4024CA Spectrum Analyzer

Programming Manual

Ceyear Technologies Co., Ltd.

VERIFICATION OF CONFORMITY

Certificate No.: CTL2107013135-EC

Applicant	:	Ceyear Technologies Co.,Ltd
Address	:	No.98, Xiangjiang Rd, Huangdao District, Qingdao City, Shandong Province
Product	:	Spectrum Analyzer
Trademark	:	思仪/Ceyear
Model(s)	:	4024D, 4024, 4024A, 4024B, 4024C, 4024E, 4024F, 4024G, 4024H, 4024L, 4024CA
Manufacturer	:	Ceyear Technologies Co.,Ltd
Address	:	No.98, Xiangjiang Rd, Huangdao District, Qingdao City, Shandong Province
Test Report	:	CTL2107013135-E

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

EN 61326-1: 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
Oct. 28, 2021

Shenzhen CTL Testing Technology Co., Ltd.
Add.: Floor 1-A, Baisha Technology Park, No.3011, Shahexi Road, Nanshan District, Shenzhen, China 518055
Tel: 86-755-89485194 E-Mail: ctl@ctl-lab.com web: www.ctl-lab.com

VERIFICATION OF CONFORMITY

Certificate No.: CTL2107013135-SC

Applicant	:	Ceyear Technologies Co.,Ltd
Address	:	No.98, Xiangjiang Rd, Huangdao District, Qingdao City, Shandong Province , China
Product	:	Spectrum Analyzer
Trademark	:	思仪/Ceyear
Model(s)	:	4024, 4024A, 4024B, 4024C, 4024D, 4024E, 4024F, 4024G, 4024H, 4024L, 4024CA
Manufacturer	:	Ceyear Technologies Co.,Ltd
Address	:	No.98, Xiangjiang Rd, Huangdao District, Qingdao City, Shandong Province , China
Test Report	:	CTL2107013135-S

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

EN 61010-1:2010+A1:2019

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
Oct 29, 2021

Shenzhen CTL Testing Technology Co., Ltd.
Add.: Floor 1-A, Baisha Technology Park, No.3011, Shahexi Road, Nanshan District, Shenzhen, China 518055
Tel: 86-755-89486194 E-Mail: ctl@ctl-lab.com web: www.ctl-lab.com

VERIFICATION OF CONFORMITY

Certificate No. : CTL2107013135-RC

Applicant	:	Ceyear Technologies Co.,Ltd
Address	:	No.98, Xiangjiang Rd, Huangdao District, Qingdao City, Shandong Province, China.
Product	:	Spectrum Analyzer
Trademark	:	思仪/Ceyear
Model(s)	:	4024D, 4024, 4024A, 4024B, 4024C, 4024E, 4024F, 4024G, 4024H, 4024L, 4024CA
Manufacturer	:	Ceyear Technologies Co.,Ltd
Address	:	No.98, Xiangjiang Rd, Huangdao District, Qingdao City, Shandong Province, China.
Test Report	:	CTL2107013135-R

Complies with the requirements of the
EC RoHS Directive 2011/65/EU & Amendment 2015/863
Test Standards:

IEC 62321-1:2013, IEC 62321-2:2013
IEC 62321-3-1:2013, IEC 62321-3-2:2013
IEC62321-4: 2013+A1:2017, IEC 62321-5:2013
IEC 62321-6:2015, IEC 62321-7-1: 2015
IEC 62321-7-2: 2017, IEC 62321-8: 2017

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
Oct. 28, 2021

Shenzhen CTL Testing Technology Co., Ltd.
Add.: Floor 1-A, Baisha Technology Park, No.3011, Shahexi Road, Nanshan District, Shenzhen, China 518055
Tel: 86-755-89486194 E-Mail: ctl@ctl-lab.com web: www.ctl-lab.com

Foreword

Thank you for choosing and using 4024CA spectrum analyzer developed and produced by Ceyear Technologies Co., Ltd.!

We will take the responsibility to meet your needs and provide you with high-quality instruments and good after-sales service. We aim to provide "high quality and considerate service", and operate on the principle of making customers satisfactory with our products and services. If you have any questions or doubts, please contact us:

Service Consultation: 0532-86889847 400-1684191

Technical Support: 0532--86880796

Quality Supervision: 0532--86886614

Fax:0532--86889056

Website: www.ceyear.com

E-mail: techbb@ceyear.com

Address: No. 98, Xiangjiang Road, Huangdao District, Qingdao, ： Shandong Province

Zip Code: 266555

This manual mainly introduces the programming method of 4024CA spectrum analyzers developed and produced by Ceyear Technologies Co., Ltd., and how the analyzer is program-controlled by external control computers via LAN interfaces or USB interfaces, to help you get familiar with and master the program control methods and commands of the analyzer.

Due to our continuous efforts to enhance the performance of our analyzers by updating and improving our hardware and firmware, the operation and control methods of the analyzer described in this manual may be updated. For the latest technical information about this manual, please feel free to contact our Technical Support.

Due to tight schedule and limited ability of the author, there might be some inevitable errors or omissions in this guide, so please do not hesitate to give your commends if you find such problems! We apologize for any inconvenience possibly caused by our mistakes.

Declaration:

This manual is the first edition of the Programming Manual for 4024CA Spectrum Analyzers.

The contents of this manual are subject to change without notice.

The contents and terms used in this manual are interpreted by Ceyear Technologies Co., Ltd.

Compiler

April 2022

Table of Contents

Chapter I Description of SCPI Commands

Section I Introduction to SCPI Command Operations

SCPI (Standard Commands for Programmable Instruments) is a new command language for controlling equipment in accordance with the IEEE488.2 standard. Its main purpose is to make the same kind of equipment have the same program control commands, so as to realize the standardization of program control commands.

This chapter contains all SCPI command information identified and executed by 4024CA spectrum analyzer. It includes the introduction to and description of IEEE488.2 general commands and measurement commands.

Each measurement command has an applicable mode. In the non applicable mode, an error ID will be returned for the query command.

Section II IEEE 488.2 Commands

*CLS - Clear Status

Clear the instrument status, namely: empty error queue and all event register. At the same time cancel all pending *OPC commands and query commands.

*IDN? - Identify

Return the unique instrument identification string, which varies with the different models. E.g.: “CEYEAR, 4024CA, SN, 1.00”.

*OPC - Operation Complete Command

Set OPC bit of the standard event status register, after completing all pending overlapping commands (e.g.: primary sweep or Default command, etc.).

*OPC? - Operation Complete Query

Return letter “1” after completing all pending overlapping commands.

*RST - Reset

Perform reset operation, cancel all pending *OPC commands or query commands. The nonvolatile memory contents of the instrument are not lost.

*WAI - Wait

The instrument processes the new commands after completion of processing all the pending overlapping commands.

Section III Measurement commands

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

(Read-write) query or set audio alarm on/off upon limits.

Applicable Mode	Spectrum Analysis, Orientation Analysis, Real-time Spectrum
Parameter	Audible alarm ON/OFF
	OFF(0) Audible alarm off
	ON(1) Audible alarm on
Set Syntax	:CALC:LIM:BEEP ON
Query Syntax	:CALC:LIM:BEEP?
Default	OFF
Return Type	Value (bool) or character

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

(Read-write) Query or set lower limit display ON/OFF.

Applicable Mode	Spectrum Analysis, Real-time Spectrum
Parameter	Lower limit display ON/OFF
	OFF(0) Display off
	ON(1) Display on
Set Syntax	:CALC:LIM:LOW:DISP ON
Query Syntax	:CALC:LIM:LOW:DISP?
Default	OFF
Return Type	Value (bool) or character

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

(Write only) All points for lower limit.

Applicable Mode	Spectrum Analysis, Real-time Spectrum
Parameter	None
Set Syntax	:CALC:LIM:LOW:EDIT:ADD
Query Syntax	None
Default	None
Return Type	None

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

(Write only) Delete all edit points for lower limit.

Applicable Mode	Spectrum Analysis, Real-time Spectrum
Parameter	None
Set Syntax	:CALC:LIM:LOW:EDIT:CLE
Query Syntax	None
Default	None

Return Type None

:CALCulate[:SELected]:LIMit:LOWer:EDIT:DATA <I00>,<FF0>,<F00>

(Read-write) Edit points for lower limit.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter Point index (int, starting from 0)
Limit Frequency: (0~9.1GHz)
Limit Value (-174dBm ~50dBm)

Set Syntax :CALC:LIM:LOW:EDIT:DATA 1,1000000,-10

Query Syntax :CALC:LIM:LOW:EDIT:DATA? 1

Default 1,0,-75

Return Type Value (float) or character

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

(Write only) Delete current point for lower limit

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter None

Set Syntax :CALC:LIM:LOW:EDIT:DEL

Query Syntax None

Default None

Return Type None

:CALCulate[:SELected]:LIMit:LOWer:MARGin <F00>

(Read-write) Query or set lower limit margin value

Applicable Mode Spectrum Analyzer

Parameter Lower limit margin value (dBm)
Parameter Range: 0.00~40.00

Set Syntax :CALC:LIM:LOW:MARG 0

Query Syntax :CALC:LIM:LOW:MARG?

Default 0

Return Type Value (float) or character

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

(Read-write) Query or set lower limit test ON/OFF.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter Lower limit test ON/OFF
OFF(0) Test OFF
ON (1) Test ON

Set Syntax :CALC:LIM:LOW:TEST ON

Query Syntax :CALC:LIM:LOW:TEST?

Default OFF

Return Type String or value (int)

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

(Read-write) Query or set upper limit display ON/OFF.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter Upper limit display ON/OFF

OFF(0) Display off

ON(1) Display on

Set Syntax :CALC:LIM:UPP:DISP ON

Query Syntax :CALC:LIM:UPP:DISP?

Default OFF

Return Type String or value (int)

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

(Write only) Add upper limit default point.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter None

Set Syntax :CALC:LIM:UPP:EDIT:ADD

Query Syntax None

Default None

Return Type None

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

(Write only) Delete all upper limit edit points.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter None

Set Syntax :CALC:LIM:UPP:EDIT:CLE

Query Syntax None

Default None

Return Type None

:CALCulate[:SELected]:LIMit:UPPer:EDIT:DATA <I00>,<FF0>,<F00>

(Read-write) Upper limit edit points.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter Point index (int, starting from 0)

Limit Frequency:(0~9.1GHz)

Limit Value (-174dBm ~ 50dBm)

Set Syntax :CALC:LIM:UPP:EDIT:DATA 1,1000000,-10

Query Syntax :CALC:LIM:UPP:EDIT:DATA? 1

Default 1,0,-25

Return Type Value (float) or character

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

(Write only) Delete current point for upper limit.

Applicable Mode Real-time Spectrum

Parameter None

Set Syntax :CALC:LIM:UPP:EDIT:DEL

Query Syntax None

Default None

Return Type None

:CALCulate[:SELected]:LIMit:UPPer:MARGin <F00>

(Read-write) Query or set upper limit margin value.

