4041 Series Spectrum Analyzer

Programming Manual

China Electronics Technology Instruments Co.,Ltd

VERIFICATION-ПРОВЕРКА-検証-VÉRIFICATION-검증-التحقق

Verification of Conformity

Certificate No.	:	CTL1803263043-EC
Applicant	:	China Electronic Technology Instruments Co.,Ltd
Address	:	NO.98, Xiangjiang Road, Huangdao District, Qingdao, China.
Product	:	SPECTRUM ANALYZER
Trademark	:	Ceyear
Model(s)	:	4041
Manufacturer	:	China Electronic Technology Instruments Co.,Ltd
Address	:	NO.98, Xiangjiang Road, Huangdao District, Qingdao, China.
Test Report	:	CTL1803263043-E

Complies with the requirements of the
EC EMC directive 2014/30/EU with amendments .
Test Standards:

EN 61326-1: 2013
EN 61000-3-2: 2014
EN 61000-3-3: 2013

Remarks:
Based on the voluntary assessment of the product sample and technical file, we confirm that the above-mentioned product meets the requirements of the EC directive.
The CE mark as show below can be used, under the responsibility of the manufacturer or the importer, after completion of an EC declaration of conformity and compliance with all relevant EC directives

CE

For Chief Executive
Jul. 30, 2018

Shenzhen CTL Testing Technology Co., Ltd.
Add:Floor 1-A,Baisha Technology Park,No.3011,Shahexi Road,Nanshan District,Shenzhen,China 518055
Tel:(86)0755-89486194 Web:www.ctl-lab.com E-mail:ctl@ctl-lab.com

VERIFICATION-ПРОВЕРКА-検証-VÉRIFICATION-검증-التحقق

Verification of Conformity

Certificate No.	:	CTL1803263043-SC
Applicant	:	China Electronics Technology Instruments Co.,Ltd
Address	:	NO.98, Xiangjiang Road, Huangdao District, Qingdao, China
Product	:	SPECTRUM ANALYZER
Trademark	:	Ceyear
Model(s)	:	4041
Manufacturer	:	China Electronics Technology Instruments Co.,Ltd
Address	:	NO.98, Xiangjiang Road, Huangdao District, Qingdao, China
Test Report	:	CTL1803263043-S

Complies with the requirements of the
EC LVD directive 2014/35/EU
Test Standards:

EN 61010-1:2010

Remarks:
Based on the voluntary assessment of the product sample and technical file, we confirm that the above-mentioned product meets the requirements of the EC directive.
The CE mark as show below can be used, under the responsibility of the manufacturer or the importer, after completion of an EC declaration of conformity and compliance with all relevant EC directives.

CE

For Chief Executive
April 25, 2018

Shenzhen CTL Testing Technology Co., Ltd.
Add:Floor 1-A,Baisha Technology Park,No.3011,Shahexi Road,Nanshan District,Shenzhen,China 518055
Tel:(86)0755-89486194 Web:www.ctl-lab.com E-mail:ctl@ctl-lab.com

VERIFICATION-ПРОВЕРКА-檢証-VÉRIFICATION-검증-التحقق

Verification of Conformity

Certificate No.	:	CTL1803263043-RC
Applicant	:	China Electronic Technology Instruments Co.,Ltd
Address	:	NO.98, Xiangjiang Road, Huangdao District, Qingdao, China.
Product	:	SPECTRUM ANALYZER
Trademark	:	Ceyear
Model(s)	:	4041
Manufacturer	:	China Electronic Technology Instruments Co.,Ltd
Address	:	NO.98, Xiangjiang Road, Huangdao District, Qingdao, China.
Test Report	:	CTL1803263043-R

Complies with the requirements of the
EC RoHS Directive 2011/65/EU
Test Standards:

IEC 62321-7-2:2017
IEC 62321-4:2013
IEC 62321-5:2013
IEC 62321-6:2015

Remarks:
Based on the voluntary assessment of the product sample and technical file, we confirm that the above-mentioned product meets the requirements of the EC directive.
The CE mark as show below can be used, under the responsibility of the manufacturer or the importer, after completion of an EC declaration of conformity and compliance with all relevant EC directives.

CE

For Chief Executive
Jul. 30, 2018

Shenzhen CTL Testing Technology Co., Ltd.
Add:Floor 1-A,Baisha Technology Park,No.3011,Shahexi Road,Nanshan District,Shenzhen,China 518055
Tel:(86)0755-89486194 Web:www.ctl-lab.com E-mail:ctl@ctl-lab.com

Foreword

Thank you for choosing the 4041 series spectrum analyzer developed and produced by China Electronics Technology Instruments Co.,Ltd. This product features high measurement accuracy, high speed and cost performance, as well as various interface options. Please read this Manual carefully for your convenience.

We devote ourselves to meeting customer’s demands, providing customers with high-quality measuring instrument and the best after-sales service. We persist with "superior quality and considerate service", and are committed to offering satisfactory products and service for our customers. If you have any questions, please don't hesitate to call us:

Tel:	**+86-0532-86896691**
Website:	**www.ceyear.com**
E-mail:	**sales@ceyear.com**
Add.:	**NO. 98 Xiangjiang Rd., Qingdao City, China**
Zip code:	**266555**

This Manual mainly describes the programming methods of 4041 series spectrum analyzer produced by China Electronics Technology Instruments Co.,Ltd, including how to realize programming control of this device using an externally controlled computer through LAN interface or USB interface, thus helping you quickly understand and master the programing control methods and commands of this device.

We strive for continuous improvement to the performance of our instruments and products by persistently upgrading and improving the hardware and firmware supplied by us. Therefore, the instructions to use and control the instrument described in this manual maybe updated from time to time. To receive the latest updates of this manual, please contact our Technical Support Department.

Due to the limit of time and knowledge of the writer, there may be some omission or error, and you are welcome to correct any such error. We deeply apologize for the mistakes which may cause your inconvenience.

This is the Third version of 4041 Series Spectrum Analyzer Programming Manual with the version number being AV2.731.1075SCCN/A.3 and is suitable for host software version 2.1.40 and above.

Compiler

Aug. 2018

Table of Contents

Chapter I Introduction to SCPI Commands

Section I Introduction of SCPI command operation

SCPI (Standard Commands for Programmable Instruments) is a new command language for instrument control developed in accordance with IEEE488.2 standard. The main purpose of SCPI is to ensure that same program control commands are applicable to the same type of instrument, so as to realize standardization of program control commands.

This chapter contains the details of all SCPI commands identified and executed by the 4041 series spectrum analyzer. Introduction and description of IEEE488.2 general commands and measurement commands are also included in this chapter.

Each measurement command has its applicable mode. Under the context when the mode is not applicable, an error identification will be returned for query command.

Section II IEEE 488.2 commands

*CLS – Clear the state

Clear the state of the instrument, i.e.: Clear wrong queue and all event registers. And clear all *OPC commands and query commands to be processed.

*IDN? - Identification

Return the unique identification character string of the device and the identification varies with models. For example: “CEYEAR, 4041, S/N, 1.00”.

*OPC - Operation complete command

After completing all overlapping commands to be processed (for example: One sweep or Default command, etc.), set the OPC bit of standard event status register.

*OPC? - Operation complete query

Number “1” will be returned when all overlapping commands to be processed are completed.

*RST - Reset

Perform reset operation and cancel all *OPC commands or query commands to be processed. The contents in the nonvolatile memory of the instrument will not be lost.

*WAI - Wait

The instrument will process all overlapping commands before processing new commands.

Section III Measurement commands

:CALCulate[:SELected]:LIMit:BEEP <bool>

(Read and write) query or set limit beep on/off.

Applicable mode Spectrum Analysis, Power Meter, Field Strength

Parameter Beep on/off.

OFF(0)	Beep off
ON(1)	Beep on

Example :CALC:LIM:BEEP ON

Query syntax :CALC:LIM:BEEP?

Default OFF

Return type Numeric value (bool) or character

:CALCulate[:SELected]:LIMit:LOWer:DISPlay <bool>

(Read and write) query or set lower limit display on/off.

Applicable mode Spectrum Analysis

Parameter Lower limit display on/off.

OFF(0)	Display off
ON(1)	Display on

Example :CALC:LIM:LOW:DISP ON

Query syntax :CALC:LIM:LOW:DISP?

Default OFF

Return type Numeric value (bool) or character

:CALCulate[:SELected]:LIMit:LOWer:MARGin <num>

(Read and write) query or set lower limit margin.

Applicable mode Spectrum Analysis

Parameter Lower limit margin.

Parameter range: 0~40 Unit: dBm

Example :CALC:LIM:LOW:MARG 10

Query syntax :CALC:LIM:LOW:MARG?

Default 0

Return type Numeric value (float) or character

:CALCulate[:SELected]:LIMit:LOWer:TEST <bool>

(Read and write) query or set lower limit test on/off.

Applicable mode Spectrum Analysis

Parameter Lower limit test on/off.

OFF(0)	Test off
ON(1)	Test on

Example :CALC:LIM:LOW:TEST ON

Query syntax :CALC:LIM:LOW:TEST?

Default OFF

Return type Numeric value (bool) or character

:CALCulate[:SELected]:LIMit:LOWer:EDIT:ADD

(Write only) add default point to lower limit, and the default setting of default point is: limit

frequency: 44.1GHz, limit value: 0dBm.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:LOW:EDIT:ADD
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:LIMit:LOWer:EDIT:DELete

(Write only) delete current point in lower limit.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:UPP:EDIT:DEL
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:LIMit:LOWer:EDIT:CLEar

(Write only) delete all edit points in lower limit.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:LOW:EDIT:CLE
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:LIMit:LOWer:EDIT:DATA

(Write only) lower limit edit point.

Applicable mode	Spectrum Analysis
Parameter	Point index (int, start from 0), limit frequency (0~44.1GHz), limit value (-174dBm~50dBm)
Example	:CALC:LIM:LOW:EDIT:DATA
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:LIMit:UPPer:DISPlay <bool>

(Read and write) query or set upper limit display on/off.

Applicable mode	Spectrum Analysis
Parameter	Lower limit display on/off. OFF(0) Display off ON(1) Display on
Example	:CALC:LIM:UPP:DISP ON
Query syntax	:CALC:LIM:UPP:DISP?
Default	OFF
Return type	Numeric value (bool) or character

:CALCulate[:SELected]:LIMit:UPPer:MARGin <num>

(Read and write) query or set upper limit margin.

Applicable mode	Spectrum Analysis
Parameter	Upper limit margin. **Parameter** range: -40~0, unit: dBm
Example	:CALC:LIM:UPP:MARG -10
Query syntax	:CALC:LIM:UPP:MARG?
Default	0
Return type	Numeric value (float) or character

:CALCulate[:SELected]:LIMit:UPPer:TEST <bool>

(Read and write) query or set upper limit test on/off.

Applicable mode	Spectrum Analysis	
Parameter	Upper limit test on/off.	
	OFF(0)	Test off
	ON(1)	Test on
Example	:CALC:LIM:UPP:TEST ON	
Query syntax	:CALC:LIM:UPP:TEST?	

:CALCulate[:SELected]:LIMit:UPPer:EDIT:ADD

(Write only) add default point in lower limit, and default setting of default point is: Limit frequency: 44.1GHz, limit value: 0dBm

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:UPP:EDIT:ADD
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:LIMit:UPPer:EDIT:CLEar

(Write only) delete all edit points in upper limit

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CALC:LIM:UPP:EDIT:CLE
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:LIMit:UPPer:EDIT:DELete

(Write only) delete current point in upper limit.

Applicable mode	Spectrum Analyser
Parameter	None
Example	:CALC:LIM:UPP:EDIT:DEL
Query syntax	None
Default	None
Return type	None
Default	OFF
Return type	Numeric value (bool) or character

:CALCulate[:SELected]:LIMit: UPPer:EDIT:DATA

(Write only) upper limit edit point.

Applicable mode	Spectrum Analysis

Parameter Point index (int, start from 0), limit frequency (0~44.1GHz), limit value (-174dBm~50dBm)
Example :CALC:LIM:UPP:EDIT:DATA
Query syntax None
Default None
Return type None

:CALCulate[:SELected]:LIST:EDIT:ADD**(Options)**

(Write only) edit list to add default segments and set the default of default segments as follows:

Spectrum Analysis		Field Strength	
Starting frequency	1GHz	Freqneucy	500MHz
Stop frequency	2GHz	Bandwidth	30kHz
Sweep Points	51	Detector	Average
Resolution Bandwidth	1MHz	Demodulaton	Continuous wave
Video Bandwidth	30kHz	Limit	-174dBm
On/off	Off	Limit display	Off

Applicable mode Spectrum Analysis, Field Strength
Parameter None
Example :CALC:LIST:EDIT:ADD
Query syntax None
Default None
Return type None

:CALCulate[:SELected]:LIST:EDIT:ADD:SEGment

(Write only) edit list to add segments
Applicable mode Spectrum Analysis, Field Strength
Parameter Spectrum Analysis Mode: start frequency (0~44.1GHz), stop frequency (0~44.1GHz), sweep point (51~501), resolution bandwidth (1Hz~10MHz), video bandwidth (1Hz~10MHz), state (ON OFF)
Field Strength Mode: frequency (1MHz~44.1GHz), bandwidth (150Hz~150kHz), detector (average, real time, peak), demodulation, limit (-174dBm~50dBm), limit on/off
Example :CALC:LIST:EDIT:ADD:SEG 100000000,200000000,501,3000,1000,ON
Query syntax None
Default None
Return type None

:CALCulate[:SELected]:LIST:EDIT:CLEar

(Write only) edit list to clear the list.
Applicable mode Spectrum Analysis, Field Strength
Parameter None
Example :CALC:LIST:EDIT:CLE
Query syntax None
Default None
Return type None

:CALCulate[:SELected]:LIST:EDIT:DELete

(Write only) edit list to delete segment.

Applicable mode	Spectrum Analysis, Field Strength
Parameter	Segment index (int), start from 0
Example	:CALC:LIST:EDIT:DEL 1
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:LIST:EDIT: SEGment

(Write only) edit list to edit segment

Applicable mode	Spectrum Analysis, Field Strength
Parameter	Segment index (start from 0), start frequency (0~44.1GHz), stop frequency (0~44.1GHz), sweep point (51~501), resolution bandwidth (1Hz~10MHz), video bandwidth (1Hz~10MHz), state (ON OFF) Field Strength Mode: Segment index (start from 0), frequency (1MHz~44.1GHz), bandwidth (150Hz~150kHz), detector (average, real time, peak), demodulation, limit (-174dBm~50dBm), limit on/off
Example	:CALC:LIST:EDIT:SEG 1,100000000,200000000,501,3000,1000,ON
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:MARKer<n>[:STATe] <string>

(Read and write) set or query marker state.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation	
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.	
<string>	Marker state.	
	OFF(0)	Marker off.
	NORM(1)	Normal marker on.
	DELTa(2)	Delta marker on.
Example	:CALC:MARK1 NORM	
Query syntax	:CALC:MARK1?	
Default	OFF	
Return type	Numeric value (int) or character	

:CALCulate[:SELected]:MARKer<n>:ACTivate

(Write only) activate marker n.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:ACT

Query syntax None
Default None
Return type None

:CALCulate[:SELected]:MARKer<n>:SET <string>

(Write only) set marker function (marker ->)**(if Marker is off, then enable the marker first).**

Applicable mode Spectrum Analysis, Interference Analysis
Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.
<string> Marker function.
Parameters and functions matching with span and non-zero span modes

Instrument mode	Parameter	Function
Non-zero span	STARt,STOP,CENTer,STEP	Set the start, stop, center and step frequencies as current marker frequency
Zero span	STARt,STOP,CENTer,STEP	Set marker index as 0, max index, center index; set step frequency as current marker frequency.

Example :CALC:MARK1:SET STAR
Query syntax None
Default None
Return type None

:CALCulate[:SELected]:MARKer:AOFF

(Write only) disable all markers.
Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter None.
Example :CALC:MARK:AOFF
Query syntax None
Default None
Return type None

:CALCulate[:SELected]:MARKer<n>:X <num>

(Read and write) set or query the value of marker X **(invalid if the marker is off).**
Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.
<num> Value of marker X (see table below for unit). If delta marker is selected, the X value can be negative.

Instrument mode	Parameter unit
Spectrum Analysis (non-zero span)	Hz
Spectrum Analysis (zero span)	ms
Interference Analysis (non-zero span)	Hz
Interference Analysis (zero span)	ms
AM-FM-PM Demodulation (RF spectrum)	Hz
AM-FM-PM Demodulation (audio spectrum)	Hz
AM-FM-PM Demodulation (audio waveform)	ms

Time unit: ms. Frequency unit: Hz.

Example :CALC:MARK1:X 10000
Query syntax :CALC:MARK1:X?
Default When creating the marker, the marker should be set as the center index point.
Return type Numeric value (double) or character

:CALCulate[:SELected]:MARKer<n>:Y?

(Read only) query Y value of marker **(0 will be returned if the marker is off).**

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n> Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6.
If not marked, then n represents 1.
Example :CALC:MARK1:Y?
Query syntax :CALC:MARK1:Y?
Default None
Return type Numeric value (float) or character

Instrument mode	Return parameter	Unit
Spectrum Analysis (single float value)	Amplitude	dBm
Interference Analysis (single float value)	Amplitude	dBm
AM-FM-PM Demodulation: RF spectrum (single float value)	Amplitude	dBm
AM-FM-PM Demodulation: Audio spectrum (single float value)	Amplitude	dBm
AM-FM-PM Demodulation: Audio waveform (single float value)	Amplitude	AM: Percentage FM:Hz PM:Rad

:CALCulate[:SELected]:MARKer<n>:FCOunt[:STATe] <bool>

(Read and write) set or query counter on/off **(the marker will be set as common marker).**

Applicable mode	Spectrum Analysis	
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.	
<bool>	Counter on/off.	
	OFF(0)	Counter off.
	ON(1)	Counter on.

Notes: Only one marker counter can be enabled now.

Example	:CALC:MARK1:FCO ON
Query syntax	:CALC:MARK1:FCO?
Default	OFF
Return type	Numeric value (bool) or character

:CALCulate[:SELected]:MARKer<n>:FCOunt:X?

(Read only) query the frequency count of counter **(0 will be returned if no count is made).**

Applicable mode	Spectrum Analysis
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Parameter	None.
Example	:CALC:MARK1:FCO:X?
Query syntax	:CALC:MARK1:FCO:X?
Default	None
Return type	Numeric value (double) or character

:CALCulate[:SELected]:MARKer<n>:FUNCtion:MAXimum

(Write only) maximum value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:MAX
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:MARKer<n>:FUNCtion:MINimum

(Write only) minimum value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:MIN

Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:MARKer<n>:FUNCtion:PEAK

(Write only) peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:PEAK
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:MARKer<n>:FUNCtion:PLEFt

(Write only) left peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:PLEF
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:MARKer<n>:FUNCtion:PNEXt

(Write only) secondary peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:PNEX
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:MARKer<n>:FUNCtion:PRIGht

(Write only) right peak value of marker search **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter <n>	Marker number, the number can be 1,2,3,4,5,6, indicating Markers 1,2,3,4,5,6. If not marked, then n represents 1.
Example	:CALC:MARK1:FUNC:PRIG
Query syntax	None
Default	None
Return type	None

:CALCulate[:SELected]:MARKer<n>:NOISe[:STATe] <bool>

(Read and write) set or query noise marker **(enable the marker first if the marker is off).**

Applicable mode	Spectrum Analysis, Interference Analysis	
Parameter	Marker number, the number can be 1,2,3,4,5,6, indicating	
<n>	Markers 1,2,3,4,5,6.	
	If not marked, then n represents 1.	
<bool>	Noise marker on/off.	
	OFF(0)	Noise marker off.
	ON(1)	Noise marker on.
Example	:CALC:MARK1:NOIS ON	
Query syntax	:CALC:MARK1:NOIS?	
Default	OFF	
Return type	Numeric value (bool) or character	

:CALCulate[:SELected]:RELative[:MAGNitude]?

(Read only) query saved relative magnitude.

Applicable mode	Power meter
Parameter	None
Example	:CALC:REL?
Query syntax	:CALC:REL?
Default	None
Return type	Numeric value (float) or character

:CALCulate[:SELected]:RELative[:MAGNitude]:AUTO <bool>

(Read and write) set or query relative magnitude on/off.

Applicable mode	Power meter	
Parameter	Relative magnitude on/off.	
	OFF(0)	Relative magnitude off.
	ON(1)	Relative magnitude on.
Example	:CALC:REL:AUTO ON	
Query syntax	:CALC:REL:AUTO?	
Default	OFF	
Return type	Numeric value (bool) or character	

:CALCulate[:SELected]:PEAK:TRAC <bool>

(Read and write) set or query peak trace on/off.

Applicable mode	Spectrum Analysis	
Parameter	Peak trace on/off.	
	OFF(0)	Peak trace off.
	ON(1)	Peak trace on.
Example	:CALC:PEAK:TRAC ON	
Query syntax	:CALC:PEAK:TRAC?	
Default	OFF	
Return type	Numeric value (bool) or character	

:CALibration:ZERO

(Write only) Power measurement calibration **(do not repeatedly calibrate during calibration).** This is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode Power meter
Parameter None
Example :CAL:ZERO;*OPC?
Query syntax None
Default None
Return type None

:CALibration:ZERO:STATe?

(Read only) query if Power measurement calibration is successful **(query is not available during calibration).**

Applicable mode Power meter
Parameter None
Example :CAL:ZERO:STAT?
Query syntax :CAL:ZERO:STAT?
Default 0
Return type Numeric value (int) or character
0: no calibration
1: calibration succeed
2: calibration failed

:DISPlay:WINDow:ANALog:LOWer <num>

(Read and write) set or query minimum scale value.

Applicable mode Power meter
Parameter Mininum scale value (-70dBm~25dBm).
Example :DISP:WIND:ANAL:LOW -60
Query syntax :DISP:WIND:ANAL:LOW?
Default -70dBm
Return type Numeric value (float) or character

:DISPlay:WINDow:ANALog:UPPer <num>

(Read and write) set or query maximum scale value.

Applicable mode Power meter
Parameter Maximum scale value (-65dBm~30dBm)
Example :DISP:WIND:ANAL:UPP 20
Query syntax :DISP:WIND:ANAL:UPP?
Default 30 dBm
Return type Numeric value (float) or character

:DISPlay:WINDow:TRACe:Y[:SCALe]:AUTO

(Write only) set as auto scale.

Applicable mode Power meter
Parameter None
Example :DISP:WIND:TRAC:Y:AUTO
Query syntax None
Default None
Return type None

:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision <num>

(Read and write) query or set scale/division

Applicable mode Spectrum Analysis (only available for logarithmic scale type), Interference Analysis, AM-FM-PM Demodulation, Channel Scanning
Parameter Scale/division. Range: 0.1dB~20dB

Example	:DISP:WIND:TRAC:Y:PDIV 0.1
Query syntax	:DISP:WIND:TRAC:Y:PDIV?
Default	10dB
Return type	Numeric value (float) or character

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <num>

(Read and write) query or set reference level.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Channel Sweep
Parameter	Reference level value (reference value). Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required. Range: -120 dBm ~+40 dBm
Example	:DISP:WIND:TRAC:Y:RLEV -10
Query syntax	:DISP:WIND:TRAC:Y:RLEV?
Default	0dBm
Return type	Numeric value (float) or character

:DISPlay:WINDow:TRACe:Y[:SCALe]:RPOSition <num>

(Read and write) set or query reference position.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Reference position (no unit) Scope -10~10.
Example	:DISP:WIND:TRAC:Y:RPOS 1
Query syntax	:DISP:WIND:TRAC:Y:RPOS?
Default	0
Return type	Numeric value (int) or character

:DISPlay:TITLe <string>

(Write only) set the title.

Applicable mode	All modes
Parameter	Title, 10-digit letter or number at maximum.
Example	:DISP:TITL SA_MEASURE
Query syntax	None
Default	None
Return type	None

:DISPlay:TITLe:STATe <bool>

(Read and write) set or query title state.

Applicable mode	All modes	
Parameter	Title state	
	OFF(0)	Title off.
	ON(1)	Title on.
Example	:DISP:TITL:STAT ON	
Query syntax	:DISP:TITL:STAT?	
Default	OFF	
Return type	Numeric value (bool) or character	

:DISPlay:MODE <string>

(Read and write) set or query display mode.

Applicable mode	All modes	
Parameter	Display mode.	
	DEFA(0)	Default mode
	OUT(1)	Black and white mode
	NIGHT(2)	Night vision mode
Example	:DISP:MODE DEFA	
Query syntax	:DISP:MODE?	
Default	DEFA	
Return type	Numeric value (int) or character	

:DISPlay:BRIG <int>

(Read and write) set or query brightness level.

Applicable mode	All modes
Parameter	Brightness level. Range: 0~4
Example	:DISP:BRIG 1
Query syntax	:DISP:BRIG?
Default	3
Return type	Numeric value (int) or character

:DISPlay:BRIG:AUTO <bool>

(Read and write) set or query brightness auto adjustment on/off.

Applicable mode	All modes	
Parameter	Brightness auto on/off.	
	OFF(0)	Brightness auto off.
	ON(1)	Brightness auto on.
Example	:DISP:BRIG:AUTO ON	
Query syntax	:DISP:BRIG:AUTO?	
Default	OFF	
Return type	Numeric value (bool) or character	

:DISPlay:TIME:FMT <string>

(Write only) set time format.

