n></td><td></td></tr>
<tr><td>| :CURVe</td><td></td><td>Vol 4-5.4.2.3</td></tr>
<tr><td>| | [:TYPE]</td><td>POLYnomial<n>|</td><td>Vol 4-5.4.2.3.1</td></tr>
<tr><td>| |</td><td>SRATional<n>,<numeric_value>{,<numeric_value>}</td><td></td></tr>
<tr><td>| | :ZFORce</td><td><Boolean></td><td>Vol 4-5.4.2.3.2</td></tr>
<tr><td>| :VERify</td><td></td><td>Vol 4-5.4.2.4</td></tr>
<tr><td>| | :ACQuire</td><td></td><td>Vol 4-5.4.2.4.1</td></tr>
<tr><td>| | :TOLerance</td><td><numeric_value></td><td>Vol 4-5.4.2.4.2</td></tr>
<tr><td>| | :TYPE</td><td>CCURve | NCURve</td><td>Vol 4-5.4.2.4.3</td></tr>
</table>

5.1.2.2 CONTrol subsystem

CONTrol commands for emissions benches are as follows:

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:CONTrol</td><td></td><td>Vol 2-6</td></tr>
<tr><td>:EBENch</td><td></td><td>Vol 2-6.6</td></tr>
<tr><td>| :CLEan</td><td></td><td>Vol 2-6.6.1</td></tr>
<tr><td>| | :DURation</td><td><numeric_value></td><td>Vol 2-6.6.1.2</td></tr>
<tr><td>| | [:INIT]</td><td></td><td>Vol 2-6.6.1.1</td></tr>
</table>

5.1.2.3 DIAGnostic subsystem

DIAGnostic commands for emissions benches, detailed in section 5.5 of this volume, are as follows:

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:DIAGnostic</td><td></td><td>Vol 4-5.5</td></tr>
<tr><td>:HUP</td><td></td><td>Vol 4-5.5.1</td></tr>
<tr><td>| :ACQuire</td><td></td><td>Vol 4-5.5.1.1</td></tr>
<tr><td>| :CALCulate</td><td></td><td>Vol 4-5.5.1.2</td></tr>
<tr><td>:LEAK</td><td></td><td>Vol 4-5.5.2</td></tr>
<tr><td>| :ACQuire</td><td></td><td>Vol 4-5.5.2.1</td></tr>
<tr><td>| :CALCulate</td><td></td><td>Vol 4-5.5.2.2</td></tr>
<tr><td>:NEFFiciency</td><td></td><td>Vol 4-5.5.3</td></tr>
<tr><td>| :ACQuire</td><td></td><td>Vol 4-5.5.3.1</td></tr>
<tr><td>| :CALCulate</td><td></td><td>Vol 4-5.5.3.2</td></tr>
</table>

5.1.2.4 INSTrument subsystem

INSTrument commands for emissions benches are as follows:

```
KEYWORD             PARAMETER FORM                     SCPI Reference
:INSTrument                                            Vol 2-12
  :CATalog?                                            Vol 2-12.1
  |   :FULL?                                           Vol 2-12.1.1
  :DEFine                                              Vol 2-12.3
  |   :GROup        <identifier>, <identifier_list>    Vol 2-12.3.1
  |  [:NAME]        <identifier>,<numeric_value>       Vol 2-12.3.2
  :DELete                                              Vol 2-12.4
  |   :ALL                                             Vol 2-12.4.1
  |  [:NAME]        <identifier>                       Vol 2-12.4.2
 [:SELect]          <identifier>                       Vol 2-12.6
```

5.1.2.5 MEMory subsystem

MEMory commands for emissions benches are as follows:

```
KEYWORD             PARAMETER FORM                          SCPI Reference
:MEMory                                                     Vol 2-13
  :TABLe                                                    Vol 2-13.11
  |   :BNUMber<n>   <numeric_value> {,<numeric_value>}      Vol 2-13.11.1
  |   |   :POINts?                                          Vol 2-13.11.1.1
  |   :CCURve       <numeric_value> {,<numeric_value>}      Vol 2-13.11.2
  |   |   :POINts?                                          Vol 2-13.11.2.1
  |   :CONCentration <numeric_value> {,<numeric_value>}     Vol 2-13.11.3
  |   |   :POINts?                                          Vol 2-13.11.3.1
  |   :CPOint       <numeric_value> {,<numeric_value>}      Vol 2-13.11.5
  |   |   :POINts?                                          Vol 2-13.11.5.1
  |   :DEFine       <structure_string> [,<numeric_value>]   Vol 2-13.11.11
  |   :DFACtory     <numeric_value> {,<numeric_value>}      Vol 2-13.11.7
  |   |   :POINts?                                          Vol 2-13.11.7.1
  |   :DLASt        numeric_value> {,<numeric_value>}       Vol 2-13.11.8
  |   |   :POINts?                                          Vol 2-13.11.8.1
  |   :DLINearize   <numeric_value> {,<numeric_value>}      Vol 2-13.11.9
  |   |   :POINts?                                          Vol 2-13.11.9.1
  |   :EXPected     <numeric_value> {,<numeric_value>}      Vol 2-13.11.10
  |   |   :POINts?                                          Vol 2-13.11.10.1
  |   :LABel        <string> {,<string>}                    Vol 2-13.11.14
  |   |   :POINts?                                          Vol 2-13.11.14.1
  |   :LOG          <string> {,<string>}                    Vol 2-13.11.16
  |   |   :POINts?                                          Vol 2-13.11.16.1
  |   :NCURve       <numeric_value> {,<numeric_value>}      Vol 2-13.11.18
  |   |   :POINts?                                          Vol 2-13.11.18.1
```

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>NOTES</th></tr>
<tr><td>| :RAW</td><td><numeric_value> {,<numeric_value>}</td><td>Vol 2-13.11.20</td></tr>
<tr><td>| | :POINts?</td><td></td><td>Vol 2-13.11.20.1</td></tr>
<tr><td>| :SELect</td><td><identifier></td><td>Vol 2-13.11.22</td></tr>
<tr><td>| :TOLerance</td><td><numeric_value> {,<numeric_value>}</td><td>Vol 2-13.11.25</td></tr>
<tr><td>| | :POINts?</td><td></td><td>Vol 2-13.11.25.1</td></tr>
<tr><td>| :WFACtor</td><td><numeric_value> {,<numeric_value>}</td><td>Vol 2-13.11.28</td></tr>
<tr><td>| | :POINts?</td><td></td><td>Vol 2-13.11.28.1</td></tr>
</table>

5.1.2.6 ROUTe subsystem

ROUTe commands for emissions benches are as follows:

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:ROUTe</td><td></td><td>Vol 2-17</td></tr>
<tr><td>:SAMPle</td><td></td><td>Vol 2-17.4.3.1</td></tr>
<tr><td>| :CATalog?</td><td></td><td>Vol 2-17.5.1</td></tr>
<tr><td>| [:OPEN]</td><td>BAG | DILute | PRE | POST | MID
| CEFFiciency | NONE | ZERO
| SPAN | VERify | MANifold<n></td><td>Vol 2-17.5.2</td></tr>
</table>

Note that some emissions benches may not implement all of the above sample paths.

5.1.2.7 SENSe subsystem

SENSe commands for emissions benches are as follows:

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>:SENSe</td><td></td><td>Vol 2-18</td></tr>
<tr><td>:AVERage</td><td></td><td>Vol 2-18.2</td></tr>
<tr><td>| :COUNt</td><td><numeric_value></td><td>Vol 2-18.2.2</td></tr>
<tr><td>| :TCONtrol
| |</td><td>EXPonential | MOVing |
NORMal | REPeat</td><td>Vol 2-18.2.4</td></tr>
<tr><td>| :TYPE
| |</td><td>COMPlex | ENVelope | MAXimum |
MINimum | RMS | SCALar</td><td>Vol 2-18.2.5</td></tr>
<tr><td>:CONCentration</td><td></td><td>Vol 2-18.4</td></tr>
<tr><td>| :CSET
|</td><td><numeric_value>,<numeric_value>
{,<numeric_value>,<numeric_value>,...}</td><td>Vol 2-18.4.1</td></tr>
<tr><td>| :LOWer</td><td><numeric_value>{,<numeric_value>}</td><td>Vol 2-18.4.2</td></tr>
<tr><td>| :LSET
|
|
|
|</td><td>POLYnomial<n> |
SRATional<n>,<numeric_value>
{,<numeric_value>},{POLYnomial<n>
| SRATional<n>,<numeric_value>
{,<numeric_value>}}</td><td>Vol 2-18.4.3</td></tr>
<tr><td>| :RANGe</td><td></td><td>Vol 2-18.4.4</td></tr>
<tr><td>| | :AUTO</td><td></td><td>Vol 2-18.4.4.1</td></tr>
<tr><td>| | | :LOWer</td><td><numeric_value>{,<numeric_value>}</td><td>Vol 2-18.4.4.1.1</td></tr>
<tr><td>| | | [:STATe]</td><td><Boolean></td><td>Vol 2-18.4.4.1.2</td></tr>
<tr><td>| | | :UPPer</td><td><numeric_value>{,<numeric_value>}</td><td>Vol 2-18.4.4.1.3</td></tr>
<tr><td>| | [:FIXed]</td><td><numeric_value>{,<numeric_value>}</td><td>Vol 2-18.4.4.2</td></tr>
</table>

```
   |  :TALign          <numeric_value> {,<numeric_value>}  Vol 2-18.4.5
   |  :UPPer           <numeric_value>{,<numeric_value>}   Vol 2-18.4.6
   :CORRection                                          Vol 2-18.6
   |  :AUTO                                             Vol 2-18.6.1
   |  :CALCulate                                        Vol 2-18.6.2
   |  :SPOint                                           Vol 2-18.6.10
   |  |  :ACQuire                                       Vol 2-18.6.10.1
   |  |  :DTOLerance   <numeric_value>{,<numeric_value>}   Vol 2-18.6.10.2
   |  :STATe           < Boolean >                      Vol 2-18.6.11
   |  :ZERO                                             Vol 2-18.6.12
   |  |  :ACQuire                                       Vol 2-18.6.12.1
   |  |  :DTOLerance   <numeric_value>{,<numeric_value>}   Vol 2-18.6.12.2
   :DATA?                                               Vol 2-18.13.1
   :FUNCtion           <function_name> {,<function_name>} Vol 2-18.13.2
   |  |                where function_name can be:
   |  |                CONCentration<n>,TIMer:COUNt,
   |  |                CONCentration<n>:SDEViation,
   |  |                CONCentration<n>:TALign,
   |  |                CONCentration<n>:RAW
   |  :CONCurrent      <Boolean>                        Vol 2-18.13.2.1
   :STABilize          Vol 2-18.22
   |  :NTOLerance      <numeric_value>{,<numeric_value>}   Vol 2-18.22.1
   | [:STATe]          <Boolean>                        Vol 2-18.22.2
   |  :TIME<n>         <numeric_value>{,<numeric_value>}   Vol 2-18.22.3
```

NOTE:
:SENSe:CORRection:CALCulate calculates the zero/span adjustment of the form

$$y = mx + b$$

where
$x = \%FS_{RAW}$ Data Value
$y = \%FS_{ADJ}$ Data Value
m = Zero/Span Adjustment Slope Coefficient
b = Zero/Span Adjustment Intercept Coefficient

and

$$m = \frac{S_0 - Z_0}{S_m - Z_m}$$

$$b = \frac{Z_0 S_m - Z_m S_0}{S_m - Z_m}$$

where
Z_0 = Zero Gas Expected
Z_m = Zero Gas Measured
S_0 = Span Gas Expected
S_m = Span Gas Measured

5

5.1.2.8 SYSTem subsystem

SYSTem commands for emissions benches are as follows:

KEYWORD	PARAMETER FORM	SCPI Reference
:SYSTem		Vol 2-21
:COMMunicate		Vol 2-21.4
| :SOCKet<n>		Vol 2-21.4.4
| | :ADDRess	<string> (IPaddress|Host Name)	Vol 2-21.4.4.1
| | :CONNect		Vol 2-21.4.4.2
| | :DISConnect		Vol 2-21.4.4.3
| | :FEED	<data_handle>{,<data_handle>}	Vol 2-21.4.4.4
| | | :OCONdition	<event_handle>	Vol 2-21.4.4.4.1
| | | :SCONdition	<event_handle>	Vol 2-21.4.4.4.2
| | :LISTen		Vol 2-21.4.4.5
| | :PORT	<NRf>|<non-decimal numeric>	Vol 2-21.4.4.6
| | :TYPE	TCP|UDP	Vol 2-21.4.4.7
:DATE	<year>,<month>,<day>	Vol 2-21.7
:ERRor		Vol 2-21.8
| :ALL?		Vol 2-21.8.4
| :CODE		Vol 2-21.8.5
| | :ALL?		Vol 2-21.8.5.1
| | [:NEXT]?		Vol 2-21.8.5.2
| :ENABle		Vol 2-21.8.7
| | :ADD	<numeric list>	Vol 2-21.8.7.1
| | :DELete	<numeric list>	Vol 2-21.8.7.2
| | [:LIST]		Vol 2-21.8.7.3
| [:NEXT]?		Vol 2-21.8.8
:LOCK		Vol 2-21.14
| :RELease		Vol 2-21.14.2
| :REQuest?		Vol 2-21.14.3
| :OWNer?		Vol 2-21.14.1
:TIME		Vol 2-21.19
| :TIMer		Vol 2-21.19.1
| | :COUNt	<numeric_value>	Vol 2-21.19.1.1
| | [:STATe]	<Boolean>	Vol 2-21.19.1.2

5.1.2.9 TRIGger subsystem

TRIGger commands for emissions benches are as follows:

KEYWORD	PARAMETER FORM	SCPI Reference
:TRIGger		Vol 2-24.8
[:SEQuence]		Vol 2-24.8.1
| :ECOunt	<numeric_value>	Vol 2-24.8.1.7
| :LINK	<event_handle>	Vol 2-24.8.1.13
| :SOURce	TIMer	Vol 2-24.8.1.17
| :TIMer	<numeric_value>	Vol 2-24.8.1.18

5.1.3 Base Status Reporting

Volume 1, Syntax and Style, Status Reporting, describes the status reporting structure used in all SCPI instruments. Volume 2, Command Reference, STATus Subsystem, defines the commands that control the status reporting structure.