Applicable Mode Spectrum Analyzer

Parameter Upper limit margin value (dBm)

Parameter Range: -40.00~0.00

Set Syntax :CALC:LIM:UPP:MARG -10

Query Syntax :CALC:LIM:UPP:MARG?

Default 0

Return Type Value (float) or character

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

(Read-write) Query or set upper limit test ON/OFF.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter Upper limit test ON/OFF

OFF(0) Test OFF

ON (1) Test ON

Set Syntax :CALC:LIM:UPP:TEST ON

Query Syntax :CALC:LIM:UPP:TEST?

Default OFF

Return Type String or value (int)

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

(Write only) Edit the list to add a default segment, which is set as:

Start Freq	1GHz
Stop Freq	2GHz

RBW	1MHz
VBW	30kHz
Sweep Points	201
ON/OFF	On

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax :CALC:LIST:EDIT:ADD

Query Syntax None

Default None

Return Type None

:CALCulate[:SELected]:LIST:EDIT:ADD:SEGMent <FF0>,<FF0>,<I00>,<FF0>,<FF0>,<E>

(Write only) Edit list - add segment

Applicable Mode Spectrum Analyzer

Parameter Start frequency (0~9.1GHz)
Stop frequency (0~9.1GHz)
Sweep points (201~1001)
Resolution bandwidth (1Hz~40kHz)
Video bandwidth (1Hz~40MHz)
ON/OFF (ON OFF)

Set Syntax :CALC:LIST:EDIT:ADD:SEG 100,1000,201,1,1,ON

Query Syntax None

Default 1000000000,2000000000,201,1000000,30000,ON

Return Type None

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

(Write only) Edit list - clear list.

Applicable Mode Spectrum Analysis, Field Strength Measurement

Parameter None

Set Syntax CALC:LIST:EDIT:CLE

Query Syntax None

Default None

Return Type None

:CALCulate[:SELected]:LIST:EDIT:DELete <I00>

(Write only) Delete a segment of the list.

Applicable Mode Spectrum Analyzer

Parameter Segment index (int, starting from 0)

Set Syntax :CALC:LIST:EDIT:DEL 1

Query Syntax None

Default 1

Return Type None

:CALCulate[:SELected]:LIST:EDIT:SEGMent <I00>,<FF0>,<FF0>,<I00>,<FF0>,<FF0>,<E>

(Read-write) Edit list editor segment

Applicable Mode Spectrum Analyzer

Parameter Segment index (starting from 0)

Start frequency (0~9.1GHz)

Stop frequency (0~9.1GHz)

Sweep points (201~1001)

Resolution bandwidth (1Hz~40kHz)

Video bandwidth (1Hz~40MHz)

ON/OFF (ON OFF)

Set Syntax :CALC:LIST:EDIT:SEG 1,100,1000,201,1,1,ON

Query Syntax None

Default 1,1000000000,2000000000,201,1000000.30000,ON

Return Type None

:CALCulate[:SELected]:MARKer[1]|2-6:ACTivate

(Write only) Activate marker n.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter<n> Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Parameter None

Set Syntax :CALC:MARK1:ACT

Query Syntax None

Default None

Return Type None

:CALCulate[:SELected]:MARKer[1]|2-6:AOFF

(Write only) Turn off all markers.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter None

Set Syntax :CALC:MARK:AOFF

Query Syntax None

Default None

Return Type None

:CALCulate[:SELected]:MARKer[1]|2-6:FCOunt[:STATe] <E>

(Read-write) Query or set counter ON/OFF**(The marker will be set to the normal marker)**.

Applicable Mode Spectrum Analyzer

Parameter<n> Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Parameter Counter ON/OFF

OFF(0) Counter OFF.

ON(1) Counter ON.

Note: Only one marker counter can be simultaneously turned on currently.

Set Syntax :CALC:MARK1:FCO ON

Query Syntax :CALC:MARK1:FCO?

Default OFF

Return Type String or value (int)

:CALCulate[:SELected]:MARKer[1]|2-6:FCOunt:X

(Read only) Query count frequency on the counter **(return to 0 if counter is not started)**.

Applicable Mode Spectrum Analyzer

Parameter<n> Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Parameter None

Set Syntax None

Query Syntax :CALC:MARK1:FCO:X?

Default None

Return Type Value (double) or character

:CALCulate[:SELected]:MARKer[1]|2-6:FUNCtion:MAXimum

(Write only) search for max. value of Marker **(turn on marker if it's not turned on)**.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter<n> Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Parameter None

Set Syntax :CALC:MARK1:FUNC:MAX

Query Syntax None

Default None

Return Type	None

:CALCulate[:SELected]:MARKer[1]|2-6:FUNCtion:MINimum

(Write only) search for min. value of Marker **(turn on marker if it's not turned on)**.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter<n>	Marker number n It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.
Parameter	None
Set Syntax	:CALC:MARK1:FUNC:MIN
Query Syntax	None
Default	None
Return Type	None

:CALCulate[:SELected]:MARKer[1]|2-6:FUNCtion:PEAK

(Write only) search for peak value of marker **(turn on marker if it's not turned on)**.

Applicable Mode	Spectrum Analysis, Interference Analysis, Orientation Analysis, Real-time Spectrum
Parameter<n>	Marker number n It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.
Parameter	None
Set Syntax	:CALC:MARK1:FUNC:PEAK
Query Syntax	None
Default	None
Return Type	None

:CALCulate[:SELected]:MARKer[1]|2-6:FUNCtion:PLEFt

(Write only) search for left peak of marker **(turn on marker if it's not turned on)**.

Applicable Mode	Spectrum Analysis, Interference Analysis, Orientation Analysis, Real-time Spectrum
Parameter<n>	Marker number n It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.
Parameter	None
Set Syntax	:CALC:MARK1:FUNC:PLEF
Query Syntax	None
Default	None
Return Type	None

:CALCulate[:SELected]:MARKer[1]|2-6:FUNCtion:PNEXt

(Write only) search for Sub Peak of marker **(turn on marker if it's not turned on)**.

Applicable Mode	Spectrum Analysis, Interference Analysis, Orientation Analysis, Real-time Spectrum
Parameter<n>	Marker number n It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.
Parameter	None
Set Syntax	:CALC:MARK1:FUNC:PNEX
Query Syntax	None
Default	None
Return Type	None

:CALCulate[:SELected]:MARKer[1]|2-6:FUNCtion:PRIGht

(Write only) search for right peak of marker **(turn on marker if it's not turned on)**.

Applicable Mode	Spectrum Analysis, Interference Analysis, Orientation Analysis, Real-time Spectrum
Parameter<n>	Marker number n It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.
Parameter	None
Set Syntax	:CALC:MARK1:FUNC:PRIG
Query Syntax	None
Default	None
Return Type	None

:CALCulate[:SELected]:MARKer[1]|2-6:NOISe[:STATe] <E>

(Read-write) Query or set noise marker**(turn on marker if it is not turned on)**.

Applicable Mode	Spectrum Analysis, Interference Analysis
Parameter<n>	Marker number n It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.
Parameter	Noise marker ON/OFF OFF(0) Noise marker OFF. ON(1) Noise marker ON.
Set Syntax	:CALC:MARK1:NOIS ON
Query Syntax	:CALC:MARK1:NOIS?
Default	OFF
Return Type	String or value (int)

:CALCulate[:SELected]:MARKer[1]|2-6:SET <E>

(Write only) Set marker function (marker->)**(turn on marker if it's not turned on)**.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter<n>	Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Parameter Marker function

Span, non-zero span modes & parameters and function correspondence:

Instrument mode	Parameter	Function
Non-zero span	STARt,STOP,CENTer,STEP 0,1,2,3	Set the start, stop, center, and step frequencies to the current marker frequency value
Span Zero	STARt,STOP,CENTer,STEP 0,1,2,3	Set the marker index to 0, maximum index, center index; set step frequency to the current marker frequency value.

Set Syntax :CALC:MARK1:SET STAR

Query Syntax None

Default STARt

Return Type None

:CALCulate[:SELected]:MARKer[1]|2-6[:STATe] <E>

(Read-write) query or set state of Marker.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter<n> Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Parameter Marker state

OFF(0) Marker OFF

NORM(1) Normal marker ON

DELTa(2) Delta marker ON

Set Syntax :CALC:MARK1 NORM

Query Syntax :CALC: CALC:MARK1?

Default OFF

Return Type String or value (int)

:CALCulate[:SELected]:MARKer[1]|2-6:X <FF0>

(Read-write) Query or set X value of the marker**(Invalid if the marker is not turned on)**。

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter<n> Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Parameter Marker X value (see the following table for the corresponding unit).

X value can be negative when the marker is a delta marker.

Instrument mode	Parameter unit
Spectrum analysis (non-zero span)	Hz
Spectrum analysis (zero span)	ns
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Real-time spectrum	Hz

Time is in ns and frequency is in Hz.

Set Syntax :CALC:MARK1:X 10000

Query Syntax :CALC:MARK1:X?

Default 4.55GHz

Return Type Value (double) or character

:CALCulate[:SELected]:MARKer[1]|2-6:Y

(Read only) Query marker Y value **(return to 0 if marker is not turned on)**.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter<n> Marker number n

It can be set to 1,2,3,4,5 and 6, respectively indicating the marker 1,2,3,4,5 and 6. n is 1 if not specified.

Set Syntax None

Query Syntax :CALC:MARK1:Y?

Default None

Return Type String or block

Instrument mode	Return parameters	Unit
Spectrum Analyzer (single float value)	Amplitude value	dBm
IA (single float value)	Amplitude value	dBm
Real-time spectrum (single float value)	Amplitude value	dBm

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

(Read-write) Query or set peak tracking ON/OFF.

Applicable Mode Spectrum Analyzer

Parameter Peak tracking ON/OFF

OFF(0) Peak tracking OFF

ON(1) Peak tracking ON

Set Syntax :CALC:PEAK:TRAC ON

Query Syntax :CALC:PEAK:TRAC?

Default OFF

Return Type String or value (int)

:DISPlay:BRIG <I00>

(Read-write) Query or set brightness level.

Applicable Mode all modes

Parameter Brightness level

Range: 1～5.

Set Syntax :DISP:BRIG 1

Query Syntax :DISP:BRIG?

Default 4

Return Type Value (int) or character

:DISPlay:BRIG:AUTO <E>

(Read-write) Query or set brightness auto ON/OFF.

Applicable Mode all modes

Parameter Brightness auto ON/OFF

OFF(0) brightness auto OFF

ON(1) brightness auto ON

Set Syntax :DISP:BRIG:AUTO ON

Query Syntax :DISP:BRIG:AUTO?

Default OFF

Return Type String or value (int)

:DISPlay:MODE <E>

(Read-write) Query or set display mode.