Applicable mode	All modes	
Parameter	Time format.	
	YMD(0)	Year/month/date.
	MDY(1)	Month/date/year.
	DMY(2)	Date/month/year.
Example	:DISP:TIME:FMT YMD	
Query syntax	None	
Default	YMD	
Return type	Numeric value (int) or character	

:FORM[:DATA] <string>

(Read and write) set or query data format.

Applicable mode	All modes
Parameter	Data format. ASC(0) refers to character format. HEX(1) refers to numerical format. If character format is selected, the data format returned after query will be character format with character as the unit If numerical format is selected, the data format returned after query will

	be numerical format with byte as the unit
Example	:FORM ASC
Query syntax	:FORM?
Default	ASC
Return type	Numeric value (bool) or character

:INITiate:CONTinuous <bool>

(Read and write) query or set sweep type. This is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Sweep type. OFF or 0 refers to single sweep. ON or 1 refers to continuous sweep.
Example	:INIT:CONT OFF;***OPC?**
Query syntax	:INIT:CONT?
Default	ON
Return type	Numeric value (bool) or character

:INITiate

(Write only) trigger one single sweep (only valid for single sweep). This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	None.
Example	:INIT;*OPC?
Query syntax	None
Default	None
Return type	None

:INSTrument:CATalog?

(Read only) query available instrument working mode. Use :INST:CAT?to query available instrument working mode.

Applicable mode	All modes
Parameter	None
Example	:INST:CAT?
Query syntax	:INST:CAT?
Default	0x01
Return type	Numeric value (int) or character The 0 bit is the bit for spectrum analysis test, 1 (mandatory) The 1 bit is the bit for AM-FM-PM Demodulation test, 1 is settable (optional) and 0 is not settable The 2 bit is the bit for interference analysis test, 1 is settable (optional) and 0 is not settable The 3 bit is the bit for Power measurement test, 1 is settable (optional) and 0 is not settable The 4 bit is the bit for channel sweep test, 1 is settable (optional) and 0 is not settable

:INSTrument[:SELect] <string>

(Read and write) query or set current instrument working mode. Use :INST:CAT? to query available instrument working modes. This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	**All modes**	
Parameter	**Instrument mode.**	
	SA(1)	Spectrum analysis mode
	IA(2)	Interference analysis mode **(option)**
	DM(3)	**AM-FM-PM Demodulation mode (option)**
	PM(4)	**Power measurement mode (option, and USB power probe connection required)**
	CS(5)	**Channel sweep mode (option)**
Example	**:INST SA;*OPC?**	
Query syntax	**:INST?**	
Default	SA	
Return type	**Numeric value (int) or character**	

:MMEMory:DELete:ANTenna <string>

(Write only) delete antenna file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of antenna file.
Example	:MMEM:DEL:ANT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:ANTenna:ALL

(Write only) delete all antenna files under current mode.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:MMEM:DEL:ANT:ALL
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:LIMit <string>

(Write only) delete limit file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of limit file
Example	:MMEM:DEL:LIM set1
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:LIMit:ALL

(Write only) delete all limit files under current mode.

Applicable mode	Spectrum Analysis
Parameter	None
Example	:MMEM:DEL:LIM:ALL
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:LIST <string>

(Write only) delete list file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of list file.
Example	:MMEM:DEL:LIST set1
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:LIST:ALL

(Write only) delete all list files under current mode.

Applicable mode	Spectrum Analysis
Parameter	Name of list file.
Example	:MMEM:DEL:LIST:ALL
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:STATe <string>

(Write only) delete state file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	All modes
Parameter	Name of state file.
Example	:MMEM:DEL:STAT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:STATe:ALL

(Write only) delete all state files under current mode.

Applicable mode	All modes
Parameter	None
Example	:MMEM:DEL:STAT:ALL
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:DATA <string>

(Write only) delete data file under current mode.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Channel Sweep
Parameter	None
Example	:MMEM:DEL:DATA set1
Query syntax	None
Default	None
Return type	None

:MMEMory:DELete:DATA:ALL

(Write only) delete all data files under current mode.

Applicable	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation,

mode Channel Sweep
Parameter None
Example :MMEM:DEL:DATA:ALL
Query syntax None
Default None
Return type None

:MMEMory:LOAD:ANTenna <string>

(Write only) select antenna factor for field strength function **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode Spectrum Analysis
Parameter Name of antenna factor file
Example :MMEM:LOAD:ANT 89101A
Query syntax None
Default Antenna factor not selected
Return type None

:MMEMory:LOAD:LIMit <string>

(Write only) recall limit line **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode Spectrum Analysis
Parameter Name of limit line file
Example :MMEM:LOAD:LIM set1
Query syntax None
Default None
Return type None
Default None
Return type None

:MMEMory:LOAD:SEM <string>

(Write only) recall Spectrum Emission Mask file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode Spectrum Analysis
Parameter Name of SEM file
Example :MMEM:LOAD:SEM set1
Query syntax None
Default None
Return type None
Default None
Return type None

:MMEMory:LOAD:LIST <string>

(Write only) recall list file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode Spectrum Analysis
Parameter Name of list file.
Example :MMEM:LOAD:LIST set1
Query syntax None
Default None
Return type None

:MMEMory:LOAD:STATe <string>

(Write only) recall state file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	All modes
Parameter	Name of state file.
Example	:MMEM:LOAD:STAT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:LOAD:DATA <string>

(Write only) recall data file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	All modes
Parameter	Name of data file.
Example	:MMEM:LOAD:DATA set1
Query syntax	None
Default	None
Return type	None

:MMEMory:LOCation <string>

(Read and write) query or set storage location.

Applicable mode	All modes	
Parameter	Location.	
	INT(0)	Internal
	SD(1)	SD card
	USB(2)	USB
Example	:MMEM:LOC USB	
Query syntax	:MMEM:LOC?	
Default	INT	
Return type	Numeric value (int) or character	

:MMEMory:STORe:ANTenna <string>

(Write only) store antenna factor file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of antenna factor file.
Example	:MMEM:STOR:ANT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:STORe:LIMit <string>

(Write only) store current limit line as file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of limit file
Example	:MMEM:STOR:LIM set1
Query syntax	None
Default	None
Return type	None

:MMEMory:STORe:LIST <string>

(Write only) store current list data in file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis
Parameter	Name of list file.
Example	:MMEM:STOR:LIST set1
Query syntax	None
Default	None
Return type	None

:MMEMory:STORe:SCReen <string>

(Write only) screen copy, and save current screenshot as file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	All modes
Parameter	Name of screen copy file.
Example	:MMEM:STOR:SCR pic1
Query syntax	None
Default	None
Return type	None

:MMEMory:STORe:STATe <string>

(Write only) store state under current mode as a file **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	All modes
Parameter	Name of state file.
Example	:MMEM:STOR:STAT set1
Query syntax	None
Default	None
Return type	None

:MMEMory:STORe:DATA <string>

(Write only) store data file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Channel Sweep
Parameter	Name of data file.
Example	:MMEM:STOR:DATA set1
Query syntax	None
Default	None
Return type	None

[:SENSe]:ACPower:ADJChbw <num>

(Read and write) set or query ACPR adjacent channel bandwidth.

Applicable mode	Spectrum Analysis
Parameter	ACPR adjacent channel bandwidth (Hz). Scope: 300Hz~20MHz
Example	:ACP:ADJC 3000000
Query syntax	:ACP:ADJC?
Default	3MHz
Return type	Numeric value (int) or character

[:SENSe]:ACPower:LIMit[:STATe] <bool>

(Read and write) set or query ACPR door limit state.

Applicable mode	Spectrum Analysis	
Parameter	Door limit test state.	
	OFF(0)	Door limit test off
	ON(1)	Door limit test on
Example	:ACP:LIM OFF	
Query syntax	:ACP:LIM?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:ACPower:MAINchbw <num>

(Read and write) set or query ACPR main channel bandwidth.

Applicable mode	Spectrum Analysis
Parameter	ACPR main channel bandwidth (Hz). Scope: 300Hz~20MHz
Example	:ACP:MAIN 3000000
Query syntax	:ACP:MAIN?
Default	3MHz
Return type	Numeric value (int) or character

[:SENSe]:ACPower:OFFSet:LLIMit <num>

(Read and write) set or query ACPR lower adjacent channel limit.

Applicable mode	Spectrum Analysis	
Parameter	ACPR lower adjacent channel limit (dB). Scope: -200dB~200dB	
Example	:ACP:OFFS:LLIM	0
Query syntax	:ACP:OFFS:LLIM?	
Default	0	
Return type	Numeric value (float) or character	

[:SENSe]:ACPower:OFFSet:ULIMit <num>

(Read and write) set or query ACPR upper adjacent channel limit.

Applicable mode	Spectrum Analysis	
Parameter	ACPR upper adjacent channel limit (dB). Scope: -200dB~200dB	
Example	:ACP:OFFS:ULIM	0
Query syntax	:ACP:OFFS:ULIM?	
Default	0	
Return type	Numeric value (float) or character	

[:SENSe]:ACPower:SPACe <num>

(Read and write) set or query ACPR channel space.

Applicable mode	Spectrum Analysis
Parameter	ACPR channel space (Hz). Scope: 0Hz~45MHz
Example	:ACP:SPAC 3000000
Query syntax	:ACP:SPAC?
Default	3MHz
Return type	Numeric value (int) or character

[:SENSe]:ACPower[:STATe] <bool>

(Read and write) set or query ACPR state **(Other functional measurements will be disabled after this function is enabled)**, or command [:SENSe]:MEASurement can be used.

Applicable mode	Spectrum Analysis	
Parameter	ACPR state.	
	OFF(0)	ACPR off
	ON(1)	ACPR on
Example	:ACP ON	
Query syntax	:ACP?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:ACPower:UPPer?

(Read only) query ACPR upper adjacent channel power ratio.

Applicable mode	Spectrum Analysis
Parameter	None
Query syntax	:ACP:UPP?
Default	0
Return type	Numeric value (float) or character

[:SENSe]:ACPower:LOWer?

(Read only) query ACPR lower adjacent channel power ratio.

Applicable mode	Spectrum Analysis
Parameter	None
Query syntax	:ACP:LOW?
Default	0
Return type	Numeric value (float) or character

[:SENSe]:AFPanalyzer:DEMod:TYPE <string>

(Read and write) set or query demodulation type of AM-FM-PM Demodulation.

Applicable mode	**AM-FM-PM Demodulation**
Parameter	Demod type.
	AM(0)
	FM(1)
	PM(2)
Example	:AFP:DEM:TYPE AM
Query syntax	:AFP:DEM:TYPE?
Default	AM
Return type	Numeric value (int) or character

[:SENSe]:AFPanalyzer:DEMod:MODE <string>

(Read and write) set or query mode of AM-FM-PM Demodulation.

Applicable mode	**AM-FM-PM Demodulation**
Parameter	Demodulation mode.
	RF(0) RF spectrum
	AF(1) audio spectrum
	AW(2) audio waveform
	ALL(3) all spectrums
Example	AFP:DEM:MODE RF
Query syntax	:AFP:DEM:MODE?
Default	ALL
Return type	Numeric value (int) or character

[:SENSe]:AFPanalyzer:SPAN <num>

(Read and write) set or query demodulation audio spectrum span of AM-FM-PM Demodulation.

Applicable mode	**AM-FM-PM Demodulation**
Parameter	Audio spectrum span.
Example	:AFP:SPAN 10000
Query syntax	:AFP:SPAN?
Default	100000
Return type	Numeric value (double) or character

[:SENSe]:AFPanalyzer:SCALe <num>

(Read and write) set or query audio spectrum scale/division of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	Audio spectrum scale/division
Example	:AFP:SCAL 10
Query syntax	:AFP:SCAL?
Default	10
Return type	Numeric value (double) or character

[:SENSe]:AFPanalyzer:SWEep:TIME <num>

(Read and write) set or query audio wave sweep time of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	Audio waveform sweep time (us).
Example	:AFP:SWE:TIME 1000
Query syntax	:AFP:SWE:TIME?
Default	None
Return type	Numeric value (double) or character

[:SENSe]:AFPanalyzer:IFBW <num>

(Read and write) set or query IF bandwidth of AM-FM-PM Demodulation.

Applicable mode	AM-FM-PM Demodulation
Parameter	intermediate-frequency bandwidth. Range: 10kHz~300kHz
Example	:AFP:IFBW 100000
Query syntax	:AFP:IFBW?
Default	300000
Return type	Numeric value (double) or character

[:SENSe]:AFPanalyzer:TRACe <string>

(Write only) set the trace of marker selection.

Applicable mode	AM-FM-PM Demodulation
Parameter	Type of marker selection trace. RF(0) RF spectrum AF(1) Audio spectrum AW(2) Audio waveform
Example	:AFP:TRAC RF
Query syntax	None
Default	RF
Return type	Numeric value (int) or character

[:SENSe]:AMPLitude:ALIGnment:NOW

(Read only) zero frequency calibration **(do not repeatedly carry out zero frequency calibration during zero frequency calibration).** This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode Spectrum Analysis
Parameter None
Example :AMPL:ALIG:NOW;*OPC?
Query syntax None
Default None
Return type None

[:SENSe]:AMPLitude:CORRections:ANTenna:OFF

(Write only) set antenna factor loading off and set antenna factor free state.

Applicable mode Spectrum Analysis, Field Strength
Parameter None
Example :AMPL:CORR:ANT:OFF
Query syntax None
Default OFF
Return type None

[:SENSe]:AMPLitude:CORRections[:STATe] <bool>

(Read and write) set or query field strength function measurement state **(Other functional measurements will be disabled after this function is enabled),** or command [:SENSe]:MEASurement can be used.

Applicable mode Spectrum Analysis
Parameter field strength state.

OFF(0)	field strength off
ON(1)	field strength on

Example :AMPL:CORR:STAT ON
Query syntax :AMPL:CORR?
Default OFF
Return type Numeric value (bool) or character

[:SENSe]:AMPLitude:SCALe<string>

(Read and write) query or set scale type.

Applicable mode Spectrum Analysis
Parameter Scale Type.

LOG(0)	Logarithm
LIN(1)	Linear

Example :AMPL:SCAL LOG
Query syntax :AMPL:SCAL?
Default LOG
Return type Numeric value (int) or character

[:SENSe]:AMPLitude:UNIT <string>

(Read and write) query or set amplitude unit.

Applicable mode Spectrum Analysis, Field Strength
Parameter Amplitude unit.

Spectrum Analysis	DBM(0)	In dBm.
	DBMV(1)	Unit: dBmV
	DBUV(2)	Unit: dBuV

	V(3)	Unit: Volts
	W(4)	Unit: Walts

Example :AMPL:UNIT DBM
Query syntax :AMPL:UNIT?
Default BM
Return type Numeric value (int) or character

[:SENSe]:AMPLitude:CORRections:ANTenna:EDIT:ADD

(Write only) edit antenna factor to add default point
Applicable mode Spectrum Analysis, Field Strength
Parameter None
Example :AMPL:CORR:ANT:EDIT:ADD
Query syntax None
Default None
Return type None

[:SENSe]:AMPLitude:CORRections:ANTenna:EDIT:ADD:DATA

(Write only) edit antenna factor to add point
Applicable mode Spectrum Analysis, Field Strength
Parameter Frequency (0~44.1GHz), antenna factor value (-200dB~200dB)
Example :AMPL:CORR:ANT:EDIT:ADD:DATA 1000000000,5.0
Query syntax None
Default None
Return type None

[:SENSe]:AMPLitude:CORRections:ANTenna:EDIT:DEL <int>

(Write only) edit antenna factor to delete point
Applicable mode Spectrum Analysis, Field Strength
Parameter Point index (start from 0)
Example :AMPL:CORR:ANT:EDIT:DEL 1
Query syntax None
Default None
Return type None

[:SENSe]:AMPLitude:CORRections:ANTenna:EDIT:DATA

(Write only) edit antenna factor to edit point
Applicable mode Spectrum Analysis, Field Strength
Parameter Point index (start from 0), frequency value (0~44.1GHz), antenna factor value (-200dB~200dB)
Example :AMPL:CORR:ANT:EDIT:DATA 1,1000000000,5.0
Query syntax None
Default None
Return type None

[:SENSe]:AVERage:COUNt <num>

(Read and write) query or set average count.
Applicable mode Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter Average.
Scope: 1~1000.

Example	:AVER:COUN 16
Query syntax	:AVER:COUN?
Default	16
Return type	Numeric value (int) or character

[:SENSe]:AVERage:CLEar

(Write only) start current average count from 0.

Applicable mode	Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter	None.
Example	:AVER:CLE
Query syntax	None
Return type	None

[:SENSe]:AVERage:STATe <bool>

(Read and write) query or set average state.

Applicable mode	Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter	Average ON/OFF. OFF or 0 refers to off. ON or 1 refers to on.
Example	:AVER:STAT OFF
Query syntax	:AVER:STAT?
Default	Off
Return type	Numeric value (bool) or character

[:SENSe]:AVERage:CURC?

(Read only) query current average count.

Applicable mode	Spectrum analysis, AM-FM-PM Demodulation, interference analysis and power meter
Parameter	None.
Example	:AVER:CURC?
Query syntax	:AVER:CURC?
Return type	None

[:SENSe]:BANDwidth[:RESolution] <num>

(Read and write) query or set resolution bandwidth.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Resolution bandwidth (Hz). Scope: 1Hz~10MHz
Example	:BAND 300000
Query syntax	:BAND?
Default	3MHz
Return type	Numeric value (double) or character

[:SENSe]:BANDwidth[:RESolution]:AUTO <bool>

(Read and write) query or set resolution bandwidth auto on/off.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Resolution bandwidth auto on/off. When set as auto, the resolution bandwidth will be adaptable to the bandwidth based on the ratio of SPAN/RBW. OFF or 0 refers to manual.

	ON or 1 refers to auto.
Example	:BAND:AUTO ON
Query syntax	:BAND:AUTO?
Default	ON
Return type	Numeric value (bool) or character

[:SENSe]:BANDwidth[:RESolution]:RATio <num>

(Read and write) query or set span/resolution bandwidth ratio.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	SPAN/RBW value. Scope: 1~500.
Example	:BAND:RAT 100
Query syntax	:BAND:RAT?
Default	100
Return type	Numeric value (int) or character

[:SENSe]:BANDwidth:VIDeo <num>

(Read and write) query or set video bandwidth.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Video bandwidth value (Hz). Scope: 1Hz~10MHz
Example	:BAND:VID 300000
Query syntax	:BAND:VID?
Default	3MHz
Return type	Numeric value (int) or character

[:SENSe]:BANDwidth:VIDeo:AUTO <bool>

(Read and write) query or set video bandwidth auto on/off.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Video bandwidth auto on/off. When set as auto, the video bandwidth will be adaptable to the resolution bandwidth based on the ratio of SPAN/VBW. OFF or 0 refers to manual. ON or 1 refers to auto.
Example	:BAND:VID:AUTO ON
Query syntax	:BAND:VID:AUTO?
Default	ON
Return type	Numeric value (bool) or character

[:SENSe]:BANDwidth:VIDeo:RATio <num>

(Read and write) set or query resolution bandwidth/video bandwidth ratio.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	RBW/VBW value. Scope: 1~100.
Example	:BAND:VID:RAT 1
Query syntax	:BAND:VID:RAT?
Default	3
Return type	Numeric value (int) or character

[:SENSe]:BANDwidth:VIDeo:TYPE <BOOL>

(Read and write) set or query video type.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Video type LIN(0), line, LOG(1), logarithm
Example	: BAND:VID:TYPE LOG
Query syntax	: BAND:VID:TYPE?
Default	LIN
Return type	Numeric value (int) or character

[:SENSe]:CMEasurement:IBW <num>

(Read and write) set or query channel power bandwidth.

Applicable mode	Spectrum Analysis
Parameter	Channel power bandwidth (Hz), scope: 100Hz~44.1GHz.
Example	:CME:IBW 1000000
Query syntax	:CME:IBW?
Default	2MHz
Return type	Numeric value (double) or character

[:SENSe]:CMEasurement:PSDR?

(Read only) query channel power density under channel power function measurement **(valid when the channel power is on and after one sweep).**

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CME:PSDR?
Query syntax	:CME:PSDR?
Default	None
Return type	Numeric value (float) or character

[:SENSe]:CMEasurement[:STATe] <bool>

(Read and write) set or query channel power function measurement state **(Other functional measurements will be disabled after this function is enabled)**, or use command [:SENSe]:MEASurement.

Applicable mode	Spectrum Analysis	
Parameter	Channel power state.	
	OFF(0)	Channel power off
	ON(1)	Channel power on
Example	:CME ON	
Query syntax	:CME?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:CMEasurement:TPWR?

(Read only) query channel power value under channel power function measurement **(valid when the channel power is on and after one sweep).**

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CME:TPWR?
Query syntax	:CME:TPWR?
Default	None
Return type	Numeric value (float) or character

[:SENSe]:CORRection:GAIN <num>

(Read and write) set or query offset value.

Applicable mode	Power meter
Parameter	Offset value (-50dB~30dB).
Example	:CORR:GAIN -5
Query syntax	:CORR:GAIN?
Default	0dB
Return type	Numeric value (float) or character

[:SENSe]:CORRection:GAIN:STATe <bool>

(Read and write) set or query offset state.

Applicable mode	Power meter	
Parameter	Offset state.	
	OFF(0)	Offset off.
	ON(1)	Offset on.
Example	:CORR:GAIN:STAT ON	
Query syntax	:CORR:GAIN:STAT?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:CNRatio[:STATe] <bool>

(Read and write) set or query CNR ratio measurement state **(Other functional measurements will be disabled after this function is enabled)**, or use command [:SENSe]:MEASurement.

Applicable mode	Spectrum Analysis	
Parameter	CNR measurement state.	
	OFF(0)	Off.
	ON(1)	On.
Example	:CNR ON	
Query syntax	:CNR?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:CNRatio:CBW <num>

(Read and write) set or query CNR carrier bandwidth.

Applicable mode	Spectrum Analysis
Parameter	CNR carrier bandwidth. Range: 300Hz~20MHz
Example	:CNR:CBW 1000000
Query syntax	:CNR:CBW?
Default	3000000
Return type	Numeric value (double) or character

[:SENSe]:CNRatio:NBW <num>

(Read and write) set or query CNR noise bandwidth.

Applicable mode	Spectrum Analysis
Parameter	CNR noise bandwidth. Range: 300Hz~20MHz
Example	:CNR:NBW 1000000
Query syntax	:CNR:NBW?
Default	3000000
Return type	Numeric value (double) or character

[:SENSe]:CNRatio:CNSPace <num>

(Read and write) set or query CNR frequency offset.

Applicable mode	Spectrum Analysis
Parameter	CNR frequency offset. Range: 0Hz~100MHz
Example	:CNR:CNSP 1000000
Query syntax	:CNR:CNSP?
Default	3000000
Return type	Numeric value (double) or character

[:SENSe]:CNRatio:CNRatio?

(Read only) query CNR under CNR measurement **(valid when CNR is open and after one sweep).**

Applicable mode	Spectrum Analysis
Parameter	None
Example	:CNR:CNR?
Query syntax	:CNR:CNR?
Default	None
Return type	Numeric value (float) or character

[:SENSe]:CS:DISPlay <string>

(Read and write) set or query channel sweep graph/table display.

Applicable mode	Channel sweep
Parameter	Channel sweep graph/table display. GRAPH(0) graph display TABLE(1) table display
Example	:CS:DISP TABLE
Query syntax	:CS:DISP?
Default	GRAPH
Return type	Numeric value (int) or character

[:SENSe]:CS:MAXHold <bool>

(Read and write) set or query channel sweep maximum hold state.

Applicable mode	Channel sweep
Parameter	maximum hold state. OFF(0) maximum hold off ON(1) maximum hold on
Example	:CS:MAXH ON
Query syntax	:CS:MAXH?
Default	OFF
Return type	Numeric value (bool) or character

[:SENSe]:CS:UNIT <string>

(Read and write) set or query channel sweep unit.

Applicable mode	Channel sweep
Parameter	channel sweep unit. CHAN(0) channel FREQ(1) frequency
Example	:CS:UNIT CHAN
Query syntax	:CS:UNIT?
Default	CHAN
Return type	Numeric value (int) or character

[:SENSe]:CS:PWR <string>

(Read and write) set or query channel sweep power display mode.

Applicable mode	Channel sweep
Parameter	channel sweep power display mode. NOW(0) real-time MAX(1) maximum
Example	:CS:PWR NOW
Query syntax	:CS:PWR?
Default	NOW
Return type	Numeric value (int) or character

[:SENSe]:CS:COLOur <string>

(Read and write) set or query channel sweep graph color code display mode.

Applicable mode	Channel sweep
Parameter	Color display mode. SING(0) single color DUAL(1) dual color
Example	:CS:COLO SING
Query syntax	:CS:COLO?
Default	SING
Return type	Numeric value (int) or character

[:SENSe]:CS:ORIEntal <string>

(Read and write) set or query vertical or horizontal display of channel sweep graph.

Applicable mode	Channel sweep
Parameter	Vertical or horizontal display of graph. VERT(0) vertical display HORI(1) horizontal display
Example	:CS:ORIE VERT
Query syntax	:CS:ORIE?
Default	VERT
Return type	Numeric value (int) or character

[:SENSe]:CS:MODE <string>

(Read and write) set or query scan mode of channel sweep.