5.1.3.1 OPERation Status

:STATus:OPERation commands for benches utilize bit 9 of the OPERation status register (see SCPI Volume 1 figure 9-1) for bench-wide status and assume a parallel structure to the :STATus:QUEStionable commands referred to in section 5.1.3.2 below. Therefore, bit 10 of an instrument summary operations register is used for instrument-specific status.

```
KEYWORD                    PARAMETER FORM                SCPI Reference
:STATus                                                  Vol 2-20
  :OPERation                                             Vol 2-20.1
  |  :BIT9                                               Vol 2-20.1.1
  |  |  :CONDition?                                      Vol 2-20.1.2
  |  |  :ENABle            <NRf>|<non-decimal numeric>   Vol 2-20.1.3
  |  | [:EVENt]?                                         Vol 2-20.1.4
  |  |  :NTRansition       <NRf>|<non-decimal numeric>   Vol 2-20.1.6
  |  |  :PTRansition       <NRf>|<non-decimal numeric>   Vol 2-20.1.7
  |  :CONDition?                                         Vol 2-20.1.2
  |  :ENABle               <NRf>|<non-decimal numeric>   Vol 2-20.1.3
  | [:EVENt]?                                            Vol 2-20.1.4
  |  :INSTrument
  |  |  :CONDition?                                      Vol 2-20.1.2
  |  |  :ENABle            <NRf>|<non-decimal numeric>   Vol 2-20.1.3
  |  | [:EVENt]?                                         Vol 2-20.1.4
  |  |  :ISUMmary<n>
  |  |  |  :BIT10                                        Vol 2-20.1.1
  |  |  |  |  :CALibrating
  |  |  |  |  |  :CONDition?                             Vol 2-20.1.2
  |  |  |  |  |  :ENABle   <NRf>|<non-decimal numeric>   Vol 2-20.1.3
  |  |  |  |  | [:EVENt]?                                Vol 2-20.1.4
  |  |  |  |  |  :NTRansition<NRf>|<non-decimal numeric> Vol 2-20.1.6
  |  |  |  |  |  :PTRansition <NRf>|<non-decimal numeric> Vol 2-20.1.7
  |  |  |  |  :CONDition?                                Vol 2-20.1.2
  |  |  |  |  :ENABle      <NRf>|<non-decimal numeric>   Vol 2-20.1.3
  |  |  |  | [:EVENt]?                                   Vol 2-20.1.4
  |  |  |  |  :NTRansition <NRf>|<non-decimal numeric>   Vol 2-20.1.6
  |  |  |  |  :PTRansition <NRf>|<non-decimal numeric>   Vol 2-20.1.7
  |  |  |  :CONDition?                                   Vol 2-20.1.2
  |  |  |  :ENABle         <NRf>|<non-decimal numeric>   Vol 2-20.1.3
  |  |  | [:EVENt]?                                      Vol 2-20.1.4
  |  |  |  :NTRansition    <NRf>|<non-decimal numeric>   Vol 2-20.1.6
  |  |  |  :PTRansition    <NRf>|<non-decimal numeric>   Vol 2-20.1.7
  |  |  :NTRansition       <NRf>|<non-decimal numeric>   Vol 2-20.1.6
  |  |  :PTRansition       <NRf>|<non-decimal numeric>   Vol 2-20.1.7
  |  :NTRansition          <NRf>|<non-decimal numeric>   Vol 2-20.1.6
  |  :PTRansition          <NRf>|<non-decimal numeric>   Vol 2-20.1.7
```

5.1.3.2 QUEStionable Status

:STATus:QUEStionable commands for benches utilize bit 9 of the QUEStionable status register (see SCPI Volume 1 figure 9-1) for bench-wide status and bit 10 of the instrument summary status register (see SCPI volume 1 figure 9-3) for instrument-specific status as follows:

```
KEYWORD                        PARAMETER FORM                   SCPI Reference
:STATus                                                         Vol 2-20
  :QUEStionable                                                 Vol 2-20.3
  |  :BIT9                                                      Vol 2-20.3.1
  |  |  :CONDition?                                             Vol 2-20.3.2
  |  |  :ENABle                <NRf>|<non-decimal numeric>      Vol 2-20.3.3
  |  | [:EVENt]?                                                Vol 2-20.3.4
  |  |  :NTRansition           <NRf>|<non-decimal numeric>      Vol 2-20.3.6
  |  |  :PTRansition           <NRf>|<non-decimal numeric>      Vol 2-20.3.7
  |  :CONDition?                                                Vol 2-20.3.2
  |  :ENABle                   <NRf>|<non-decimal numeric>      Vol 2-20.3.3
  | [:EVENt]?                                                   Vol 2-20.3.4
  |  :INSTrument
  |  |  :CONDition?                                             Vol 2-20.3.2
  |  |  :ENABle                <NRf>|<non-decimal numeric>      Vol 2-20.3.3
  |  | [:EVENt]?                                                Vol 2-20.3.4
  |  |  :ISUMmary<n>
  |  |  |  :BIT10                                               Vol 2-20.3.1
  |  |  |  |  :CALibration
  |  |  |  |  |  :CONDition?                                    Vol 2-20.3.2
  |  |  |  |  |  :ENABle       <NRf>|<non-decimal numeric>      Vol 2-20.3.3
  |  |  |  |  | [:EVENt]?                                       Vol 2-20.3.4
  |  |  |  |  |  :NTRansition  <NRf>|<non-decimal numeric>      Vol 2-20.3.6
  |  |  |  |  |  :PTRansition  <NRf>|<non-decimal numeric>      Vol 2-20.3.7
  |  |  |  |  :CONDition?                                       Vol 2-20.3.2
  |  |  |  |  :ENABle          <NRf>|<non-decimal numeric>      Vol 2-20.3.3
  |  |  |  | [:EVENt]?                                          Vol 2-20.3.4
  |  |  |  |  :NTRansition     <NRf>|<non-decimal numeric>      Vol 2-20.3.6
  |  |  |  |  :PTRansition     <NRf>|<non-decimal numeric>      Vol 2-20.3.7
  |  |  |  :CONDition?                                          Vol 2-20.3.2
  |  |  |  :ENABle             <NRf>|<non-decimal numeric>      Vol 2-20.3.3
  |  |  | [:EVENt]?                                             Vol 2-20.3.4
  |  |  |  :NTRansition        <NRf>|<non-decimal numeric>      Vol 2-20.3.6
  |  |  |  :PTRansition        <NRf>|<non-decimal numeric>      Vol 2-20.3.7
  |  |  :NTRansition           <NRf>|<non-decimal numeric>      Vol 2-20.3.6
  |  |  :PTRansition           <NRf>|<non-decimal numeric>      Vol 2-20.3.7
  |  :NTRansition              <NRf>|<non-decimal numeric>      Vol 2-20.3.6
  |  :PTRansition              <NRf>|<non-decimal numeric>      Vol 2-20.3.7
```

5.2 Additional functionality

All functionality described for emissions benches is considered base functionality at this time.

5.3 Examples

5.3.1 Zero/Span/Zero Procedure

5

The following illustrates a possible command sequence for performing a zero/span/zero procedure:

```
SCPI Command            Notes
:INST:DEFine:GROup "BENCH","CONC1","CONC2"
:INST:SELect "BENCH"
                        Select desired analyzers
:SENSe:FUNCtion:CONCurrent ON
:SENSe:FUNCtion "CONC1","CONC1:SDEV","CONC2","CONC2:SDEV"
                        Applies to selected analyzers; used in :SENS:DATA
:SENSe:CORRection:ZERO:DTOLerance 5
                        Allow 5% tolerance for zero drift
:SENSe:CORRection:SPOint:DTOLerance 5
                        Allow 5% tolerance for set point (span) drift
:SENSe:CORRection:STATe OFF
                        New Z/S/Z needs uncorrected values;
                        drift check would require :SENS:CORR:STATE ON
:SENSe:CORRection:ZERO:ACQuire
                        request and execute zero procedure
:STATus:OPERation:INSTrument:ISUMmary1:BIT10:CALibrating?
                        Keep polling while bit 1 (zero) is set (CONC1)
:STATus:OPERation:INSTrument:ISUMmary2:BIT10:CALibrating?
                        Keep polling while bit 1 (zero) is set (CONC2)
:STATus:QUEStionable:INSTrument:ISUMmary1:BIT10:CALibrating?
                        0 = successful zero; Bit 1 set = failure
:STATus:QUEStionable:INSTrument:ISUMmary2:BIT10:CALibrating?
                        0 = successful zero; Bit 1 set = failure
:SENSe:DATA?            Retrieve Zero Concentration Readings and SDEVs
:MEMory:DATA?ZDRift     Retrieve Zero Drifts
:SENSe:CORRection:SPOint:ACQuire
                        execute SPOint (span) procedure
*WAI                    Alternative to polling;
                        wait for SPOint (span) operation to complete
:STATus:QUEStionable:INSTrument:ISUMmary1:BIT10:CALibrating?
                        0 = successful zero; Bit 2 set = failure
:STATus:QUEStionable:INSTrument:ISUMmary2:BIT10:CALibrating?
                        0 = successful zero; Bit 2 set = failure
:SENSe:DATA?            Retrieve Span Concentration Readings and SDEV's
:MEMory:DATA?SDRift     Retrieve Span Drifts
:SENSe:CORRection:CALCulate
                        Calculate new m, b corrections (skip this if drift check)
:SENSe:CORRection:ZERO:ACQuire
                        do re-zero
```

```
:STATus:OPERation:INSTrument:ISUMmary1:BIT10:CALibrating?
                   Keep polling while bit 1 (zeroing) is set
                   (CONC1)
:STATus:OPERation:INSTrument:ISUMmary2:BIT10:CALibrating?
                   Keep polling while bit 1 (zeroing) is set
                   (CONC2)
:STATus:QUEStionable:INSTrument:ISUMmary1:BIT10:CALibrating?
                   0 = successful zero; Bit 1 set = failure
:STATus:QUEStionable:INSTrument:ISUMmary2:BIT10:CALibrating?
                   0 = successful zero; Bit 1 set = failure
:SENSe:DATA?       Retrieve Zero Readings and SDEVs
:MEMory:DATA?ZDRift Retrieve Zero Drifts
:SENSe:CORRection:ZERO:DTOLerance 2
                   Allow 2% tolerance for zero drift
:SENSe:CORRection:STATe[:ON]
                   start applying new m, b
:SENSe:CONC:CSET?  Retrieve m,b for this range of this analyzer
```


5.3.2 Bag Read Procedure

The following command sequence illustrates a possible command sequence for bag reads:

SCPI Command *Notes*

```
:SYSTem:LOCK:REQuest?
                   Assert host control of bench if 1 is returned
:INSTrument:DEFine:GROUp "Bag_Analyzers","CONC1", "CONC2",
"CONC3", "CONC4"   Sets up the group "Bag_Analyzers to use
                   four concentrations
:INSTrument:SELect "Bag_Analyzers"
:CONTrol:EBENch:CLEan:DURation 60
                   Set a gas line "cleanup" for 60 seconds
:CONTrol:EBENch:CLEan[:INIT]
                   Start CLEan procedure
:SENSE:FUNCtion:CONCurrent ON
:SENSe:FUNCtion "CONC1", "CONC1:SDEV","CONC2","CONC2:SDEV",
"CONC3","CONC3:SDEV", "CONC4", "CONC4:SDEV"
:TRIGger:SEQ1:SOURce TIMer
:TRIGger:SEQ1:TIMer 0.1
                   ".1" means .1 seconds
:SENSe:STABilized:STATE ON
                   Use Stabilized mode
```

Start "Sniff"

```
:SENSe:CONC:RANGe:AUTO ON       Auto range to find correct range
```

Host command to sampling system that selects bag

```
:ROUTe:SAMPle BAG  Cold_Transient_Sample
:INITiate:IMMediate:SEQuence1
:STATus:OPERation:BIT9?
                   Keep polling while bit 5 (sampling) is set
:STATus:QUEStionable:BIT9?
```

```
                            0 = successful bag sample
    :SENSe:DATA?            Retrieve four Sniff concentrations
    :SENSe:CONC:RANGe:AUTO OFF
                            Leave in selected range
    :SENSe:CONC:RANGe:FIXed?
                            Retrieve range numbers (n,n,n,n)
    :ROUTe:SAMPle NONE      Turn off gas flow
```

Perform ZERO/SPAN/ZERO PROCEDURE - See Section 5.3.1

Host command to sampling system that selects bag

```
    :ROUTe:SAMPle BAG   Cold_Transient_Sample
    :INITiate:IMMediate:SEQuence1
    :STATus:OPERation:BIT9?
                            Keep polling while bit 5 (sampling) is set
    :STATus:QUEStionable:BIT9?
                            0 = successful bag sample
    :SENSe:DATA?            Retrieve the four Sample Bag concentrations and
                            SDEVs
```

Host command to sampling system that selects bag

```
    :INITiate:IMMediate:SEQuence1
    :STATus:OPERation:BIT9?
                            Keep polling while bit 5 (sampling) is set
    :STATus:QUEStionable:BIT9?
                            0 = successful bag sample
    :SENSe:DATA?            Retrieve the four Ambient Bag concentrations and
                            SDEV s
    :ROUTe:SAMPle NONE      Turn off gas flow
```

Perform Z/S Drift check procedure - similar to section 5.3.1

```
    :SYSTem:LOCK:RELease
                            Release control of bench
```

5.3.3 Start of Diesel Test Procedure

The following command sequence illustrates a possible command sequence for starting a diesel test:

SCPI Command *Notes*

Setup Phase

```
    :SYSTem:LOCK:REQuest?
                            Assert host control of bench
    :INSTrument:DEFine:GROup “All_Bag_Analyzers”,"CONC1", “CONC2",
    ”CONC3", CONC4"         Sets up the groups to use
    :INSTrument:SELect “All_Bag_Analyzers”
    :CONTrol:EBENch:CLEan:DURation 60
                            Do a gas line “cleanup” for 60 seconds
    :CONTrol:EBENch:CLEan[:INIT]
                            Do a gas line “cleanup” for 60 seconds
    :SENSe:FUNCtion:CONCurrent ON
                            Applies to selected analyzers
    :SENSe:FUNCtion “CONC1", ”CONC1:SDEV","CONC2","CONC2:SDEV",
```

```
"CONC3","CONC3:SDEV", "CONC4", "CONC4:SDEV"
                    Applies to selected analyzers
:SENSe:STABilization:STATE ON
                    Applies to selected analyzers
:SENSe:STABilization:TIME1 60,50,45,55
                    Stabilization rise time
:SENSe:STABilization:TIME2 15,15,15,15
                    Stabilization evaluation period
:SENSe:STABilization:TIME3 5,5,5,5
                    Stabilization averaging period
:SENSe:STABilization:TIME4 120,120,120,120
                    Stabilization maximum duration
:SENSe:STABilization:TOLerance 4,4,4,4
                    Stabilization tolerance in %
:INSTrument:DEFine:GROUp
"Bag_With_Out_Fid","CONC2","CONC3","CONC4"
:INSTrument:SELect "Bag_With_Out_Fid"
:SENSE:FUNCtion:CONCurrent ON
                    Applies to selected analyzers
:SENSe:FUNCtion"CONC2","CONC2:SDEV",
"CONC3","CONC3:SDEV","CONC4","CONC4:SDEV"
                    Applies to selected analyzers
:SENSe:STABilization:STATE ON
                    Applies to selected analyzers
:SENSe:STABilization:TIME1 50,45,55
                    Stabilization rise time
:SENSe:STABilization:TIME2 15,15,15
                    Stabilization evaluation period
:SENSe:STABilization:TIME3 5,5,5
                    Stabilization averaging period
:SENSe:STABilization:TIME4 120,120,120
                    Stabilization maximum duration
:SENSe:STABilization:TOLerance 4,4,4
                    Stabilization tolerance in %
:INSTrument:DEFine:GROUp "Just_Heated Fid","CONC1"
:INSTrument:SELect "Just_Heated Fid"
:SENSe:AVERage1:COUNT 10
:SENSe:AVERage1:TCONtrol MOVing
:SENSe:AVERage1:TYPE SCALar
:SENSe:AVERage1:STATe ON
:SENSe:FUNCtion:CONCurrent ON
:SENSe:FUNCtion "CONC1 ON AVERage1","CONC1:SDEV ON AVERage1"
:SENSe:STABilization:STATe ON
:SENSe:STABilization:TIME1 60
                    Stabilization rise time
:SENSe:STABilization:TIME2 15
                    Stabilization evaluation period
:SENSe:STABilization:TIME3 5
                    Stabilization averaging period
:SENSe:STABilization:TIME4 120
```

5

```
                        Stabilization maximum duration
   :SENSe:STABilization:TOLerance 4
                        Stabilization tolerance in %
   :TRIGger:SEQ1:SOURce TIMer
                        Set up 10 Hz Sampling
   :TRIGger:SEQ1:TIMer 0.1
                        Set up integration rate
   :TRIGger:SEQ2:LINK "TRIGger:SEQence1"
                        Seq 2 slaved to Seq1
   :TRIGger:SEQ2:ECOunt 10
                        Seqence 2 once per second
   :SYSTem:COMMunicate:SOCKet1:ADDRess "123.456.789.1"
                        Separate Data feed
   :SYSTem:COMMunicate:SOCKet1:PORT 701
                        to the host
   :SYSTem:COMMunicate:SOCKet1:TYPE TCP
                        A TCP Socket
   :SYSTem:COMMunicate:SOCKet1:FEED:OCON "TRIG:SEQ1"
                        Collect 10/sec
   :SYSTem:COMMunicate:SOCKet1:FEED:SCON "TRIG:SEQ2"
                        Send once/sec
   :SYSTem:COMMunicate:SOCKet1:CONNect
```

Pretest function

```
   :INSTrument:SELect "Just_Heated_Fid"
   :SENSe:CONCentration:RANGe:FIXed n
                        Alternate form for 1 analyzer
```

Perform ZERO/SPAN/ZERO PROCEDURE - See Section 5.3.1

```
   :SENSe:STABilization:STATe OFF
                        For command channel "reads"
   :SENSe:FUNCtion "TIMer:COUNt", "CONC1:TALign", "TEMP1"
   :SYSTem:COMMunicate:SOCKet1:FEED "TIMer:COUNt", "CONC1:TAL",
   "TEMP1"              For socket feed
   :ROUTe:SAMPle DILute
```

Phase 1

```
   :INITiate:CONTinuous ON
                        Start socket feed
```

drive phase 1

5.4 CALibration Subsystem (Bench Commands)

```
KEYWORD               PARAMETER FORM                 NOTES
:CALibration
  :LINearize
  |  :ACCept                                         [event]
  |  :ACQuire                                        [event]
  |  :AUTO            ONCE                           [event]
  |  :CALCulate       AUTO | POLYnomial<n>
  |                   | SRATional<n>                 [event]
  |  :CURve
  |  |  [:TYPE]       POLYnomial<n>
  |  |                | SRATional<n>, <numeric_value>{,<numeric_value>}
  |  |  :ZFORce       <Boolean>
  |  :VERify
  |  |  :ACQuire                                     [event]
  |  |  :TOLerance    <numeric_value>
  |  |  :TYPE        CCURve | NCURve
```

5.4.1 :LINearize

CALibration:LINearize

The LINearize subsystem is used to produce a calibration curve used to linearize a non-linear instrument. An instance of this subsystem exists for each range in an instrument. The data defined and queried and the events will only be performed on the currently selected range and instrument.

5.4.1.1 :ACCept

CALibration:LINearize:ACCept

ACCept causes an already-calculated or supplied set of coefficients to be placed into regular use by an instrument. This consists of replacing the “current” values with the “new” values. The current values can be obtained by querying :SENSe:CONCentration:RANGe:LSET?.

5.4.2 :ACQuire

CALibration:LINearize:ACQuire

ACQuire causes the automatic collection of linearization data POINts for the currently selected range of the currently selected analyzer to begin. Results are placed in a predefined table as they are collected. This procedure stores the expected gas concentration, causes gas flow for the defined cutPOINts, waits for a stabilized reading of each one, and records the measured values. :SENSe:STABilize:STATe must be ON when this command is issued. Gases are selected for the duration of the procedure.

5.4.2.1 :AUTO ONCE

CALibration:LINearize:AUTO

This command shall perform linearization based on a vendor defined procedure, including :CALibrate:LINearize:ACQuire and possibly other procedures, for example :SENSe:CORRection:ZERO and :SPOint. :SENSe:STABilize:STATe must be ON when this command is issued. Gases are selected for the duration of the procedure. This command is provided to allow the instrument to perform an automated linearization procedure according to the manufacturer's recommended practice.

5.4.2.2 :CALCulate AUTO|POLYnomial<n>|SRATional<n>

CALibration:LINearize:CALCulate

Using the results data stored in a predefined linearization table, calculate a linearization curve. The results are placed in that table. A curve fit of type AUTO allows the device to calculate the curve fit according to the manufacturer's recommended practice. A POLYnomial<n> parameter shall limit the calculation to the specified polynomial curve type. A SRATional<n> parameter shall limit the linearization to the specified simple rational curve fit. The <n> designates the order of the curve. A curve fit can automatically be forced through the origin, if :CALibration:LINearize:CURVe:ZFORce has been turned on. The results can be stored in the instrument at a later time (see ACCept). An example of the predefined table follows:

LSI01R01

CPOint (%FS)	EXPected	RAW	CCURve	NCURve
6.7	10000	464646	10015	10001
20.0	30000	484848	30109	30000
33.3	50000	515151	49905	49999
46.7	70000	525252	71004	70001
60	90000	555555	89950	90005
73.3	110000	565656	111070	110100
86.7	130000	575859	129051	129999
100	150000	590101	150031	149000

5.4.2.3 :CURVe

CALibration:LINearize:CURVe

This node controls analyzer curve fitting.

5.4.2.3.1 [:TYPE] POLYnomial<n> | SRATional<n>, <numeric_value>{,<numeric_value>}

CALibration:LINearize:CURVe[:TYPE]

[:TYPE] sets or retrieves the information used to determine a calibration curve.

The first parameter is the curve type, which consists of either POLYnomial for a polynomial curve fit or SRATional for a simple rational polynomial curve fit, followed by a positive integer <n>, which is the curve order.

The additional parameters are the curve coefficients. These are listed starting with the 0th order term. Note that the number of coefficients depends on the order of the curve - there are exactly n+1 coefficients for a curve of order n. For example:

```
:CALibration:LINearize:CURVe:TYPE
POLY3,-0.117,-7.43,1.02,-0.034
```

defines a calibration curve of type 3rd order polynomial and the four coefficients used.

As a query, TYPE returns latest new pending curve fit type and calculated curve coefficients.

If, as in the above example,

$$f(x) = -0.117 - 7.43*x + 1.02*x^2 - 0.034*x^3$$

results from the curve fit, the returned curve values shall be:

```
POLY3,-0.117,-7.43,1.02,-0.034
```

A third order simple rational polynomial

$$f(x) = \frac{x}{0.81 - 2.0*x + 1.17*x^2 + 0.02*x^3}$$

shall be represented by the parameters

```
SRAT3,0.81,-2.0,1.17,0.02
```

5

5.4.2.3.2 :ZFORce <Boolean>

CALibration:LINearize:CURVe:ZFORce

The setting of ZFORce (Zero FORce) determines whether or not a newly calculated curve will be fit such that the scalar polynomial term is 0. This affects the :CALibration:LINearize:CALCulate command.