Applicable Mode all modes

Parameter Display mode

DEFA(0) Default mode

OUT(1) Outdoor mode

NIGHT(2) Night vision mode

Set Syntax :DISP:MODE DEFA

Query Syntax : DISP:MODE?

Default DEFA

Return Type String or value (int)

:DISPlay:TIME:FMT <E>

(Write only) Set time format.

Applicable Mode	all modes
Parameter	Time format
	YMD(0) Year/Month/Day
	MDY(1) Month/Day/Year
	DMY(2) Day/Month/Year
Set Syntax	:DISP:TIME:FMT YMD
Query Syntax	None
Default	YMD
Return Type	Value (int) or character

:DISPlay:TITLe <S>

(Write only) Set title.

Applicable Mode	all modes
Parameter	Title, up to 10 letters or numbers.
Set Syntax	:DISP:TITL "OBWMEASURE"
Query Syntax	None
Default	None
Return Type	None

:DISPlay:TITLe:STATe <E>

(Read-write) Query or set title ON/OFF.

Applicable Mode	all modes
Parameter	title ON/OFF
	OFF(0) Title OFF
	ON (1) Title ON
Set Syntax	:DISP:TITL:STAT ON
Query Syntax	:DISP:TITL:STAT?
Default	OFF
Return Type	String or value (int)

:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision <F00>

(Read-write) query or set Scale/Div.

Applicable Mode	Spectrum Analysis (Settable only in logarithmic scale type), Interference Analysis, Orientation Analysis, Real-time Spectrum, 4G, 5G
Parameter	Scale/Div (dB)
	Range: 0.01dB-100dB.
Set Syntax	:DISP:WIND:TRAC:Y:PDIV 0.1

Query Syntax	:DISP:WIND:TRAC:Y:PDIV?
Default	10dB
Return Type	Value (float) or character

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <F00>

(Read-write) Query or set reference level.

Applicable Mode	Spectrum Analysis, Interference Analysis, Orientation Analysis, Real-time Spectrum, 4G, 5G
Parameter	Reference level value (reference value). The reference level value is related to the current amplitude unit, the setting range corresponds to dBm, and conversion is required. Range: -150dBm~ +30dBm
Set Syntax	:DISP:WIND:TRAC:Y:RLEV -10
Query Syntax	:DISP:WIND:TRAC:Y:RLEV?
Default	-0dBm
Return Type	Value (float) or character

:DISPlay:WINDow:TRACe:Y[:SCALe]:RPOSition <I00>

(Read-write) Query or set reference position.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter	Reference position (no unit) Range: -10～10.
Set Syntax	:DISP:WIND:TRAC:Y:RPOS 1
Query Syntax	:DISP:WIND:TRAC:Y:RPOS?
Default	0
Return Type	Value (int) or character

:FORM[:DATA] <E>

(Read-write) query or set data format.

Applicable Mode	all modes
Parameter	Data format ASC(0) Character format HEX(1) Numeric format If set to character format, the query returns data in character format, in characters If set to numeric format, the query returns data in numeric format, in bytes.
Set Syntax	:FORM ASC
Query Syntax	:FORM?
Default	ASC
Return Type	Value (bool) or character

:INITiate

(Write only) trigger sweep once (valid only in Swp Once.). This command is an interleaved command. Use ***OPC?**to query whether the command is completed before sending other commands.

Applicable Mode	all modes
Parameter	None
Set Syntax	:INIT;*OPC?
Query Syntax	None
Default	None
Return Type	None

:INITiate:CONTinuous <E>

(Read-write) query or set sweep type. Set the single sweep command to an interleaved command. Use ***OPC?**to query whether the command is completed before sending other commands.

Applicable Mode	all modes
Parameter	Sweep type OFF(0) single sweep ON(1) continuous sweep
Set Syntax	:INIT:CONT OFF;*OPC?
Query Syntax	:INIT:CONT?
Default	ON
Return Type	String or value (int)

:INSTrument:CATalog

(Read only) Query the available instrument operating modes, using **:INST:CAT ?** Available instrument operating modes can be queried.

Applicable Mode	all modes
Parameter	None
Set Syntax	None
Query Syntax	:INST:CAT?
Default	0x01
Return Type	Value (int) or character Bit 0 is the spectrum analysis test bit, which is 1 (required); Bit 1 is the interference analysis test bit, with 1 being settable (optional) and 0 being non-settable; Bit 2 is the directional analysis test bit, with 1 being settable (optional) and 0 being non-settable; Bit 3 is the GSM/EDGE test bit, with 1 being settable (optional) and 0 being non-settable; Bit 4 is the 5GNR test bit, with 1 being settable (optional) and 0 being non-settable;

Bit 5 is the LTE analysis test bit, 1 is settable (optional), 0 is not settable.

:INSTrument[:SELect] <E>

(Read-write) Query or set current instrument operating mode, using :INST:CAT? The available instrument operating modes can be queried. This command is an interleaved command. Use ***OPC?**to query whether the command is completed before sending other commands.

Applicable Mode	all modes	
Parameter	Instrument mode	
	SA(1)	Spectrum analysis
	IA(2)	Interference analysis
	COMPASS(3)	Orientation Analysis
	RTSA(4)	Real-time spectrum analysis
	GSM(5)	GSM/EDGE
	NR(6)	5GNR
	LTE(7)	LTE analysis
Set Syntax	:INST SA	
Query Syntax	:INST?	
Default	SA	
Return Type	String or value (int)	

:MMEMory:DELete:ANTenna

(Write only) delete antenna file in the current mode (**the command is invalid if the file does not exist and is valid only for the current storage location**).

Applicable Mode	Spectrum Analysis, Orientation Analysis
Parameter	Antenna file name
Set Syntax	:MMEM:DEL:ANT "AV89101A"
Query Syntax	None
Default	None
Return Type	None

:MMEMory:DELete:ANTenna:ALL

(Write only) Delete all antenna files in the current mode.

Applicable Mode	Spectrum Analysis, Orientation Analysis
Parameter	None
Set Syntax	:MMEM:DEL:ANT:ALL
Query Syntax	None
Default	None
Return Type	None

:MMEMory:DELete:DATA <S>

(Write only) Delete data file in the current mode.

Applicable Mode all modes

Parameter None

Set Syntax :MMEM:DEL:DATA "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:DELete:DATA:ALL

(Write only) Delete all data files in the current mode.

Applicable Mode all modes

Parameter None

Set Syntax :MMEM:DEL:DATA:ALL

Query Syntax None

Default None

Return Type None

:MMEMory:DELete:LIMit <S>

(Write only) delete limit file in the current mode (**the command is invalid if the file does not exist and is valid only for the current storage location**).

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter Limit file name

Set Syntax :MMEM:DEL:LIM "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:DELete:LIMit:ALL

(Write only) Delete all limit files in the current mode.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter None

Set Syntax :MMEM:DEL:LIM:ALL

Query Syntax None

Default None

Return Type None

:MMEMory:DELete:LIST <S>

(Write only) delete list file in the current mode (**the command is invalid if the file does not exist and is valid only for the current storage location**).

Applicable Mode Spectrum Analyzer

Parameter List file name

Set Syntax :MMEM:DEL:LIST "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:DELete:LIST:ALL

(Write only) Delete all list files in the current mode.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax :MMEM:DEL:LIST:ALL

Query Syntax None

Default None

Return Type None

:MMEMory:DELete:STATe <S>

(Write only) delete state file in the current mode(**the command is invalid if the file does not exist and is valid only for the current storage location**).

Applicable Mode all modes

Parameter state file name

Set Syntax :MMEM:DEL:STAT "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:DELete:STATe:ALL

(Write only) delete all state files in the current mode.

Applicable Mode all modes

Parameter None

Set Syntax :MMEM:DEL:STAT:ALL

Query Syntax None

Default None

Return Type None

:MMEMory:LOAD:ANTenna <S>

(Write only) Select antenna factor in field strength measurement**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Applicable Mode Spectrum Analysis, Orientation Analysis

Parameter Antenna factor file name

Set Syntax :MMEM:LOAD:ANT "AV89101A"

Query Syntax None

Default Antenna factor not selected

Return Type None

:MMEMory:LOAD:DATA <S>

(Write only) Recall date files in the current mode**(the command is invalid if the file does not exist and is valid only for the current storage location)**。

Applicable Mode	all modes
Parameter	Data file name
Set Syntax	:MMEM:LOAD:DATA "set1"
Query Syntax	None
Default	None
Return Type	None

:MMEMory:LOAD:FILE <E>,<S>

(Read only) Download files

Applicable Mode	all modes
Parameter	File type. The optional file types are ANT(0), LIST(1), SALIMIT(2), RTSALIMIT(3), SASTATE(4), IASTATE(5), DMSTATE(6), LASTATE(7), RTSASTATE(8), NRSTATE(9), GSMSTATE(10), LTESTATE(11), ERROR(12), SADATA(13), IADATA(14), DMDATA(15), LADATA(16), RTSADATA(17), NRDATA(18), GSMDATA(19), LTEDATA(20), SADANL(21), SARRES(22). Document title
Set Syntax	None
Query Syntax	:MMEMory:LOAD:FILE? ANT,"AV89101.ant"
Default	ANT
Return Type	Block

:MMEMory:LOAD:FILE:NAME <E>

(Read only) Download file name list

Applicable Mode	all modes
Parameter	File type. The optional file types are ANT(0), LIST(1), SALIMIT(2), RTSALIMIT(3), SASTATE(4), IASTATE(5), DMSTATE(6), LASTATE(7), RTSASTATE(8), NRSTATE(9), GSMSTATE(10), LTESTATE(11), SADATA(12), IADATA(13), DMDATA(14), LADATA(15), RTSADATA(16), NRDATA(17), GSMDATA(18), LTEDATA(19).
Set Syntax	None
Query Syntax	:MMEMory:LOAD:FILE:NAME? ANT
Default	ANT
Return Type	Block

:MMEMory:LOAD:LIMit <S>

(Write only) Load limit line **(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Applicable Mode	Spectrum Analysis, Real-time Spectrum

Parameter Limit line file name

Set Syntax :MMEM:LOAD:LIM "NAME"

Query Syntax None

Default None

Return Type None

:MMEMory:LOAD:LIST <S>

(Write only) Load list file **(the** command **is invalid if the file does not exist and is valid only for the current storage location)**。

Applicable Mode Spectrum Analyzer

Parameter List file name

Set Syntax :MMEM:LOAD:LIST "NAME"

Query Syntax None

Default None

Return Type None

:MMEMory:LOAD:SEM <S>

(Write only) Load spurious emission mask file **(the command is invalid if the file does not exist and is valid only for the current storage location)**。

Applicable Mode Spectrum Analyzer

Parameter Spurious emission mask file name

Set Syntax :MMEM:LOAD:SEM a.lmt

Query Syntax None

Default None

Return Type None

:MMEMory:LOAD:STATe <S>

(Write only) Recall state file in the current mode(**the command is invalid if the file does not exist and is valid only for the current storage location**).