Applicable mode	Channel sweep
Parameter	Sweep mode. CHAN(0) channel sweep FREQ(1) frequency sweep LIST(2) list sweep
Example	:CS:MODE CHAN
Query syntax	:CS:MODE?
Default	CHAN
Return type	Numeric value (int) or character

[:SENSe]:CS:CHANnel:NUMber <num>

(Read and write) set or query number of channels swept by channel sweep

Applicable mode	Channel sweep
Parameter	Channel number (1~20).
Example	:CS:CHAN:NUM 10
Query syntax	:CS:CHAN:NUM?
Default	10

Return type Numeric value (int) or character

[:SENSe]:CS:CHANnel:STEP <num>

(Read and write) set or query channel steps swept by channel sweep

Applicable mode Channel sweep
Parameter Channel step (1~25).
Example :CS:CHAN:STEP 10
Query syntax :CS:CHAN:STEP?
Default 1
Return type Numeric value (int) or character

[:SENSe]:CS:CHANnel:BANDwidth <num>

(Read and write) set or query channel bandwidth swept by channel sweep

Applicable mode Channel sweep
Parameter channel bandwidth (Hz).
Range: 1kHz~20MHz
Example :CS:CHAN:BAND 100000
Query syntax :CS:CHAN:BAND?
Default 200000
Return type Numeric value (double) or character

[:SENSe]:CS:FREQuency:STARt <num>

(Read and write) set or query start frequency of frequency sweep.

Applicable mode Channel sweep
Parameter start frequency (Hz)
Range: 0Hz~44.1GHz
Example :CS:FREQ:STAR 1000000
Query syntax :CS:FREQ:STAR?
Default 890000000
Return type Numeric value (double) or character

[:SENSe]:CS:FREQuency:STEP <num>

(Read and write) set or query step frequency of frequency sweep in channel sweep.

Applicable mode Channel sweep
Parameter step frequency (Hz)
Range: 1Hz~5GHz
Example :CS:FREQ:STEP 1000000
Query syntax S:FREQ:STEP?
Default 200000
Return type Numeric value (double) or character

[:SENSe]:CS:FREQuency:BANDwidth <num>

(Read and write) set or query channel bandwidth of frequency sweep in channel sweep.

Applicable mode Channel sweep
Parameter channel bandwidth (Hz).
Range: 1kHz~20MHz
Example :CS:FREQ:BAND 1000000
Query syntax :CS:FREQ:BAND?
Default 200000
Return type Numeric value (double) or character

[:SENSe]:CS:FREQuency:NUMber <num>

(Read and write) set or query number of channels swept by frequency sweep in channel sweep.

Applicable mode Channel sweep
Parameter Channel number (1~20).
Example :CS:FREQ:NUM 10
Query syntax :CS:FREQ:NUM?
Default 10
Return type Numeric value (int) or character

[:SENSe]:CS:LIST:NUMber <num>

(Read and write) set or query number of channels swept by list sweep in channel sweep

Applicable mode Channel sweep
Parameter Channel number (1~20).
Example :CS:LIST:NUM 10
Query syntax :CS:LIST:NUM?
Default 10
Return type Numeric value (int) or character

[:SENSe]:DETector:FUNCtion <string>

(Read and write) set or query detector type.

Applicable mode Spectrum Analysis, Interference Analysis, Field Strength
Parameter Detector type

Spectrum Analysis, Interference Analysis		Field strength	
POSitive(0)	Positive Peak	AVERage(0)	Average
NEGative(1)	Negative Peak		
SAMPle(2)	Sample	POSitive(1)	Peak
NORMal(3)	Standard (Rosenfeld)		
AVERage(4)	Average	SAMPle(2)	Real time
RMS(5)	Root mean square		

Example :DET:FUNC NORM
Query syntax :DET:FUNC?
Default NORM
Return type Numeric value (int) or character

[:SENSe]:DETector:FUNCtion:AUTO <bool>

(Read and write) set or query detector auto on/off.

Applicable mode Spectrum Analysis, Interference Analysis
Parameter Detector auto on/off. When the detector is set as auto, the instrument will automatically select detector type based on different measurement.
OFF(0) refers to manual.
ON(1) refers to auto.
Example :DET:FUNC:AUTO OFF
Query syntax :DET:FUNC:AUTO?

Default ON
Return type Numeric value (bool) or character

[:SENSe]:EMISsion[:STATe] <bool>

(Read and write) set or query SEM state **(Other functional measurements will be disabled after this function is enabled)**, or use command [:SENSe]:MEASurement.

Applicable mode Spectrum Analysis
Parameter SEM measurement state
OFF(0), off.
ON(1), on.
Example :EMIS OFF
Query syntax :EMIS?
Default OFF
Return type Numeric value (bool) or character

[:SENSe]:EMISsion:CBW <num>

(Read and write) set or query SEM reference channel bandwidth.

Applicable mode Spectrum Analysis
Parameter reference channel bandwidth (Hz).
Range: 1kHz~44.1GHz
Example :EMIS:CBW 1000000
Query syntax :EMIS:CBW?
Default 1000000
Return type Numeric value (double) or character

[:SENSe]:EMISsion:RTYPe <string>

(Read and write) set or query SEM reference power type.

Applicable mode Spectrum Analysis
Parameter reference power type.
PEAK(0) peak
CHANnel(1) channel
Example :EMIS:RTYP PEAK
Query syntax :EMIS:RTYP?
Default PEAK
Return type Numeric value (int) or character

[:SENSe]:EMISsion:MARKer <bool>

(Read and write) set or query SEM peak marker state.

Applicable mode Spectrum Analysis
Parameter Peak marker on/off.
OFF(0) off
ON(1) on
Example :EMIS:MARK ON
Query syntax :EMIS:MARK?
Default OFF
Return type Numeric value (bool) or character

[:SENSe]:EMISsion:Fail?

(Read only) query if SEM test fails. The return value 1 means failed and 0 succeed.

Applicable mode Spectrum Analysis
Parameter None

Example	:EMIS:Fail?
Query syntax	:EMIS:Fail?
Default	None
Return type	Numeric value (int) or character

[:SENSe]:FREQuency:CENTer <num>

(Read and write) set or query center frequency.

Applicable mode	Spectrum Analysis, Interference Analysis**,** AM-FM-PM Demodulation, Power Meter
Parameter	Center frequency value (Hz). Frequency range of spectrum analysis, 0Hz~44.1GHz. Frequency range of interference analysis, 0Hz~44.1GHz. Frequency range of AM-FM-PM analysis, 500Hz~44.1GHz. Frequency range of power meter, 10MHz~40GHz.
Example	:FREQ:CENT 10000
Query syntax	:FREQ:CENT?
Default	22.05GHz (Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation) 10MHz (power meter)
Return type	Numeric value (double) or character

[:SENSe]:FREQuency:CENTer:STEP <num>

(Read and write) set or query step frequency.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Step frequency (Hz). Scope of step frequency, 1Hz~5GHz.
Example	:FREQ:CENT:STEP 10000
Query syntax	:FREQ:CENT:STEP?
Default	1MHz
Return type	Numeric value (double) or character

[:SENSe]:FREQuency:CENTer:STEP:AUTO <bool>

(Read and write) set or query step frequency state.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Step frequency auto state. OFF(0) refers to auto off. ON(1) refers to auto on.
Example	:FREQ:CENT:STEP:AUTO OFF
Query syntax	:FREQ:CENT:STEP:AUTO?
Default	ON
Return type	Numeric value (bool) or character

[:SENSe]:FREQuency:SPAN <num>

(Read and write) set or query span.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter	Span value (Hz). Frequency range of spectrum analysis span, 0Hz~44.1GHz. Frequency range of interference analysis span, 0Hz~44.1GHz. Frequency range of AM-FM-PM analysis span, 0Hz~44.1GHz.
Example	:FREQ:SPAN 10000
Query syntax	:FREQ:SPAN?
Default	44.1GHz (Spectrum Analysis, Interference Analysis) 3MHz (AM-FM-PM Demodulation)

Return type Numeric value (double) or character

[:SENSe]:FREQuency:SPAN:FULL

(Write only) set as full span.

Applicable mode Spectrum Analysis, Interference Analysis
Parameter None
Example :FREQ:SPAN:FULL
Query syntax None
Return type None

[:SENSe]:FREQuency:SPAN:PREVious

(Write only) set as previous span.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter None
Example :FREQ:SPAN:PREV
Query syntax None
Return type None

[:SENSe]:FREQuency:SPAN:ZERO

(Write only) set as zero span.

Applicable mode Spectrum Analysis, Interference Analysis
Parameter None
Example :FREQ:SPAN:ZERO
Query syntax None
Return type None

[:SENSe]:FREQuency:STARt <num>

(Read and write) query or set start frequency.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter Start frequency value (Hz).
Start frequency range of spectrum analysis, 0Hz~44.1GHz.
Start frequency range of interference analysis, 0Hz~44.1GHz.
Example :FREQ:STAR 10000
Query syntax :FREQ:STAR?
Default 0Hz (spectrum analysis, interference analysis)
Return type Numeric value (double) or character

[:SENSe]:FREQuency:STOP <num>

(Read and write) query or set stop frequency.

Applicable mode Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation
Parameter Stop frequency value (Hz).
Stop frequency range of spectrum analysis, 0Hz~44.1GHz.
Stop frequency range of interference analysis, 0Hz~44.1GHz.
Example :FREQ:STOP 10000
Query syntax :FREQ:STOP?
Default 44.1GHz (Spectrum Analysis, Interference Analysis)
Return type Numeric value (double) or character

[:SENSe]:FREQuency:SIGStandard:NAME <string>

(Read and write) query or set signal standard.

Applicable Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation,

mode	Channel Sweep
Parameter	Name of signal standard.
Example	:FREQ:SIGS:NAME P-GSM UL
Query syntax	:FREQ:SIGS:NAME?
Default	None
Return type	Character string

[:SENSe]:FREQuency:SIGStandard:CHANnel <num>

(Read and write) query or set channel number.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Channel Sweep
Parameter	Channel number.
Example	:FREQ:SIGS:CHAN 1
Query syntax	:FREQ:SIGS:CHAN?
Default	Initial channel of signal standard
Return type	Numeric value (int) or character

[:SENSe]:FREQuency:SIGnal:SEARch

(Write only) signal search

Applicable mode	Spectrum Analysis
Parameter	None
Example	:FREQ:SIG:SEAR
Query syntax	:None
Default	None
Return type	None

[:SENSe]:FREQuency:SIGnal:TRAC <BOOL>

(Read and write) Signal track on/off

Applicable mode	Spectrum Analysis
Parameter	ON(1) OFF(0)
Example	: FREQ:SIG:TRAC ON
Query syntax	: FREQ:SIG:TRAC?
Default	OFF
Return type	Numeric value (int) or character

[:SENSe]:FREQuency:RESolution <num>

(Read and write) query or set Power measurement frequency resolution.

Applicable mode	Power meter
Parameter	Resolution (0~3).
Example	:FREQ:RES 1
Query syntax	:FREQ:RES?
Default	2
Return type	Numeric value (int) or character

[:SENSe]:IAMeasure:MODE <string>

(Read and write) query or set interference analysis measurement mode.

Applicable mode	Interference Analysis
Parameter	NORMAL(0) spectrum measurement G(1) waterfall graph RSSI(2) RSSI

Example	:IAM:MODE SG
Query syntax	:IAM:MODE?
Default	SG
Return type	Numeric value (int) or character

[:SENSe]:IAMeasure:TRACe:SPAN <num>

(Read and write) query or set interference analysis span time.

Applicable mode	Interference Analysis
Parameter	Span time (min) Range: 0~1440 min
Example	:IAM:TRAC:SPAN 1
Query syntax	:IAM:TRAC:SPAN?
Default	0
Return type	Numeric value (int) or character

[:SENSe]:IAMeasure:TRACe:SAVE <bool>

(Read and write) query or set interference analysis auto save on/off.

Applicable mode	Interference Analysis
Parameter	Auto save on/off OFF(0) off ON(1) on
Example	:IAM:TRAC:SAVE ON
Query syntax	:IAM:TRAC:SAVE?
Default	OFF
Return type	Numeric value (bool) or character

[:SENSe]:IAMeasure:TRACe:CURS <num>

(Write only) set time cursor.

Applicable mode	Interference Analysis
Parameter	Time cursor (int) Range: 0~288
Example	:IAM:TRAC:CURS 1
Query syntax	None
Default	0
Return type	None

[:SENSe]:IAMeasure:TRACe:RESTart

(Write only) set restart measurement.

Applicable mode	Interference Analysis
Parameter	None
Example	:IAM:TRAC:REST
Query syntax	None
Default	None
Return type	None

[:SENSe]:IAMeasure:TRACe:INTErval <num>

(Read and write) query or set sweep interval of interference analysis.

Applicable mode	Interference Analysis
Parameter	Sweep interval (ms)
Example	:IAM:TRAC:INTE 1000
Query syntax	:IAM:TRAC:INTE?
Default	0

Return type Numeric value (double) or character

[:SENSe]:IF:OUT <bool>

(Read and write) query or set IF output state.

Applicable mode Spectrum Analysis
Parameter IF output state
OFF(0) off
ON(1) on
Example :IF:OUT ON
Query syntax :IF:OUT?
Default OFF
Return type Numeric value (bool) or character

[:SENSe]:IF:SELect <string>

(Read and write) query or set IF selection.

Applicable mode Spectrum Analysis
Parameter 3IF(0) 3IF output
4IF(1) 4IF output
Example :IF:SEL 3IF
Query syntax :IF:SEL?
Default 3IF
Return type Numeric value (int) or character

[:SENSe]:IQ:CAPture[:STATe] <bool>

(Read and write) query or set IQ capture state **(Other functional measurements will be disabled after this function is enabled)**, or use command [:SENSe]:MEASurement.

Applicable mode Spectrum Analysis
Parameter IQ capture measurement state
OFF(0), off.
ON(1), on.
Example :IQ:CAP OFF
Query syntax :IQ:CAP?
Default OFF
Return type Numeric value (bool) or character

[:SENSe]:IQ:CAPture:STARt

(Write only) start IQ capture.

Applicable mode Spectrum Analysis
Parameter None
Example :IQ:CAP:STAR
Query syntax None
Default None
Return type None

[:SENSe]:IQ:CAPture:STOP

(Write only) stop IQ capture.

Applicable mode Spectrum Analysis
Parameter None
Example :IQ:CAP:STOP
Query syntax None
Default None
Return type None

[:SENSe]:IQ:CAPture:TIME <num>

(Read and write) query or set IQ capture time.

Applicable mode	Spectrum Analysis
Parameter	IQ capture time (us)
Example	:IQ:CAPture:TIME 10
Query syntax	:IQ:CAPture:TIME?
Default	40us
Return type	Numeric value (double) or character

[:SENSe]:IQ:CAPture:MODE <num>

(Read and write) query or set IQ capture mode.

Applicable mode	Spectrum Analysis
Parameter	IQ capture mode SING(0) single capture CONT(1) continuous capture
Example	:IQ:CAPture:MODE SING
Query syntax	:IQ:CAPture:MODE?
Default	SING
Return type	Numeric value (int) or character

[:SENSe]:IQ:CAPture:SAMPle <num>

(Read and write) query or set IQ capture sample rate.

Applicable mode	Spectrum Analysis
Parameter	IQ capture sample rate (Hz) Scope:
Example	:IQ:CAP:SAMP 500000
Query syntax	:IQ:CAP:SAMP?
Default	5MHz
Return type	Numeric value (double) or character

[:SENSe]:IQ:CAPture:NAME <string>

(Read and write) query or set IQ capture save name.

Applicable mode	Spectrum Analysis
Parameter	IQ capture save name
Example	:IQ:CAP:NAME ABCD
Query syntax	:IQ:CAP:NAME?
Default	IQCapture
Return type	Character string

[:SENSe]:IQ:CAPture:TRIG <string>

(Read and write) query or set IQ capture trigger type.

Applicable mode	Spectrum Analysis
Parameter	Trigger Type FREE(0) free trigger EXTR(1) external trigger
Example	:IQ:CAP:TRIG FREE
Query syntax	:IQ:CAP:TRIG?
Default	FREE
Return type	Numeric value (int) or character

[:SENSe]:IQ:CAPture:TRIG:SLOPe <string>

(Read and write) query or set IQ capture trigger polarity.

Applicable mode Spectrum Analysis
Parameter Trigger Polarity
POS(0) positive
NEG(1) negative
Example :IQ:CAP:TRIG:SLOP POS
Query syntax :IQ:CAP:TRIG:SLOP?
Default NEG
Return type Numeric value (int) or character

[:SENSe]:IQ:CAPture:TRIG:DELAy <num>

(Read and write) query or set IQ capture trigger delay.

Applicable mode Spectrum Analysis
Parameter Trigger delay (us)
Scope: 1us~500ms
Example :IQ:CAP:TRIG:DELA 10
Query syntax :IQ:CAP:TRIG:DELA?
Default 1us
Return type Numeric value (double) or character

[:SENSe]:IQ:CAPture:TRIG:AMPlitude <num>

(Read and write) query or set IQ capture trigger level.

Applicable mode Spectrum Analysis
Parameter trigger level (mv)
Scope: 0~5V
Example :IQ:CAP:TRIG:AMP 10
Query syntax :IQ:CAP:TRIG:AMP?
Default 1us
Return type Numeric value (double) or character

[:SENSe]:MEASurement <string>

(Read and write) query or set function measurement type, or set through Power measurement on/off. Only one function measurement is available at a time.

Applicable mode Spectrum Analysis
Parameter Function measurement type.

Parameter Setting	Measuring type
NONE(0)	Normal spectrum measurement
FST(1)	Field strength measurement
CHP(2)	Channel power meter
OBW(3)	Occupied bandwidth measurement
ACPR(4)	Adjacent channel power ratio measurement
DEMOD(5)	Audio demodulation measurement
EM(6)	Emission mask measurement
CNR(7)	Carrier to noise

	ratio measurement
IQ(8)	IQ capture measurement

Example :MEAS NONE
Query syntax :MEAS?
Default NONE
Return type Numeric value (int) or character

[:SENSe]:MEASurement:AOFF

(Write only) disable function measurement and switch to normal spectrum measurement.
Applicable mode Spectrum Analysis
Parameter None
Example :MEAS:AOFF
Query syntax None
Default None
Return type None

[:SENSe]:OBW:METHod <string>

(Read and write) set or query method for occupied bandwidth function measurement.
Applicable mode Spectrum Analysis
Parameter Occupied bandwidth measurement method.

PPOW(0)	Percentage
XDB(1)	XdB

Example :OBW:METH XDB
Query syntax :OBW:METH?
Default PPOW
Return type Numeric value (int) or character

[:SENSe]:OBW:OBW?

(Read only) query occupied bandwidth value **(valid after the occupied bandwidth is on and after one sweep).**
Applicable mode Spectrum Analysis
Parameter None
Example :OBW:OBW?
Query syntax :OBW:OBW?
Default None
Return type Number (double) or character (Hz)

[:SENSe]:OBW:PPOW <num>

(Read and write) set or query occupied bandwidth percentage.
Applicable mode Spectrum Analysis
Parameter occupied bandwidth percentage (no unit).
Scope: 10.00%~99.99%
Example :OBW:PPOW 90
Query syntax :OBW:PPOW?
Default 99%
Return type Numeric value (float) or character

[:SENSe]:OBW[:STATe] <bool>

(Read and write) set or query occupied bandwidth function measurement state **(Other functional measurements will be disabled after this function is enabled),** or use command

[:SENSe]:MEASurement.

Applicable mode	Spectrum Analysis	
Parameter	Occupied bandwidth state.	
	OFF(0)	Occupied bandwidth off
	ON(1)	Occupied bandwidth on
Example	:OBW ON	
Query syntax	:OBW?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:OBW:XDB <num>

(Read and write) set or query occupied bandwidth XdB value.

Applicable mode	Spectrum Analysis
Parameter	occupied bandwidth XdB (dB). Scope: -100.0dB~-0.1dB
Example	:OBW:XDB -3
Query syntax	:OBW:XDB?
Default	-3dB
Return type	Numeric value (float) or character

[:SENSe]:PMManager:LIMit:STATe <bool>

(Read and write) set or query Power measurement limit state.

Applicable mode	Power meter	
Parameter	Limit state.	
	OFF(0)	Off
	ON(1)	On
Example	:PMM:LIM:STAT ON	
Query syntax	:PMM:LIM:STAT?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:PMManager:LIMit:UPPer <num>

(Read and write) set or query Power measurement upper limit value.

Applicable mode	Power meter
Parameter	Upper limit value. Scope: -65dBm~30dBm
Example	:PMM:LIM:UPP 10
Query syntax	:PMM:LIM:UPP?
Default	30 dBm
Return type	Numeric value (float) or character

[:SENSe]:PMManager:LIMit:LOWer <num>

(Read and write) set or query Power measurement lower limit value.

Applicable mode	Power meter
Parameter	Lower limit value. Scope: -70dBm~25dBm
Example	:PMM:LIM:LOW 10
Query syntax	:PMM:LIM:LOW?
Default	-70dBm
Return type	Numeric value (float) or character

[:SENSe]:PMManager:MAXHold <bool>

(Read and write) set or query Power measurement maximum hold state.

Applicable mode	Power meter	
Parameter	maximum hold state.	
	OFF(0)	Off
	ON(1)	On
Example	:PMM:MAXH ON	
Query syntax	:PMM:MAXH?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:ROSC:SOUR <string>

(Read and write) set or query 10MHz frequency reference source mode.

Applicable mode	All modes	
Parameter	Frequency reference type.	
	OFF(0)	Frequency reference is internal and off.
	INTernal(0)	Frequency reference is internal and on.
	EXTernal(1)	Frequency reference is external.
Example	:ROSC:SOUR EXTernal	
Query syntax	:ROSC:SOUR?	
Default	INTernal	
Return type	Numeric value (int) or character	

[:SENSe]:POWer[:RF]:ATTenuation <num>

(Read and write) set or query attenuation value.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Field Strength
Parameter	Attenuation value Seven scales covering 0, 10, 20, 30, 40, 50, 60
Example	:POW:ATT 20
Query syntax	:POW:ATT?
Default	10
Return type	Numeric value (int) or character

[:SENSe]:POWer[:RF]:ATTenuation:AUTO <bool>

(Read and write) set or query attenuation auto on/off.

Applicable mode	Spectrum Analysis, Interference Analysis, AM-FM-PM Demodulation, Field Strength
Parameter	Attenuation auto on/off. When the attenuation auto is on, the instrument will automatically set relevant attenuation value based on reference value. OFF(0) refers to manual. ON(1) refers to auto.
Example	:POW:ATT:AUTO ON
Query syntax	:POW:ATT:AUTO?
Default	ON
Return type	Numeric value (bool) or character

[:SENSe]:POWer[:RF]:GAIN[:STATe] <bool>

(Read and write) query or set pre-amplifier state.

Applicable mode	Spectrum Analysis, Field Strength
Parameter	Pre-amplifier ON OFF. OFF(0) refers to off. ON(1) refers to on.
Example	:POW:GAIN OFF
Query syntax	:POW:GAIN?
Default	Off
Return type	Numeric value (bool) or character

[:SENSe]:SWEep:MODE <num>

(Read and write) query or set sweep mode, including linear sweep and list sweep.

Applicable mode	Spectrum Analysis	
Parameter	Sweep mode.	
	LIN(0)	Linear sweep
	LIST(1)	List Sweep
Example	:SWE:MODE LIN	
Query syntax	:SWE:MODE?	
Default	LIN	
Return type	Numeric value (int) or character	

[:SENSe]:SWEep:TIME <num>

(Read and write) query or set sweep time of linear sweep. The sweep time is the time required for selected frequency interval for local oscillator tuning. The sweep time directly affects the time for completing one test, excluding the dead time between one sweep and the next sweep. The sweep time generally changes with the span, resolution bandwidth and video bandwidth. The sweep time is not available when the resolution bandwidth is <=1kHz under spectrum analysis mode.

Applicable mode	Spectrum Analysis, Interference Analysis
Parameter	Sweep time under linear sweep mode (unit: ms).
Example	:SWE:TIME 100
Query syntax	:SWE:TIME?
Default	None
Return type	Numeric value (double) or character

[:SENSe]:SWEep:TIME:AUTO <bool>

(Read and write) set or query sweep time auto on/off of linear sweep. When auto is on, the instrument will use quick sweep speed as possible, or the manual mode is available to increase the sweep time to meet certain measurement needs. The sweep time for manual setting must be equal to or greater than auto sweep time.

Applicable mode	Spectrum Analysis, Interference Analysis	
Parameter	Sweep time auto on/off under linear sweep mode.	
	OFF(0)	Sweep time manual.
	ON(1)	Sweep time auto.
Example	:SWE:TIME:AUTO ON	
Query syntax	:SWE:TIME:AUTO?	
Default	ON	
Return type	Numeric value (bool) or character	

[:SENSe]:SWEep:TRIG <string>

(Read and write) set or query trigger type, including free trigger, video trigger or external trigger.

Applicable mode	Spectrum Analysis	
Parameter	Trigger Type.	
	FREE(0)	Free Trigger
	VIDEO(1)	Video Trigger
	EXTRA(2)	External triggering
Example	:SWE:TRIG FREE	
Query syntax	:SWE:TRIG?	
Default	FREE	
Return type	Numeric value (int) or character	

[:SENSe]:SWEep:TRIG:EXTRa:AMPlitude <num>

(Read and write) set or query external trigger level.

Applicable mode	Spectrum Analysis
Parameter	External trigger level (mv). Scope: 0~5V
Example	:SWE:TRIG:EXTR:AMP 1
Query syntax	:SWE:TRIG:EXTR:AMP?
Default	1.5V
Return type	Numeric value (float) or character

[:SENSe]:SWEep:TRIG:EXTRa:SLOP <string>

(Read and write) set or query external trigger polarity.