At *RST, ZFORce is not changed.

5.4.2.4 :VERify

CALibration:LINearize:VERify

VERify causes the instrument to check the validity of a set of linearization coefficients by checking the difference from the concentration measured with the curve to the expected concentration of the verification gas.

This command is an event, and therefore has no *RST value associated with it.

5.4.2.4.1 :ACQuire

CALibration:LINearize:VERify:ACQuire

ACQuire causes the flow and stabilization of a verification gas to occur. First, verification gas flow is enabled, next a stabilized reading is taken, and finally the verification gas flow is disabled. The concentration determined from this procedure is compared against a known concentration stored in a designated table. If the difference between the expected concentration of the verification gas and the concentration measured using the curve coefficients is greater than the tolerance value set by :CALibration:LINearize:VERify:TOL, then the appropriate questionable status bit is set.

This command is an event, and therefore has no *RST value associated with it.

5.4.2.4.2 :TOLerance <numeric_value>

CALibration:LINearize:VERify:TOLerance

TOLerance sets the allowable tolerance between the expected concentration of the verification gas and the concentration measured using the curve coefficients in % of full scale of the active range of the selected analyzer.

At *RST, TOLerance is not changed.

5.4.2.4.3 :TYPE

CALibration:LINearize:VERify:TYPE

TYPE selects the linearization coefficients to be verified. Possible types are:

- CCURve - The curve coefficients currently in effect
- NCURve - A newly calculated or downloaded set of coefficients that are not yet in use.

At *RST, TYPE is set to CCURve.

5.5 DIAGnostic Subsystem (Bench Commands)

KEYWORD	PARAMETER FORM	NOTES
:DIAGnostic		
:HUP		
| :ACQuire		[no query]
| :CALCulate		[no query]
:LEAK		
| :ACQuire		[no query]
| :CALCulate		[no query]
:NEFFiciency		
| :ACQuire		[no query]
| :CALCulate		[no query]

5.5.1 :HUP

DIAGnostic:HUP

Commands to perform a diesel HC HangUP check.

5.5.1.1 :ACQuire

DIAGnostic:HUP:ACQuire

This procedure performs the procedure found the 1996 Code of Federal Regulations, 86.1340-90 Exhaust Sample Analysis, Paragraph (e) (3); it first takes a zero direct reading;, then a zero overflow reading. This command has no query form.

This command is an event, and therefore has no *RST value associated with it.

5.5.1.2 :CALCulate

DIAGnostic:HUP:CALCulate

Compare the direct zero value with the overflow zero value. These values and their difference are stored in memory table HUPResults. The difference is compared with the value in memory table HUPTolerance. Error bit 4 of :STATus:QUEStionable:INSTrument:ISUMmary5:BIT10 is set if the value of the difference is outside of the tolerance values stored in HUPTolerance. These tolerance values are upper and lower tolerance in ppm.

This command is an event, and therefore has no *RST value associated with it.

5.5.2 :LEAK

DIAGnostic:LEAK

Commands to help perform a bench leak check.

5.5.2.1 :ACQuire

DIAGnostic:LEAK:ACQuire

Initiate a bench leak check. This causes the bench perform a device-dependent leak check procedure. This command has no query form.

This command is an event, and therefore has no *RST value associated with it.

5.5.2.2 :CALCulate

DIAGnostic:LEAK:CALCulate

This command performs a device-dependent leak check calculation to indicate success or failure of the leak check.

Error bit 4 of :STATus:QUEStionable:INSTrument:ISUMmary5:BIT10 is set if any of the calculated values are outside of these tolerance values. The lower and upper tolerance values are stored in table LTOLerance; the value which is compared against these tolerances is stored in table LRESults. This command has no query form.

This command is an event, and therefore has no *RST value associated with it.

5.5.3 :NEFFiciency

DIAGnostic:NEFFiciency

Commands to initiate the NOx efficiency diagnostic procedure and its calculations.

5.5.3.1 :ACQuire

DIAGnostic:NEFFiciency:ACQuire

Initiate the NOx efficiency procedure. This is a six-step test and is performed according to the EPA's 1996 code of Federal Regulations, section 86.123-78.

This command is an event, and therefore has no *RST value associated with it.

5.5.3.2 :CALCulate

:DIAGnostic:NEFFiciency:CALCulate

Calculate the NOx efficiency results, per the EPA's 1996 code of Federal Regulations, section 86.123-78. This command has no query form. The results of the calculations are placed in the memory table NEResults. Error bit 4 of :STATus:QUEStionable:INSTrument:ISUMmary5:BIT10 is set if any of the calculated values are outside of the tolerance values resident in the memory table NETolerance.

This command is an event, and therefore has no *RST value associated with it.

6 Emission Test Cell

An Emission Test Cell consists of one or more computers and a set of instruments that are used to perform emissions testing. The instruments which are included in an ETCELL may include Emission Benches, Dynamometers, Sampling Systems, and other emissions measurement or other devices. In order to standardize the interfaces within the cell it is desirable that all instrument implement a core set of commands. These commands are described in this chapter.

All SCPI compliant Emission Test Cells shall implement the commands listed in Syntax & Style sections 4.2.

6.1 Base Functionality

This section describes the base functionality for an Emission Test Cell.

The <bf_keyword> for an Emissions Test Cell is ETCELL.

6.1.1 Base Measurement Instructions

The base functionality for an ETCELL contains no commands from the MEASure subsystem.

6.1.2 Base Device-oriented Functions

ETCELL compliant instruments shall include SYSTem commands, which set up and cause communication between instruments and the site computer and the set up and transfer of data from instruments.

6.1.2.1 SYSTem subsystem

The SYSTem subsystem of an Emission Test Cell shall include the following commands.

<table>
<tr><th>KEYWORD</th><th>PARAMETER FORM</th><th>SCPI Reference</th></tr>
<tr><td>SYSTem</td><td></td><td>V2-21</td></tr>
<tr><td>:CAPability?</td><td></td><td>V2-21.3</td></tr>
<tr><td>:COMMunicate</td><td></td><td>V2-21.4</td></tr>
<tr><td>| :SOCKet<n></td><td></td><td>V2-21.4.4</td></tr>
<tr><td>| | :ADDRess</td><td><string>(Ipaddress | Host Name)</td><td>V2-21.4.4.1</td></tr>
<tr><td>| | :CONNect</td><td></td><td>V2-21.4.4.2</td></tr>
<tr><td>| | :DISConnect</td><td></td><td>V2-21.4.4.3</td></tr>
<tr><td>| | :FEED</td><td><data_handle>{,<data_handle>}</td><td>V2-21.4.4.4</td></tr>
<tr><td>| | | :OCONdition</td><td><event_handle></td><td>V2-21.4.4.4.1</td></tr>
<tr><td>| | | :SCONdition</td><td><event_handle></td><td>V2-21.4.4.4.2</td></tr>
<tr><td>| | :LISTen</td><td></td><td>V2-21.4.4.5</td></tr>
<tr><td>| | :PORT</td><td><NRf>|<non-decimal numeric></td><td>V2-21.4.4.6</td></tr>
<tr><td>| | :TYPE</td><td>TCP | UDP</td><td>V2-21.4.4.7</td></tr>
<tr><td>:DATE</td><td><year>,<month>,<day></td><td>V2-21.7</td></tr>
<tr><td>:ERRor</td><td></td><td>V2-21.8</td></tr>
<tr><td>| :ALL?</td><td></td><td>V2-21.8.4</td></tr>
<tr><td>| :CODE?</td><td></td><td>V2-21.8.5</td></tr>
<tr><td>| | :ALL?</td><td></td><td>V2-21.8.5.1</td></tr>
<tr><td>| | :[NEXT]?</td><td></td><td>V2-21.8.5.2</td></tr>
</table>

KEYWORD	PARAMETER FORM	SCPI Reference
	:ENABle		V2-21.8.7
		:ADD	<numeric_list>	V2-21.8.7.1
		:DELete	<numeric_list>	V2-21.8.7.2
		[:LIST]	<numeric_list>	V2-21.8.7.3
	[:NEXT]?		V2-21.8.8
:LOCK		V2-21.14
	:OWNer?		V2-21.14.1
	:RELease		V2-21.14.2
	:REQuest?		V2-21.14.3
:TIME		V2-21.19
	:TIMer		V2-21.19.1
		:COUNt	<numeric_value>	V2-21.19.1.1
		[:STATe]	<Boolean>	V2-21.19.1.2
:VERSion?		V2-21.21

6.1.2.2 TRIGger subsystem

TRIGger commands for ETCELLs are as follows:

KEYWORD	PARAMETER FORM	
:TRIGger		V2-24.8
[:SEQuence]		V2-24.8.1
	:ECOunt	<numeric_value>	V2-24.8.1.7
	:LINK	<event_handle>	V2-24.8.1.13
	:SOURce	TIMer	V2-24.8.1.17
	:TIMer	<numeric_value>	V2-24.8.1.18

6.2 Additional Functionality

Not applicable.

6.3 Programming Examples

This section discusses how a user might use commands in an Emission Test Cell

6.3.1 Instrument Capability and Version Example

:SYSTem:CAPability? *A chassis dynamometer for an emissions test cell returns:*
(ETCELL & CDYNO)<NL>

:SYSTem:VERSion? *A chassis dynamometer might return:*
1999.0<NL>

6.3.2 Command Channel Example

Assume the Site computer wants to configure an Emission Test Cell instrument to perform a reading and return the data. The following commands might be issued to the instrument to perform this reading.

Commands

```
SENSE:FUNCtion:CONCurrent ON    Configure to sense data from
                                more than one analyzer.

SENSe:FUNCtion “CONC1:TAL”,"CONC2:TAL", “CONC3:TAL”,"CONC4:TAL"
                   Select the required data.
SENSe:CONCentration:RANGe:AUTO ON
                   Set instrument to automatic range mode.
INITiate           Start the measurement
SENSe:DATA?        Return data values for CONC1:TAL etc
```

By default, the result is returned in ASCII. The FORMat subsystem can be used to configure the instrument to send the data in various binary formats.

6.3.3 Two Channel Example

This example illustrates how to configure the instrument to send continuously acquired data on a separate channel from the SCPI command channel. This allows results to be continuously streamed back from the instrument to the controller.

Assume the controller wants the instrument to take data every 0.1 second, average the data over one second, buffer it then feed it out a second channel every 5 seconds.

```
Setup Sense functions to get modal data averaged over a second
SENSe:FUNCtion:CONCurrent ON
SENSe:FUNCtion “CONC1 ON AVERage”,"CONC2 ON AVERage",
“CONC3 ON AVERage”,"CONC4 ON AVERage"
Setup averaging parameter for the modal data
SENSe:AVERage:COUNt 10
SENSe:AVERage:TCONtrol REPeat
SENSe:AVERage:TYPE SCALAR
Turn the Averaging ON inside of the Sense block
SENSe:AVERage:STATe ON
Trigger Sequence will setup rates to sense the modal data
TRIGger:SEQuence1:SOURce TIMer
TRIGger:SEQuence1:TIMer 0.1      set sampling rate of 0.1sec
TRIGger:SEQuence2:LINK “TRIGger:SEQuence1"
TRIGger:SEQuence2:ECOunt 10      set integration rate
                                 10 times 0.1=1sec
TRIGger:SEQuence3:LINK ”TRIGger:SEQuence1"
TRIGger:SEQuence3:ECOunt 50      set transmission rate
                                 50 times 0.1=5sec
Setup and connect on the communication socket
SYStem:COMMunicate:SOCKet:ADDRess “123.45.67.89"
SYStem:COMMunicate:SOCKet:PORT 701
SYStem:COMMunicate:SOCKet:TYPE TCP
```

```
SYStem:COMMunicate:SOCKet:CONNect
Setup the Socket feed to data channel
SYStem:COMMunicate:SOCKet:FEED ”SYSTem:TIME:TEST","CONC1:TAL
ON AVERage".........
SYStem:COMMunicate:SOCKet:OCONdition “TRIGger:SEQuence2"
                    collects data into the socket every 1 sec
SYStem:COMMunicate:SOCKet:SCONdition ”TRIGger:SEQuence3"
                    sends data out the socket every 5 sec
```


Note: you can feed the data from two logical instruments into the same socket. Here for example, RT analyzer data and RE analyzer data can both be sent on SOCKet.

```
Initiate all sequences for continuous sampling
:INITiate:CONtinuous ON
The data is sent every 5 seconds on the data channel.
```

6.3.4 Returning Data using a Table Example

Assuming that the table MYTABLE has already been defined with items FORCe, SPEed,and TIME:

```
MEMory:TABLe:SELect MYTABLE     Select the table containing
                                the data.
MEMory:DATA? SPEed              Retrieve the speed data from the table.
```

6.3.5 Time Example

Standard SCPI time commands shall be used to synchronize each of the instrument clocks to the Site computer clock. This synchronization should be done during a period in the test when the instruments are in a quiescent state.

```
SYSTem:TIME 07,45,00.010
```

This command is used to control internal timer/counters in Emission Test Cell instruments. The Timer is used to timestamp data returned by Emission Test Cell instruments and to track time during an emission test. Set or reset the instrument timer to 0 at the beginning of a test.

```
SYSTem:TIME:TIMer:COUNt 0
Start the instrument timer.
SYSTem:TIME:TIMer:STATe ON
Query the value of the instrument timer.
SYSTem:TIME:TIMer:COUNt?
```

7 Power Supplies

A power supply is a basic sourcing instrument. It typically supplies a constant voltage or current at significant power levels to energize an electrical circuit. Because a power supply is primarily a source, the SOURce root node is optional.