Applicable Mode all modes

Parameter state file name

Set Syntax :MMEM:LOAD:STAT "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:LOCation <E>

(Read-write) query or set storage location.

Applicable Mode all modes

Parameter Storage location

INT(0) Internal

SD(1) SD card

USB(2) USB

Set Syntax :MMEM:LOC USB

Query Syntax :MMEM:LOC?

Default INT

Return Type String or value (int)

:MMEMory:STORe:ANTenna <S>

(Write only) Store antenna factor file**(the file will overwrite an existing file and is valid only for the current storage location)**.

Applicable Mode Spectrum Analysis, Orientation Analysis

Parameter Antenna factor file name

Set Syntax :MMEM:STOR:ANT "AV89101A"

Query Syntax None

Default None

Return Type None

:MMEMory:STORe:DATA <S>

(Write only) Store data files in current mode**(the file will overwrite an existing file and is valid only for the current storage location)**.

Applicable Mode all modes

Parameter Data file name

Set Syntax :MMEM:STOR:DATA "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:STORe:FDATa <S>

(Write only) Store CSV data file in current mode**(the file will overwrite an existing file and is valid only for the current storage location)**.

Applicable Mode all modes

Parameter Data file name

Set Syntax :MMEM:STOR:FDAT "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:STORe:LIMit <S>

(Write only) Store current limit line as a file**(the file will overwrite an existing file and is valid only for the current storage location)**.

Applicable Mode Spectrum Analysis, Real-time Spectrum

Parameter Limit file name

Set Syntax :MMEM:STOR:LIM "NAME"

Query Syntax None

Default None

Return Type None

:MMEMory:STORe:LIST <S>

(Write only) Store current list data to file**(the file will overwrite an existing file and is valid only for the current storage location)**.

Applicable Mode Spectrum Analyzer

Parameter List file name

Set Syntax :MMEM:STOR:LIST "NAME"

Query Syntax None

Default None

Return Type None

:MMEMory:STORe:SCReen <S>

(Write only) copy the screen and save the snapshot of the current screen into a file (**the file will overwrite an existing file and is valid only for the current storage location**).

Applicable Mode all modes

Parameter Screen shot file name

Set Syntax :MMEM:STOR:SCR "pic1"

Query Syntax None

Default None

Return Type None

:MMEMory:STORe:STATe <S>

(Write only) save the state in the current mode as a file (**the file will overwrite an existing file and is valid only for the current storage location**).

Applicable Mode all modes

Parameter state file name

Set Syntax :MMEM:STOR:STAT "set1"

Query Syntax None

Default None

Return Type None

:MMEMory:UPLoad:ANT <B>

(Write only) Upload antenna factor file.

Applicable Mode all modes

Parameter Data block

Set Syntax :MMEM:UPL:ANT BLOCK

Query Syntax None

Default None

Return Type None

:MMEMory:UPLoad:LIMit <B>

(Write only) Upload limit line file.

Applicable Mode all modes

Parameter Data block

Set Syntax :MMEM:UPL:LIM BLOCK

Query Syntax None

Default None

Return Type None

:MMEMory:UPLoad:LIST <B>

(Write only) Upload list file.

Applicable Mode all modes

Parameter Data block

Set Syntax :MMEM:UPL:LIST BLOCK

Query Syntax None

Default None

Return Type None

[:SENSe]:ACPower:ADJChbw <FF0>

(Read-write) Query or set adjacent channel bandwidth in the adjacent channel power ratio.

Applicable Mode Spectrum Analyzer

Parameter ACPR ACBW (Hz)

Range： 300Hz～20MHz.

Set Syntax :ACP:ADJC 3000000

Query Syntax :ACP:ADJC?

Default 3MHz

Return Type Value (double) or character

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

(Read-write) Query or set limit test ON/OFF in the adjacent channel power ratio.

Applicable Mode Spectrum Analyzer

Parameter Limit test ON/OFF

OFF(0) limit test OFF

ON (1) limit test ON

Set Syntax :ACP:LIM OFF

Query Syntax : ACP:LIM?

Default OFF

Return Type String or value (int)

[:SENSe]:ACPower:LOWer

(Read only) Query lower adjacent channel power ratio in the adjacent channel power ratio.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax None

Query Syntax : ACP:LOW?

Default None

Return Type Value (float) or character

[:SENSe]:ACPower:MAINchbw <FF0>

(Read-write) Query or set adjacent channel power ratio to the main channel bandwidth.

Applicable Mode Spectrum Analyzer

Parameter ACPR main channel bandwidth (Hz)

Range： 300Hz～20MHz.

Set Syntax :ACP:MAIN 3000000

Query Syntax :ACP:MAIN?

Default 10MHz

Return Type Value (double) or character

[:SENSe]:ACPower:OFFSet:LLIMit <F00>

(Read-write) Query or set adjacent channel power ratio and lower adjacent channel limit.

Applicable Mode Spectrum Analyzer

Parameter ACPR lower adjacent channel limit (dB)

Range：-200dB~200dB.

Set Syntax :ACP:OFFS:LLIM 0

Query Syntax :ACP:OFFS:LLIM?

Default 0

Return Type Value (float) or character

[:SENSe]:ACPower:OFFSet:ULIMit <F00>

(Read-write) Query or set adjacent channel power ratio and upper adjacent channel limit.

Applicable Mode Spectrum Analyzer

Parameter ACPR upper adjacent channel limit (dB)

Range：-200dB ~200dB.

Set Syntax :ACP:OFFS:ULIM 0

Query Syntax :ACP:OFFS:ULIM?

Default 0

Return Type Value (float) or character

[:SENSe]:ACPower:SPACe <FF0>

(Read-write) Query or set channel spacing of adjacent channel power ratio.

Applicable Mode	Spectrum Analyzer
Parameter	ACPR channel spacing (Hz) Range： 0Hz～ 45MHz.
Set Syntax	:ACP:SPAC 3000000
Query Syntax	:ACP:SPAC?
Default	3MHz
Return Type	Value (double) or character

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

(Read-write) Query or set adjacent channel power ratio ON/OFF**(other measurement functions will be disabled if this function is enabled)**. [:SENSe]:MEASurement command is also available.

Applicable Mode	Spectrum Analyzer
Parameter	ACPR ON/OFF OFF(0) ACPR OFF ON(1) ACPR ON
Set Syntax	:ACP ON
Query Syntax	:ACP?
Default	OFF
Return Type	String or value (int)

[:SENSe]:ACPower:UPPer

(Read only) query ration between the ACPR and upper ACPR.

Applicable Mode	Spectrum Analyzer
Parameter	None
Set Syntax	None
Query Syntax	:ACP:UPP?
Default	None
Return Type	Value (float) or character

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

(Write only) Field strength measurement function, edit antenna factor and add default point

Applicable Mode	Spectrum Analysis, Orientation Analysis
Parameter	None
Set Syntax	:AMPL:CORR:ANT:EDIT:ADD
Query Syntax	None
Default	None
Return Type	None

[:SENSe]:AMPLitude:CORRections:ANTenna:EDIT:ADD:DATA <FF0>,<FF0>

(Write only) Field strength measurement function, edit antenna factor and add points.

Applicable Mode Spectrum Analysis, Orientation Analysis

Parameter Frequency (0~9.1GHz)

Antenna factor value (-200dB ~200dB)

Set Syntax :AMPL:CORR:ANT:EDIT:ADD:DATA 10000000,20

Query Syntax None

Default 1000000000,0

Return Type None

[:SENSe]:AMPLitude:CORRections:ANTenna:EDIT:DATA <I00>,<FF0>,<F00>

(Write only) Field strength measurement function, edit antenna factor and edit point.

Applicable Mode Spectrum Analysis, Orientation Analysis

Parameter Point index (starting from 0)

Frequency value (0-9.1GHz)

Antenna factor value (-200dB~200dB)

Set Syntax :AMPL:CORR:ANT:EDIT:DATA 1,10000000,20

Query Syntax :AMPL:CORR:ANT:EDIT:DATA? 1

Default 0,1000000000,0

Return Type None

[:SENSe]:AMPLitude:CORRections:ANTenna:EDIT:DELete <I00>

(Write only) Field strength measurement function, delete antenna factor and edit point.

Applicable Mode Spectrum Analysis, Orientation Analysis

Parameter Edit point index

Set Syntax :AMPL:CORR:ANT:EDIT:DEL 1

Query Syntax None

Default 1

Return Type None

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

(Write only) turn off antenna loading and set to No-antenna state.

Applicable Mode Spectrum Analysis, Orientation Analysis

Parameter None

Set Syntax :AMPL:CORR:ANT:OFF

Query Syntax None

Default OFF

Return Type None

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

(Read-write) Query or set field strength measurement function ON/OFF**(other measurement functions will be disabled if this function is enabled)**. [:SENSe]:MEASurement command is also available.

Applicable Mode Spectrum Analyzer

Parameter Field strength ON/OFF

OFF(0) Field strength OFF

ON(1) Field strength ON

Set Syntax :AMPL:CORR ON

Query Syntax : AMPL:CORR?

Default OFF

Return Type String or value (int)

[:SENSe]:AMPLitude:SCALe <E>

(Read-write) query or set scale type.

Applicable Mode Spectrum Analyzer

Parameter Scale type

LOG(0) Logarithm

LIN(1) Linear

Set Syntax :AMPL:SCAL LOG

Query Syntax :AMPL:SCAL?

Default LOG

Return Type String or value (int)

[:SENSe]:AMPLitude:UNIT <E>

(Read-write) query or set amplitude unit.

Applicable Mode Spectrum Analysis, Interference Analysis, Orientation Analysis

Parameter Amplitude unit

Spectrum Analyzer mode. The optional amplitude units are DBM(0), DBMV(1), DBUV(2), V(3), W(4), A(5), DBW(6), DBV(7), DBA(8), DBMA(9), DBUA(10).

IA mode. The amplitude unit is dBm(0).

Orientation Analysis Mode The optional amplitude units are DBM(0), DBMV(1), DBUV(2), V(3), W(4), A(5), DBW(6), DBV(7), DBA(8), DBMA(9), DBUA(10).

Set Syntax :AMPL:UNIT DBM

Query Syntax :AMPL:UNIT?

Default DBM

Return Type String or value (int)

[:SENSe]:AVERage:CLEar

(Write only) count current averaging from 0.

Applicable Mode	Spectrum Analysis,Interference Analysis
Parameter	None
Set Syntax	:AVER:CLE
Query Syntax	None
Default	None
Return Type	None

[:SENSe]:AVERage:COUNt <I00>

(Read-write) query or set average count.