Applicable mode	Spectrum Analysis	
Parameter	External trigger polarity.	
	POS(0)	Positive
	NEG(1)	Negative
Example	:SWE:TRIG:EXTR:SLOP POS	
Query syntax	:SWE:TRIG:EXTR:SLOP?	
Default	POS	
Return type	Number (int) or character	

[:SENSe]:SWEep:TRIG:EXTRa:DELAy <num>

(Read and write) set or query external trigger delay.

Applicable mode	Spectrum Analysis
Parameter	External trigger delay (us). Scope: 1us~500ms
Example	:SWE:TRIG:EXTR:DELA 1
Query syntax	:SWE:TRIG:EXTR:DELA?
Default	1us
Return type	Numeric value (float) or character

[:SENSe]:SWEep:TRIG:VIDEo:AMPlitude <num>

(Read and write) set or query video trigger level.

Applicable mode	Spectrum Analysis
Parameter	Video trigger level (dBm). Scope: -120dBm~40dBm
Example	:SWE:TRIG:VIDE:AMP 10
Query syntax	:SWE:TRIG:VIDE:AMP?
Default	-25dBm

Return type Numeric value (float) or character

[:SENSe]:SWEep:POINts <num>

(Read and write) set or query sweep points.

Applicable mode	Spectrum Analysis, Interefence Analysis
Parameter	Sweep points. Scope: 201, 501, 1001, 2001 and 4001
Example	: SWE:POIN 501
Query syntax	: SWE:POIN?
Default	1001
Return type	Numeric value (int) or character

[:SENSe]:TAListen:AVOLume <num>

(Read and write) set or query video demodulation measurement volume.

Applicable mode	Spectrum Analysis
Parameter	Demodulation volume (no unit) Scope: 0~100
Example	:TAL:AVOL 80
Query syntax	:TAL:AVOL?
Default	60
Return type	Numeric value (int) or character

[:SENSe]:TAListen:DMODe <string>

(Read and write) set or query demodulation mode for audio demodulation measurement. The intermittent mode refers to the mode which demodulation will be stopped as per the demodulation time after sweeping one screen of data before sweeping the next screen of data, and will continue this process over and over again; continuous mode refers to the mode that the instrument will continuously demodulate after sweeping one screen of data and will not sweep data again.

Applicable mode	Spectrum Analysis	
Parameter	Demodulation mode.	
	INTer(0)	Continuous OFF
	CONT(1)	Continuous
Example	:TAL:DMOD CONT	
Query syntax	:TAL:DMOD?	
Default	CONT	
Return type	Numeric value (int) or character	

[:SENSe]:TAListen:DSTate <bool>

(Read and write) set or query audio demodulation measurement state **(Other functional measurements will be disabled after this function is enabled)**, or use command [:SENSe]:MEASurement.

Applicable mode	Spectrum Analysis	
Parameter	Demodulation state.	
	OFF(0)	Demodulation off
	ON(1)	Demodulation on
Example	:TAL:DST ON	
Query syntax	:TAL:DST?	
Default	OFF	
Return type	Numeric value (bool) or character	

[:SENSe]:TAListen:DTYPe <string>

(Read and write) set or query audio demodulation measurement demodulation type.

Applicable mode	Spectrum Analysis	
Parameter	Demod type.	
	FM(0)	Frequency modulation
	AM(1)	Amplitude modulation
	USB(2)	Upper-sideband
	LSB(3)	Lower-sideband
Example	:TAL:DTYP FM	
Query syntax	:TAL:DTYP?	
Default	FM	
Return type	Numeric value (int) or character	

[:SENSe]:TAListen:LTIMe <num>

(Read and write) set or query demodulation time for audio demodulation measurement. This **parameter** is valid when the demodulation mode is intermittent mode and the demodulation time refers to the time for staying in demodulation state after one sweep.

Applicable mode	Spectrum Analysis
Parameter	Demodulation time (ms). Scope: 1us~400s
Example	:TAL:LTIM 100
Query syntax	:TAL:LTIM?
Default	100ms
Return type	Numeric value (double) or character

[:SENSe]:FREQ:POIN <num>

(Read and write) set or query point frequency.
Applicable mode Field Strength
Parameter point frequency(Hz).
range:1MHz~44.1GHz
Example :FREQ:POIN 1000000
Query syntax :FREQ:POIN?
Default 500MHz
Return type Numeric value (double) or character

[:SENSe]:FREQ:TRAC <BOOL>

(Read and write) set or query frequency track on.
Applicable mode Field Strength
Parameter frequency track on (Hz).
Example :FREQ:TRAC ON
Query syntax :FREQ:TRAC?
Default OFF
Return type Numeric value (BOOL) or character

[:SENSe]:FREQ:FSCan:STARt <num>

(Read and write) set or query start frequency of frequency scanner.
Applicable mode Field Strength
Parameter start frequency (Hz).
range:1MHz~44.1GHz
Example :FREQ:FSC:STAR 1000000
Query syntax :FREQ:FSC:STAR?
Default 500MHz

Return type Numeric value (double) or character

[:SENSe]:FREQ:FSCan:STEP <num>

(Read and write) set or query step frequency of frequency scanner.
Applicable mode Field Strength
Parameter step frequency (Hz).
range:0Hz~5GHz
Example :FREQ:FSC:STEP 1000000
Query syntax :FREQ:FSC:STEP?
Default 10MHz
Return type Numeric value (double) or character

[:SENSe]:FREQ:FSCan:POINts <num>

(Read and write) set or query scan points of frequency scanner.
Applicable mode Field Strength
Parameter scan points.
range:2~58
Example :FREQ:FSC:POIN 10
Query syntax :FREQ:FSC:POIN?
Default 58
Return type Numeric value (int) or character

[:SENSe]:SWEep:DWELl <num>

(Read and write) set or query dwell time**.**
Applicable mode Field Strength
Parameter dwell time (us).
range:1ms~40s
Example :SWE:DWEL 10000
Query syntax :SWE:DWEL?
Default 100ms
Return type Numeric value (double) or character

[:SENSe]:SWEep:DWELl:INFinite <bool>

(Read and write) set or query dwell time auto on**.**
Applicable mode Field Strength
Parameter dwell time auto on.
Example :SWE:DWEL:INF ON
Query syntax :SWE:DWEL:INF?
Default ON
Return type Numeric value (BOOL) or character

[:SENSe]:SWEep:STAYtime<num>

(Read and write) set or query stay time.
Applicable mode Field Strength
Parameter stay time (us).
range:1ms~40s
Example :SWE:STAY 10000
Query syntax :SWE:STAY?
Default 100ms
Return type Numeric value (double) or character

[:SENSe]:SWEep:STAYtime:STATe <bool>

(Read and write) set or query stay time auto on**.**
Applicable mode Field Strength
Parameter stay time auto on**.**
Example :SWE:STAY:STAT OFF
Query syntax :SWE:STAY:STAT?
Default OFF
Return type Numeric value (BOOL) or character

[:SENSe]:POWer:LIMit:STATe <bool>

(Read and write) set or query limit state.
Applicable mode Field Strength
Parameter limit state.
Example :POW:LIM:STAT OFF
Query syntax :POW:LIM:STAT?
Default OFF
Return type Numeric value (BOOL) or character

[:SENSe]:POWer:LIMit <num>

(Read and write) set or query limit.
Applicable mode Field Strength
Parameter limit (dBm).
range:-174dBm~50dBm
Example :POW:LIM10000
Query syntax :POW:LIM?
Default 0dBm
Return type Numeric value (double) or character

[:SENSe]:DMODe <string>

(Read and write) set or query demodulation mode.
Applicable mode Field Strength
Parameter demodulation mode.
CW(0) continuous wave
FM(1) frequency modulation
AM(2) ampltitude modulation
USB(3) upper sideband
LSB(4) lower sideband
SPEAK(5) speak
Example :DMOD CW
Query syntax :DMOD?
Default CW
Return type Numeric value (int) or character

[:SENSe]:DMODe:VOLume <num>

(Read and write) set or query demodulation volume.
Applicable mode Field Strength
Parameter demodulation volume (0~100).
Example :DMOD:VOLume 50
Query syntax :DMOD:VOLume?
Default 95
Return type Numeric value (int) or character

[:SENSe]:IFBWidth <num>

(Read and write) set or query bandwidth.
Applicable mode Field Strength
Parameter bandwidth (150Hz~150kHz).
Example :IFBW 1000
Query syntax :IFBW?
Default 30kHz
Return type Numeric value (double) or character

[:SENSe]:DATA:POTF:AMPL?

(Read only) query amplitude of point frequency measurment**.**
Applicable mode Field Strength
Parameter amplitude.
Example :DATA:POTF:AMPL?
Query syntax :DATA:POTF:AMPL?
Default None
Return type Numeric value (double) or character

[:SENSe]:DATA:POTF:FIELd?

(Read only) query field strength of point frequency measurement.
Applicable mode Field Strength
Parameter field strength.
Example :DATA:POTF:FIEL?
Query syntax :DATA:POTF:FIEL?
Default None
Return type Numeric value (double) or character

[:SENSe]:DATA:FSCan:AMPL?

(Read only) query amplitude of frequency scanner
Applicable mode Field Strength
Parameter amplitude.
Example :DATA:FSC:AMPL?
Query syntax :DATA:FSC:AMPL?
Default None
Return type Numeric value or character
The character format is xx.xx,xx.xx,…xx.xx\n, in which xx.xx is float data, numbers are separated by "," to indicate interval end and "\n" to indicate end.

[:SENSe]:DATA:FSCan:FIELd?

(Read only) query field strength of frequency scanner**.**
Applicable mode Field Strength
Parameter field strength.
Example :DATA:FSC:FIEL?
Query syntax :DATA:FSC:FIEL?
Default None
Return type Numeric value or character
The character format is xx.xx,xx.xx,…xx.xx\n, in which xx.xx is float data, numbers are separated by "," to indicate interval end and "\n" to indicate end.

[:SENSe]:DATA:LSCan:AMPL?

(Read only) query amplitude of list scanner.

Applicable mode Field Strength
Parameter amplitude.
Example :DATA:LSC:AMPL?
Query syntax :DATA:LSC:AMPL?
Default None
Return type Numeric value or character
The character format is xx.xx,xx.xx,…xx.xx\n, in which xx.xx is float data, numbers are separated by "," to indicate interval end and "\n" to indicate end.

[:SENSe]:DATA:LSCan:FIELd?

(Read only) query field strength of list scanner**.**
Applicable mode Field Strength
Parameter field strength.
Example :DATA:LSC:FIEL?
Query syntax :DATA:LSC:FIEL?
Default None
Return type Numeric value or character
The character format is xx.xx,xx.xx,…xx.xx\n, in which xx.xx is float data, numbers are separated by "," to indicate interval end and "\n" to indicate end.

[:SENSe]:FREQuency:FCOunt?

(Read only) query offset of point frequency measurement.
Applicable mode Field Strength
Parameter offset.
Example :FREQ:FCO?
Query syntax :FREQ:FCO?
Default None
Return type Numeric value (double) or character

[:SENSe]:FST:MEASurement <string>

(Read and write) set or query measurement type.
Applicable mode Field Strength
Parameter measurement type.
POTF(0) point frequency mode
FREQ(1) frequency scanning mode
LIST(2) list scanning mode
Example :FST:MEAS POTF
Query syntax :FST:MEAS?
Default POTF
Return type Numeric value (int) or character

[:SENSe]:FST:PEAK

(Write only) set the peak value of marker**.**
Applicable mode Field Strength
Parameter None.
Example :FST:PEAK
Query syntax None
Default None
Return type None

[:SENSe]:FST:MARKer <bool>

(Read and write) set or query marker state.

Applicable mode Field Strength
Parameter marker state.
Example :FST:MARK ON
Query syntax :FST:MARK?
Default OFF
Return type Numeric value (BOOL) or character

[:SENSe]:FST:INDEx <int>

(Read and write) set or query marker index**.**
Applicable mode Field Strength
Parameter marker index (0~57).
Example :FST:INDEx 20
Query syntax :FST:INDEx?
Default 29
Return type Numeric value (BOOL) or character

:SYSTem:BATTery:STAT?

(Read only) query battery state.
Applicable mode all modes
Parameter None
Example :SYST:BATT:STAT?
Query syntax :SYST:BATT:STAT?
Default None
Return type Numeric value (int) 1 battery and external power source 2 external power source only 3 battery only

:SYSTem:BATTery:VOLume?

(Read only) query battery state.
Applicable mode all modes
Parameter None
Example :SYST:BATT:VOL?
Query syntax :SYST:BATT:VOL?
Default None
Return type Numeric value (int) or character

:SYSTem:GPS <bool>**(Options)**

(Read and write) set or query GPS on/off. When on, the screen will display the longitude and latitude and sea level collected by GPS. This command is an overlapping command. Use ***OPC?** before sending other commands to query if this command is completed.

Applicable mode	All modes	
Parameter	GPS on/off	
	OFF(0)	GPS off
	ON(1)	GPS on
Example	:SYST:GPS ON;	
Query syntax	:SYST:GPS?	
Default	OFF	
Return type	Numeric value (bool) or character	

:SYSTem:GPS:DATA?

(Read only) return current GPS data in the format below: ”<longitude>,<latitude>,<sea level>,<time UTC>”
Applicable mode All modes

Parameter None
Example :SYST:GPS:DATA?
Return
38 28'11.22" N,122 42'13.23" W,152,06/28/2010 23:35:38\n
Return --,--,--,-- n if no data
Query syntax :SYST:GPS:DATA?
Default None
Return type Character

:SYSTem:GPS:RECeive[:STATe]?

(Read only) query GPS receiver state.
Applicable mode All modes
Parameter None
Example :SYST:GPS:REC?
Query syntax :SYST:GPS:REC?
Default None
Return type Numeric value (int) or character
0: receiver off
1: receiver on

:SYSTem:GPS:RST

(Write only) GPS reset. In areas with poor reception where GPS signal cannot be received after a long time even when moving to another place. Under this condition, reset can run new GPS positioning to quickly start search. In this case, reset can be selected to enable the module to find galaxy positioning again.
Applicable mode All modes
Parameter None
Example :SYST:GPS:RST
Query syntax None
Default None
Return type None

:SYSTem:GPS:STATe?

(Read only) query GPS state.
Applicable mode All modes
Parameter None
Example :SYST:GPS:STAT?
Query syntax :SYST:GPS:STAT?
Default None
Return type Numeric value (int) or character
0: Invalid
1: No Differential Fix
2: Differential Fix
3: Invalid PPS
4: Estimated Reckoning

:SYSTem:INFO?

(Read only) query system information.
Applicable mode all modes
Parameter None
Example :SYST:INFO?
Query syntax :SYST:INFO?
Default None

Return type character string

:SYSTem:PWR:SHUTdown <num>

(Read and write) query or set shutdown time.

Applicable mode	All modes
Parameter	Shutdown time. Scope: [1,240] min
Example	:SYST:PWR:SHUT 20
Query syntax	:SYST:PWR:SHUT?
Default	20 minutes
Return type	Numeric value (int) or character

:SYSTem:PWR:SHUTdown:STATe <bool>

(Read and write) query or set shutdown time state. When on, the instrument will automatically shut down when reaching the set shutdown time **(any operation will result in re-timing of shutdown time).**

Applicable mode	All modes
Parameter	Auto shutdown on/off. OFF(0) refers to off. ON(1) refers to on.
Example	:SYST:PWR:SHUT:STAT OFF
Query syntax	:SYST:PWR:SHUT:STAT?
Default	Off
Return type	Numeric value (bool) or character

:SYSTem:PWR:SLEep <num>

(Read and write) query or set sleep time.

Applicable mode	All modes
Parameter	Sleep time Scope: [1,240] min
Example	:SYST:PWR:SLE 20
Query syntax	:SYST:PWR:SLE?
Default	240 minutes
Return type	Numeric value (int) or character

:SYSTem:PWR:SLEEp:STATe <bool>

(Read and write) query or set sleep time state. When on, the instrument will automatically in sleep state when reaching the set sleep time **(any operation will result in re-timing of sleep time).**

Applicable mode	All modes
Parameter	Auto sleep on/off. OFF(0) refers to off. ON(1) refers to on.
Example	:SYST:PWR:SLEE:STAT OFF
Query syntax	:SYST:PWR:SLEE:STAT?
Default	Off
Return type	Numeric value (bool) or character

:SYSTem:TIME <num>,<num>,<num>,<num>,<num>

(Read and write) query or set time.

Applicable mode	All modes
Parameter	Time

	Date & time
Example	:SYST:TIME 2015,12,30,10,30
Query syntax	:SYST:TIME?
Default	None
Return type	Character

:TRACe<n>:DATA?

(Read only) query trace data.

Applicable mode	Spectrum Analysis, Interefence Analysis, AM-FM-PM Analysis
Parameter <n>	Trace number can be set as 1,2 or 3, indicating trace 1, 2, or 3. If not marked, then n represents 1. Trace number is not necessary under Interefence Analysis and AM-FM-PM Analyzer modes.
Example	:TRAC1:DATA?
Query syntax	:TRAC1:DATA?
Default	None
Return type	Numeric value or character

Number format: "#NXXXX data" XXXX is the size of binary data. N is the no. n digit of XXXX. For example: #3512…. means that the size of binary data is 3 and the 3 digits after 3 is 512, indicating this data is followed by a binary data with 512 bytes. The data type of each point of the trace is float Tye, occupying 4 digits.

The character format is xx.xx,xx.xx,…xx.xx/n, in which xx.xx is float data, numbers are separated by "," to indicate interval end and "/n" to indicate end.

Only one trace data is sent each time under Spectrum Analysis and Interferance Analysis modes. Data containing three traces can be sent each time under AM-FM-PM Analyzer mode, totaling 3003 points. Each trace contains 1001 points. The three traces are RF Spectrum, Audio Spectrum and Audio Waveform.

:TRACe<n>:TYPE <string>

(Read and write) query or set trace state.

Applicable mode	Spectrum Analysis
Parameter <n>	Trace number can be set as 1,2 or 3, indicating trace 1, 2, or 3. If not marked, then n represents 1.
<string>	Trace types. CLRW(0), Clear Write MAXH(1), maximum hold MINH(2), minimum hold VIEW(3), view BLANk(4), conceal
Example	:TRAC2:TYPE CLRW
Query syntax	:TRAC2:TYPE?
Default	CLRW
Return type	Numeric value (int) or character

[:SENSe]:GEN:MODE < string >

(Read-write) Query or set the generator mode ON/OFF
Applicable mode Spectrum analysis
Type Read-write
Parameter CW(0) Mode OFF
TRACK(1) Mode ON

Example :GEN:MODE CW
Query syntax :GEN:MODE?
Default CW
Return type Number (int) or character

[:SENSe]:GEN:FREQ:OFFSet <num>

(Read-write) Query or set the generator frequency offset
Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:FREQ:OFFS 1000000
Query syntax :GEN:FREQ:OFFS?
Default 0
Return type Number (double) or character

[:SENSe]:GEN:FREQ:POTF <num>

(Read-write) Query or set the generator frequency
Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:FREQ:POTF 1000000000
Query syntax :GEN:FREQ:POTF?
Default 1000000000
Return type Number (double) or character

[:SENSe]:GEN:AMPL:OUT <num>

(Read-write) Query or set the generator output power
Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:AMPL:OUT -2
Query syntax :GEN:AMPL:OUT?
Default 0
Return type Number (float) or character

[:SENSe]:GEN:AMPL:OFFS <num>

(Read-write) Query or set the generator power offset
Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:AMPL:OFFS 10
Query syntax :GEN:AMPL:OFFS?
Default 0
Return type Number (float) or character

[:SENSe]:GEN:NORMZ:STAT <bool>

(Read-write) Query or set the generator normalization ON/OFF
Applicable mode Spectrum analysis
Type Read-write
Parameter ON(1) Normalization ON

OFF(0) Normalization OFF
Example :GEN:NORM:STAT ON
Query syntax :GEN:NORM:STAT?
Default OFF(0)
Return type Number (bool) or character

[:SENSe]:GEN:NORMZ:RLEV <num>

(Read-write) Query or set the reference level of generator normalization
Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:NORM:RLEV 10
Query syntax :GEN:NORM:RLEV?
Default 0
Return type Number (float) or character

[:SENSe]:GEN:NORM:RPOS <NUM>

(Read-write) Query or set the reference position of generator normalization
Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:NORM:RPOS 5
Query syntax :GEN:NORM:RPOS?
Default 0
Return type Number (int) or character

[:SENSe]:GEN:NORM:PDIV <num>

(Read-write) Query or set the scale/division of generator normalization
Applicable mode Spectrum analysis
Type Read-write
Parameter No
Example :GEN:NORM:PDIV 10
Query syntax :GEN:NORM:PDIV?
Default 0
Return type Number (float) or character

[:SENSe]:GEN:NORM:RTRA < num >

(Read-write) Query or set the generator reference trace ON/OFF
Applicable mode Spectrum analysis
Type Read-write
Parameter SHOW(1) Reference trace ON
HIDE(0) Reference trace OFF
Example :GEN:NORM:RTRA SHOW
Query syntax :GEN:NORM:RTRA?
Default HIDE(0)
Return type Number (int) or character

Section IV Programing instances

This chapter describes how to use different I/O libraries and programming design languages to explain the control of spectrum analyzer. Instrument control is realized through LAN for communication (when using USB port communication, first the USB drive should e installed. Refer to Section I of Chapter II for the instructions of USB drive installation. After USB drive is successfully installed, specific implementation step will be the same as LAN port communication).

1. C/C++ instance

PC should at least have the following configuration:
windows XP operating system
VC6.0 integrated development environment
VISA library of NI
Network card

2. Running of C/C++ design program

To run the C/C++ program, VC6.0 should include the corresponding library file.
The following steps should be followed to use VISA library:
Add the visa.h file to the header file.
Add visatype.h to header file
Add visa32.lib to the project

3. Network design example

To correctly use the instances below, please first confirm the IP address of 4041 Spectrum Analyzer.

1） Use socket and C++ to realize frequency setting and query

Specific realization codes are as follows (the code can be removed, and this section provides only an implementation instance):

Establish dialog box-based MFC project and add the code below in the program:

```
void CSocketTestDlg::Test()
{
    CSocket sockClient;
    bool flag;
    char buff[100];
    if(!AfxSocketInit())
    {
        AfxMessageBox(_T("Initialization failed!" ));
    }
    else
    {
        flag = sockClient.Create();
        if(flag)
        {
            AfxMessageBox(_T("socket creation succeeded!" ));
        }
        else
        {
            AfxMessageBox(_T("socket creation failed!" ));
            sockClient.Close();
        }
    }
    flag = sockClient.Connect(name,5000); //* name is the IP address of spectrum analyzer
    flag = sockClient.Send(":FREQ:STAR 1000000\n",100,0);
    if(!flag)
    {
        AfxMessageBox(_T("send failed!" ));
        exit(0);
```

```
        }
        flag = sockClient.Send("FREQ:STAR?\n",12,0);
        if(!flag)
        {
            AfxMessageBox(_T("send failed!" ));
            exit(0);
        }
        flag = sockClient.Receive(buff,100,0);
        if(!flag)
        {
            AfxMessageBox(_T("receive failed!" ));
            exit(0);
        }
        sockClient.Close();
    }
```

2） Use VISA library and C++ to realize setting and query instructions

The following files should be include

```
#include<visa.h>
#include<afxsock.h>
#include<visa.h>
extern char ResourceStr[50];
ViSession DftRM;
ViSession vi;
//* open the device
ViStatus AV4041_OpenDevice(BOOL bUsb)
{
    char ResourceStr[50];
    if(bUsb)
    {
        strcpy(ResourceStr, "USB0::0x8086::0xA6CD::NI-VISA-0::RAW");
    }
    else
    {
        strcpy(ResourceStr, "TCPIP::x.x.x.x::5000::SOCKET"); //* x.x.x.x is the IP address of spectrum analyzer.
    }
    ViStatus nReturnStatus = 0;
    nReturnStatus = viOpenDefaultRM(&DftRM);
    nReturnStatus = viOpen(DftRM, ResourceStr, VI_NULL,VI_NULL, &vi);
    if(!bUsb)
    {
        viSetAttribute(vi, VI_ATTR_SUPPRESS_END_EN, FALSE);
    }
    return nReturnStatus;
}
//*Set center frequency
ViStatus AV4041_SetFqCent(double Fq)
{
    ViChar Buf[64];
    sprintf(Buf,"%s %.3Lf;", ":FREQ:CENT", Fq);
    ViUInt32 returnCount = 0;
    if( strlen(Buf)! = 0)
        return viWrite(vi, (ViBuf)Buf, strlen(Buf), &returnCount);
    else
        return -1;
}
//*Query center frequency
ViStatus AV4041_QueryFqCent(double& Fq)
{
    ViChar CmdBuf [64];
    ViChar RcvBuf[64];
    ViUInt32 returnCount = 0;
    ViUInt32 actualCount = 0;
    ViStatus nStatus = TRUE;
```

```
        sprintf(CmdBuf,"%s?\n;", ":FREQ:CENT");
        nStatus |= viWrite(vi, (ViBuf)CmdBuf, strlen(CmdBuf), &returnCount);
        nStatus |= viRead(vi, (ViBuf)RcvBuf, 100, &actualCount);
        Fq = *(reinterpret_cast<double*>(RcvBuf));
        return TRUE;
}
```

Chapter II Function Description of Secondary Development Library

For users' convenience, we encapsulate SCPI commands and make it into a dynamic linking library. Users can conveniently set or query 4041 by calling this dynamic linking library, which is suitable for them to create auto test system. (Note: This dynamic library is generated under LabWindows/CVI 2010 programming environment and the communication interface is VISA library of NI.