All SCPI compliant products implement the commands listed in Syntax & Style section 4.2.

Figure 7-1 shows an instrument model for a power supply. It is derived from Command Reference, Chapter 2. It is essentially the same as Figure 2-1 in Command Reference with various parts shaded or dashed. The shaded boxes are not described here. While a power supply may have a MEMory subsystem, its operation is not described. The TRIGger block is dashed because not all power supplies have that functionality. While a power supply contains FORMat and Signal Routing functionality, their behaviors generally follow the *RST setting. These blocks are not described so they are dashed.

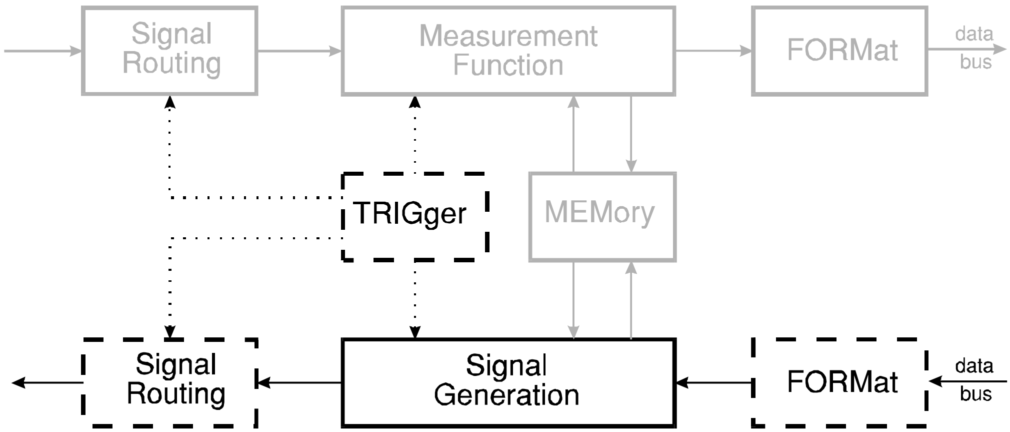


Figure 7-1 Simplified Model for a Power Supply

Figure 7-2 shows how the “Signal Generation” block in Figure 7-1 is expanded in a power supply. It is the same as Figure 2-5 in Command Reference with some shaded and dashed boxes. The MEMory block is shaded because while a power supply may contain this function, this chapter does not address it. The CALCulate block is dashed and its behavior not described. It behaves according to the *RST value.

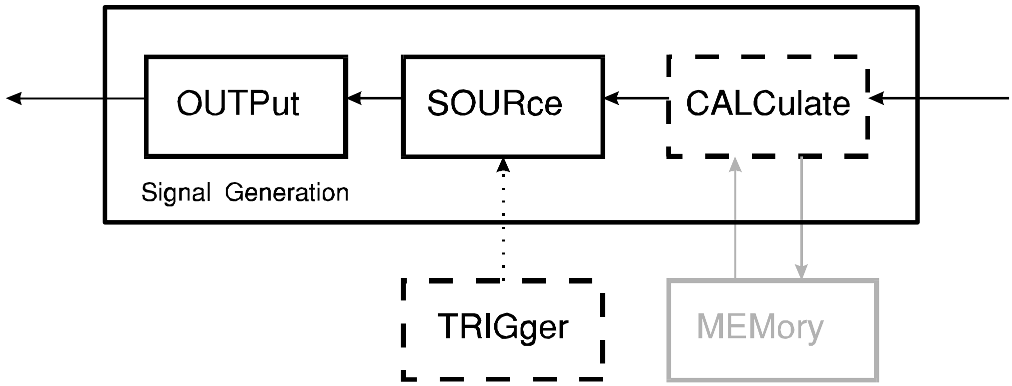


Figure 7-2 Expanded Signal Generation Model

7.1 Base Functionality

This section describes the base functionality for power supplies.

The <bf_keyword> for a DC power supply is DCPSUPPLY.

7.1.1 Base Measurement Instructions

The base functionality for a power supply contains no measurement instructions.

7.1.2 Base Device-oriented Functions

7.1.2.1 Outputs

A power supply shall have at least one output which can be turned on and off. This OUTPut subsystem command shall be implemented.

KEYWORD	PARAMETER FORM	SCPI Reference
OUTPut		Vol 2-15
[:STATe]	<Boolean>	Vol 2-15.12

As stated in Command Reference, *RST sets the output state to off.

7.1.2.2 SOURce subsystem

A power supply shall provide control of voltage and current by implementing these commands.

KEYWORD	PARAMETER FORM	SCPI Reference
[SOURce]		Vol 2-19
:CURRent		Vol 2-19.5
[:LEVel]		Vol 2-19.5.4
[:IMMediate]		Vol 2-19.5.4.1
[:AMPLitude]	<numeric_value>	Vol 2-19.5.4.1.1
:VOLTage		Vol 2-19.23
[:LEVel]		Vol 2-19.23.4
[:IMMediate]		Vol 2-19.23.4.1
[:AMPLitude]	<numeric_value>	Vol 2-19.23.4.1.1

A power supply can operate in one of two modes: voltage source or current source. The mode of operation depends on the load connected to it. When a power supply is operating as a voltage source, the :VOLTage command sets the output voltage and the :CURRent command sets the current limit. When a power supply is operating as a current source, the :VOLTage command sets the voltage limit and the :CURRent command sets the output current.

If the :CURRent and :VOLTage commmands are implemented as overlapped commands, the associated Pending-Operation flags shall be reported in the No-Operation-Pending flag. The operation started by the :CURRent command completes when the output current has reached the programmed current value or when the voltage limit has been reached. Likewise, the operation started by the :VOLTage command completes when the output voltage has reached the programmed voltage value or the current limit has been reached.

At *RST a power supply is required to set the voltage and current to “safe” conditions. This is generally achieved by setting them to their values closest to zero.

7.1.3 Base Status Reporting

All SCPI power supplies shall implement the status reporting structure described in Syntax & Style, Status Reporting. Command Reference, STATus Subsystem defines the commands which shall be used to control the status reporting structure.

7.1.3.1 QUEStionable

For a power supply, the bits of interest in the QUEStionable status structure are VOLTage and CURRent. When a power supply is operating as a voltage source, bit 1 (CURRent) shall be set. When a power supply is operating as a current source, bit 0 (VOLTage) shall be set. When the output is unregulated, both bits shall be set (for example, while the output is changing to a new programmed value).

7.1.3.2 OPERation

The base functionality of a power supply adds no requirements for this register.

7.2 Additional Functionality

This section describes additional functionality for power supplies.

7.2.1 Measurement capability

While the basic function of a power supply is to source voltage and current, this additional functionality provides for the measurement of the actual voltage and current delivered to the load.

The <af_keyword> for measurement capability is MEASURE.

7.2.1.1 MEASURE Measurement Instructions

The following MEASure:<function>? queries shall be implemented.

```
KEYWORD              PARAMETER FORM
MEASure
  [:SCALar]
     :VOLTage[:DC]?    [<expected_value>[,<resolution>]]
     :CURRent[:DC]?    [<expected_value>[,<resolution>]]
```

These queries read the actual voltage and or current at the power supply's output. They control a sensing function that is completely separate from the power supply itself. Executing these queries shall have no effect on the power supply's settings. This functionality further require no additional Measurement Instruction commands and has no interaction with the power supply's triggering system, if implemented.

The <expected_value> and <resolution> parameters are included solely for compatibility with the Command Reference. While parameters shall be accepted for syntactic compatibility, they are not required by MEASURE and may be ignored by the device.

7.2.1.2 MEASURE Device-oriented Functions

MEASURE adds no device-oriented functions.

7.2.1.3 MEASURE Status Reporting

MEASURE adds no Status Reporting requirements.

7.2.2 Multiple supplies

A power supply may contain multiple independent power supplies.

The <af_keyword> for multiple supplies is MULTIPLE.

7.2.2.1 MULTIPLE Measurement Instructions

MULTIPLE adds no measurement instructions.

7.2.2.2 MULTIPLE Device-oriented Functions

An instrument which contains multiple power supplies shall use the INSTrument:NSELect and INSTrument[:SELect] commands to switch among them.

KEYWORD	PARAMETER FORM	SCPI Reference
:INSTrument		Vol 2-12
:NSELect	<numeric_value>	Vol 2-12.5
[:SELect]	<identifier>	Vol 2-12.6

Valid values of the <numeric_value> shall range from 1 to the number of power supplies.

7.2.2.3 MULTIPLE Status Reporting

Section 9.5 of Syntax & Style describes the implications of multiple instruments on the status reporting model. The INSTrument Summary bits in the QUEStionable and OPERation registers shall be used to summarize the status from all logical instruments. The following STATus subsystem commands are required.

KEYWORD	PARAMETER FORM	SCPI Reference
STATus		Vol 2-20
:OPERation		Vol 2-20.1
:INSTrument		
:CONDition?		Vol 2-20.1.2
:ENABle	<NRf>	Vol 2-20.1.3
[:EVENt]?		Vol 2-20.1.4
:ISUMmary<n>		
:CONDition?		Vol 2-20.1.2
:ENABle	<NRf>	Vol 2-20.1.3
[:EVENt]?		Vol 2-20.1.4
:QUEStionable		Vol 2-20.3
:INSTrument		
:CONDition?		Vol 2-20.3.2
:ENABle	<NRf>	Vol 2-20.3.3
[:EVENt]?		Vol 2-20.3.4
:ISUMmary<n>		
:CONDition?		Vol 2-20.1.2
:ENABle	<NRf>	Vol 2-20.1.3
[:EVENt]?		Vol 2-20.1.4

7.2.3 Triggering Capability

Some power supplies are able to change their voltage and current based on the arrival of a command or external signal. Trigger commands, used in conjunction with source commands, control when the output changes.

The <af_keyword> for trigger capability is TRIGGER.

7.2.3.1 TRIGGER Measurement Instructions

TRIGGER adds no measurement instructions.

7.2.3.2 TRIGGER Device-oriented Functions

Power supplies which have TRIGGER additional functionality shall implement the following SOURce and TRIGger commands.

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:IMMediate]		Vol 2-24.7.2
[:ALL]		Vol 2-24.7.2.1
[SOURce]		Vol 2-19
:CURRent		Vol 2-19.5
[:LEVel]		Vol 2-19.5.4
:TRIGgered		Vol 2-19.5.4.2
[:AMPLitude]	<numeric_value>	Vol 2-19.5.4.2.1
:VOLTage		Vol 2-19.23
[:LEVel]		Vol 2-19.23.4
:TRIGgered		Vol 2-19.23.4.2
[:AMPLitude]	<numeric_value>	Vol 2-19.23.4.2.1
TRIGger		Vol 2-24.8
[:SEQuence]		Vol 2-24.8.1
[:IMMediate]		Vol 2-24.8.1.11
:SOURce	{BUS | EXTernal | ...}	Vol 2-24.8.1.17

The trigger commands associated with the TRIGGER additional functionality shall have no interaction with the MEASURE additional functionality.

Requiring BUS as a parameter to TRIGger:SOURce has the side effect of requiring DT1 capability in an IEEE 488.1 device. It also means the power supply implements the *TRG common command.

A VXIbus device with a trigger function is required to implement the *Trigger* command. When TRIGger:SOURce is set to BUS, a VXIbus power supply uses the *Trigger* command as the TRIGger source.