Applicable Mode	Spectrum Analysis,Interference Analysis
Parameter	Average count Range: 1～1000.
Set Syntax	:AVER:COUN 16
Query Syntax	:AVER:COUN?
Default	16
Return Type	Value (int) or character

[:SENSe]:AVERage:CURC

(Read only) Query current average count.

Applicable Mode	Spectrum Analysis,Interference Analysis
Parameter	None
Set Syntax	None
Query Syntax	:AVER:CURC?
Default	16
Return Type	Value (int) or character

[:SENSe]:AVERage:STATe <E>

(Read-write) query or set averaging on/off.

Applicable Mode	Spectrum Analysis,Interference Analysis
Parameter	Average ON/OFF OFF(0) means OFF ON(1) means ON
Set Syntax	:AVER:STAT OFF
Query Syntax	:AVER:STAT?
Default	OFF
Return Type	String or value (int)

[:SENSe]:BANDwidth[:RESolution] <FF0>

(Read-write) query or set resolution bandwidth (RBW).

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter	Resolution bandwidth value (Hz) Range： 1Hz～20MHz.
Set Syntax	:BAND 300000
Query Syntax	:BAND?
Default	3MHz
Return Type	Value (double) or character

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

(Read-write) query or set RBW Auto On/Off.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter	Set RBW to be auto On/Off. When set to Auto, RBW will adjust RBW by bandwidth according to the ratio of SPAN/RBW. OFF(0) means manual ON(1) means automatic
Set Syntax	:BAND:AUTO ON
Query Syntax	:BAND:AUTO?
Default	ON
Return Type	String or value (int)

[:SENSe]:BANDwidth[:RESolution]:RATio <I00>

(Read-write) query or set Span/RBW value.

Applicable Mode	Spectrum Analysis, Interference Analysis
Parameter	SPAN/RBW value Range: 1～500.
Set Syntax	:BAND:RAT 100
Query Syntax	:BAND:RAT?
Default	100
Return Type	Value (int) or character

[:SENSe]:BANDwidth:VIDeo <FF0>

(Read-write) query or set video bandwidth (VBW).

Applicable Mode	Spectrum Analysis, Interference Analysis
Parameter	Video bandwidth value (Hz) Range： 1Hz～20MHz.
Set Syntax	:BAND:VID 300000
Query Syntax	:BAND:VID?
Default	3MHz

Return Type Value (double) or character

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

(Read-write) query or set query or set VBW Auto On/Off.

Applicable Mode Spectrum Analysis, Interference Analysis

Parameter Set auto on/off of VBW. When set to Auto, VBW will adjust RBW by resolution bandwidth according to the ratio of RBW/VBW.

OFF(0) means manual

ON(1) means automatic

Set Syntax :BAND:VID:AUTO ON

Query Syntax :BAND:VID:AUTO?

Default ON

Return Type String or value (int)

[:SENSe]:BANDwidth:VIDeo:RATio <I00>

(Read-write) query or set RBW/VBW value.

Applicable Mode Spectrum Analysis, Interference Analysis

Parameter RBW/VBW value

Range: 1～100.

Set Syntax :BAND:VID:RAT 1

Query Syntax :BAND:VID:RAT?

Default 1

Return Type Value (int) or character

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

(Read-write) Query or set video type.

Applicable Mode Spectrum Analysis, Interference Analysis

Parameter Video type

LIN(0) Linear

LOG(1) Logarithm

Set Syntax :BAND:VID:TYPE LOG

Query Syntax :BAND:VID:TYPE?

Default LIN

Return Type String or value (int)

[:SENSe]:CMEasurement:IBW <FF0>

(Read-write) Query or set channel power bandwidth.

Applicable Mode Spectrum Analyzer

Parameter Channel power bandwidth (Hz)

Range： 5Hz～ 9.1GHz.

Set Syntax :CME:IBW 1000000

Query Syntax :CME:IBW?

Default 10MHz

Return Type Value (double) or character

[:SENSe]:CMEasurement:PSDR

(Read only) Query channel power density value under the channel power function measurement**(Valid after channel power is turned on and swept once)**.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax None

Query Syntax :CME:PSDR?

Default None

Return Type Value (float) or character

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

(Read-write) Query or set channel power function measurement ON/OFF**(other measurement functions will be disabled if this function is enabled)** . [:SENSe]:MEASurement command is also available.

Applicable Mode Spectrum Analyzer

Parameter Channel power ON/OFF

OFF(0) Channel power OFF

ON(1) Channel power ON

Set Syntax :CME ON

Query Syntax :CME?

Default OFF

Return Type String or value (int)

[:SENSe]:CMEasurement:TPWR

(Read only) Query channel power value under the channel power function measurement**(Valid after channel power is turned on and swept once)**.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax None

Query Syntax :CME:TPWR?

Default None

Return Type Value (float) or character

[:SENSe]:CNRatio:CBW <FF0>

(Read-write) Query or set CNR carrier bandwidth.

Applicable Mode Spectrum Analyzer

Parameter CNR carrier bandwidth (Hz)

Range： 300Hz～ 20MHz.

Set Syntax :CNR:CBW 3000000

Query Syntax :CNR:CBW?

Default 3000000

Return Type Value (double) or character

[:SENSe]:CNRatio:CNRatio

(Read only) Query carrier-to-noise ratio value under carrier-to-noise ratio measurement**(Valid after CNR is turned on and swept once)**。

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax None

Query Syntax :CNR:CNR?

Default None

Return Type Value (float) or character

[:SENSe]:CNRatio:CNSPace <FF0>

(Read-write) Query or set CNR frequency offset.

Applicable Mode Spectrum Analyzer

Parameter CNR frequency offset (Hz)

Range： 0Hz～ 100MHz.

Set Syntax :CNR:CNSP 3000000

Query Syntax :CNR:CNSP?

Default 3000000

Return Type Value (double) or character

[:SENSe]:CNRatio:NBW <FF0>

(Read-write) Query or set CNR noise bandwidth.

Applicable Mode Spectrum Analyzer

Parameter CNR noise bandwidth (Hz)

Range： 300Hz～ 20MHz.

Set Syntax :CNR:NBW 3000000

Query Syntax :CNR:NBW?

Default 3000000

Return Type Value (double) or character

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

(Read-write) Query or set carrier-to-noise ratio measurement ON/OFF**(other measurement functions will be disabled if this function is enabled)**. [:SENSe]:MEASurement command is also available.

Applicable Mode Spectrum Analyzer

Parameter CNR measurement ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :CNR ON

Query Syntax :CNR?

Default OFF

Return Type String or value (int)

[:SENSe]:DETector:FUNCtion <E>

(Read-write) query or set detection type.

Applicable Mode Spectrum Analysis, Interference Analysis, Orientation Analysis

Parameter Detector type.

Spectrum Analysis, Interference Analysis	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Sample
	NORMal(3)	Normal (Rosenfeld)
	AVERage(4)	Average
	RMS(5)	Rms
Orientation analysis	AVERage(0)	Average
	POSitive(1)	Peak
	SAMPle(2)	Real-time
Real-time spectrum	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Smp
	AVERage(4)	Average

Set Syntax :DET:FUNC NORM

Query Syntax :DET:FUNC?

Default Spectrum analysis, interference analysis NORM

Orientation analysis AVER

Real-time spectrum POS

Return Type String or value (int)

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

(Read-write) Query or set detection auto ON/OFF.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter	Auto detector ON/OFF. When the detector is set to auto, the instrument will automatically select the detector type according to different measurements. OFF(0) means manual ON(1) means automatic
Set Syntax	:DET:FUNC:AUTO OFF
Query Syntax	:DET:FUNC:AUTO?
Default	ON
Return Type	String or value (int)

[:SENSe]:DMODe <E>

(Read-write) Query or set demodulation mode.

Applicable Mode	Orientation Analysis
Parameter	Demodulation mode CW(0) OFF FM(1) Frequency modulation AM(2) Amplitude modulation USB(3) Upper sideband LSB(4) Lower sideband
Set Syntax	:DMOD FM
Query Syntax	:DMOD?
Default	CW
Return Type	String or value (int)

[:SENSe]:DMODe:SPEak:STATe <E>

(Read-write) Query or set audio speaker ON/OFF.

Applicable Mode	Orientation Analysis
Parameter	Audio speaker ON/OFF OFF(0) OFF ON (1) ON
Set Syntax	:DMOD:SPE:STAT ON
Query Syntax	:DMOD:SPE:STAT?
Default	OFF
Return Type	String or value (int)

[:SENSe]:DMODe:VOLume <I00>

(Read-write) Query or set demodulation volume.

Applicable Mode	Orientation Analysis
Parameter	Demodulation volume

Range: 0～ 100.

Set Syntax :DMOD:VOL 80

Query Syntax :DMOD:VOL?

Default 95

Return Type Value (int) or character

[:SENSe]:EMISsion:CBW <FF0>

(Read-write) Query or set spectrum emission mask reference channel bandwidth.

Applicable Mode Spectrum Analyzer

Parameter Reference channel bandwidth (Hz)

Range： 1kHz～ 9.1GHz.

Set Syntax :EMIS:CBW 3000000

Query Syntax :EMIS:CBW?

Default 1000000

Return Type Value (double) or character

[:SENSe]:EMISsion:FAIL

(Read only) Query whether the SEM test passes. The return value 1 means the test fails, and 0 means the test passes.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax None

Query Syntax :EMIS:FAIL?

Default None

Return Type Value (int) or character

[:SENSe]:EMISsion:MARKer <E>

(Read-write) Query or set spurious emission mask peak marker ON/OFF.

Applicable Mode Spectrum Analyzer

Parameter Peak marker ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :EMIS:MARK ON

Query Syntax :EMIS:MARK?

Default OFF

Return Type String or value (int)

[:SENSe]:EMISsion:RTYPe <E>

(Read-write) Query or set spurious emission mask reference power type.

Applicable Mode Spectrum Analyzer

Parameter Reference power type

PEAK(0) peak value

CHANnel(1) channel

Set Syntax :EMIS:RTYP PEAK

Query Syntax :EMIS:RTYP?

Default PEAK

Return Type String or value (int)

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

(Read-write) Query or set spectrum emission ON/OFF**(other measurement functions will be disabled if this function is enabled)**. [:SENSe]:MEASurement command is also available.

Applicable Mode Spectrum Analyzer

Parameter Spectrum emission ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :EMIS OFF

Query Syntax :EMIS?

Default OFF

Return Type String or value (int)

[:SENSe]:FREQuency:CENTer <FF0>

(Read-write) query or set center frequency.

Applicable Mode all modes

Parameter Center frequency value (Hz)

SA	5Hz~9.099999995GHz
IA	5Hz~9.099999995GHz
COMPASS	1MHz~9.1GHz
RTSA	1831Hz~9.09999817GHz
GSM/EDGE	0Hz~9.1GHz
5GNR	0Hz~9.1GHz
LTE	5Hz~9.099999995GHz

Set Syntax :FREQ:CENT 10000

Query Syntax :FREQ:CENT?