Section I Drive installation

If using cross-over cable for connection, only IO library above NI-VISA Runtime4.4.1 and above will be installed **(not compatible to VISA library of Agilent).** If using USB cable for connection, then ,in addition to install IO library above NI-VISA Runtime4.4.1 and above, a USB drive should also be installed. 4041USBSETUP.inf should be installed for xp operating system and 4041USBSETUP_vista.inf for win7 operating system. Specific installation process is as follows:

1. click Setup.exe under Volume directory in the disk, and click Add/Remove to start install NI-VISA library, click next until installation is completed.
2. for USB drive, right click 4041USBSETUP.inf installation file and select install until completion.
3. the computer will realize communication by connecting to 4041 through a cross-over cable or USB cable.

Section II Function description

Operating Instructions of Dynamic Linking Library

This dynamic linking library includes three files which are 4041.h, 4041.dll and 4041.lib Under LabWindows/CVI programing environment, by adding these three files into the project, the user can realize control of the instrument by using functions in 4041.h.

Instrument connection – Open device

ViStatus _VI_FUNC AV4041_init(ViRsrc resourceName, ViBoolean IDQuery, ViBoolean resetDevice, ViSession* instrumentHandle)

Function purpose:

Open device

This function is the first function to be called for accessing the instrument as this function completes the initialization operation below:

Open the handle of the module based on the interface designated by **parameter** resourceName and logical address information to establish data channel with the spectrum analyzer.

The returned instrumentHandle is for identifying the module in the future recall of instrument drive functions.

Parameter list:

resourceName

Instrument resource character

4041 resource character string under USB cable connection state can be obtained by the following mode:

```
ViChar ResourceName[256];
ViSession defaultRM;
ViFindList fList;
ViUInt32 num;
viOpenDefaultRM(&defaultRM);
```

viFindRsrc(defaultRM,"USB?::0x045E::0x00CE::?*::RAW",&fList,&num,ResourceName);
The resource character string stored in the ResourceName is the string queried.
TCP connection resource character string is “TCPIP::172.141.11.202::5000::SOCKET”, in which the underlined part is the default IP address of the instrument. If the IP of the instrument is changed, the underlined part shall be the actual IP of the instrument.

IDQuery

ID query parameter. If set as VI_TRUE, the function will query instrument ID and check compliance with the drive.
resetDevice
If this parameter is set as VI_TRUE, the function will reset the device, same as sending *RST command.
instrumentHandle
Instrument handle returned by the function for communication with the instrument.
Return value:
The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Instrument connection – close instrument

ViStatus _VI_FUNC AV4041_close (ViSession instrumentHandle)

Function purpose:

Close the instrument. This function is called to close the instrument after the control is done.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

IEEE488.2 function

Clear instrument state

ViStatus _VI_FUNC AV4041_CLS (ViSession instrumentHandle)

Function purpose:

Clear the state of the instrument, i.e.: Clear error queue and all event registers and cancel *OPC command and query command to be processed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Query instrument identification number

ViStatus _VI_FUNC AV4041_QueryIDN (ViSession instrumentHandle, char IDN[])

Function purpose:

Inquire the character string of the instrument identification number.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

IDN

Instrument identification character string sent from the instrument, in the form of "CEYEAR,4041,XXXXXX,X.X.X" under normal state. XXXXXX is the serial number of the instrument, X.X.X is the version number of current host program.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Operation complete command.

ViStatus _VI_FUNC AV4041_OPC (ViSession instrumentHandle)

Function purpose:

After completing all overlapping commands to be processed **(for example: trigger one sweep command)**, set the OPC bit of standard event state register.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Operation complete query

ViStatus _VI_FUNC AV4041_QueryOPC (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

1 will be returned when all overlapping commands to be processed are completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Query value after operation. If the value is 1, this means that overlapping commands are completed.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Reset

ViStatus _VI_FUNC AV4041_Reset (ViSession instrumentHandle)

Function purpose:

Restore current work mode of the instrument to known default state known as default state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Wait

ViStatus _VI_FUNC AV4041_WAI (ViSession instrumentHandle)

Function purpose:

Wait for processing of all overlapping commands before processing of new commands.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Universal functions for all measurement modes

Mode – query available instrument mode

ViStatus _VI_FUNC AV4041_QueryInstCatalog (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query available instrument mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Available instrument mode.

The 0 bit is the bit for spectrum analysis test, 1 (mandatory)

The 1 bit is the bit for AM-FM-PM demodulation test, 1 is settable (optional) and 0 is not settable

The 2 bit is the bit for interference analysis test, 1 is settable (optional) and 0 is not settable

The 3 bit is the bit for power measurement test, 1 is settable (optional) and 0 is not settable

The 4 bit is the bit for channel sweep test, 1 is settable (optional) and 0 is not settable

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Mode – set instrument mode

ViStatus _VI_FUNC AV4041_SetInstSel (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set instrument mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Instrument mode.

1: spectrum analysis mode

2: interference analysis mode

3: AM-FM-PM analysis mode

4: Power measurement mode

5: channel sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Mode – query instrument mode

ViStatus _VI_FUNC AV4041_QueryInstSel (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query instrument mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Instrument mode.

1: spectrum analysis mode

2: interference analysis mode

3: AM-FM-PM analysis mode

4: Power measurement mode

5: channel sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – set data type

ViStatus _VI_FUNC AV4041_SetFormat (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set data type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Data type.

0: character type

1: binary type

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query data type

ViStatus _VI_FUNC AV4041_QueryFormat (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query data type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Data type.

0: character type

1: binary type

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete state file

ViStatus _VI_FUNC AV4041_DeleteStateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete state file under current mode **(if the file does not exist, the command will be invalid.**

The command be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all state files

ViStatus _VI_FUNC AV4041_DeleteAllStateFile (ViSession instrumentHandle)

Function purpose:

delete all state files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall state file

ViStatus _VI_FUNC AV4041_LoadStateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall state file under current mode **(this command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – save state file

ViStatus _VI_FUNC AV4041_StoreStateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Store state file under current mode **(if the file does not exist, the command will be invalid.**

The command be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – set storage location

ViStatus _VI_FUNC AV4041_SetLocation (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set storage location.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: internal.

1: SD card.

2: USB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – query storage location

ViStatus _VI_FUNC AV4041_QueryLocation (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query current storage location

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: internal.

1: SD card.

2: USB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store screen copy

ViStatus _VI_FUNC AV4041_StoreScreen (ViSession instrumentHandle, char chStr[])

Function purpose:

Screen copy, save current screenshot as file **(file will be overwritten if exist and only valid to current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – set frequency reference

ViStatus _VI_FUNC AV4041_SetRoscSourc (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set 10MHz frequency reference source mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Frequency reference type.

0: frequency reference is internal

1: frequency reference is external.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – query frequency reference

ViStatus _VI_FUNC AV4041_QueryRoscSource (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query 10MHz frequency reference source mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Frequency reference type.

0: frequency reference is internal

1: frequency reference is external.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS –set GPS on/off

ViStatus _VI_FUNC AV4041_SetGPSOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set GPS on/off, when on, the screen will display the longitude and latitude and sea level collected by GPS chip. This command is an overlapping command. Before sending other commands, use **AV4041_QueryOPC()** to query if this command is completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – query GPS on/off

ViStatus _VI_FUNC AV4041_QueryGPSOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query GPS on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – query GPS state

ViStatus _VI_FUNC AV4041_QueryGPSState (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query GPS state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

GPS state.

0: Invalid

1: No Differential Fix

2: Differential Fix

3: Invalid PPS

4: Estimated Reckoning

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – query GPS receiver state

ViStatus _VI_FUNC AV4041_QueryGPSReceiveState (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query GPS receiver state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

GPS receiver state.

0: no receiver data

1: receiver data

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – GPS reset

ViStatus _VI_FUNC AV4041_GPSReset (ViSession instrumentHandle)

Function purpose:

GPS reset. In areas with poor reception where GPS signal cannot be received after a long time even when moving to another place. Under this condition, reset can run new GPS positioning to quickly start search. In this case, reset can be selected to enable the module to find galaxy positioning again.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – GPS – query GPS data

ViStatus _VI_FUNC AV4041_QueryGPSData (ViSession instrumentHandle, ViChar chStr[])

Function purpose:

Query data collected by GPS chip, and return current GPS data in the format below: ”<longitude>,<latitude>,<sea level>,<timeUTC>”.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Returned GPS data.

Example: Return ”38 28’11.22” N,122 42’13.23” W,152,06/28/2010 23:35:38 n” when there is data

Return ”--,--,--,-- n” if no data

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – shutdown – set auto shutdown on/off

ViStatus _VI_FUNC AV4041_SetAutoShutdownOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set auto shutdown on/off. When on, the instrument will shutdown automatically once reaching the set shutdown time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – shutdown – query auto shutdown on/off

ViStatus _VI_FUNC AV4041_QueryAutoShutdownOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query shutdown auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – shutdown – set shutdown time

ViStatus _VI_FUNC AV4041_SetShutdown (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set shutdown time. When shutdown auto is on, the instrument will automatically shutdown once reaching the set shutdown time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Shutdown time (minute), scope: 1~240 min.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – shutdown – query shutdown time

ViStatus _VI_FUNC AV4041_QueryShutdown (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query shutdown time

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Shutdown time (min)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – sleep – set sleep auto on/off

ViStatus _VI_FUNC AV4041_SetAutoSleepOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set sleep auto on/off. When auto sleep is on, when reaching the set sleep time, the instrument will automatically go to sleep and turn off screen display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – sleep – query sleep auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoSleepOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query sleep auto on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – sleep – set sleep time

ViStatus _VI_FUNC AV4041_SetSleep (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set sleep time. When auto sleep is on, when reaching the set sleep time, the instrument will automatically go to sleep and turn off screen display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sleep time (minute), scope: 1~240 min.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – sleep – query sleep time

ViStatus _VI_FUNC AV4041_QuerySleep (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query sleep time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sleep time (min)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – set title

ViStatus _VI_FUNC AV4041_SetTitle (ViSession instrumentHandle, char chStr[])

Function purpose:

set title.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Title name

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – set title on/off

ViStatus _VI_FUNC AV4041_SetTitleOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set title on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – query title on/off

ViStatus _VI_FUNC AV4041_QueryTitleOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query title on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – set display mode

ViStatus _VI_FUNC AV4041_SetShowMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: default mode.

1: black and white mode.

2: night vision mode.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – query display mode

ViStatus _VI_FUNC AV4041_QueryShowMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Location.

0: default mode.

1: black and white mode.

2: night vision mode.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – set brightness auto adjustment on/off

ViStatus _VI_FUNC AV4041_SetBrightOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set brightness auto adjustment on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – query brightness auto adjustment on/off

ViStatus _VI_FUNC AV4041_QueryBrightOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query brightness auto adjustment on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System - set brightness level.

ViStatus _VI_FUNC AV4041_SetBright (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set brightness level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Brightness level, scope: 0~4.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

System – query brightness level

ViStatus _VI_FUNC AV4041_QueryBright (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query brightness level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Brightness level.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Spectrum Analysis Mode Functions

Frequency – set center frequency

ViStatus _VI_FUNC AV4041_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query center frequency

ViStatus _VI_FUNC AV4041_QueryCntFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency

ViStatus _VI_FUNC AV4041_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, scope: 1Hz~5GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency

ViStatus _VI_FUNC AV4041_QueryStepFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency auto on/off

ViStatus _VI_FUNC AV4041_SetAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set step frequency auto on/off. When auto is on, the step frequency is 1MHz. when off, the step frequency can be 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query step frequency auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set span

ViStatus _VI_FUNC AV4041_SetSpan (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query span

ViStatus _VI_FUNC AV4041_QuerySpan (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – full span

ViStatus _VI_FUNC AV4041_SetFullSpan (ViSession instrumentHandle)

Function purpose:

Set as full span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – zero span

ViStatus _VI_FUNC AV4041_SetZeroSpan (ViSession instrumentHandle)

Function purpose:

Set as zero span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – previous span

ViStatus _VI_FUNC AV4041_SetLastSpan (ViSession instrumentHandle)

Function purpose:

Set as previous span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set start frequency

ViStatus _VI_FUNC AV4041_SetSttFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query start frequency

ViStatus _VI_FUNC AV4041_QuerySttFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set stop frequency

ViStatus _VI_FUNC AV4041_SetStpFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query stop frequency

ViStatus _VI_FUNC AV4041_QueryStpFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard name

ViStatus _VI_FUNC AV4041_SetSIGStandard (ViSession instrumentHandle, char* standard)

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard name

ViStatus _VI_FUNC AV4041_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard channel number

ViStatus _VI_FUNC AV4041_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard channel number

ViStatus _VI_FUNC AV4041_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set zero span IF output on/off

ViStatus _VI_FUNC AV4041_SetIFOutOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set zero span IF output on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query zero span IF output on/off

ViStatus _VI_FUNC AV4041_QueryIFOutOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query zero span IF output on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set zero span IF output IF selection

ViStatus _VI_FUNC AV4041_SetIFOutSelect (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set zero span IF output IF selection.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Intermediate frequency selection

0: 3IF, 1: 4IF.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query zero span IF output IF selection

ViStatus _VI_FUNC AV4041_QueryIFOutSelect (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query zero span IF output IF selection.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Intermediate frequency selection

0: 3IF, 1: 4IF.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference level

ViStatus _VI_FUNC AV4041_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level. Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference level

ViStatus _VI_FUNC AV4041_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value). Reference level value is related to current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference position

ViStatus _VI_FUNC AV4041_SetRefPos (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting of reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Reference position, scope: -10~10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference position

ViStatus _VI_FUNC AV4041_QueryRefPos (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

reference position value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set attenuation

ViStatus _VI_FUNC AV4041_SetAtt (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query attenuation

ViStatus _VI_FUNC AV4041_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude - set attenuation auto on/off

ViStatus _VI_FUNC AV4041_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude - query attenuation auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set scale/division

ViStatus _VI_FUNC AV4041_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division. Not available when the spectrum analysis mode is linear scale type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division (0.1dB~20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query scale/division

ViStatus _VI_FUNC AV4041_QueryScalePDiv (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set scale type

ViStatus _VI_FUNC AV4041_SetScaleType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set scale type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scale Type

0: logarithmic scale, 1: linear scale.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query scale type

ViStatus _VI_FUNC AV4041_QueryScaleType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query scale type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scale Type

0: logarithmic scale, 1: linear scale.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude –set unit

ViStatus _VI_FUNC AV4041_SetAmpUnit (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	In dBm.
DBMV(1)	Unit: dBmV
DBUV(2)	Unit: dBuV
V(3)	Unit: Volts
W(4)	Unit: Walts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query unit

ViStatus _VI_FUNC AV4041_QueryAmpUnit (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	In dBm.
DBMV(1)	Unit: dBmV
DBUV(2)	Unit: dBuV
V(3)	Unit: Volts
W(4)	Unit: Walts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set pre-amplifier on/off

ViStatus _VI_FUNC AV4041_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query pre-amplifier on/off

ViStatus _VI_FUNC AV4041_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set resolution bandwidth

ViStatus _VI_FUNC AV4041_SetRBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set resolution bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query resolution bandwidth

ViStatus _VI_FUNC AV4041_QueryRBW (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query resolution bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set video bandwidth

ViStatus _VI_FUNC AV4041_SetVBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set video bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query video bandwidth

ViStatus _VI_FUNC AV4041_QueryVBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query video bandwidth of linear sweep under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set resolution bandwidth auto on/off

ViStatus _VI_FUNC AV4041_SetAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set resolution bandwidth auto on/off. When on, the resolution bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query resolution bandwidth auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query resolution bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set video bandwidth auto on/off

ViStatus _VI_FUNC AV4041_SetAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set video bandwidth auto on/off. When on, the video bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth –query video bandwidth auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query video bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set SPAN/RBW

ViStatus _VI_FUNC AV4041_SetSR100 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set SPAN/RBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW, scope: 1~500.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query SPAN/RBW

ViStatus _VI_FUNC AV4041_QuerySR100 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query SPAN/RBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set RBW/VBW

ViStatus _VI_FUNC AV4041_SetRV300 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set RBW/VBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

RBW/VBW, scope: 1~100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query RBW/VBW

ViStatus _VI_FUNC AV4041_QueryRV300 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query RBW/VBW under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average on/off

ViStatus _VI_FUNC AV4041_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average on/off

ViStatus _VI_FUNC AV4041_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average count

ViStatus _VI_FUNC AV4041_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1~1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average count

ViStatus _VI_FUNC AV4041_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – clear average count

ViStatus _VI_FUNC AV4041_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query current average count

ViStatus _VI_FUNC AV4041_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – set detector type

ViStatus _VI_FUNC AV4041_SetDetectorType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive Peak
NEGative(1)	Negative Peak
SAMPle(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – query detector type

ViStatus _VI_FUNC AV4041_QueryDetectorType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive

	Peak
NEGative(1)	Negative Peak
SAMPle(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – set detector auto on/off

ViStatus _VI_FUNC AV4041_SetAutoDetectorOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set detector auto on/off. When on, the instrument will automatically select detector type based on different measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – query detector auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoDetectorOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query detector auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep type

ViStatus _VI_FUNC AV4041_SetSwpType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the scanning type. This is an overlapping command. Use **AV4041_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep type

ViStatus _VI_FUNC AV4041_QuerySwpType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

0: single sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – trigger single sweep

ViStatus _VI_FUNC AV4041_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use **AV4041_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep mode

ViStatus _VI_FUNC AV4041_SetSwpMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set sweep mode, including linear sweep and list sweep modes.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep mode.

0: linear sweep mode

1: list sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep mode

ViStatus _VI_FUNC AV4041_QuerySwpMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep mode, including linear sweep and list sweep modes. The user can edit the list segment to observe the signal of several sweep segments.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep mode.

0: linear sweep mode

1: list sweep mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep time

ViStatus _VI_FUNC AV4041_SetSwpTime (ViSession instrumentHandle, ViReal64

dbVal)

Function purpose:

Set sweep time under current mode. The sweep time is the time required for selected frequency interval for local oscillator tuning. The sweep time directly affects the time for completing one test, excluding the dead time between one sweep and the next sweep. The sweep time generally changes with the span, resolution bandwidth and video bandwidth. The sweep time is not available when the resolution bandwidth is <=1kHz under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep time

ViStatus _VI_FUNC AV4041_QuerySwpTime (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query sweep time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep time auto on/off

ViStatus _VI_FUNC AV4041_SetAutoSwpTimeOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set sweep time auto on/off. When auto is on, the instrument will use quick sweep speed as possible, or the manual mode is available to increase the sweep time to meet certain measurement needs. The sweep time for manual setting must be equal to or greater than auto sweep time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep time auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoSwpTimeOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query sweep time auto on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – list add default segment

ViStatus _VI_FUNC AV4041_ListAddSeg (ViSession instrumentHandle)

Function purpose:

Add default sweep segment to the list edit under current mode.

Starting frequency	1GHz
Stop frequency	2GHz
Sweep Points	51
Resolution Bandwidth	1MHz
Video Bandwidth	30kHz
On/off	Off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – list delete segment

ViStatus _VI_FUNC AV4041_ListDelSeg (ViSession instrumentHandle, ViInt32 nval)

Function purpose:

Delete segment from list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nval

Segment index

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – clear list

ViStatus _VI_FUNC AV4041_ListClear (ViSession instrumentHandle)

Function purpose:

Delete all list edit segments under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – add segment

ViStatus _VI_FUNC AV4041_ListAdd (ViSession instrumentHandle, ViReal64 startfrequency, ViReal64 stopfrequency, ViInt32 rbw, ViInt32 vbw, ViInt32 sweepPoints, ViInt32 on)

Function purpose:

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

startfrequency

Start frequency (0~44.1GHz)

stopfrequency

Stop frequency (0~44.1GHz)

rbw

Resolution bandwidth (0~10MHz)

vbw

Video bandwidth (0~10MHz)

sweepPoints

Sweep points (51~501)

on

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – edit segment

ViStatus _VI_FUNC AV4041_ListEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 startfrequency, ViReal64 stopfrequency, ViInt32 rbw, ViInt32 vbw, ViInt32 sweepPoints, ViInt32 on)

Function purpose:

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Index

Segment index

startfrequency

Start frequency (0~44.1GHz)

stopfrequency

Stop frequency (0~44.1GHz)

rbw

Resolution bandwidth (0~10MHz)

vbw

Video bandwidth (0~10MHz)

sweepPoints

Sweep points (51~501)

on

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set trigger mode

ViStatus _VI_FUNC AV4041_SetTrigType (ViSession instrumentHandle,ViInt32 nval)

Function purpose:

Set trigger mode, including free trigger, video trigger and external trigger modes.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: video trigger

2: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query trigger mode

ViStatus _VI_FUNC AV4041_QueryTrigType (ViSession instrumentHandle,ViInt32 nval[])

Function purpose:

Query trigger mode, including free trigger, video trigger and external trigger modes.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: video trigger

2: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set video trigger level

ViStatus _VI_FUNC AV4041_SetTrigVideoAMP (ViSession instrumentHandle,

ViReal64 dVal)

Function purpose:

Set video trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query video trigger level

ViStatus _VI_FUNC AV4041_QueryTrigVideoAMP (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query video trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set external trigger level

ViStatus _VI_FUNC AV4041_SetTrigExtraAMP (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set external trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (mv).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query external trigger level

ViStatus _VI_FUNC AV4041_QueryTrigExtraAMP (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query external trigger level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (mv).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep- set external trigger polarity

ViStatus _VI_FUNC AV4041_SetTrigExtraSlop (ViSession instrumentHandle,ViInt32 nval)

Function purpose:

set external trigger polarity.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query external trigger polarity

ViStatus _VI_FUNC AV4041_QueryTrigExtraSlop (ViSession instrumentHandle,ViInt32 nval[])

Function purpose:

query external trigger polarity

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set external trigger delay

ViStatus _VI_FUNC AV4041_SetTrigExtraDelay (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set external trigger delay.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query external trigger delay

ViStatus _VI_FUNC AV4041_QueryTrigExtraDelay (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query external trigger delay.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query trace data

ViStatus _VI_FUNC AV4041_QueryTraceData (ViSession instrumentHandle,ViInt32 size[], ViReal64 data[],ViInt32 index)

Function purpose:

Query trace data under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

size

Size of trace data received.

data

Trace data storage array point. The array should meet the size of trace data received.

index

Trace number can be set as 1, 2 or 3, indicating query trace data of trace 1, 2, or 3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Trace - set trace state

ViStatus _VI_FUNC AV4041_SetTraceType (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nTrace)

Function purpose:

Set trace state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

3: hold trace

4: conceal trace

nTrace

Trace number can be set as 1, 2 or 3, indicating query trace data of trace 1, 2, or 3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Trace – query trace state

ViStatus _VI_FUNC AV4041_QueryTraceType (ViSession instrumentHandle, ViInt32

nVal[],ViInt32 nTrace)

Function purpose:

Query trace state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

3: hold trace

4: conceal trace

nTrace

Trace number can be set as 1, 2 or 3, indicating query trace data of trace 1, 2, or 3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set alarm on/off

ViStatus _VI_FUNC AV4041_SetAlarmOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails, the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query alarm on/off

ViStatus _VI_FUNC AV4041_QueryAlarmOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails,

the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set lower limit display on/off

ViStatus _VI_FUNC AV4041_SetLowLmtDispOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set lower limit display on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query lower limit display on/off

ViStatus _VI_FUNC AV4041_QueryLowLmtDispOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query lower limit display on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set upper limit display on/off

ViStatus _VI_FUNC AV4041_SetUppLmtDispOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set upper limit display on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query upper limit display on/off

ViStatus _VI_FUNC AV4041_QueryUppLmtDispOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query upper limit display on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set lower limit test on/off

ViStatus _VI_FUNC AV4041_SetLowLmtTestOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set lower limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query lower limit test on/off

ViStatus _VI_FUNC AV4041_QueryLowLmtTestOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query lower limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set upper limit test on/off

ViStatus _VI_FUNC AV4041_SetUppLmtTestOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set upper limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query upper limit test on/off

ViStatus _VI_FUNC AV4041_QueryUppLmtTestOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query upper limit test on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set lower limit margin

ViStatus _VI_FUNC AV4041_SetLowLmtMargin (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

set lower limit margin.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin (0dB~40dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query lower limit margin

ViStatus _VI_FUNC AV4041_QueryLowLmtMargin (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query lower limit margin.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set upper limit margin

ViStatus _VI_FUNC AV4041_SetUppLmtMargin (ViSession instrumentHandle,ViReal64 dbVal)

Function purpose:

set upper limit margin.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin (-40dB~0dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query upper limit margin

ViStatus _VI_FUNC AV4041_QueryUppLmtMargin (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

query upper limit margin.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Margin

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – lower limit add default point

ViStatus _VI_FUNC AV4041_LowLmtAddPt (ViSession instrumentHandle)

Function purpose:

lower limit add default point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – lower limit delete default point

ViStatus _VI_FUNC AV4041_LowLmtDelPt (ViSession instrumentHandle)

Function purpose:

lower limit delete default point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – clear all lower limit points

ViStatus _VI_FUNC AV4041_LowLmtClear (ViSession instrumentHandle)

Function purpose:

Clear all edit points in lower limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – lower limit edit point

ViStatus _VI_FUNC AV4041_LowerLimitEdit (ViSession instrumentHandle,ViInt32 index, ViReal64 frequency, ViReal64 amplitude)

Function purpose:

Set lower limit edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Limit point index.

frequency

Frequency (Hz)(0~44.1GHz).

amplitude

Amplitude (dBm)(-174~50dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – upper limit add default point

ViStatus _VI_FUNC AV4041_UppLmtAddPt (ViSession instrumentHandle)

Function purpose:

upper limit add default point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – upper limit delete current point

ViStatus _VI_FUNC AV4041_UppLmtDelPt (ViSession instrumentHandle)

Function purpose:

Upper limit delete current point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – upper limit point clear

ViStatus _VI_FUNC AV4041_UppLmtClear (ViSession instrumentHandle)

Function purpose:

Clear all edit points in the upper limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – lower limit edit point

ViStatus _VI_FUNC AV4041_UpperLimitEdit (ViSession instrumentHandle,ViInt32 index, ViReal64 frequency, ViReal64 amplitude)

Function purpose:

Set upper limit edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Limit point index.

frequency

Frequency (Hz)(0~44.1GHz).

amplitude

Amplitude (dBm)(-174~50dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set market state

ViStatus _VI_FUNC AV4041_SetMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState)

Function purpose:

set marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker state

ViStatus _VI_FUNC AV4041_QueryMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState[])

Function purpose:

Query marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker- activate marker

ViStatus _VI_FUNC AV4041_SetMkrActive (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Activate marker under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker- marker function (marker ->)

ViStatus _VI_FUNC AV4041_SetMkrTo (ViSession instrumentHandle, ViInt32 nVal ,ViInt32 nSetIdx)

Function purpose:

Set marker function under current mode (marker -> for spectrum analysis mode).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nSetIdx

Instrument mode	nSetIdx	Function
Spectrum Analysis (non-zero span)	0	Marker -> start frequency (set marker frequency as start frequency)
	1	Marker -> stop frequency (set marker frequency as stop frequency)
	2	Marker -> center frequency (set marker frequency as center frequency)
	3	Marker -> step frequency (set marker frequency as

		step frequency)
Spectrum Analysis (zero span)	0	Marker -> start frequency (set marker index as minimum index)
	1	Marker -> stop frequency (set marker index as maximum index)
	2	Marker -> center frequency (set marker index as center index)
	3	Marker -> step frequency (set marker frequency as step frequency)

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – disable all markers

ViStatus _VI_FUNC AV4041_DisableAllMarkers (ViSession instrumentHandle)

Function purpose:

Disable all markers under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker –set marker X value

ViStatus _VI_FUNC AV4041_MoveMarker (ViSession instrumentHandle, ViInt32 nVal, ViReal64 dbVal,)

Function purpose:

set marker X value under current mode, when marker is an offset marker, the X value can be negative.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Marker X value, time in ms and frequency in Hz.