When TRIGger:SOURce is set to IMMediate, an INITiate commmand immediately transfers the TRIGgered[:AMPLitude] value to [:IMMediate][:AMPLitude].

7.2.3.3 TRIGGER Status Reporting

Bit 5 (Waiting for TRIG) in the OPERation Status Register shall be used to indicate the status of the trigger model.

7.3 Programming Examples

This section discusses how a user might program a power supply for certain applications.

7.3.1 Simple

To use a power supply as a constant voltage supply the programmer could send:

```
*RST
SOURce:VOLTage:LEVel:IMMediate:AMPLitude 5V
SOURce:CURRent:LEVel:IMMediate:AMPLitude MAXimum
OUTPut:STATe ON
```

or

```
*RST;VOLT 5V;CURR MAX;OUTP ON
```

7.3.2 Time Critical

With a power supply containing the TRIGGER additional functionality, an application could change the voltage at a specific time by sending:

```
*RST
SOURce:CURRent:LEVel:IMMediate:AMPLitude MAXimum
OUTPut:STATe ON
TRIGger:SEQuence:SOURce EXTernal
SOURce:VOLTage:LEVel:IMMediate:AMPLitude 7.2V
SOURce:VOLTage:LEVel:TRIGgered:AMPLitude 9.6V
INITiate:IMMediate
```

or

```
*RST;CURR MAX;OUTP ON;TRIG:SOUR EXT
VOLT:IMM 7.2V;TRIG 9.6V
INIT
```

The power supply sources 7.2 volts immediately and waits for a signal on the external trigger. When the trigger event occurs, the output voltage changes to 9.6 volts.

7.3.3 Level Verification

With a power supply containing the MEASURE additional functionality, an application could verify the level by sending:

```
*RST
OUTPut:STATe ON
SOURce:CURRent:LEVel:IMMediate:AMPLitude MAXimum
SOURce:VOLTage:LEVel:IMMediate:AMPLitude 7.2V
*OPC?
```

or

```
*RST
OUTP ON;CURR MAX;VOLT 7.2;*OPC?
```

The application reads the ‘1’ returned by the supply when it completes the operation of changing the level.

Now the actual voltage and current supplied to the load are measured with:

```
:MEASure:VOLTage:DC?;:MEASure:CURRent:DC?
```

or

```
MEAS:VOLT?;CURR?
```

The supply returns the measured voltage and current, which the application can then compare with expected results.

7.3.4 Multiple Supplies

A device containing two power supplies sets supply #1 to 5 Volts and supply #2 to -5 volts with the following commands:

```
*RST
INSTrument:NSELect 1
SOURce:VOLTage:LEVel:IMMediate:AMPLitude 5V
OUTPut:STATe ON
INSTrument:NSELect 2
SOURce:VOLTage:LEVel:IMMediate:AMPLitude -5V
OUTPut:STATe ON
```


or

```
*RST
INST:NSEL 1;VOLT 5;OUTP ON
INST:NSEL 2;VOLT -5;OUTP ON
```

To check that both supplies are acting as voltage sources and are not current limited, the QUEStionable Condition Register is queried using :

```
STATus:QUEStionable:CONDition?
```

If bit 0 is 0 and bit 1 is 1, both supplies are in the voltage source mode.

To check each supplies separately, the application subsequently reads the regulation status with these commands sent in the same program message:

```
STATus:QUEStionable:INSTrument:ISUMmary1:CONDition?;
:STATus:QUEStionable:INSTrument:ISUMmary2:CONDition?
```

or

```
STAT:QUES:INST:ISUM1:COND?;:STAT:QUES:INST:ISUM2:COND?
```

A possible response would be 2;3 indicating the CURRent bit but not the VOLTage bit has been set on supply #1 (current limited, voltage regulated) and both bits set on supply #2 (unregulated).

8 RF & Microwave Sources

RF sources and microwave sources are basic sourcing instruments. They typically generate a single sinusoidal signal at one frequency and at constant output level. The frequency is typically in the range from above 10 MHz up to several GHz. Their output impedance is normally 50 Ohms and matched impedance is assumed. The output level is specified as power into a matched load rather than as voltage. Signal generators with modulation and sweepers are also covered by this chapter. Their special attributes are described as additional functionalities.

Excluded from this chapter are optical sources, function generators, arbitrary waveform generators, pulse generators and data pattern generators.

All SCPI compliant products implement the commands listed in Syntax & Style section 4.2.

Figure 8-1 shows an instrument model for RF & microwave sources. It is derived from Command Reference, chapter 2. It is essentially the same as Figure 2-1 in Command Reference with various parts shaded or dashed. The bold boxes and lines, solid and dashed are described in this chapter.

8

The FORMat and ROUTe blocks are shown with dashed lines because many RF & microwave sources use their default settings. *RST sets FORMat[:DATA] to ASCii and ROUTe:TERMinals to FRONt. The shaded boxes are not described here.

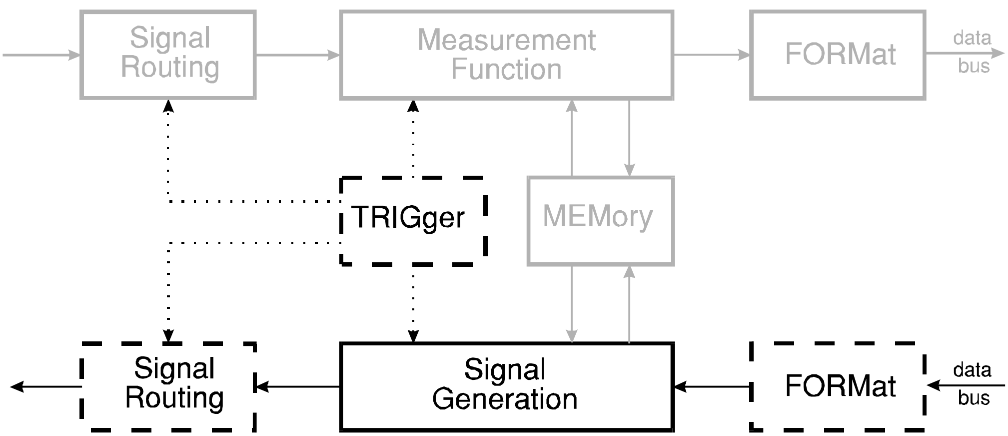


Figure 8-1 Simplified Model of an RF & Microwave Source

Figure 8-2 shows how the “Signal Generation” block in Figure 8-1 is expanded in RF & microwave sources. It is the same as Figure 2-5 in Command Reference with some dashed and shaded boxes. The CALCulate block is dashed because after *RST this block has no effect on the data. This chapter does not describe any CALCulate functions. The MEMory block is shaded because while RF & microwave sources may contain this function, this chapter does not address it.

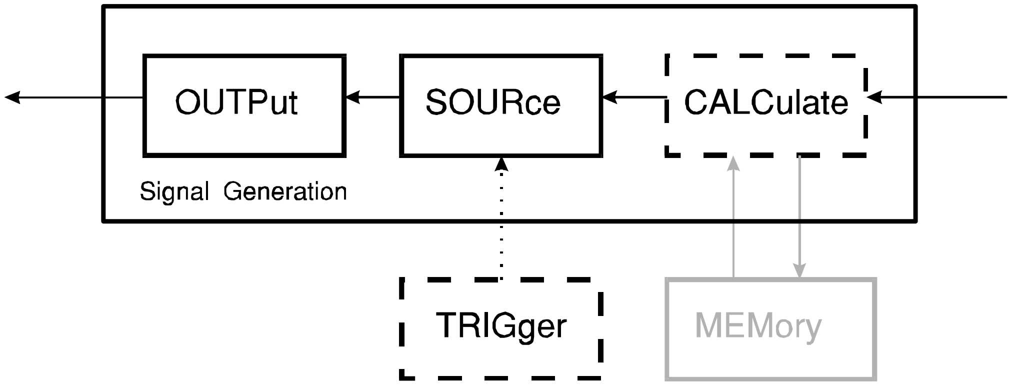


Figure 8-2 Expanded Signal Generation Model

8.1 Base Functionality

This section describes the base functionality for RF & microwave signal sources.

The <bf_keyword> for RF & microwave sources is RFSOURCE.

8.1.1 Base Measurement Instructions

The base functionality for RF & microwave sources contains no measurement instructions.

8.1.2 Base Device-oriented Functions

8.1.2.1 SOURce subsystem

RF & microwave sources shall provide control of its output frequency and level by implementing these commands.

KEYWORD	PARAMETER FORM	SCPI Reference
[SOURce]		Vol 2-19
:FREQuency		Vol 2-19.9
[:CW|:FIXed]	<numeric_value>	Vol 2-19.9.2
:POWer		Vol 2-19.15
:ALC		Vol 2-19.15.2
[:STATe]	<Boolean>	Vol 2-19.15.2.1
[:LEVel]		Vol 2-19.15.4
[:IMMediate]		Vol 2-19.15.4.1
[:AMPLitude]	<numeric_value>	Vol 2-19.15.4.1.1

If the :FREQuency and :POWer commands are implemented as overlapped commands, the associated Pending-Operation flags shall be reported in the No-Operation-Pending flag. The operation started by the :FREQuency command completes when the frequency has reached the programmed frequency value. Likewise, the operation started by the :POWer command completes when the output level has reached the programmed power value.

At *RST a RF & microwave source should be set to a “safe” condition. This is generally achieved by setting the amplitude to its minimum value in conjunction with *RST setting OUTPut:STATe to OFF.

8.1.2.2 OUTPut Subsystem

RF & microwave sources that have at least one output which can be turned on and off and shall implement the following command.

KEYWORD	PARAMETER FORM	SCPI Reference
OUTPut		Vol 2-15
[:STATe]	<Boolean>	Vol 2-15.12

As stated in Command Reference, *RST sets the output state to off.

8.1.2.3 UNIT Subsystem

RF & microwave sources shall provide control of its UNITs for the output power by implementing this command.

KEYWORD	PARAMETER FORM	SCPI Reference
UNIT		Vol 2-25
:POWer	W|V|DBM|DB<suffix mult.>W|...	Vol 2-25.2

At *RST, this value shall be set to DBM.

8.1.3 Base Status Reporting

All SCPI RF & microwave sources shall implement the status reporting structure described in Syntax & Style, Status Reporting. Command Reference, STATus Subsystem defines the commands which shall be used to control the status reporting structure.

8.1.3.1 QUEStionable Status

RF & microwave sources shall set bit 3 (POWer) or bit 5 (FREQuency) in the QUEStionable Status Register when the RF & microwave source suspects the power or the frequency is of questionable accuracy.

8.1.3.2 OPERation Status

The base functionality of a RF & microwave source adds no requirements to the OPERation Status register.

8.2 Additional functionality

The basic function of a RF & microwave source is to generate a signal at fixed frequency and power. Typical signal generators and typical sweepers may include additional functionality as modulation, sweep of frequency or power, marker and trigger function and selection of the reference oscillator. This section describes this additional functionality for RF & microwave sources.

8.2.1 Amplitude Modulation

This additional functionality provides amplitude modulation of the RF output signal delivered to the load either by an internally generated modulation signal or by an external signal input.

The <af_keyword> for amplitude modulation is AM.

8.2.1.1 AM Measurement Instructions

AM adds no Measurement Instructions.

8.2.1.2 AM Device Oriented Commands

RF & microwave sources which have the AM additional functionality shall provide control of amplitude modulation by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
[SOURce]		Vol 2-19
:AM		Vol 2-19.2
[:DEPTh]	<numeric_value>	Vol 2-19.2.2
:EXTernal		Vol 2-19.2.3
:COUPling	AC|DC|...	Vol 2-19.2.3.1
:INTernal		Vol 2-19.2.4
:FREQuency	<numeric_value>	Vol 2-19.2.4.1
:SOURce	EXTernal|INTernal	Vol 2-19.2.8
:STATe	<Boolean>	Vol 2-19.2.9

8.2.1.3 AM Status Reporting

When the RF & microwave source suspects the additional function AM is of questionable accuracy, bit 7 (MODulation) shall be set in the QUEStionable Status Register.

8.2.2 Frequency Modulation

This additional functionality provides frequency modulation of the RF output signal delivered to the load either by an internally generated modulation signal or by an external signal input.

The <af_keyword> for frequency modulation is FM.

8.2.2.1 FM Measurement Instructions

FM adds no Measurement Instructions.