Default 4.55GHz (Spectrum analysis, interference analysis, real-time spectrum, GSM/EDGE,5GNR, LTE analysis)

500MHz (COMPASS)

Return Type Value (double) or character

[:SENSe]:FREQuency:CENTer:STEP <FF0>

(Read-write) Query or set step frequency.

Applicable Mode all modes

Parameter Step frequency value (Hz)

Range： 1Hz～ 5GHz.

Set Syntax :FREQ:CENT:STEP 10000

Query Syntax :FREQ:CENT:STEP?

Default 10MHz

Return Type Value (double) or character

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

(Read-write) Query or set step frequency ON/OFF.

Applicable Mode all modes

Parameter Step frequency auto ON/OFF

OFF(0) auto OFF

ON (1) auto ON

Set Syntax :FREQ:CENT:STEP:AUTO OFF

Query Syntax :FREQ:CENT:STEP:AUTO?

Default ON

Return Type String or value (int)

[:SENSe]:FREQuency:SIGNal:TRAC <E>

(Read-write) Query or set signal tracking ON/OFF.

Applicable Mode Spectrum Analyzer

Parameter Tracking ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :FREQ:SIGN:TRAC ON

Query Syntax :FREQ:SIGN:TRAC?

Default OFF

Return Type String or value (int)

[:SENSe]:FREQuency:SIGStandard:CHANnel <I00>

(Read-write) Query or set channel number.

Applicable Mode Spectrum Analysis, Interference Analysis

Parameter Channel number

Set Syntax :FREQ:SIGS:CHAN 1

Query Syntax :FREQ:SIGS:CHAN?

Default 0

Return Type Value (int) or character

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

(Read-write) Query or set signal standard.

Applicable Mode Spectrum Analysis, Interference Analysis

Parameter Signal standard name

Set Syntax :FREQ:SIGS:NAME "GSM"

Query Syntax :FREQ:SIGS:NAME?

Default None

Return Type String

[:SENSe]:FREQuency:SPAN <FF0>

(Read-write) query or set Span.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter Span value (Hz)

SA 0Hz~9.1GHz

IA 0Hz~9.1GHz

RTSA 3.662kHz~120MHz

Set Syntax :FREQ:SPAN 10000

Query Syntax :FREQ:SPAN?

Default 9.1GHz (Spectrum analysis, interference analysis)

120MHz (RTSA)

Return Type Value (double) or character

[:SENSe]:FREQuency:SPAN:FULL

(Write only) Set to full span or maximum span.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter None

Set Syntax :FREQ:SPAN:FULL

Query Syntax None

Default None

Return Type None

[:SENSe]:FREQuency:SPAN:MIN

(Write only) Set minimum span.

Applicable Mode Real-time Spectrum

Parameter None

Set Syntax :FREQ:SPAN:MIN

Query Syntax None

Default None

Return Type None

[:SENSe]:FREQuency:SPAN:PREVious

(Write only) set to the previous Span.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter None

Set Syntax :FREQ:SPAN:PREV

Query Syntax None

Default None

Return Type None

[:SENSe]:FREQuency:SPAN:ZERO

(Write only) set to zero Span.

Applicable Mode Spectrum Analysis, Interference Analysis

Parameter None

Set Syntax :FREQ:SPAN:ZERO

Query Syntax None

Default None

Return Type None

[:SENSe]:FREQuency:STARt <FF0>

(Read-write) query or set start frequency.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter Start frequency value (Hz)

Range： 0Hz～9.1GHz.

Set Syntax :FREQ:STAR 10000

Query Syntax :FREQ:STAR?

Default 0Hz (Spectrum analysis, interference analysis)

4.49GHz (RTSA)

Return Type Value (double) or character

[:SENSe]:FREQuency:STOP <FF0>

(Read-write) query or set stop frequency.

Applicable Mode Spectrum Analysis, Interference Analysis, Real-time Spectrum

Parameter Stop frequency value (Hz)

Range： 0Hz～9.1GHz.

Set Syntax :FREQ:STOP 10000

Query Syntax :FREQ:STOP?

Default 9.1GHz (Spectrum analysis, interference analysis)

4.61GHz (RTSA)

Return Type Value (double) or character

[:SENSe]:GATE:DELay <F00>

(Read-write) Query or set time gate delay.

Applicable Mode Spectrum Analyzer

Parameter Delay time (s)

Range：0s～20s.

Set Syntax :GATE:DEL 10

Query Syntax :GATE:DEL?

Default 1ms

Return Type Value (double) or character

[:SENSe]:GATE:LENGth <F00>

(Read-write) Query or set time gate length.

Applicable Mode Spectrum Analyzer

Parameter Time gate length (ms)

Range：1μs～20s.

Set Syntax :GATE:LENG 10

Query Syntax :GATE:LENG?

Default 5ms

Return Type Value (double) or character

[:SENSe]:GATE:SLOP <E>

(Read-write) Query or set time gate edge slope.

Applicable Mode Spectrum Analyzer

Parameter Edge slope

FALLING(0) negative

RISING(1) positive

Set Syntax :GATE:SLOP FALLING

Query Syntax :GATE:SLOP?

Default RISING

Return Type String or value (int)

[:SENSe]:GATE[:STATe] <E>

(Read-write) Query or set time gate ON/OFF.

Applicable Mode Spectrum Analyzer

Parameter Time gate ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :GATE ON

Query Syntax :GATE?

Default OFF

Return Type String or value (int)

[:SENSe]:GATE:TRIGger <E>

(Read-write) Query or set time gate trigger source.

Applicable Mode Spectrum Analyzer

Parameter Trigger source

EXT(0) External

GPS(1) GPS

Set Syntax :GATE:TRIG GPS

Query Syntax :GATE:TRIG?

Default EXT

Return Type String or value (int)

[:SENSe]:GATE:VIEW[:STATe] <E>

(Read-write) Query or set time gate view ON/OFF.

Applicable Mode Spectrum Analyzer

Parameter View ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :GATE:VIEW ON

Query Syntax :GATE:VIEW?

Default OFF

Return Type String or value (int)

[:SENSe]:GATE:VIEW:TIME <F00>

(Read-write) Query or set time gate view time.

Applicable Mode Spectrum Analyzer

Parameter View time (ms)

Range：3.472μs～6000s.

Set Syntax :GATE:VIEW:TIME 20

Query Syntax :GATE:VIEW:TIME?

Default 10

Return Type Value (double) or character

[:SENSe]:GSM:BAND:CHANnel <I00>

(Read-write) Query or set band channel number.

Applicable Mode GSM/EDGE

Parameter Channel number

Set Syntax :GSM:BAND:CHAN 1

Query Syntax :GSM:BAND:CHAN?

Default None

Return Type Value (int) or character

[:SENSe]:GSM:BAND:NAME <S>

(Read-write) Query or set band name.

Applicable Mode GSM/EDGE

Parameter Band name

Set Syntax :GSM:BAND:NAME "NAME"

Query Syntax :GSM:BAND:NAME?

Default None

Return Type String

[:SENSe]:GSM[:MEASurement] <E>

(Read-write) Query or set measurement mode.

Applicable Mode GSM/EDGE

Parameter Measurement mode

DEMod(0) Constellation diagram

SUMMary(1) Demodulation summary

PFail(2) Test pass

Set Syntax :GSM SUMM

Query Syntax :GSM?

Default DEMod

Return Type String or value (int)

[:SENSe]:GSM[:MEASurement]:DEMod:DATA

(Read only) Acquire demodulation data.

Applicable Mode GSM/EDGE

Parameter None

Set Syntax None

Query Syntax :GSM:DEM:DATA?

Default None

Return Type String or block ,the returned data is:

Modulation mode(1:8PSK 0:GMSK)
BSIC
NCC
BCC
C/I
Frequency error (Hz)
Frequency error (ppm)
Phase error (rms)
Phase error (peak)
EVM(rms)
EVM(peak)
Origin Offset
Amplitude error

[:SENSe]:GSM[:MEASurement]:DEMod:DATA:IQ

(Read only) Acquire demodulated constellation data.

Applicable Mode	GSM/EDGE
Parameter	None
Set Syntax	None
Query Syntax	:GSM:DEM:DATA:IQ?
Default	None
Return Type	String or block

[:SENSe]:GSM[:MEASurement]:DEMod[:VIEW] <E>

(Read-write) Query or set IQ display mode.

Applicable Mode	GSM/EDGE
Parameter	Display mode
	CONStant(0) scalar
	VECTor(1) vector
Set Syntax	:GSM:DEM VECT
Query Syntax	:GSM:DEM?
Default	CONStant
Return Type	String or value (int)

[:SENSe]:GSM[:MEASurement]:PFAil:CONFig <I00>,<F00>,<F00>,<E>

(Read-write) Configure test items.

Applicable Mode	GSM/EDGE
Parameter	Test item index 1~10
	Maximum Different test items have different value ranges
	Minimum Different test items have different Min ranges
	ON/OFF ON/OFF
Set Syntax	:GSM:PFA:CONF 1,0,-10,ON
Query Syntax	:GSM:PFA:CONF? 1
Default	None
Return Type	String or block

[:SENSe]:GSM[:MEASurement]:PFAil:DATA

(Read only) Get pass test data.

Applicable Mode	GSM/EDGE
Parameter	None
Set Syntax	None
Query Syntax	:GSM:PFA:DATA?
Default	None
Return Type	String or block ,the returned data is:

The channel power
C/I
Frequency error(Hz)
Frequency error(ppm)
Phase error(rms)
Phase error(peak)
EVM(rms)
EVM(peak)
Origin Offset
Amplitude error

[:SENSe]:GSM[:MEASurement]:PFAil:LOAD <S>

(Write only) Load test items.

Applicable Mode GSM/EDGE

Parameter None

Set Syntax :GSM:PFA:LOAD

Query Syntax None

Default None

Return Type None

[:SENSe]:GSM[:MEASurement]:PFAil:RESet

(Write only) Reset test item.

Applicable Mode GSM/EDGE

Parameter None

Set Syntax :GSM:PFA:RES

Query Syntax None

Default None

Return Type None

[:SENSe]:GSM[:MEASurement]:PFAil:SAVE <S>

(Write only) Save test item.

Applicable Mode GSM/EDGE

Parameter None

Set Syntax :GSM:PFA:SAVE

Query Syntax None

Default None

Return Type None

[:SENSe]:GSM[:MEASurement]:SUMMary:DATA

(Read only) Get summary data.

Applicable Mode GSM/EDGE

Parameter None

Set Syntax None

Query Syntax :GSM:SUMM:DATA?

Default None

Return Type String or block, the returned data is:

Modulation mode(1:8PSK 0:GMSK)
BSIC
NCC
BCC
C/I
Frequency error (Hz)
Frequency error (ppm)
Phase error (rms)
Phase error (peak)
EVM(rms)
EVM(peak)
Origin Offset
Amplitude error

[:SENSe]:HARMonics:FUNDamental <FF0>

(Read-write) Query or set harmonic distortion fundamental frequency.