Instrument mode	Parameter unit

Spectrum Analysis (non-zero span)	Hz
Spectrum Analysis (zero span)	us

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker X value

ViStatus _VI_FUNC AV4041_QueryMarker (ViSession instrumentHandle, ViInt32 markerIndex, ViReal64 markerPosition[],ViReal64 markerAmplitude[])

Function purpose:

Query marker X value and Y value under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

markerPosition

Marker X value, time in ms and frequency in Hz.

Instrument mode	Parameter unit
Spectrum Analysis (non-zero span)	Hz
Spectrum Analysis (zero span)	us

markerIndex

Marker index, 1~6 available.

markerAmplitude

Marker Y value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker - search

ViStatus _VI_FUNC AV4041_SetMkrSearch (ViSession instrumentHandle, ViInt32 nVal,ViInt32 type)

Function purpose:

Move marker to maximum, minimum, peak, secondary peak, left adjacent peak and right adjacent peak.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index can be set from 1 to 6 1~6 available, indicating marker 1, 2, 3 and 4.

type

Search type.

1 Maximum value

2 Minimum value

3 Peak

4 Secondary peak

5 left adjacent peak

6 right adjacent peak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker –set marker count on

ViStatus _VI_FUNC AV4041_SetMkrCountOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn)

Function purpose:

Set the marker count state under current mode and set marker will be switched to normal marker state.

Notes: Only one marker counter can be enabled now.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker –query marker count on

ViStatus _VI_FUNC AV4041_QueryMkrCountOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn[])

Function purpose:

Query the marker count state under current mode

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker count frequency

ViStatus _VI_FUNC AV4041_QueryMkrCountFreq (ViSession instrumentHandle,ViInt32 nVal,
ViReal64 dbVal[])

Function purpose:

Query marker count frequency **(invalid if the count is disabled or does not start count).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

dbVal

Returned counter frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set noise marker on

ViStatus _VI_FUNC AV4041_SetMkrNoiseOn (ViSession instrumentHandle,ViInt32 nVal,ViBoolean bOn)

Function purpose:

Set noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query noise marker on

ViStatus _VI_FUNC AV4041_QueryMkrNoiseOn (ViSession instrumentHandle, ViInt32 nVal, ViBoolean bOn[])

Function purpose:

Query noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set peak track on/off

ViStatus _VI_FUNC AV4041_SetPeakTrack (ViSession instrumentHandle,ViBoolean bOn)

Function purpose:

Set peak track on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query peak track on/off

ViStatus _VI_FUNC AV4041_QueryPeakTrack (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query peak track on/off under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set function measurement

ViStatus _VI_FUNC AV4041_SetMeasFunc (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set function measurement type, or the function measurement type can be directly set by the function measurement state. Only one function measurement is allowed at a time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Function measurement type.

Parameter Setting	Measuring type
NONE(0)	Normal spectrum measurement
FST(1)	Field strength measurement
CHP(2)	Channel power meter
OBW(3)	Occupied bandwidth measurement
ACPR(4)	Adjacent channel power ratio measurement
DEMOD(5)	Audio demodulation measurement
EM(6)	Emission mask measurement
CNR(7)	Carrier to noise ratio measurement
IQ(8)	IQ capture measurement

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query function measurement.

ViStatus _VI_FUNC AV4041_QueryMeasFunc (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query function measurement type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Function measurement type.

Parameter Setting	Measuring type
NONE(0)	Normal spectrum measurement
FST(1)	Field strength measurement
CHP(2)	Channel power meter
OBW(3)	Occupied bandwidth measurement
ACPR(4)	Adjacent channel power ratio measurement
DEMOD(5)	Audio demodulation measurement
EM(6)	Emission mask measurement
CNR(7)	Carrier to noise ratio measurement
IQ(8)	IQ capture measurement

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – turn of measurement

ViStatus _VI_FUNC AV4041_SetMeasFuncAOff (ViSession instrumentHandle)

Function purpose:

Turn off current function measurement and switch to normal spectrum measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Measurement – field strength – set antenna factor off

ViStatus _VI_FUNC AV4041_SetAntOff (ViSession instrumentHandle)

Function purpose:

Set antenna factor loading off and set as antenna factor free state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – field strength – set field strength on/off

ViStatus _VI_FUNC AV4041_SetFstOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set field strength function measurement on/off, or use function **AV4041_SetMeasFunc()** to enable **(by enabling this function measurement, other function measurement will be disabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – field strength – query field strength on/off

ViStatus _VI_FUNC AV4041_QueryFstOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query field strength on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – field strength – edit antenna factor – add default point

ViStatus _VI_FUNC AV4041_AntennaAddDefault (ViSession instrumentHandle)

Function purpose:

edit antenna factor to add default point. Frequency: 1GHz antenna factor value: 0dB

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – field strength – edit antenna factor – delete point

ViStatus _VI_FUNC AV4041_AntennaDelete (ViSession instrumentHandle,ViInt32 index)

Function purpose:

edit antenna factor to delete point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – field strength – edit antenna factor – edit point

ViStatus _VI_FUNC AV4041_AntennaEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 frequency, ViReal64 factor)

Function purpose:

edit antenna factor to edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

frequency

Frequency (0~44.1GHz)

factor

Antenna factor value (-200~200dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – field strength – edit antenna factor – add point

ViStatus _VI_FUNC AV4041_AntennaAdd (ViSession instrumentHandle, ViReal64 frequency, ViReal64 factor)

Function purpose:

edit antenna factor to add point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

frequency

Frequency (0~44.1GHz)

factor

Antenna factor value (-200~200dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – set channel power on/off

ViStatus _VI_FUNC AV4041_SetCHPOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set channel power on/off, or use function **AV4041_SetMeasFunc()** to enable **(by enabling this function measurement, other function measurement will be disabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – query channel power state

ViStatus _VI_FUNC AV4041_QueryCHPOn (ViSession instrumentHandle, ViBoolean

bOn[])

Function purpose:

Query channel power state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – set channel power bandwidth

ViStatus _VI_FUNC AV4041_SetCHPIBW (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

set channel power bandwidth for channel power function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz), scope: 100Hz~44.1GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – query channel power bandwidth

ViStatus _VI_FUNC AV4041_QueryCHPIBW (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

query channel power bandwidth for channel power function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – query channel power value

ViStatus _VI_FUNC AV4041_QueryTPWR (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query channel power value for channel power function measurement under spectrum analysis mode **(valid when channel power is on and after swept).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Power value (dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – channel power – query channel power density

ViStatus _VI_FUNC AV4041_QueryPSDR (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query channel power density for channel power function measurement under spectrum analysis mode **(valid when channel power is on and after swept).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Power density (dBm/Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – set occupied bandwidth state

ViStatus _VI_FUNC AV4041_SetOBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set occupied bandwidth function measurement state, or use function **AV4041_SetMeasFunc() (Other functional measurements will be disabled after this function is enabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query occupied bandwidth state

ViStatus _VI_FUNC AV4041_QueryOBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query occupied bandwidth state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – set measurement method

ViStatus _VI_FUNC AV4041_SetOBWMethod (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set occupied bandwidth measurement method. The percentage measurement method is to obtain the x% bandwidth of total power of all span. The XdB measurement method is to obtain the xdB bandwidth less than maximum power value at both sides.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query measurement method

ViStatus _VI_FUNC AV4041_QueryOBWMethod (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query occupied bandwidth measurement method.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – set percentage

ViStatus _VI_FUNC AV4041_SetOBWPpow (ViSession instrumentHandle,ViReal64 dVal)

Function purpose:

Set occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Percentage, scope: 10.00%~99.99%.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query percentage

ViStatus _VI_FUNC AV4041_QueryOBWPpow (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

fVal

Percentage.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – set XdB

ViStatus _VI_FUNC AV4041_SetOBWXdB (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set occupied bandwidth XdB, valid when the measurement method is XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB), scope: -100dB~-0.1dB

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query XdB

ViStatus _VI_FUNC AV4041_QueryOBWXdB (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – occupied bandwidth – query occupied bandwidth

ViStatus _VI_FUNC AV4041_QueryOBWOBW (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

query occupied bandwidth

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

occupied bandwidth (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – set demodulation state

ViStatus _VI_FUNC AV4041_SetDemodOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set AF/FM function measurement state, or use function **AV4041_SetMeasFunc()(by enabling this function measurement, other functional measurement will be disabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – query demodulation state

ViStatus _VI_FUNC AV4041_QueryDemodOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query AM/FM demodulation state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – set demodulation mode

ViStatus _VI_FUNC AV4041_SetDMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demodulation mode. The intermittent mode refers to the mode which demodulation will

be stopped as per the demodulation time after sweeping one screen of data before sweeping the next screen of data, and will continue this process over and over again; continuous mode refers to the mode that the instrument will continuously demodulate after sweeping one screen of data and will not sweep data again.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation mode.

0: intermittent mode

1: continuous mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – query demodulation mode

ViStatus _VI_FUNC AV4041_QueryDMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query demodulation mode. The intermittent mode refers to the mode which demodulation will be stopped as per the demodulation time after sweeping one screen of data before sweeping the next screen of data, and will continue this process over and over again; continuous mode refers to the mode that the instrument will continuously demodulate after sweeping one screen of data and will not sweep data again.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation mode.

0: intermittent mode

1: continuous mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – set demodulation type

ViStatus _VI_FUNC AV4041_SetDType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demodulation type, frequency modulation, amplitude demodulation, upper sideband and lower sideband available.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod types.

0: frequency modulation

1: amplitude demodulation

2: upper sideband

3. lower sideband

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – query demodulation type

ViStatus _VI_FUNC AV4041_QueryDType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod type.

0: frequency modulation

1: amplitude demodulation

2: upper sideband

3. lower sideband

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – set demodulation time

ViStatus _VI_FUNC AV4041_SetDTime (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set demodulation time. This **parameter** is valid when the demodulation mode is intermittent mode and the demodulation time refers to the time for staying in demodulation state after one sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Demodulation time (ms), scope: 1us~400s

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – query demodulation time

ViStatus _VI_FUNC AV4041_QueryDTime (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query demodulation time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Demodulation time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – set volume

ViStatus _VI_FUNC AV4041_SetVolume (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set loudspeaker volume of demodulation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation volume, scope: 0~100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – audio demodulation – query volume

ViStatus _VI_FUNC AV4041_QueryVolume (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query loudspeaker volume of demodulation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation volume.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set ACPR on/off

ViStatus _VI_FUNC AV4041_SetACPROn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set ACP on, or use function **AV4041_SetMeasFunc()(Other functional measurements will be disabled after this function is enabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query ACPR on/off

ViStatus _VI_FUNC AV4041_QueryACPROn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query ACP on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set main channel bandwidth

ViStatus _VI_FUNC AV4041_SetACPRMainChBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set main channel bandwidth for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query main channel bandwidth

ViStatus _VI_FUNC AV4041_QueryACPRMainChBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query main channel bandwidth for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set adjacent channel bandwidth

ViStatus _VI_FUNC AV4041_SetACPRAdjChBW (ViSession instrumentHandle,ViReal64 dVal)

Function purpose:

Set adjacent channel bandwidth for ACPR function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Measurement – ACPR – query adjacent channel bandwidth

ViStatus _VI_FUNC AV4041_QueryACPRAdjChBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query adjacent channel bandwidth for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set channel space

ViStatus _VI_FUNC AV4041_SetACPRSpace (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set channel space for ACPR function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 0Hz~45MHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query channel space

ViStatus _VI_FUNC AV4041_QueryACPRSpace (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query channel space for ACP function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set limit test on

ViStatus _VI_FUNC AV4041_SetACPRLmtTestOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set adjacent channel power limit test on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query limit test on

ViStatus _VI_FUNC AV4041_QueryACPRLmtTestOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query adjacent channel power limit test on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set lower adjacent channel limit

ViStatus _VI_FUNC AV4041_SetACPRLowLmt (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set lower adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query lower adjacent channel limit

ViStatus _VI_FUNC AV4041_QueryACPRLowLmt (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query lower adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – set upper adjacent channel limit

ViStatus _VI_FUNC AV4041_SetACPRUppLmt (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set upper adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query upper adjacent channel limit

ViStatus _VI_FUNC AV4041_QueryACPRUppLmt (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query upper adjacent channel limit of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Adjacent channel limit (dB), scope: -200dB~200dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query upper ACPR

ViStatus _VI_FUNC AV4041_QueryACPRrateUpper (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query upper ACPR of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

ACPR (dBc).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – ACPR – query lower ACPR

ViStatus _VI_FUNC AV4041_QueryACPRrateLower (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query lower ACPR of ACPR.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

ACPR (dBc).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – set emission mask on

ViStatus _VI_FUNC AV4041_SetEMOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set emission mask on, or use function **AV4041_SetMeasFunc()** to open this option **(other functional measurements will be disabled after this measurement is enabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – query emission mask on

ViStatus _VI_FUNC AV4041_QueryEmOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query emission mask on

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – set reference channel bandwidth

ViStatus _VI_FUNC AV4041_SetEMRefCHBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference channel bandwidth of emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1kHz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Measurement – emission mask – query reference channel bandwidth

ViStatus _VI_FUNC AV4041_QueryEMRefCHBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query reference channel bandwidth of emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – set reference power type

ViStatus _VI_FUNC AV4041_SetEMRefPowerType (ViSession instrumentHandle, ViInt32 type)

Function purpose:

set reference power type for emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

type

Reference power type, 0 is peak and 1 is channel.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – query reference power type

ViStatus _VI_FUNC AV4041_QueryEMRefPowerType (ViSession instrumentHandle, ViInt32 type[])

Function purpose:

Query reference power type for emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

type

Reference power type, 0 is peak and 1 is channel.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – set peak marker on in emission mask

ViStatus _VI_FUNC AV4041_SetEMPeakMarkerOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set peak marker on in emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – query peak marker on of emission mask

ViStatus _VI_FUNC AV4041_QueryEMPeakMarkerOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query peak marker on of emission mask.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – emission mask – query if emission mask fails

ViStatus _VI_FUNC AV4041_QueryEMFail (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query if emission mask fails.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: pass, 1: fail.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – set CNR state

ViStatus _VI_FUNC AV4041_SetCNROn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set CNR state, or use function **AV4041_SetMeasFunc()** to open this item **(by opening this function measurement, other function measurement will be turned off).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR state

ViStatus _VI_FUNC AV4041_QueryCNROn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query CNR state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – set CNR carrier bandwidth

ViStatus _VI_FUNC AV4041_SetCNRCBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set CNR carrier bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR carrier bandwidth

ViStatus _VI_FUNC AV4041_QueryCNRCBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR carrier bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – set CNR noise bandwidth

ViStatus _VI_FUNC AV4041_SetCNRNBW (ViSession instrumentHandle,ViReal64 dVal)

Function purpose:

Set CNR noise bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 300Hz~20MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR noise bandwidth

ViStatus _VI_FUNC AV4041_QueryCNRNBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR noise bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – set CNR frequency offset

ViStatus _VI_FUNC AV4041_SetCNRSpace (ViSession instrumentHandle,ViReal64 dVal)

Function purpose:

Set CNR frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency value (unit: Hz), scope: 0Hz~100MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR frequency offset

ViStatus _VI_FUNC AV4041_QueryCNRSpace (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement –CNR – query CNR measurement result

ViStatus _VI_FUNC AV4041_QueryCNRValue (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query CNR measurement result.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

measurement result (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set IQ capture state

ViStatus _VI_FUNC AV4041_SetIQcaptureOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set IQ capture state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query IQ capture state

ViStatus _VI_FUNC AV4041_QueryIQCaptureOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query IQ capture state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – start capture

ViStatus _VI_FUNC AV4041_SetIQCaptureStart (ViSession instrumentHandle)

Function purpose:

Set IQ capture to start capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – stop capture

ViStatus _VI_FUNC AV4041_SetIQCaptureStop (ViSession instrumentHandle)

Function purpose:

Set IQ capture to stop capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set capture time

ViStatus _VI_FUNC AV4041_SetIQCaptureTime (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set capture time for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Capture time (us)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query capture time

ViStatus _VI_FUNC AV4041_QueryIQCaptureTime (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query capture time of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Capture time (us)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set IQ capture mode

ViStatus _VI_FUNC AV4041_SetIQCaptureMode (ViSession instrumentHandle, ViInt32 mode)

Function purpose:

Set IQ capture mode, including single capture and continuous capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

mode

0: single capture, 1: continuous capture.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query IQ capture mode

ViStatus _VI_FUNC AV4041_QueryIQCaptureMode (ViSession instrumentHandle, ViInt32 mode[])

Function purpose:

Query IQ capture mode, including single capture and continuous capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

mode

0: single capture, 1: continuous capture.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set sampling rate

ViStatus _VI_FUNC AV4041_SetIQCaptureSample (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set sampling rate of IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Sample rate.

Capture sampling rate	Capture bandwidth
12.5MHz	10MHz
5MHz	4MHz
1.25MHz	1MHz
500kHz	400kHz
125kHz	100kHz
50kHz	40kHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query sampling rate

ViStatus _VI_FUNC AV4041_QueryIQCaptureSample (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query IQ capture sampling rate.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Sample rate.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set IQ capture storage name

ViStatus _VI_FUNC AV4041_SetIQCaptureFilename(ViSession instrumentHandle, char* name)

Function purpose:

Set IQ capture file storage name.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

name

File storage name

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set trigger mode

ViStatus _VI_FUNC AV4041_SetIQCaptureTrigMode (ViSession instrumentHandle,ViInt32 nval)

Function purpose:

Set trigger mode, including free trigger and external trigger.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query trigger mode

ViStatus _VI_FUNC AV4041_QueryIQCaptureTrigMode (ViSession instrumentHandle,ViInt32 nval[])

Function purpose:

Query trigger mode, including free trigger and external trigger.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: free trigger

1: external trigger

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set external trigger polarity

ViStatus _VI_FUNC AV4041_SetIQCaptureTrigSlop (ViSession instrumentHandle,ViInt32 nval)

Function purpose:

set external trigger polarity.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query external trigger polarity

ViStatus _VI_FUNC AV4041_QueryIQCaptureTrigSlop (ViSession instrumentHandle,ViInt32 nval[])

Function purpose:

query external trigger polarity

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trigger Type.

0: positive

1: negative

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set external trigger delay

ViStatus _VI_FUNC AV4041_SetIQCaptureTrigDelay (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set external trigger delay for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query external trigger delay

ViStatus _VI_FUNC AV4041_QueryIQCaptureTrigDelay (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query external trigger delay for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

trigger delay (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – set external trigger amplitude

ViStatus _VI_FUNC AV4041_SetIQCaptureTrigAMP (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set external trigger amplitude for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (volt).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – IQ capture – query external trigger amplitude

ViStatus _VI_FUNC AV4041_QueryIQCaptureTrigAMP (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query external trigger amplitude for IQ capture.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Trigger level (volt).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measure- Generator- Set the generator ON/OFF

ViStatus _VI_FUNC AV4041_SetGenMode (ViSession instrumentHandle, ViInt32 bOn)

Purpose of function:

Set the generator ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator ON/OFF

ViStatus _VI_FUNC AV4041_QueryGenMode (ViSession instrumentHandle,ViInt32 bOn[])

Purpose of function:

Query the generator ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the frequency offset

ViStatus _VI_FUNC AV4041_SetGenFreqOffset (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the frequency offset

ViStatus _VI_FUNC AV4041_QueryGenFreqOffset (ViSession instrumentHandle, ViReal64 dVal[])

Purpose of function:

Query the generator frequency offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator frequency

ViStatus _VI_FUNC AV4041_SetGenFreq (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator frequency

ViStatus _VI_FUNC AV4041_QueryGenFreq (ViSession instrumentHandle,ViReal64 dVal[])

Purpose of function:

Query the generator frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Frequency.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator power

ViStatus _VI_FUNC AV4041_SetGenPower (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator power.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator power

ViStatus _VI_FUNC AV4041_QueryGenPower (ViSession instrumentHandle,ViReal64 dVal[])

Purpose of function:

Query the generator power.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator power offset

ViStatus _VI_FUNC AV4041_SetGenPowerOffset (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator power offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator power offset

ViStatus _VI_FUNC AV4041_QueryGenPowerOffset (ViSession instrumentHandle,ViReal64 dVal[])

Purpose of function:

Query the generator power offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Power offset.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator normalization ON/OFF

ViStatus _VI_FUNC AV4041_SetGenNormzStat (ViSession instrumentHandle, ViInt32 bOn)

Purpose of function:

Set the generator normalization ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator normalization ON/OFF

ViStatus _VI_FUNC AV4041_QueryGenNormzStat (ViSession instrumentHandle,ViInt32 bOn[])

Purpose of function:

Query the generator normalization ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator normalization reference level

ViStatus _VI_FUNC AV4041_SetGenNormzRef (ViSession instrumentHandle, ViReal64 dVal)

Purpose of function:

Set the generator normalization reference level.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Reference level.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator normalization reference level

ViStatus _VI_FUNC AV4041_QueryGenNormzRef (ViSession instrumentHandle,ViReal64 dVal[])

Purpose of function:

Query the generator normalization reference level.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Reference level.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator normalization reference position

ViStatus _VI_FUNC AV4041_SetGenNormzRefPos (ViSession instrumentHandle, ViInt32 nVal)

Purpose of function:

Set the generator normalization reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Reference position.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator normalization reference position

ViStatus _VI_FUNC AV4041_QueryGenNormzRefPos (ViSession instrumentHandle,ViInt32 nVal[])

Purpose of function:

Query the generator normalization reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

nVal

Reference position.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator normalization scale/division

ViStatus _VI_FUNC AV4041_SetGenNormzPdiv (ViSession instrumentHandle, ViReal64

dVal)

Purpose of function:

Set the generator normalization scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Scale/division.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator normalization scale/division

ViStatus _VI_FUNC AV4041_QueryGenNormzPdiv (ViSession instrumentHandle,ViReal64 dVal[])

Purpose of function:

Query the generator normalization scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

dVal

Scale/division.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Set the generator reference trace ON/OFF

ViStatus _VI_FUNC AV4041_SetGenRefTraceStat (ViSession instrumentHandle, ViInt32 bOn)

Purpose of function:

Set the generator reference trace ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Measure- Generator- Query the generator reference trace ON/OFF

ViStatus _VI_FUNC AV4041_QueryGenRefTraceStat (ViSession instrumentHandle,ViInt32 bOn[])

Purpose of function:

Query the generator reference trace ON/OFF.

Parameter list:

instrumentHandle

Instrument handle returned by the function, communicate with instrument.

bOn

0: OFF; 1: ON.

Returned value:

Returned value indicates the implementation of function: 0- succeed; <0-fail.