8.2.2.2 FM Device Oriented Commands

RF & microwave sources which have the FM additional functionality shall provide control of frequency modulation by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
[SOURce]		Vol 2-19
:FM		Vol 2-19.7
[:DEViation]	<numeric_value>	Vol 2-19.7.2
:EXTernal		Vol 2-19.7.3
:COUPling	AC|DC|...	Vol 2-19.7.3.1
:INTernal		Vol 2-19.7.4
:FREQuency	<numeric_value>	Vol 2-19.7.4.1
:SOURce	EXTernal|INTernal	Vol 2-19.7.8
:STATe	<Boolean>	Vol 2-19.7.9

8.2.2.3 Status Reporting

When the RF & microwave source suspects the additional function FM is of questionable accuracy, bit 7 (MODulation) shall be set in the QUEStionable Status Register.

8.2.3 Pulse Modulation

This additional functionality provides pulse modulation of the RF output signal delivered to the load by an external modulation signal.

The <af_keyword> for pulse modulation is PULM.

8.2.3.1 PULM Measurement Instructions

PULM adds no Measurement Instructions.

8.2.3.2 PULM Device Oriented Commands

RF & microwave sources which have the PULM additional functionality shall provide control of pulse modulation by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
[SOURce]		Vol 2-19
:PULM		Vol 2-19.16
:POLarity	NORMal|INVerted	Vol 2-19.16.4
:STATe	<Boolean>	Vol 2-19.16.6

8.2.3.3 PULM Status Reporting

When the RF & microwave source suspects the additional functionality PULM is of questionable accuracy, bit 7 (MODulation) shall be set in the QUEStionable Status Register.

8.2.4 Analog Frequency Sweeping

RF & microwave sources may provide control of analog frequency sweep functions.

The <af_keyword> for analog frequency sweeping is FASWEEP.

8.2.4.1 FASWEEP Measurement Instructions

FASWEEP adds no Measurement Instructions.

8.2.4.2 FASWEEP Device Oriented Commands

RF & microwave sources which have the FASWEEP additional functionality shall provide control of frequency sweep by implementing the following commands.

8

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:ALL]		Vol 2-24.7.1.1
[SOURce]		Vol 2-19
:FREQuency		Vol 2-19.9
:MODE	CW|FIXed|SWEep|...	Vol 2-19.9.4
:STARt	<numeric_value>	Vol 2-19.9.9
:STOP	<numeric_value>	Vol 2-19.9.10
:CENTer	<numeric_value>	Vol 2-19.9.1
:SPAN	<numeric_value>	Vol 2-19.9.8
:SWEep		Vol 2-19.21
:TIME	<numeric_value>	Vol 2-19.21.1
:AUTO	<Boolean>|ONCE	Vol 2-19.21.1.1

See the beginning of the Command Refernce, SOURce chapter for a description of the coupling among STARt, STOP, CENTer & SPAN.

Typically a RF & microwave source has just one single SWEEP source for frequency and power sweep. The setting of sweep parameters are then coupled and identical. If there is more than one sweep source they can be distinguished by use of a numeric suffix.

When the mode is changed from CW to SWEep, the instrument is not sweeping until INITiate:CONTinuous is set to ON.

8.2.4.3 FASWEEP Status Reporting

If the additional function Analog Frequency Sweep is implemented and the RF & microwave source is performing this operation, bit 3 (SWEeping) shall be set in the OPERation Status Register.

8.2.5 Stepped Frequency Sweeping

RF & microwave sources may provide control of stepped frequency sweep functions.

The <af_keyword> for stepped frequency sweeping is FSSWEEP.

8.2.5.1 FSSWEEP Measurement Instructions

FSSWEEP adds no Measurement Instructions.

8.2.5.2 FSSWEEP Device Oriented Commands

RF & microwave sources which have the FSSWEEP additional functionality shall provide control of frequency sweep by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:ALL]		Vol 2-24.7.1.1
[SOURce]		Vol 2-19
:FREQuency		Vol 2-19.9
:MODE	CW|FIXed|SWEep|...	Vol 2-19.9.4

:STARt	<numeric_value>	Vol 2-19.9.9
:STOP	<numeric_value>	Vol 2-19.9.10
:CENTer	<numeric_value>	Vol 2-19.9.1
:SPAN	<numeric_value>	Vol 2-19.9.8
:SWEep		Vol 2-19.21
:DWELl	<numeric_value>	Vol 2-19.21.2
:STEP	<numeric_value>	Vol 2-19.21.7

See the beginning of the Command Reference, SOURce chapter for a description of the coupling among STARt, STOP, CENTer & SPAN.

Typically a RF & microwave source has just one single SWEEP source for frequency and power sweep. The setting of sweep parameters are then coupled and identical. If there is more than one sweep source they can be distinguished by use of a numeric suffix.

When the mode is changed from CW to SWEep, the instrument is not sweeping until INITiate:CONTinuous is set to ON.

8

8.2.5.3 FSSWEEP Status Reporting

If the additional function Stepped Frequency Sweep is implemented and the RF & microwave source is performing this operation, bit 3 (SWEeping) shall be set in the OPERation Status Register.

8.2.6 Analog Power Sweeping

RF & microwave sources may provide control of analog power sweep functions.

The <af_keyword> for analog power sweeping is PASWEEP.

8.2.6.1 PASWEEP Measurement Instructions

PASWEEP adds no Measurement Instructions.

8.2.6.2 PASWEEP Device Oriented Commands

RF & microwave sources which have the PASWEEP additional functionality shall provide control of analog power sweep by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:ALL]		Vol 2-24.7.1.1
[SOURce]		Vol 2-19
:POWer		Vol 2-19.15
:MODE	FIXed|SWEep	Vol 2-19.15.7
:STARt	<numeric value>	Vol 2-19.15.13
:STOP	<numeric value>	Vol 2-19.15.14
:CENTer	<numeric value>	Vol 2-19.15.3
:SPAN	<numeric value>	Vol 2-19.15.12
:SWEep		Vol 2-19.21
:TIME	<numeric value>	Vol 2-19.21.1
:AUTO	<Boolean>|ONCE	Vol 2-19.21.1.1

See the beginning of the Command Reference, SOURce chapter for a description of the coupling among STARt, STOP, CENTer & SPAN.

Typically a RF & microwave source has just one single SWEEP source for frequency and power sweep. The setting of sweep parameters are then coupled and identical. If there is more than one sweep source they can be distinguished by use of a numeric suffix.

When the mode is changed from CW to SWEep, the instrument is not sweeping until INITiate:CONTinuous is set to ON.

8.2.6.3 PASWEEP Status Reporting

If the additional function Analog Power Sweep is implemented and the RF & microwave source is performing this operation, bit 3 (SWEeping) shall be set in the OPERation Status Register.

8.2.7 Stepped Power Sweeping

RF & microwave sources may provide control of stepped power sweep functions.

The <af_keyword> for stepped power sweeping is PSSWEEP.

8.2.7.1 PSSWEEP Measurement Instructions

PSSWEEP adds no Measurement Instructions.

8.2.7.2 PSSWEEP Device Oriented Commands

RF & microwave sources which have the PSSWEEP additional functionality shall provide control of stepped power sweep by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:ALL]		Vol 2-24.7.1.1
[SOURce]		Vol 2-19
:POWer		Vol 2-19.15
:MODE	FIXed|SWEep	Vol 2-19.15.7
:STARt	<numeric value>	Vol 2-19.15.13
:STOP	<numeric value>	Vol 2-19.15.14
:CENTer	<numeric value>	Vol 2-19.15.3
:SPAN	<numeric value>	Vol 2-19.15.12
:SWEep		Vol 2-19.21
:DWELl	<numeric_value>	Vol 2-19.21.2
:STEP	<numeric_value>	Vol 2-19.21.7

See the beginning of the Command Reference, SOURce chapter for a description of the coupling among STARt, STOP, CENTer & SPAN.

Typically a RF & microwave source has just one single SWEEP source for frequency and power sweep. The setting of sweep parameters are then coupled and identical. If there is more than one sweep source they can be distinguished by use of a numeric suffix.

When the mode is changed from CW to SWEep, the instrument is not sweeping until INITiate:CONTinuous is set to ON.

8.2.7.3 PSSWEEP Status Reporting

If the additional function Stepped Power Sweep is implemented and the signal generator is performing this operation, bit 3 (SWEeping) shall be set in the OPERation Status Register.

8.2.8 Frequency List

RF & microwave sources may provide control of additional frequency sweep functions by a frequency list containing the sequence of frequencies to be stepped through.

The <af_keyword> for frequency lists is FLIST.

8.2.8.1 FLIST Measurement Instructions

FLIST adds no Measurement Instructions.

8.2.8.2 FLIST Device Oriented Commands

RF & microwave sources which have the FLIST additional functionality shall provide control of frequency list by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:ALL]		Vol 2-24.7.1.1
[SOURce]		Vol 2-19
:FREQuency		Vol 2-19.9
:MODE	CW|FIXed|LIST|...	Vol 2-19.9.4
:LIST		Vol 2-19.11
:DWELl	<numeric value>	Vol 2-19.11.8
:POINts?	query only	Vol 2-19.11.8.1
:FREQuency	<numeric_value>{,<numeric_value>}	Vol 2-19.11.9
:POINts?	query only	Vol 2-19.11.9.1

This functionality only requires one parameter for DWELl, however the instrument may accept more than one.

8.2.8.3 FLIST Status Reporting

FLIST adds no requirement for status reporting.

8.2.9 Marker Function

RF & microwave sources may provide control of marker functions.

The <af_keyword> for marker function is MARKER.

8.2.9.1 MARKER Measurement Instructions

MARKER adds no Measurement Instructions.

8.2.9.2 MARKER Device Oriented Commands

RF & microwave sources which have the MARKER additional functionality shall provide control of markers by implementing the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
[SOURce]		Vol 2-19
:MARKer<n>		Vol 2-19.12
:AOFF		Vol 2-19.12.2
:FREQuency	<numeric_value>	Vol 2-19.12.3
[:STATe]	<Boolean>	Vol 2-19.12.7

RF & microwave sources may have more than one Marker. In this case a numeric suffix shall be used to distinguish between the different markers.

8.2.9.3 MARKER Status Reporting

MARKER adds no requirement for status reporting.

8.2.10 Trigger Function

RF & microwave sources uses the TRIGger Subsystem to control the start of the sweep or list function if the frequency and/or the amplitude is swept. When a trigger is received, the RF & microwave source starts and completes a sweep or list sequence and then waits for the next trigger event.

The <af_keyword> for trigger function is TRIGGER.

8.2.10.1 TRIGGER Measurement Instructions

TRIGGER adds no Measurement Instructions.

8.2.10.2 TRIGGER Device Oriented Commands

RF & microwave sources which have the TRIGGER additional functionality shall implement the following trigger commands.

KEYWORD	PARAMETER FORM	SCPI Reference
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:ALL]		Vol 2-24.7.1.1
[:IMMediate]		Vol 2-24.7.2
[:ALL]		Vol 2-24.7.2.1
ABORt		Vol 2-24.5
TRIGger		Vol 2-24.8
[:SEQuence]		Vol 2-24.8.1
[:IMMediate]		Vol 2-24.8.1.11
:SOURce	BUS | IMMediate | ... †	Vol 2-24.8.1.17

† In addition to the TRIGger:SOURce parameters listed, a RF & microwave source with FSSWEEP, PSSWEEP or FLIST shall implement an external trigger, such as EXTernal, ECLTrg<n> or TTLTrg<n>.

Requiring BUS as a legal parameter to TRIGger:SOURce has the side effect of requiring DT1 capability in a 488.1 device. It also means the RF & microwave source implements the *TRG common command.

A VXIbus device with a trigger function is required to implement the *Trigger* command. When TRIGger:SOURce is set BUS, a VXIbus RF & microwave source shall use the *Trigger* command as the TRIGger source.

8.2.10.3 TRIGGER Status Reporting

If the additional function TRIGGER is implemented and the RF & microwave source is performing this operation, bit 5 (Waiting for TRIGger) shall be set in the OPERation Status Register.

8.2.11 Reference Oscillator

RF & microwave sources may have the functionality to select which reference oscillator is in use and to define the frequency of the external reference oscillator.

The <af_keyword> for reference oscillator function is REFERENCE.

8.2.11.1 REFERENCE Measurement Instructions

REFERENCE adds no Measurement Instructions.

8.2.11.2 REFERENCE Device Oriented Commands

RF & microwave sources which have the additional functionality to select the reference oscillator shall implement the following commands.

KEYWORD	PARAMETER FORM	SCPI Reference
[:SOURce]		Vol 2-19
:ROSCillator		Vol 2-19.19
:EXTernal		Vol 2-19.19.2
:FREQuency	<numeric_value>	Vol 2-19.19.2.1
:SOURce	INTernal|EXTernal	Vol 2-19.19.3

8.2.11.3 REFERENCE Status Reporting

RF & microwave sources shall set bit 5 (FREQuency) in the QUEStionable Status Register when there is no appropriate reference signal available and the RF & microwave source suspects the output frequency is of questionable accuracy.