Applicable Mode Spectrum Analyzer

Parameter Fundamental frequency (Hz)

Range：0~9.1GHz.

Set Syntax :HARM:FUND 3000000

Query Syntax :HARM:FUND?

Default 1000000000

Return Type Value (double) or character

[:SENSe]:HARMonics:NUM <I00>

(Read-write) Query or set harmonic distortion harmonic number.

Applicable Mode Spectrum Analyzer

Parameter Harmonic number

Range: 1～10.

Set Syntax :HARM:NUM 3

Query Syntax :HARM:NUM?

Default 4

Return Type Value (int) or character

[:SENSe]:HARMonics:RBW <FF0>

(Read-write) Query or set harmonic distortion resolution bandwidth.

Applicable Mode Spectrum Analyzer

Parameter Resolution bandwidth (Hz)

Range： 1Hz～20MHz.

Set Syntax :HARM:RBW 3000000

Query Syntax :HARM:RBW?

Default 3000000

Return Type Value (double) or character

[:SENSe]:HARMonics:STATe <E>

(Read-write) Query or set harmonic distortion status.

Applicable Mode Spectrum Analyzer

Parameter Harmonic distortion ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :HARM:STAT 1

Query Syntax :HARM:STAT?

Default OFF

Return Type String or value (int)

[:SENSe]:HARMonics:VBW <FF0>

(Read-write) Query or set harmonic distortion video bandwidth.

Applicable Mode Spectrum Analyzer

Parameter Video bandwidth (Hz)

Range： 1Hz～20MHz.

Set Syntax :HARM:VBW 300000

Query Syntax :HARM:VBW?

Default 3000000

Return Type Value (double) or character

[:SENSe]:IAMeasure:MODE <E>

(Read-write) Query or set interference analysis measurement mode.

Applicable Mode Interference Analysis

Parameter NORMAL(0) Spectrum measurement

SG(1) Waterfall plot

RSSI(2) RSSI

Set Syntax :IAM:MODE SG

Query Syntax : IAM:MODE?

Default SG

Return Type String or value (int)

[:SENSe]:IAMeasure:TRACe:CURS <I00>

(Write only) Set time cursor.

Applicable Mode Interference Analysis

Parameter Time cursor (int)

Range: 1～ 290.

Set Syntax :IAM:TRAC:CURS 1

Query Syntax None

Default 0

Return Type None

[:SENSe]:IAMeasure:TRACe:INTerval <F00>

(Read-write) Query or set interference analysis sweep interval.

Applicable Mode Interference Analysis

Parameter Sweep time (ms)

Range：0s～1000s.

Set Syntax :IAM:TRAC:INT 1000

Query Syntax :IAM:TRAC:INT?

Default 0ns

Return Type Value (double) or character

[:SENSe]:IAMeasure:TRACe:RESTart

(Write only) Set restart measurement.

Applicable Mode Interference Analysis

Parameter None

Set Syntax :IAM:TRAC:REST

Query Syntax None

Default None

Return Type None

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

(Read-write) Query or set interference analysis automatic stave ON/OFF.

Applicable Mode Interference Analysis

Parameter Automatic stave ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :IAM:TRAC:SAVE ON

Query Syntax :IAM:TRAC:SAVE?

Default OFF

Return Type Value (bool) or character

[:SENSe]:IAMeasure:TRACe:SPAN <I00>

(Read-write) Query or set interference analysis span time.

Applicable Mode Interference Analysis

Parameter Span time (min)

Range：1~2880000 min.

Set Syntax :IAM:TRAC:SPAN 1

Query Syntax :IAM:TRAC:SPAN?

Default 0

Return Type Value (int) or character

[:SENSe]:IF:OUT <E>

(Read-write) Query or set IF output ON/OFF.

Applicable Mode Spectrum Analyzer

Parameter IF output ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :IF:OUT ON

Query Syntax :IF:OUT?

Default OFF

Return Type String or value (int)

[:SENSe]:IF:SELect <E>

(Read-write) Query or set IF selection.

Applicable Mode Spectrum Analyzer

Parameter 3IF(0) Three IF output

4IF(1) Four IF output

Set Syntax :IF:SEL 3IF

Query Syntax :IF:SEL?

Default -3IF

Return Type Value (int) or character

[:SENSe]:IFBWidth <FF0>

(Read-write) Query or set bandwidth.

Applicable Mode Orientation Analysis

Parameter Bandwidth (Hz)

Range： 150Hz～ 150kHz.

Set Syntax :IFBW 1000

Query Syntax :IFBW?

Default 10MHz

Return Type Value (double) or character

[:SENSe]:IQ:CAPTure:MODE <E>

(Read-write) Query or set IQ capture mode.

Applicable Mode Spectrum Analyzer

Parameter	IQ capture mode SING(0) single capture CONT(1) continuous capture
Set Syntax	:IQ:CAPT:MODE CONT
Query Syntax	:IQ:CAPT:MODE?
Default	SING
Return Type	String or value (int)

[:SENSe]:IQ:CAPTure:NAME <S>

(Read-write) Query or set IQ capture storage name.

Applicable Mode	Spectrum Analyzer
Parameter	IQ capture storage name
Set Syntax	:IQ:CAPT:NAME "str"
Query Syntax	:IQ:CAPT:NAME?
Default	IQCapture
Return Type	String

[:SENSe]:IQ:CAPTure:SAMPle <FF0>

(Read-write) Query or set IQ capture sampling rate.

Applicable Mode	Spectrum Analyzer
Parameter	IQ capture sampling rate (Hz) Range： 5.625kHz～184.32MHz.
Set Syntax	:IQ:CAPT:SAMP 5625
Query Syntax	:IQ:CAPT:SAMP?
Default	92.16MHz
Return Type	Value (double) or character

[:SENSe]:IQ:CAPTure:STARt

(Write only) Start IQ capture.

Applicable Mode	Spectrum Analyzer
Parameter	None
Set Syntax	:IQ:CAPT:STARt
Query Syntax	None
Default	None
Return Type	None

[:SENSe]:IQ:CAPTure[:STATe] <E>

(Read-write) Query or set IQ capture ON/OFF**(other measurement functions will be disabled if this function is enabled)**. [:SENSe]:MEASurement command is also available.

Applicable Mode	Spectrum Analyzer
Parameter	IQ capture measurement ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :IQ:CAPT ON

Query Syntax :IQ:CAPT?

Default OFF

Return Type String or value (int)

[:SENSe]:IQ:CAPTure:STOP

(Write only) Stop IQ capture.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax :IQ:CAPT:STOP

Query Syntax None

Default None

Return Type None

[:SENSe]:IQ:CAPTure:TIME <F00>

(Read-write) Query or set IQ capture time.

Applicable Mode Spectrum Analyzer

Parameter IQ capture time (μs)

Range：1μs～98.739ms.

Set Syntax :IQ:CAPT:TIME 10

Query Syntax :IQ:CAPT:TIME?

Default -1μs

Return Type Value (double) or character

[:SENSe]:IQ:CAPTure:TRIGger <E>

(Read-write) Query or set IQ capture trigger type.

Applicable Mode Spectrum Analyzer

Parameter Trigger type

FREE(0) Free trigger

EXTR(1) External trigger

Set Syntax :IQ:CAPT:TRIG EXTR

Query Syntax :IQ:CAPT:TRIG?

Default FREE

Return Type String or value (int)

[:SENSe]:IQ:CAPTure:TRIGger:DELay <F00>

(Read-write) Query or set IQ capture trigger delay.

Applicable Mode Spectrum Analyzer

Parameter	Trigger delay (μs) Range：1μs～500ms.
Set Syntax	:IQ:CAPT:TRIG:DEL 100
Query Syntax	:IQ:CAPT:TRIG:DEL?
Default	1
Return Type	Value (double) or character

[:SENSe]:IQ:CAPTure:TRIGger:SLOPe <E>

(Read-write) Query or set IQ capture trigger polarity.

Applicable Mode	Spectrum Analyzer
Parameter	Trigger polarity POS(0) positive NEG(1) negative
Set Syntax	:IQ:CAPT:TRIG:SLOP POS
Query Syntax	:IQ:CAPT:TRIG:SLOP?
Default	NEG
Return Type	String or value (int)

[:SENSe]:LA:DATA:HORScan:AMPLitude

(Read only) Query horizontal scan amplitude value.

Applicable Mode	Orientation Analysis
Parameter	None
Set Syntax	None
Query Syntax	:LA:DATA:HORS:AMPL?
Default	None
Return Type	Value (float) or character

[:SENSe]:LA:DATA:HORScan:AMPLitude:ALL

(Read only) Query array of horizontal scan amplitude value.

Applicable Mode	Orientation Analysis
Parameter	None
Set Syntax	None
Query Syntax	:LA:DATA:HORS:AMPL:ALL?
Default	None
Return Type	String or block

[:SENSe]:LA:DATA:HORScan:AMPLitude:MAX

(Read only) Query maximum horizontal scan amplitude.

Applicable Mode	Orientation Analysis
Parameter	None

Set Syntax None

Query Syntax :LA:DATA:HORS:AMPL:MAX?

Default None

Return Type Value (float) or character

[:SENSe]:LA:DATA:HORScan:FIELd

(Read only) Query horizontal scan field strength value.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:HORS:FIEL?

Default None

Return Type Value (float) or character

[:SENSe]:LA:DATA:HORScan:FIELd:ALL

(Read only) Query array of horizontal scan field strength values.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:HORS:FIEL:ALL?

Default None

Return Type String or block

[:SENSe]:LA:DATA:HORScan:FIELd:MAX

(Read only) Query maximum horizontal scan field strength value.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:HORS:FIEL:MAX?

Default None

Return Type Value (float) or character

[:SENSe]:LA:DATA:MAP:FIELd

(Read only) Query map positioning field strength value.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:MAP:FIEL?

Default None

Return Type Value (float) or character

[:SENSe]:LA:DATA:POTScan:AMPLitude

(Read only) Query amplitude value.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:POTS:AMPL?

Default None

Return Type Value (float) or character

[:SENSe]:LA:DATA:POTScan:AMPLitude:MAX

(Read only) Query maximum amplitude value.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:POTS:AMPL:MAX?

Default None

Return Type Value (float) or character

[:SENSe]:LA:DATA:POTScan:FIELd

(Read only) Query field strength value.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:POTS:FIEL?

Default None

Return Type Value (float) or character

[:SENSe]:LA:DATA:POTScan:FIELd:MAX

(Read only) Query maximum field strength value.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:DATA:POTS:FIEL:MAX?

Default None

Return Type Value (float) or character

[:SENSe]:LA:FREQuency:POSition:DATA[1]|2-6

(Read only) Query storage position data.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:FREQ:POS:DATA1?