Zero calibration – execute

ViStatus _VI_FUNC AV4041_SetAligNow (ViSession instrumentHandle)

Function purpose:
Zero calibration **(please do not repeatedly calibrate during calibration).** This command is an overlapping command. Use **AV4041_QueryOPC()** to query if this command is completed before sending other commands.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall antenna factor

ViStatus _VI_FUNC AV4041_LoadAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall antenna factor in field strength function measurement under spectrum analysis mode **(the command will be invalid if the file does not exist and be only valid for current storage location)** so that this antenna factor can be weighted when opening relevant measurement function.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store antenna factor

ViStatus _VI_FUNC AV4041_StoreAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Store antenna factor edited in field strength function measurement under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete antenna factor

ViStatus _VI_FUNC AV4041_DelAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

delete antenna factor file.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all antenna factors

ViStatus _VI_FUNC AV4041_DelAllAntFile (ViSession instrumentHandle)

Function purpose:

delete all antenna factor files.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

File – emission mask recall limit line

ViStatus _VI_FUNC AV4041_LoadLmtFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall limit line as the mask for emission mask measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store list to file

ViStatus _VI_FUNC AV4041_StoreList (ViSession instrumentHandle, char chStr[])

Function purpose:

Store the list as list file (the command will be invalid if the file does not exist and will only be valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall list file

ViStatus _VI_FUNC AV4041_LoadList (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall list file **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete list file

ViStatus _VI_FUNC AV4041_DelList (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete list file **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all list files

ViStatus _VI_FUNC AV4041_DelAllList (ViSession instrumentHandle)

Function purpose:

Delete all list files.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete limit file

ViStatus _VI_FUNC AV4041_DelLimit (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete limit file **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store limit file

ViStatus _VI_FUNC AV4041_StoreLimit (ViSession instrumentHandle,

char chStr[])

Function purpose:

Store limit line file as limit file **(the command will be invalid if the file does not exist and will only be valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall limit line

ViStatus _VI_FUNC AV4041_LoadLimit (ViSession instrumentHandle,

char chStr[])

Function purpose:

Recall limit file to limit line **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - delete all limit files

ViStatus _VI_FUNC AV4041_DelAllLimit (ViSession instrumentHandle)

Function purpose:

delete all limit files

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete data file

ViStatus _VI_FUNC AV4041_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all data files

ViStatus _VI_FUNC AV4041_DeleteAllDataFile (ViSession instrumentHandle)

Function purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall data file

ViStatus _VI_FUNC AV4041_LoadDateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store data file

ViStatus _VI_FUNC AV4041_StoreDataFile (ViSession instrumentHandle,

char chStr[])

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Interference Analysis Mode Functions

Frequency – set center frequency

ViStatus _VI_FUNC AV4041_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query center frequency

ViStatus _VI_FUNC AV4041_QueryCntFreq (ViSession instrumentHandle,
ViReal64 dbVal[])

Function purpose:

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set span

ViStatus _VI_FUNC AV4041_SetSpan (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query span

ViStatus _VI_FUNC AV4041_QuerySpan (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – full span

ViStatus _VI_FUNC AV4041_SetFullSpan (ViSession instrumentHandle)

Function purpose:

Set as full span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – zero span

ViStatus _VI_FUNC AV4041_SetZeroSpan (ViSession instrumentHandle)

Function purpose:

Set as zero span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – previous span

ViStatus _VI_FUNC AV4041_SetLastSpan (ViSession instrumentHandle)

Function purpose:

Set as previous span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set start frequency

ViStatus _VI_FUNC AV4041_SetSttFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query start frequency

ViStatus _VI_FUNC AV4041_QuerySttFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set stop frequency

ViStatus _VI_FUNC AV4041_SetStpFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope of interference analysis frequency 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query stop frequency

ViStatus _VI_FUNC AV4041_QueryStpFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency

ViStatus _VI_FUNC AV4041_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, scope: 1Hz~5GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency

ViStatus _VI_FUNC AV4041_QueryStepFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency auto on/off

ViStatus _VI_FUNC AV4041_SetAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set step frequency auto on/off. When auto is on, the step frequency is 1MHz. when off, the step frequency can be 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query step frequency auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard name

ViStatus _VI_FUNC AV4041_SetSIGStandard (ViSession instrumentHandle, char* standard)

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard name

ViStatus _VI_FUNC AV4041_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard channel number

ViStatus _VI_FUNC AV4041_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard channel number

ViStatus _VI_FUNC AV4041_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference level

ViStatus _VI_FUNC AV4041_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level. Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference level

ViStatus _VI_FUNC AV4041_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value). Reference level value is related to current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference position

ViStatus _VI_FUNC AV4041_SetRefPos (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting of reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Reference position, scope: -10~10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference position

ViStatus _VI_FUNC AV4041_QueryRefPos (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

reference position value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set attenuation

ViStatus _VI_FUNC AV4041_SetAtt (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query attenuation

ViStatus _VI_FUNC AV4041_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude set attenuation auto on/off

ViStatus _VI_FUNC AV4041_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude query attenuation auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set scale/division

ViStatus _VI_FUNC AV4041_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale (0.1dB~20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query scale/division

ViStatus _VI_FUNC AV4041_QueryScalePDiv (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set pre-amplifier on/off

ViStatus _VI_FUNC AV4041_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query pre-amplifier on/off

ViStatus _VI_FUNC AV4041_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set resolution bandwidth

ViStatus _VI_FUNC AV4041_SetRBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set resolution bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query resolution bandwidth

ViStatus _VI_FUNC AV4041_QueryRBW (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query resolution bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set video bandwidth

ViStatus _VI_FUNC AV4041_SetVBW (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set video bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz), scope: 1Hz~10MHz, step: 1-3-10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query video bandwidth

ViStatus _VI_FUNC AV4041_QueryVBW (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query video bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set resolution bandwidth auto on/off

ViStatus _VI_FUNC AV4041_SetAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set resolution bandwidth auto on/off. When on, the resolution bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query resolution bandwidth auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoRBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query resolution bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set video bandwidth auto on/off

ViStatus _VI_FUNC AV4041_SetAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set video bandwidth auto on/off. When on, the video bandwidth will automatically adapter the resolution bandwidth as per SPAN/RBW based on span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Bandwidth –query video bandwidth auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoVBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query video bandwidth auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set SPAN/RBW

ViStatus _VI_FUNC AV4041_SetSR100 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set SPAN/RBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW, scope: 1~500.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query SPAN/RBW

ViStatus _VI_FUNC AV4041_QuerySR100 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query SPAN/RBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set RBW/VBW

ViStatus _VI_FUNC AV4041_SetRV300 (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set RBW/VBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

RBW/VBW, scope: 1~100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – query RBW/VBW

ViStatus _VI_FUNC AV4041_QueryRV300 (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query RBW/VBW value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

SPAN/RBW value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average on/off

ViStatus _VI_FUNC AV4041_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average on/off

ViStatus _VI_FUNC AV4041_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average count

ViStatus _VI_FUNC AV4041_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1~1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average count

ViStatus _VI_FUNC AV4041_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – clear average count

ViStatus _VI_FUNC AV4041_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query current average count

ViStatus _VI_FUNC AV4041_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – set detector type

ViStatus _VI_FUNC AV4041_SetDetectorType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive Peak
NEGative(1)	Negative Peak
SAMPle(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – query detector type

ViStatus _VI_FUNC AV4041_QueryDetectorType (ViSession instrumentHandle, ViInt32 nVal [])

Function purpose:

query detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(0)	Positive Peak
NEGative(1)	Negative Peak
SAMPle(2)	Sample
NORMal(3)	Standard (Rosenfeld)
AVERage(4)	Average
RMS(5)	Root mean square

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – set detector auto on/off

ViStatus _VI_FUNC AV4041_SetAutoDetectorOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set detector auto on/off. When on, the instrument will automatically select detector type

based on different measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – query detector auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoDetectorOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query detector auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep type

ViStatus _VI_FUNC AV4041_SetSwpType (ViSession instrumentHandle, ViBoolean nVal)

Function purpose:

Set the scanning type. This is an overlapping command. Use **AV4041_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Sweep – query sweep type

ViStatus _VI_FUNC AV4041_QuerySwpType (ViSession instrumentHandle, ViBoolean nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – trigger single sweep

ViStatus _VI_FUNC AV4041_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use **AV4041_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep time

ViStatus _VI_FUNC AV4041_SetSwpTime (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set sweep time under current mode. The sweep time is the time required for selected frequency interval for local oscillator tuning. The sweep time directly affects the time for completing one test, excluding the dead time between one sweep and the next sweep. The sweep time generally changes with the span, resolution bandwidth and video bandwidth. The sweep time is not available when the resolution bandwidth is <=1kHz under spectrum analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep time

ViStatus _VI_FUNC AV4041_QuerySwpTime (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query sweep time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep time auto on/off

ViStatus _VI_FUNC AV4041_SetAutoSwpTimeOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set sweep time auto on/off. When auto is on, the instrument will use quick sweep speed as possible, or the manual mode is available to increase the sweep time to meet certain measurement needs. The sweep time for manual setting must be equal to or greater than auto sweep time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep time auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoSwpTimeOn (ViSession instrumentHandle,

ViBoolean bOn[])

Function purpose:

query sweep time auto on/off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Trace - set trace state

ViStatus _VI_FUNC AV4041_SetTraceType (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nTrace)

Function purpose:

Set trace state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

Set nTrace value as 1

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Trace – query trace state

ViStatus _VI_FUNC AV4041_QueryTraceType (ViSession instrumentHandle, ViInt32 nVal[], ViInt32 nTrace)

Function purpose:

Query trace state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Trace types.

0: refresh trace

1: maximum hold

2: minimum hold

Set nTrace value as 1

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set market state

ViStatus _VI_FUNC AV4041_SetMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState)

Function purpose:

set marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker state

ViStatus _VI_FUNC AV4041_QueryMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState[])

Function purpose:

Query marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker- activate marker

ViStatus _VI_FUNC AV4041_SetMkrActive (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Activate marker under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker- marker function (marker ->)

ViStatus _VI_FUNC AV4041_SetMkrTo (ViSession instrumentHandle, ViInt32 nVal , ViInt32 nSetIdx)

Function purpose:

Set marker function under current mode (marker -> for spectrum analysis mode).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nSetIdx

Instrument mode	nSetIdx	Function
Interference Analysis (non-zero span)	0	Marker -> start frequency (set marker frequency as start frequency)
	1	Marker -> stop frequency (set marker frequency as

		stop frequency)
	2	Marker -> center frequency (set marker frequency as center frequency)
	3	Marker -> step frequency (set marker frequency as step frequency)
Interference Analysis (zero span)	0	Marker -> start frequency (set marker index as minimum index)
	1	Marker -> stop frequency (set marker index as maximum index)
	2	Marker -> center frequency (set marker index as center index)
	3	Marker -> step frequency (set marker frequency as step frequency)

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – disable all markers

ViStatus _VI_FUNC AV4041_DisableAllMarkers (ViSession instrumentHandle)

Function purpose:

Disable all markers under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker –set marker X value

ViStatus _VI_FUNC AV4041_MoveMarker (ViSession instrumentHandle, ViInt32 nVal, ViReal64 dbVal,)

Function purpose:

set marker X value under current mode, when marker is an offset marker, the X value can be negative.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Marker X value, time in ms and frequency in Hz.

Instrument mode	**Parameter** unit
Interference Analysis (non-zero span)	Hz
Interference Analysis (zero span)	ms

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker X value

ViStatus _VI_FUNC AV4041_QueryMarker (ViSession instrumentHandle,
ViInt32 markerIndex,
ViReal64 markerPosition[],
ViReal64 markerAmplitude[])

Function purpose:

Query marker X value and Y value under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

markerPosition

Marker X value, time in ms and frequency in Hz.

Instrument mode	**Parameter** unit
Interference Analysis (non-zero span)	Hz
Interference Analysis (zero span)	ms

markerIndex

Marker index, 1~6 available.

markerAmplitude

Marker Y value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker - search

ViStatus _VI_FUNC AV4041_SetMkrSearch (ViSession instrumentHandle, ViInt32 nVal,ViInt32 type)

Function purpose:

Move marker to maximum, minimum, peak, secondary peak, left adjacent peak and right adjacent peak.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index can be set from 1 to 6 1~6 available, indicating marker 1, 2, 3 and 4.

type

Search type.

1 Maximum value

2 Minimum value

3 Peak

4 Secondary peak

5 left adjacent peak

6 right adjacent peak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set noise marker on

ViStatus _VI_FUNC AV4041_SetMkrNoiseOn (ViSession instrumentHandle,ViInt32 nVal,
ViBoolean bOn)

Function purpose:

Set noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Marker – query noise marker on

ViStatus _VI_FUNC AV4041_QueryMkrNoiseOn (ViSession instrumentHandle, ViInt32 nVal,
ViBoolean bOn[])

Function purpose:

Query noise marker on under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

State, 0 is off and 1 is on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set measurement mode.

ViStatus _VI_FUNC AV4041_SetIAMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the measurement mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement mode.

0: spectrum measurement.

1: waterfall graph.

2: RSSI

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query measurement mode.

ViStatus _VI_FUNC AV4041_QueryIAMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query the measurement mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement mode.

0: spectrum measurement.

1: waterfall graph.

2: RSSI

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – set span time

ViStatus _VI_FUNC AV4041_SetTraceTimeSpan (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set span time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Span time (0~1440 min)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – query span time

ViStatus _VI_FUNC AV4041_QueryTraceTimeSpan (ViSession instrumentHandle, ViInt32 nVal [])

Function purpose:

Query span time.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Span time.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – set auto save state

ViStatus _VI_FUNC AV4041_SetIASaveOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set auto save state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – query auto save state

ViStatus _VI_FUNC AV4041_QueryIASaveOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query auto save state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – set time cursor

ViStatus _VI_FUNC AV4041_SetIACursor (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set time cursor.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Time cursor (0~288)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – set sweep interval

ViStatus _VI_FUNC AV4041_SetIAInterval (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set sweep interval.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep interval (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – query sweep interval

ViStatus _VI_FUNC AV4041_QueryIAInterval (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query sweep interval

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep interval (ms).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Auto save – restart measurement

ViStatus _VI_FUNC AV4041_SetIARestart (ViSession instrumentHandle)

Function purpose:

restart measurement.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete data file

ViStatus _VI_FUNC AV4041_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all data files

ViStatus _VI_FUNC AV4041_DeleteAllDataFile (ViSession instrumentHandle)

Function purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall data file

ViStatus _VI_FUNC AV4041_LoadDateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store data file

ViStatus _VI_FUNC AV4041_StoreDataFile (ViSession instrumentHandle,

char chStr[])

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query trace data

ViStatus _VI_FUNC AV4041_QueryTraceData (ViSession instrumentHandle,ViInt32 size[], ViReal64 data[],

ViInt32 index)

Function purpose:

Query trace data under Interference Analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

size

Number of trace data received.

data

Trace data stores array pointer and the arrya should match with the size of trace data received.

index

Trace number, and the definit value is 1 under Interference Analysis mode.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

AM-FM-PM Analyzer Mode Functions

Frequency – set center frequency

ViStatus _VI_FUNC AV4041_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Frequency range of spectrum analysis, 0Hz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query center frequency

ViStatus _VI_FUNC AV4041_QueryCntFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency

ViStatus _VI_FUNC AV4041_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, scope: 1Hz~5GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency

ViStatus _VI_FUNC AV4041_QueryStepFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query step value of center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency auto on/off

ViStatus _VI_FUNC AV4041_SetAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set step frequency auto on/off. When auto is on, the step frequency is 1MHz. when off, the step frequency can be 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query step frequency auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set span

ViStatus _VI_FUNC AV4041_SetSpan (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Scope: 1kHz~10MHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query span

ViStatus _VI_FUNC AV4041_QuerySpan (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query span under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – previous span

ViStatus _VI_FUNC AV4041_SetLastSpan (ViSession instrumentHandle)

Function purpose:

Set as previous span.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set start frequency

ViStatus _VI_FUNC AV4041_SetSttFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query start frequency

ViStatus _VI_FUNC AV4041_QuerySttFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query start frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set stop frequency

ViStatus _VI_FUNC AV4041_SetStpFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query stop frequency

ViStatus _VI_FUNC AV4041_QueryStpFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query stop frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard name

ViStatus _VI_FUNC AV4041_SetSIGStandard (ViSession instrumentHandle, char* standard)

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard name

ViStatus _VI_FUNC AV4041_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set signal standard channel number

ViStatus _VI_FUNC AV4041_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query signal standard channel number

ViStatus _VI_FUNC AV4041_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference level

ViStatus _VI_FUNC AV4041_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level. Reference level is related to current amplitude unit, and the setting scope corresponds to dBm. Conversion is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference level

ViStatus _VI_FUNC AV4041_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value). Reference level value is related to current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set reference position

ViStatus _VI_FUNC AV4041_SetRefPos (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Setting of reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Reference position, scope: -10~10.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference position

ViStatus _VI_FUNC AV4041_QueryRefPos (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query reference position.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

reference position value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set attenuation

ViStatus _VI_FUNC AV4041_SetAtt (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query attenuation

ViStatus _VI_FUNC AV4041_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude set attenuation auto on/off

ViStatus _VI_FUNC AV4041_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude query attenuation auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set scale/division

ViStatus _VI_FUNC AV4041_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale (0.1dB~20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query scale/division

ViStatus _VI_FUNC AV4041_QueryScalePDiv (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set pre-amplifier on/off

ViStatus _VI_FUNC AV4041_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query pre-amplifier on/off

ViStatus _VI_FUNC AV4041_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average on/off

ViStatus _VI_FUNC AV4041_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average on/off

ViStatus _VI_FUNC AV4041_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average count

ViStatus _VI_FUNC AV4041_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1~1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average count

ViStatus _VI_FUNC AV4041_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – clear average count

ViStatus _VI_FUNC AV4041_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Average – query current average count

ViStatus _VI_FUNC AV4041_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep type

ViStatus _VI_FUNC AV4041_SetSwpType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the scanning type. This is an overlapping command. Use **AV4041_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep type

ViStatus _VI_FUNC AV4041_QuerySwpType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – trigger single sweep

ViStatus _VI_FUNC AV4041_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use **AV4041_QueryOPC()** to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set market state

ViStatus _VI_FUNC AV4041_SetMkrState (ViSession instrumentHandle, ViInt32 nVal, ViInt32 nState)

Function purpose:

set marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Marker – query marker state

ViStatus _VI_FUNC AV4041_QueryMkrState (ViSession instrumentHandle, ViInt32 nVal,
ViInt32 nState[])

Function purpose:

Query marker state under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nState

Marker state.

0: marker off.

1: normal marker on.

2: offset marker on.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker- activate marker

ViStatus _VI_FUNC AV4041_SetMkrActive (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Activate marker under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – disable all markers

ViStatus _VI_FUNC AV4041_SetMkrAOff (ViSession instrumentHandle)

Function purpose:

Disable all markers under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker –set marker X value

ViStatus _VI_FUNC AV4041_MoveMarker (ViSession instrumentHandle,
ViInt32 nVal, ViReal64 dbVal,)

Function purpose:

set marker X value under current mode, when marker is an offset marker, the X value can be negative.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Marker X value, time in ms and frequency in Hz.

nVal

Marker index, 1~6 available.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker X value

ViStatus _VI_FUNC AV4041_QueryMarker (ViSession instrumentHandle,
ViInt32 markerIndex,
ViReal64 markerPosition[],
ViReal64 markerAmplitude[])

Function purpose:

Query marker X value and Y value under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

markerPosition

Marker X value, time in ms and frequency in Hz.

markerIndex

Marker index, 1~6 available.

markerAmplitude

Marker Y value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker - search

ViStatus _VI_FUNC AV4041_SetMkrSearch (ViSession instrumentHandle, ViInt32 nVal,ViInt32 type)

Function purpose:

Move marker to maximum, minimum, peak, secondary peak, left adjacent peak and right adjacent peak.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Marker index can be set from 1 to 6 1~6 available, indicating marker 1, 2, 3 and 4.

type

Search type.

1 Maximum value

2 Minimum value

3 Peak

4 Secondary peak

5 left adjacent peak

6 right adjacent peak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – set occupied bandwidth state

ViStatus _VI_FUNC AV4041_SetOBWOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set occupied bandwidth function measurement state, or use function **AV4041_SetMeasFunc() (Other functional measurements will be disabled after this function is enabled).**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – query occupied bandwidth state

ViStatus _VI_FUNC AV4041_QueryOBWOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query occupied bandwidth state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – set measurement method

ViStatus _VI_FUNC AV4041_SetOBWMethod (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set occupied bandwidth measurement method. The percentage measurement method is to obtain the x% bandwidth of total power of all span. The XdB measurement method is to obtain the xdB bandwidth less than maximum power value at both sides.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – query measurement method

ViStatus _VI_FUNC AV4041_QueryOBWMethod (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query occupied bandwidth measurement method.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement Method.

0: percentage measurement method

1: XdB measurement method

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – set percentage

ViStatus _VI_FUNC AV4041_SetOBWPpow (ViSession instrumentHandle,ViReal64 dVal)

Function purpose:

Set occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Percentage, scope: 10.00%~99.99%.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – query percentage

ViStatus _VI_FUNC AV4041_QueryOBWPpow (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth percentage.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

fVal

Percentage.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – set XdB

ViStatus _VI_FUNC AV4041_SetOBWXdB (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set occupied bandwidth XdB, valid when the measurement method is XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB), scope: -100dB~-0.1dB

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

RF spectrum – occupied bandwidth – query XdB

ViStatus _VI_FUNC AV4041_QueryOBWXdB (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query occupied bandwidth XdB.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

XdB value (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set demodulation type

ViStatus _VI_FUNC AV4041_SetDMType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demodulation type, including AM, F and PM.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod type.

0: AM demodulation

1: FM demodulation

2: PM demodulation

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query demodulation type

ViStatus _VI_FUNC AV4041_QueryDMType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demod type.

0: AM demodulation

1: FM demodulation

2: PM demodulation

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set display mode

ViStatus _VI_FUNC AV4041_SetDMMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set demonstration mode, including RF, AF, AW and ALL

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Display mode.

0: RF spectrum.

1: audio spectrum.

2: audio waveform.

3: display all.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means

failed.

Measurement – query display mode

ViStatus _VI_FUNC AV4041_QueryDMMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Display mode.

0: RF spectrum.

1: audio spectrum.

2: audio waveform.

3: display all.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio spectrum – set span

ViStatus _VI_FUNC AV4041_SetDMAFSpan (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set span for audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Span (500Hz~150KHz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio spectrum – query span

ViStatus _VI_FUNC AV4041_QueryDMAFSpan (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query span of audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Span.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio spectrum – set scale

ViStatus _VI_FUNC AV4041_SetDMScale (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set scale of audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio spectrum – query scale

ViStatus _VI_FUNC AV4041_QueryDMScale (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query scale of audio spectrum.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio waveform – set sweep time

ViStatus _VI_FUNC AV4041_SetDMTime (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the sweep time for audio waveform.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio waveform – query sweep time

ViStatus _VI_FUNC AV4041_QueryDMTime (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query the sweep time for audio waveform.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

sweep time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Bandwidth – set IF bandwidth

ViStatus _VI_FUNC AV4041_SetDMIBW (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set IF bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Bandwidth (Hz), scope: 10KHz~300KHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Audio waveform – query IF bandwidth

ViStatus _VI_FUNC AV4041_QueryDMIBW (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

query IF bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Bandwidth (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – select trace

ViStatus _VI_FUNC AV4041_SetDMTrace (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the trace selected for the marker.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Display mode.

0: RF spectrum.

1: audio spectrum.

2: audio waveform.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete data file

ViStatus _VI_FUNC AV4041_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all data files

ViStatus _VI_FUNC AV4041_DeleteAllDataFile (ViSession instrumentHandle)

Function purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall data file

ViStatus _VI_FUNC AV4041_LoadDateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store data file

ViStatus _VI_FUNC AV4041_StoreDataFile (ViSession instrumentHandle,

char chStr[])

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query trace data

ViStatus _VI_FUNC AV4041_QueryTraceData (ViSession instrumentHandle,ViInt32 size[], ViReal64 data[],

ViInt32 index)

Function purpose:

Query trace data under demodulation analysis mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

size

Number of trace data received.

data

Trace data stores array pointer and the arrya should match with the size of trace data received. The returned data of 3003 points are RF spectrum trace data, audio waveform trace data and audio spectrum trace data and each trace contains data of 1001 points.

index

Trace number, and the definit value is 1 under demodulation analysis mode.

Return value:

The return value represents the function execution: 0 means succeeded, and less than 0 means failed.

Power Meter Mode Functions

Frequency – set center frequency

ViStatus _VI_FUNC AV4041_SetCntFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency.

The frequency range of USB Power measurement is decided by the USB power probe model.