8.3 Programming Examples

This section discusses how a user might program RF & microwave sources for certain applications.

8.3.1 Simple

To use RF & microwave sources as a simple signal source for fixed frequency and output level the programmer could send:

```
*RST
SOURce:CW:FREQuency 123.45 MHZ
SOURce:POWer:LEVel:IMMediate:AMPLitude -27 DBM
OUTput:STATe ON
```

or

```
*RST
FREQ 123.45MHZ;POW -27DBM;OUTP ON
```


8.3.2 Modulation

To use a RF & microwave source having the functionality to generate a frequency modulated signal by use of the internal modulation source the programmer could send:

```
*RST
SOURce:CW:FREQuency 89.5 MHZ
SOURce:FM:DEViation 200 KHZ
SOURce:FM:SOURce INTernal
SOURce:FM:INTernal:FREQuency 1 KHZ
SOURce:POWer:LEVel:IMMediate:AMPLitude 100 UW
OUTPut:STATe ON
```

or

```
*RST
FREQ 89.5MHZ
FM 200KHZ;FM:SOUR INT;INT:FREQ 1KHZ
POW 100UW;OUTP ON
```

8.3.3 Analog Sweep

To use a RF & microwave source having the functionality to sweep the frequency with setting of start/stop frequency and sweep time the programmer could send:

```
*RST
SOURce:FREQuency:MODE SWEep
SOURce:FREQuency:STARt 100 MHZ
SOURce:FREQuency:STOP  200 MHZ
SOURce:SWEep:TIME 100 MS
SOURce:POWer:LEVel:IMMediate:AMPLitude -27 DBM
OUTPut:STATe ON
INITiate:CONTinuous ON
```

or

```
*RST
FREQ:MODE SWE
FREQ:STAR 100MHZ;STOP 200MHZ
SWE:TIME 100MS
POW -27 DBM;OUTP ON
INIT:CONT ON
```

8.3.4 Triggered Analog Sweep

To use a RF & microwave source as a analog sweeper in triggered single sweep mode the programmer could send:

```
*RST
SOURce:FREQuency:MODE SWEep
SOURce:FREQuency:STARt 100 MHZ
SOURce:FREQuency:STOP 200 MHZ
SOURce:SWEep:TIME 100 MS
SOURce:POWer:LEVel:IMMediate:AMPLitude -27 DBM
OUTPut:STATe ON
TRIGger:SOURce EXTernal
INITiate:IMMediate
```

or

```
*RST
FREQ:MODE SWE
FREQ:STAR 100MHZ;STOP 200MHZ
SWE:TIME 100MS
POW -27 DBM;OUTP ON
TRIG:SOUR EXT
INIT
```

Note: The SOURCE:SWEEP:COUNT and SOURCE:LIST:COUNT are set to 1 after *RST i.e. the sweeper will perform single sweep function after occurrence of an external trigger signal and then waits for the next trigger event.

8.3.5 Sweep with Marker

To use a sweeper with setting of two markers the programmer could add the following commands to example 8.3.3:

```
SOURce:MARKer1:FREQuency 125 MHZ
SOURce:MARKer2:FREQuency 170 MHZ
```

or

```
MARK1:FREQ 125MHZ;MARK2:FREQ 170MHZ
```

8.3.6 Reference oscillator

To use a RF & microwave source with its output frequency derived from an external reference frequency the programmer could send:

```
*RST
SOURce:ROSCillator:EXTernal 10 MHZ
SOURce:CW:FREQuency 2355.5 MHZ
SOURce:POWer:LEVel:IMMediate:AMPLitude -6 DBM
OUTPut:STATe ON
```

or

```
*RST
ROSC:EXT 10MHZ
FREQ 2355.5MHZ;POW -6DBM;OUTP ON
```

9 Signal Switchers

A signal switcher is a basic signal routing instrument. In its most basic form, it can make and break signal connections. These signal connections are usually referred to as channels. Making a connection is closing a channel, and breaking a signal is opening a channel. More advanced signal switchers can be programmed with sequence lists of channels to close in response to trigger events. Because a signal switcher is primarily a signal router, the ROUTe root node is optional.

Figure 9-1 shows an instrument model for a signal switcher. It is derived from Command Reference, chapter 2. It is essentially the same as Figure 2-1 in Command Reference with various parts gray or dashed. The bold boxes and lines, solid and dashed, are described in this chapter.

All SCPI compliant products implement the commands listed in Syntax & Style section 4.2.

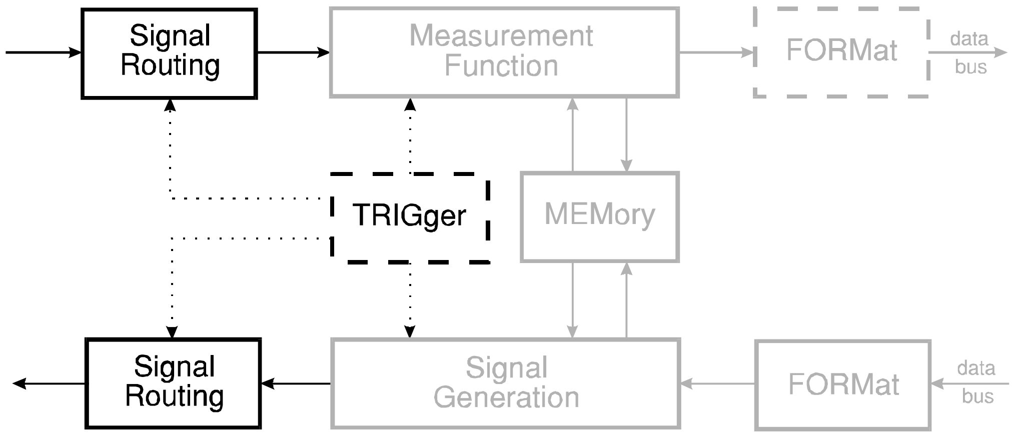


Figure 9-1 Simplified Model for a Progammable Signal Switcher

9.1 Base Functionality

The <bf_keyword> for a signal switcher is SWITCHER.

9.1.1 Base Measurement Instructions

Since a signal switcher has no measuring capability, it has no measurement instructions.

9.1.2 Base Device-oriented Functions

9.1.2.1 ROUTe Subsystem

A signal switcher can at least make and break signal connections by closing and opening channels.

The following commands shall be implemented.

KEYWORD	PARAMETER FORM	SCPI Reference
[ROUTe]		Vol 2-17
:CLOSe	<channel_list>	Vol 2-17.1
:STATe?		Vol 2-17.1.1
:OPEN	<channel_list>	Vol 2-17.3
:ALL		Vol 2-17.3.1

If the :CLOSe and :OPEN commmands are implemented as overlapped commands, the associated Pending-Operation flags shall be reported in the No-Operation-Pending flag.

9.1.3 Base Status Reporting

All SCPI signal switchers shall implement the status reporting structure described in Syntax & Style, Status Reporting. Command Reference, STATus Subsystem defines the commands which shall be used to control the status reporting structure.

The base functionality of a switcher adds no additional status reporting requirements.

9.2 Additional Functionality

9.2.1 Scanning

The <af_keyword> for scanning is SCAN.

Signal switchers which implement the SCAN additional functionality can be programmed with a list of channels to “scan”. Each time the signal switcher recognizes a trigger event, it opens the previously closed channel and closes the next channel in the list.

9.2.1.1 SCAN Measurement Instructions

SCAN adds no measurement instructions.

9.2.1.2 SCAN Device-oriented Functions

9.2.1.2.1 ROUTe Subsystem

A signal switcher which has the SCAN additional functionality shall implement the following command.

KEYWORD	PARAMETER FORM	SCPI Reference
[ROUTe]		Vol 2-17
:SCAN	<channel_list>	Vol 2-17.6

Note: The following paragraph is intended to provide guidance to designers of signal switchers. It is not intended to impose further design requirements.

The appropriate interaction between the SCAN command and the CLOSe and OPEN commands depends upon the intended application of the signal switcher. In some instances it is advantageous to be able to close a set of channels using the CLOSe command (e.g. to connect sourcing instruments to the device under test), then scan through a different set of channels (e.g. to make measurements on various nodes in the device under test) without opening those channels which were closed by the CLOSe command. In other instances it is desirable that the start of the scan operation first opens any channels which are already closed.

9.2.1.2.2 TRIGger Subsystem

A signal switcher which has the SCAN additional functionality shall implement the following trigger commands.

KEYWORD	PARAMETER FORM	SCPI Reference
ABORt		Vol 2-24.5
INITiate		Vol 2-24.7
:CONTinuous	<Boolean>	Vol 2-24.7.1
[:IMMediate]		Vol 2-24.7.2
[:ALL]		Vol 2-24.7.2.1
TRIGger		Vol 2-24.8
[:SEQuence]		Vol 2-24.8.1
:COUNt	<numeric_value>	Vol 2-24.8.1.2
:SOURce	BUS	IMMediate	... †	Vol 2-24.8.1.17

† In addition to the TRIGger:SOURce parameters listed, a signal switcher with SCAN shall implement an external trigger, such as EXTernal, ECLTrg<n> or TTLTrg<n>.

Requiring BUS as a legal parameter to TRIGger:SOURce has the side effect of requiring DT1 capability in a 488.1 device. It also means the signal switcher implements the *TRG common command.

A VXIbus device with a trigger function is required to implement the *Trigger* command. When TRIGger:SOURce is set BUS, a VXIbus switcher shall use the *Trigger* command as the TRIGger source.

9.2.1.3 SCAN Status Reporting

Bit 5 (Waiting for TRIG) in the OPERation Status Register shall be used to indicate the status of the trigger model.

9.2.2 Extended Trigger

The <af_keyword> for extended trigger is ETRIGGER. This <af_keyword> is subservient to SCAN.

9

Some signal switchers which have SCAN functionality (and so have TRIGger commands) provide two layer triggering where a separate event must occur before the TRIGger layer is entered.

9.2.2.1 ETRIGGER Measurement Instructions

ETRIGGER adds no Measurement Instructions.

9.2.2.2 ETRIGGER Device-oriented Functions

9.2.2.2.1 TRIGger Subsystem

The following ARM command shall be used to provide the ETRIGGER functionality.

KEYWORD	PARAMETER FORM	SCPI Reference
ARM		Vol 2-24.6
[:SEQuence]		Vol 2-24.6.1
[:LAYer]		Vol 2-24.6.1.2
:SOURce	{BUS | IMMediate | ... }	Vol 2-24.6.1.2.14

9.2.2.3 ETRIGGER Status Reporting

Bit 6 (Waiting for ARM) in the OPERation Status Register shall be used to indicate the status of the trigger model.

9.3 Programming Examples

This section discusses how a user might program a signal switcher for certain applications. The examples assume the signal switcher contains a single switch module which is a 10-channel multiplexer which permits any combination of the 10 channels to be closed simultaneously.

9.3.1 Making and Breaking Connections

To close signal channel #3:

```
ROUTe:CLOSe (@3)
```

or

```
CLOS (@3)
```

To query whether channel #3 is closed:

```
ROUTe:CLOSe? (@3)
```

or

```
CLOS? (@3)
```


9.3.2 Programmed Connection Sequences

A signal switcher which can be programmed with a sequence of connections can improve the throughput of test systems by reducing the communication required between the system controller and the test instruments. For example, the output of the 10-channel multiplexer which has SCAN functionality, and includes EXTernal as a valid parameter for TRIGger:SOURce, is connected to the inputs of a DMM. The "meter complete" signal from the DMM is connected to the external trigger input of the signal switcher. To configure this system to measure on all 10 channels, use this sequence of commands to the signal switcher:

```
*RST
TRIGger:SEQuence:COUNt 9;SOURce EXTernal
ROUTe:SCAN (@2,3,4,5,6,7,8,9,10);CLOSe (@1)
INITiate:IMMediate
```

or

```
*RST
TRIG:COUN 9;SOUR EXT
SCAN (@2:10);CLOS (@1)
INIT
```

The *RST command sets INITiate:CONTinuous to OFF and puts the trigger model in IDLE. After sending the INITiate:IMMediate command, the system controller triggers the DMM. When the meter is finished measuring, its "meter complete" signal activate the signal switcher's external trigger input, causing it to close channel 2. Now the system controller triggers the DMM again, and so on. When the signal switcher has finished processing the 9th trigger (closing channel 10) it returns to the trigger model IDLE layer.