Power probe model	Frequency range
87231	10MHz~18GHz
87232	50MHz~26.5GHz
87233	50MHz~40GHz

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query center frequency

ViStatus _VI_FUNC AV4041_QueryCntFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query center frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set frequency resolution

ViStatus _VI_FUNC AV4041_SetPMResolution (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set frequency resolution.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Frequency resolution, scope: 0~3.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query frequency resolution

ViStatus _VI_FUNC AV4041_QueryPMResolution (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query frequency resolution.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

frequency resolution

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – auto scale

ViStatus _VI_FUNC AV4041_SetPMAutoscale (ViSession instrumentHandle)

Function purpose:

Set current display scale as a scale suitable for observation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – set minimum value

ViStatus _VI_FUNC AV4041_SetPMLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set minimum value of scale.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Minimum scale (dBm), scope: -70dBm~25dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – query minimum value

ViStatus _VI_FUNC AV4041_QueryPMLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query minimum scale.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Minimum scale (dBm), scope: -70dBm~25dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – set maximum value

ViStatus _VI_FUNC AV4041_SetPMUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set maximum value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Maximum scale (dBm), scope: -65dBm~30dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – query maximum value

ViStatus _VI_FUNC AV4041_QueryPMUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

query maximum scale value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Maximum scale (dBm), scope: -65dBm~30dBm.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – query relative measurement value

ViStatus _VI_FUNC AV4041_QueryPMRelative (ViSession instrumentHandle, ViReal64 dVal [])

Function purpose:

Query relative measurement value

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

relative measurement value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – set relative measurement state

ViStatus _VI_FUNC AV4041_SetPMRelativeOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set relative measurement state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – query relative measurement state

ViStatus _VI_FUNC AV4041_QueryPMRelativeOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query relative measurement state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – set offset value

ViStatus _VI_FUNC AV4041_SetPMCORRection (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set offset value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Offset value (dB), scope -50dB~30dB.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – query offset value

ViStatus _VI_FUNC AV4041_QueryPMCORRection (ViSession instrumentHandle, ViReal64 dVal [])

Function purpose:

Query offset value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Offset value (dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – set offset state

ViStatus _VI_FUNC AV4041_SetPMCORRectionOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set offset state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – query offset state

ViStatus _VI_FUNC AV4041_QueryPMCORRectionOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query offset state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – set maximum hold state

ViStatus _VI_FUNC AV4041_SetPMMaxholdOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set maximum hold state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Scale/division – query maximum hold state

ViStatus _VI_FUNC AV4041_QueryPMMaxholdOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query maximum hold state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average on/off

ViStatus _VI_FUNC AV4041_SetAvgOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set up average switch.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average on/off

ViStatus _VI_FUNC AV4041_QueryAvgOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query average on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – set average count

ViStatus _VI_FUNC AV4041_SetAvgCount (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set up average frequency.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count, scope: 1~1000

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query average count

ViStatus _VI_FUNC AV4041_QueryAvgCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Average.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – clear average count

ViStatus _VI_FUNC AV4041_ClearAvgCount (ViSession instrumentHandle)

Function purpose:

Clear average count to start from 0.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Average – query current average count

ViStatus _VI_FUNC AV4041_QueryCurrentCount (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query current average count.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

average count.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Calibration - execute

ViStatus _VI_FUNC AV4041_SetPMCalibZero (ViSession instrumentHandle)

Function purpose:

USB Power measurement zero calibration **(do not repeat zero calibration during calibration).** This command function is an overlapping command function. Use **AV4041_QueryOPC()** before sending other commands function to query if this command is completed.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Calibration – query zero calibration state

ViStatus _VI_FUNC AV4041_QueryPMCalibZero (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query USB Power measurement calibration fails (cannot be queried during zero calibration).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Calibration state.

0: no calibration

1: calibration succeed

2: calibration failed

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set limit state

ViStatus _VI_FUNC AV4041_SetPMLimitOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query limit state

ViStatus _VI_FUNC AV4041_QueryPMLimitOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query limit state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set upper limit

ViStatus _VI_FUNC AV4041_SetPMLimitUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set upper limit

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

upper limit (-65~30dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query upper limit

ViStatus _VI_FUNC AV4041_QueryPMLimitUpper (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query upper limit

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Upper limit value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set lower limit

ViStatus _VI_FUNC AV4041_SetPMLimitLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set lower limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Lower limit (-70~25dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query lower limit

ViStatus _VI_FUNC AV4041_QueryPMLimitLower (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query lower limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Lower limit value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set alarm on/off

ViStatus _VI_FUNC AV4041_SetAlarmOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit audio alarm state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query alarm on/off

ViStatus _VI_FUNC AV4041_QueryAlarmOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query limit audio alarm state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Channel Sweep Mode Functions

Amplitude – set reference level

ViStatus _VI_FUNC AV4041_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set reference level.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

reference level (-120dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query reference level

ViStatus _VI_FUNC AV4041_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query reference level (reference value).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level value (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set scale/division

ViStatus _VI_FUNC AV4041_SetScalePDiv (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale (0.1dB~20dB)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query scale/division

ViStatus _VI_FUNC AV4041_QueryScalePDiv (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set graph/table display

ViStatus _VI_FUNC AV4041_SetCSDisplay (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set graph/table display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 is graph display and 1 is table display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query graph/table display

ViStatus _VI_FUNC AV4041_QueryCSDisplay (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query graph/table display.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 is graph display and 1 is table display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set maximum hold state

ViStatus _VI_FUNC AV4041_SetCSMaxhold (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set maximum hold state

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query maximum hold state

ViStatus _VI_FUNC AV4041_QueryCSMaxhold (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query maximum hold state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0: off, 1: on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set unit

ViStatus _VI_FUNC AV4041_SetCSUnit (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set unit

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel, and 1 frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query unit

ViStatus _VI_FUNC AV4041_QueryCSUnit (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel, and 1 frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set power display

ViStatus _VI_FUNC AV4041_SetCSPower (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set power display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates now, and 1 maximum.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query power display

ViStatus _VI_FUNC AV4041_QueryCSPower (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query power display mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates now, and 1 maximum.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set color

ViStatus _VI_FUNC AV4041_SetCSColour (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set color.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates single color, 1 dual color.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query color

ViStatus _VI_FUNC AV4041_QueryCSColour (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query color.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates single color, 1 dual color.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set graph/table direction

ViStatus _VI_FUNC AV4041_SetCSOriental (ViSession instrumentHandle, ViInt32

nVal)

Function purpose:

Set graph/table direction

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates vertical display, 1 horizontal display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query graph/table direction

ViStatus _VI_FUNC AV4041_QueryCSOriental (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query graph/table direction

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates vertical display, 1 horizontal display.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep mode

ViStatus _VI_FUNC AV4041_SetCSMode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set sweep mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel sweep, 1 frequency sweep and 2 list sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query sweep mode

ViStatus _VI_FUNC AV4041_QueryCSMode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query sweep mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

0 indicates channel sweep, 1 frequency sweep and 2 list sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set signal standard name

ViStatus _VI_FUNC AV4041_SetSIGStandard (ViSession instrumentHandle, char* standard)

Function purpose:

set signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – query signal standard name

ViStatus _VI_FUNC AV4041_QuerySIGstandard (ViSession instrumentHandle, char standard[])

Function purpose:

Query signal standard name under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Standard

Name of signal standard.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set signal standard channel number

ViStatus _VI_FUNC AV4041_SetChannelNum (ViSession instrumentHandle, ViInt32 channelNum)

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – query signal standard channel number

ViStatus _VI_FUNC AV4041_QueryChannelNum (ViSession instrumentHandle, ViInt32 channelNum[])

Function purpose:

Set channel number under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

channelNum

Channel number.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set channel number

ViStatus _VI_FUNC AV4041_SetCSNum (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set Channel number

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1~20).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – query channel number

ViStatus _VI_FUNC AV4041_QueryCSNum (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query channel number

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1~20)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set channel step

ViStatus _VI_FUNC AV4041_SetCSStep (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set channel step.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

channel step (1~25).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – query channel step

ViStatus _VI_FUNC AV4041_QueryCSStep (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query channel step.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

channel step (1~25)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – set channel bandwidth

ViStatus _VI_FUNC AV4041_SetCSBandwidth (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set channel bandwidth

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz~20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – channel sweep – query channel bandwidth

ViStatus _VI_FUNC AV4041_QueryCSBandwidth (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query channel bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz~20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – set start frequency

ViStatus _VI_FUNC AV4041_SetCSStartFreq (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set start frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Start frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – query start frequency

ViStatus _VI_FUNC AV4041_QueryCSStartFreq (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query start frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Start frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – set step frequency

ViStatus _VI_FUNC AV4041_SetCSStepFreq (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set step frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Step frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – query step frequency

ViStatus _VI_FUNC AV4041_QueryCSStepFreq (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query step frequency for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Step frequency.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – set channel bandwidth

ViStatus _VI_FUNC AV4041_SetCSFreqScanBandwidth (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

set channel bandwidth

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz~20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – query channel bandwidth

ViStatus _VI_FUNC AV4041_QueryCSFreqScanBandwidth (ViSession instrumentHandle,
ViReal64 dVal[])

Function purpose:

query channel bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

channel bandwidth (1kHz~20MHz)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – set channel number

ViStatus _VI_FUNC AV4041_SetCSFreqScanNum (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set channel number for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1~20).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – frequency sweep – query channel number

ViStatus _VI_FUNC AV4041_QueryCSFreqScanNum (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query channel number for frequency sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1~20)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – list sweep – set channel number

ViStatus _VI_FUNC AV4041_SetCSListScanNum **(ViSession instrumentHandle, ViInt32 nVal)**

Function purpose:

Set channel number for list sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1~20).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – list sweep – query channel number

ViStatus _VI_FUNC AV4041_QueryCSListScanNum (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query channel number for list sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Channel number (1~20)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete data file

ViStatus _VI_FUNC AV4041_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – delete all data files

ViStatus _VI_FUNC AV4041_DeleteAllDataFile (ViSession instrumentHandle)

Function purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall data file

ViStatus _VI_FUNC AV4041_LoadDateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode **(the command will be invalid if the file does not exist and be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – store data file

ViStatus _VI_FUNC AV4041_StoreDataFile (ViSession instrumentHandle,

char chStr[])

Function purpose:

Store current measurement data as data file **(the file will be overwritten if the file exists and this command be only valid for current storage location)**.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Field Strength Mode Functions

Frequency – set point frequency

ViStatus _VI_FUNC AV4041_SetPointFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1Hz~5GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query point frequency

ViStatus _VI_FUNC AV4041_QueryPointFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query the point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set step frequency

ViStatus _VI_FUNC AV4041_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the step value of point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1Hz~5GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – query step frequency

ViStatus _VI_FUNC AV4041_QueryStepFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query the step value of point frequency under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – set auto step frequency on

ViStatus _VI_FUNC AV4041_SetAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set auto step frequency on. When auto is on, the step frequency of the instrument will be 1MHz, and when off, the frequency can be set as 1Hz~5GHz.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Query auto step frequency on

ViStatus _VI_FUNC AV4041_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query auto step frequency on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Set frequency track on

ViStatus _VI_FUNC AV4041_SetFreqTrac (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set frequency track on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Query frequency track on

ViStatus _VI_FUNC AV4041_QueryAutoStepFreqOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query frequency track on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Set the start frequency for frequency scanning

ViStatus _VI_FUNC AV4041_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the start frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1MHz~44.1GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Query the start frequency for frequency scanning

ViStatus _VI_FUNC AV4041_QueryStepFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query the start frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Set the step frequency for frequency scanning

ViStatus _VI_FUNC AV4041_SetStepFreq (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set the step frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency, range: 1Hz~5GHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Query the step frequency for frequency scanning

ViStatus _VI_FUNC AV4041_QueryStepFreq (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query the step frequency for frequency scanning under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Set the scan points

ViStatus _VI_FUNC AV4041_SetFscanPoints (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the scan points.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scan points (2~58).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Frequency – Query the scan points

ViStatus _VI_FUNC AV4041_QueryFscanPoints(ViSession instrumentHandle,ViInt32 nVal[])

Function purpose:

Query the scan points.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Scan points (2~58)

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – Set the reference level

ViStatus _VI_FUNC AV4041_SetRef (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set the reference level. The reference level depends on current amplitude unit. The setting range corresponds to dBm and conversion of units is required.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level (-150dBm~40dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – Query the reference level

ViStatus _VI_FUNC AV4041_QueryRef (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

Query the reference level. The reference level depends on current amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Reference level (reference value).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set attenuation

ViStatus _VI_FUNC AV4041_SetAtt (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set attenuation, only seven scales are available, which are 0, 10, 20, 30, 40, 50, 60. Other values set will be set as the attenuation for adjacent channel.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query attenuation

ViStatus _VI_FUNC AV4041_QueryAtt (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query attenuation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Attenuation, seven scales 0, 10, 20, 30, 40, 50, 60.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude - set attenuation auto on/off

ViStatus _VI_FUNC AV4041_SetAutoAttOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set attenuation auto on/off. When on, the instrument will set relevant attenuation value automatically based on the reference value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude - query attenuation auto on/off

ViStatus _VI_FUNC AV4041_QueryAutoAttOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query attenuation auto on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set scale/division

ViStatus _VI_FUNC AV4041_SetScalePDiv (ViSession instrumentHandle, ViReal64

dVal)

Function purpose:

Set scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Set scale/division (1dB~40dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query scale/division

ViStatus _VI_FUNC AV4041_QueryScalePDiv (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query scale/division.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Scale/division.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set unit

ViStatus _VI_FUNC AV4041_SetAmpUnit (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	unit:dBm
DBMV(1)	unit:dBmV
DBUV(2)	unit:dBuV
V(3)	unit:Volts
W(4)	unit:Walts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query unit

ViStatus _VI_FUNC AV4041_QueryAmpUnit (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query amplitude unit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Amplitude unit

DBM(0)	unit:dBm
DBMV(1)	unit:dBmV
DBUV(2)	unit:dBuV
V(3)	unit:Volts
W(4)	unit:Walts

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – set pre-amplifier on/off

ViStatus _VI_FUNC AV4041_SetPreAmpOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set pre-amplifier on/off. When on, the measurement accuracy of small signal can be improved. However, when measuring large power signal, the pre-amplifier is better be off, or measurement AD overflow may occur.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Amplitude – query pre-amplifier on/off

ViStatus _VI_FUNC AV4041_QueryPreAmpOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query pre-amplifier on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – set detector type

ViStatus _VI_FUNC AV4041_SetDetectorType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type.

POSitive(0)	Peak
SAMPle(2)	Real time
AVERage(4)	Average

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Detector – query detector type

ViStatus _VI_FUNC AV4041_QueryDetectorType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query detector type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Detector type

POSitive(1)	Peak
SAMPle(2)	Real time

AVERage(0)	Average

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set sweep type

ViStatus _VI_FUNC AV4041_SetSwpType (ViSession instrumentHandle, ViBoolean nVal)

Function purpose:

Set the scanning type. This is an overlapping command. Use AV4041_QueryOPC() to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

1: continuous sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep - query sweep type

ViStatus _VI_FUNC AV4041_QuerySwpType (ViSession instrumentHandle, ViBoolean nVal[])

Function purpose:

Query sweep type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Sweep type.

0: single sweep.

0: single sweep.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep - trigger single sweep

ViStatus _VI_FUNC AV4041_TrigSingleSwp (ViSession instrumentHandle)

Function purpose:

Trigger one single sweep (only valid for single sweep). This command function is an overlapping command function. Use AV4041_QueryOPC() to query if this command is completed before sending other command.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set dwell time

ViStatus _VI_FUNC AV4041_SetTDwell (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set dwell time **under current mode** (1ms~40s).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep - query dwell time

ViStatus _VI_FUNC AV4041_QueryTDwell (ViSession instrumentHandle, ViReal64 dbVal[])

Function purpose:

Query dwell time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

Time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set dwell time auto on.

ViStatus _VI_FUNC AV4041_SetAutoSwpTimeOn (ViSession instrumentHandle,

ViBoolean bOn)

Function purpose:

set dwell time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep - query dwell time auto on.

ViStatus _VI_FUNC AV4041_QueryTDwellOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query dwell time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep - set stay time

ViStatus _VI_FUNC AV4041_SetTStay (ViSession instrumentHandle, ViReal64 dbVal)

Function purpose:

Set stay time under current mode (1ms~40s).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – query stay time

ViStatus _VI_FUNC AV4041_QueryTStay (ViSession instrumentHandle,
ViReal64 dbVal[])

Function purpose:

Query stay time under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dbVal

time (us).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep – set stay time auto on

ViStatus _VI_FUNC AV4041_SetTStayOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set stay time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Sweep - query stay time auto on

ViStatus _VI_FUNC AV4041_QueryTStayOn (ViSession instrumentHandle,
ViBoolean bOn[])

Function purpose:

query stay time auto on.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – list add default segment

ViStatus _VI_FUNC AV4041_ListAddSeg (ViSession instrumentHandle)

Function purpose:

Add default sweep segment to the list edit under current mode.

Point frequency	500MHz
Bandwidth	30kHz
Detector	Average
Demodulation	Continuous wave
Limit	-174dBm
Limit on/off	Off

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – list delete segment

ViStatus _VI_FUNC AV4041_ListDelSeg (ViSession instrumentHandle, ViInt32 nval)

Function purpose:

Delete segment from list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nval

Segment index

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – clear list

ViStatus _VI_FUNC AV4041_ListClear (ViSession instrumentHandle)

Function purpose:

Delete all list edit segments under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – add segment

ViStatus _VI_FUNC AV4041_ListAdd (ViSession instrumentHandle,
ViReal64 startfrequency,
ViReal64 stopfrequency, ViInt32 rbw,
ViInt32 vbw, ViInt32 sweepPoints,
ViInt32 on)

Function purpose:

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

startfrequency

point frequency(1MHz~44.1GHz)

stopfrequency

Bandwidth (150Hz~150kHz)

rbw

Demodulation

vbw

Limit (-174dBm~50dBm)

sweepPoints

Detector

on

Limit on/off, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

List edit – edit segment

ViStatus _VI_FUNC AV4041_ListEdit (ViSession instrumentHandle, ViInt32 index,
ViReal64 startfrequency,
ViReal64 stopfrequency, ViInt32 rbw,

ViInt32 vbw, ViInt32 sweepPoints,
ViInt32 on)

Function purpose:

Add segment in list edit under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Index

Segment index

startfrequency

point frequency(1MHz~44.1GHz)

stopfrequency

bandwidth(150Hz~150kHz)

rbw

demodulation

vbw

limit(-174dBm~50dBm)

sweepPoints

detector

on

limit on/off, 0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set alarm on/off

ViStatus _VI_FUNC AV4041_SetAlarmOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails, the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query alarm on/off

ViStatus _VI_FUNC AV4041_QueryAlarmOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query limit alarm on/off. If the alarm is on, when the limit test on/off is on and the test fails, the alarm will give “beep” tone pip after each sweep.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set limit

ViStatus _VI_FUNC AV4041_SetLimit (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

limit(-174~50dBm).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query limit

ViStatus _VI_FUNC AV4041_QueryLimit (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Query limit.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

limit.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – set limit on/off

ViStatus _VI_FUNC AV4041_SetLimitOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

Set limit on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Limit – query limit on/off

ViStatus _VI_FUNC AV4041_QueryLimitOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

query limit on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 means succeeded, and less than 0 means failed.

Demodulation – set demodulation type

ViStatus _VI_FUNC AV4041_SetFSTDmode (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Demodulation type.

CW(0)	continuous wave
FM(1)	frequency modulation
AM(2)	ampltitude modulation
USB(3)	upper sideband
LSB(4)	lower sideband
SPEAK(5)	speak

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Demodulation – query demodulation type

ViStatus _VI_FUNC AV4041_QueryFSTDmode (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query demodulation type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

demodulation type.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Demodulation – set demodulation volume

ViStatus _VI_FUNC AV4041_SetDmodeVolume (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set the speaker volumne for demodulation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

deemodulation volume，range:0~100.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

demodulation - query demodulation volume

ViStatus _VI_FUNC AV4041_QueryDmodeVolume (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query the speaker volumne for demodulation.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

deemodulation volume.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

bandwidth – set bandwidth

ViStatus _VI_FUNC AV4041_SetFSTBw (ViSession instrumentHandle, ViReal64 dVal)

Function purpose:

Set bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (unit:Hz), range:150Hz~150kHz.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

bandwidth – query bandwidth

ViStatus _VI_FUNC AV4041_QueryFSTBw (ViSession instrumentHandle,ViReal64 dVal[])

Function purpose:

Query bandwidth.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Frequency (Hz).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – set measurement type

ViStatus _VI_FUNC AV4041_SetFstMeasureType (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

Set measure type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement types.

POTF (0) point frequency mode

FREQ(1) frequency scanning mode

LIST(2) list scanning mode

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Measurement – query measurement type

ViStatus _VI_FUNC AV4041_QueryFstMeasureType (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

Query measurement type.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

Measurement type.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker - peak

ViStatus _VI_FUNC AV4041_SetFstPeak (ViSession instrumentHandle)

Function purpose:

set the peak value of marker**.**

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set marker on/off

ViStatus _VI_FUNC AV4041_SetFstMarkerOn (ViSession instrumentHandle, ViBoolean bOn)

Function purpose:

set marker on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker on/off

ViStatus _VI_FUNC AV4041_QueryFstMarkerOn (ViSession instrumentHandle, ViBoolean bOn[])

Function purpose:

Query marker on/off.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

bOn

0 is off and 1 is on.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – set marker index

ViStatus _VI_FUNC AV4041_SetFstMarkerIndex (ViSession instrumentHandle, ViInt32 nVal)

Function purpose:

set marker index.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

marker index (0~57).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Marker – query marker index

ViStatus _VI_FUNC AV4041_QueryFstMarkerIndex (ViSession instrumentHandle, ViInt32 nVal[])

Function purpose:

query marker index.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

nVal

marker index.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query point scan amplitude value

ViStatus _VI_FUNC AV4041_QueryFstPscanAmpValue (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query point scan amplitude value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

amplitude.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query point scan field strength value

ViStatus _VI_FUNC AV4041_QueryFstPscanFstValue (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query point scan field strength value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

Field strength value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query frequency scan amplitude value

ViStatus _VI_FUNC AV4041_QueryFscanAmpValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query frequency scan amplitude value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

amplitude.

Size

Number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query frequency scan field strength value

ViStatus _VI_FUNC AV4041_QueryFscanFstValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query frequency scan field strength value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

field strength value.

Size

number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query list scan amplitude value

ViStatus _VI_FUNC AV4041_QueryLscanAmpValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query list scan amplitude value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

amplitude.

Size

number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query list scan field strength value

ViStatus _VI_FUNC AV4041_QueryLscanFstValue (ViSession vi, ViReal64 data[], ViInt32 *size)

Function purpose:

query list scan field strength value.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

data

field strength value.

Size

number of measured value.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Data – query offset

ViStatus _VI_FUNC AV4041_QueryFstOffset (ViSession instrumentHandle, ViReal64 dVal[])

Function purpose:

query offset.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

dVal

offset.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Antenna – set antenna off

ViStatus _VI_FUNC AV4041_SetAntOff (ViSession instrumentHandle)

Function purpose:

Set antenna factor loading off and set antenna factor free state.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File – recall antenna factor

ViStatus _VI_FUNC AV4041_LoadAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall antenna factor in field strength function measurement under spectrum analysis mode (the command will be invalid if the file does not exist and be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - store antenna factor

ViStatus _VI_FUNC AV4041_StoreAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Store antenna factor under field strength mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - delete antenna factor

ViStatus _VI_FUNC AV4041_DelAntFile (ViSession instrumentHandle, char chStr[])

Function purpose:

delete antenna factor file.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - delete all antenna factors

ViStatus _VI_FUNC AV4041_DelAllAntFile (ViSession instrumentHandle)

Function purpose:

delete all antenna factor files.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - delete data file

ViStatus _VI_FUNC AV4041_DeleteDataFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Delete data file under current mode (the command will be invalid if the file does not exist and be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - delete all data files

ViStatus _VI_FUNC AV4041_DeleteAllDataFile (ViSession instrumentHandle)

Function purpose:

delete all data files under current mode.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - recall data file

ViStatus _VI_FUNC AV4041_LoadDateFile (ViSession instrumentHandle, char chStr[])

Function purpose:

Recall data file under current mode (the command will be invalid if the file does not exist and be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

File - store data file

ViStatus _VI_FUNC AV4041_StoreDataFile (ViSession instrumentHandle,

char chStr[])

Function purpose:

Store current measurement data as data file (the file will be overwritten if the file exists and

this command be only valid for current storage location).

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

chStr

Document name.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Edit antenna factor – add default point

ViStatus _VI_FUNC AV4041_AntennaAddDefault (ViSession instrumentHandle)

Function purpose:

edit antenna factor to add default point. Frequency: 1GHz antenna factor value: 0dB

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Edit antenna factor – delete point

ViStatus _VI_FUNC AV4041_AntennaDelete (ViSession instrumentHandle,ViInt32 index)

Function purpose:

edit antenna factor to delete point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Edit antenna factor – edit point

ViStatus _VI_FUNC AV4041_AntennaEdit (ViSession instrumentHandle, ViInt32 index, ViReal64 frequency, ViReal64 factor)

Function purpose:

edit antenna factor to edit point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

index

Point index.

frequency

frequency (0~44.1GHz).

factor

Antenna factor value (-200~200dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.

Edit antenna factor – add point

ViStatus _VI_FUNC AV4041_AntennaAdd (ViSession instrumentHandle, ViReal64 frequency, ViReal64 factor)

Function purpose:

edit antenna factor to add point.

Parameter list:

instrumentHandle

Instrument handle returned by the function for communication with the instrument.

frequency

Frequency (0~44.1GHz).

factor

Antenna factor value (-200~200dB).

Return value:

The return value represents the function execution: 0 mean succeeded, and less than 0 means failed.