Default None

Return Type String or block

Serial number
Frequency(MHz)
Amplitude
Azimuth
The dimension
Longitude

[:SENSe]:LA:FREQuency:POSition:DELete <I00>

(Write only) Delete edit points.

Applicable Mode Orientation Analysis

Parameter Edit point index

Set Syntax :LA:FREQ:POS:DEL 1

Query Syntax None

Default 1

Return Type None

[:SENSe]:LA:FREQuency:POSition:DELete:ALL

(Write only) Clear edit points.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax :LA:FREQ:POS:DEL:ALL

Query Syntax None

Default None

Return Type None

[:SENSe]:LA:FREQuency:POSition:SAVE

(Write only) Positioning save.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax :LA:FREQ:POS:SAVE

Query Syntax None

Default None

Return Type None

[:SENSe]:LA:FREQuency:POSition:SELect <I00>

(Read-write) Positioning selection.

Applicable Mode Orientation Analysis

Parameter Positioning index

Range: 1～6.

Set Syntax :LA:FREQ:POS:SEL 2

Query Syntax :LA:FREQ:POS:SEL?

Default 1

Return Type Value (int) or character

[:SENSe]:LA:HEADing

(Read only) Query heading.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:HEAD?

Default None

Return Type Value (float) or character

[:SENSe]:LA:HEADing:MAX

(Read only) Query maximum point heading.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax None

Query Syntax :LA:HEAD:MAX?

Default None

Return Type Value (float) or character

[:SENSe]:LA:MAP:MARKer:CURRent

(Write only) Mark current position on the map.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax :LA:MAP:MARK:CURR

Query Syntax None

Default None

Return Type None

[:SENSe]:LA:MAP:MARKer:DELete:ALL

(Write only) Delete all mark positions on the map.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax :LA:MAP:MARK:DEL:ALL

Query Syntax None

Default None

Return Type None

[:SENSe]:LA:MAP:MARKer:DELete:CURRent

(Write only) Delete current mark positions on the map.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax :LA:MAP:MARK:DEL:CURR

Query Syntax None

Default None

Return Type None

[:SENSe]:LA:MAP:POSition:DATA:DELete <I00>

(Write only) Delete map positioning data.

Applicable Mode Orientation Analysis

Parameter Positioning data index

Set Syntax :LA:MAP:POS:DATA:DEL 1

Query Syntax None

Default 1

Return Type None

[:SENSe]:LA:MAP:POSition:DATA:DELete:ALL

(Write only) Delete all map positioning data.

Applicable Mode Orientation Analysis

Parameter None

Set Syntax :LA:MAP:POS:DATA:DEL:ALL

Query Syntax None

Default None

Return Type None

[:SENSe]:LA:MAP:POSition:DATA:LOAD <I00>,<I00>

(Write only) Load map positioning data.

Applicable Mode Orientation Analysis

Parameter Positioning data

Set Syntax :LA:MAP:POS:DATA:LOAD 1

Query Syntax None

Default None

Return Type None

[:SENSe]:LA:MAP:TYPE <E>

(Read-write) Query or set map positioning mode.

Applicable Mode Orientation Analysis

Parameter	Positioning mode REAL(0) Real time LOAD(1) Load
Set Syntax	:LA:MAP:TYPE REAL
Query Syntax	:LA:MAP:TYPE?
Default	REAL
Return Type	String or value (int)

[:SENSe]:LA:MAP:ZOOM <E>

(Write only) Set map zoom mode.

Applicable Mode	Orientation Analysis
Parameter	Zoom mode OUT(0) zoom out IN(1) zoom in
Set Syntax	:LA:MAP:ZOOM IN
Query Syntax	None
Default	None
Return Type	None

[:SENSe]:LA:MARKer:FSTangle:MAX <F00>

(Read-write) Query or set maximum field strength angle.

Applicable Mode	Orientation Analysis
Parameter	Maximum field strength angle Range：0～359.9°.
Set Syntax	:LA:MARK:FST:MAX 90
Query Syntax	:LA:MARK:FST:MAX?
Default	0
Return Type	Value (float) or character

[:SENSe]:LA:MARKer:FSTangle:MAX:AUTO <E>

(Read-write) Query or set maximum field strength angle auto ON/OFF.

Applicable Mode	Orientation Analysis
Parameter	Maximum field strength angle auto ON/OFF OFF(0) OFF ON (1) ON
Set Syntax	:LA:MARK:FST:MAX:AUTO ON
Query Syntax	:LA:MARK:FST:MAX:AUTO?
Default	OFF
Return Type	String or value (int)

[:SENSe]:LA:MARKer:SUBimage:RANG <I00>

(Read-write) Query or set maximum field strength angle range.

Applicable Mode	Orientation Analysis
Parameter	Angle range (even) Range：2.0～180.0°
Set Syntax	:LA:MARK:SUB:RANG 180
Query Syntax	:LA:MARK:SUB:RANG?
Default	10
Return Type	Value (int) or character

[:SENSe]:LA:MEASurement <E>

(Read-write) Query or set measurement mode.

Applicable Mode	Orientation Analysis
Parameter	Measurement mode POTScan(0) Position scan HORScan(1) Horizontal scan MAP(2) Map positioning
Set Syntax	:LA:MEAS MAP
Query Syntax	:LA:MEAS?
Default	POTScan
Return Type	String or value (int)

[:SENSe]:LA:PICTch

(Read only) Query pitch angle.

Applicable Mode	Orientation Analysis
Parameter	None
Set Syntax	None
Query Syntax	:LA:PICT?
Default	None
Return Type	Value (float) or character

[:SENSe]:LA:PICTch:MAX

(Read only) Query maximum point pitch angle.

Applicable Mode	Orientation Analysis
Parameter	None
Set Syntax	None
Query Syntax	:LA:PICT:MAX?
Default	None
Return Type	Value (float) or character

[:SENSe]:LA:ROLL

(Read only) Query roll angle.

Applicable Mode	Orientation Analysis
Parameter	None
Set Syntax	None
Query Syntax	:LA:ROLL?
Default	None
Return Type	Value (float) or character

[:SENSe]:LA:ROLL:MAX

(Read only) Query maximum point roll angle.

Applicable Mode	Orientation Analysis
Parameter	None
Set Syntax	None
Query Syntax	:LA:ROLL:MAX?
Default	None
Return Type	Value (float) or character

[:SENSe]:LTE:BAND:EARFcn <I00>

(Read-write) Query or set EARFCN.

Applicable Mode	LTE Analysis
Parameter	EARFCN
Set Syntax	:LTE:BAND:EARF 1
Query Syntax	:LTE:BAND:EARF?
Default	None
Return Type	Value (int) or character

[:SENSe]:LTE:BAND:NAME <S>

(Read-write) Query or set band.

Applicable Mode	LTE Analysis
Parameter	Band name
Set Syntax	:LTE:BAND:NAME " BAND "
Query Syntax	:LTE:BAND:NAME?
Default	None
Return Type	String

[:SENSe]:LTE:CBW <FF0>

(Read-write) Query or set signal bandwidth.

Applicable Mode	LTE Analysis
Parameter	Signal bandwidth (Hz)

Range： 1.4MHz～20MHz.

Set Syntax :LTE:CBW 1000000

Query Syntax :LTE:CBW?

Default 10MHz

Return Type Value (double) or character

[:SENSe]:LTE:CONFig:CELLid <I00>

(Read-write) Query or set cell ID.

Applicable Mode LTE Analysis

Parameter Cell ID

Range: 0～503.

Set Syntax :LTE:CONF:CELLid 10

Query Syntax :LTE:CONF:CELLid?

Default 0

Return Type Value (int) or character

[:SENSe]:LTE:CONFig:CELLid:AUTO <E>

(Read-write) Automatically Query or set cell ID.

Applicable Mode LTE Analysis

Parameter Cell ID auto ON/OFF

OFF(0) auto OFF

ON (1) auto ON

Set Syntax :LTE:CONF:CELL:AUTO OFF

Query Syntax :LTE:CONF:CELL:AUTO?

Default ON

Return Type String or value (int)

[:SENSe]:LTE:CONFig:CFI <E>

(Read-write) Query or set CFI configuration.

Applicable Mode LTE Analysis

Parameter CFI configuration

AUTO(0) Automatic

MAN1(1) Manual 1

MAN2(2) Manual 2

MAN3(3) Manual 3

MAN4(4) Manual 4

Set Syntax :LTE:CONF:CFI AUTO

Query Syntax :LTE:CONF:CFI?

Default AUTO

Return Type String or value (int)

[:SENSe]:LTE:CONFig:CPRefix <E>

(Read-write) Query or set cyclic prefix.

Applicable Mode	LTE Analysis
Parameter	Cyclic prefix
	NORMal(0) Normal
	EXTend(1) Extend
Set Syntax	:LTE:CONF:CPR NORM
Query Syntax	:LTE:CONF:CPR?
Default	NORMal
Return Type	String or value (int)

[:SENSe]:LTE:CONFig:MODE <E>

(Read-write) Query or set measurement mode.

Applicable Mode	LTE Analysis
Parameter	Measurement mode
	FDD(0) FDD
	TDD(1) TDD
Set Syntax	:LTE:CONF:MODE TDD
Query Syntax	:LTE:CONF:MODE?
Default	FDD
Return Type	String or value (int)

[:SENSe]:LTE:CONFig:PHICh:EXTend <E>

(Read-write) Query or set PHICH mode.

Applicable Mode	LTE Analysis
Parameter	PHICH configuration
	OFF(0) Normal
	ON(1) Extend
Set Syntax	:LTE:CONF:PHIC:EXT OFF
Query Syntax	:LTE:CONF:PHIC:EXT?
Default	OFF
Return Type	String or value (int)

[:SENSe]:LTE:CONFig:PHICh:NG <E>

(Read-write) Query or set PHICH number.

Applicable Mode	LTE Analysis
Parameter	PHICH configuration
	ONESixth(0) 1/6
	HALF(1) 1/2
	ONE(2) 1

	TWO(3) 2
Set Syntax	:LTE:CONF:PHIC:NG TWO
Query Syntax	:LTE:CONF:PHIC:NG?
Default	ONESixth
Return Type	String or value (int)

[:SENSe]:LTE:CONFig:PORT <E>

(Read-write) Query or set number of ports.

Applicable Mode	LTE Analysis
Parameter	Number of ports
	AUTO (0) Auto
	ONE(1) 1
	TWO(2) 2
	FOUR(3) 3
Set Syntax	:LTE:CONF:PORT AUTO
Query Syntax	:LTE:CONF:PORT?
Default	AUTO
Return Type	String or value (int)

[:SENSe]:LTE:CONFig:SSUBframe <I00>

(Read-write) Query or set special subframe.

Applicable Mode	LTE Analysis
wParameter	Special subframe number
	Range: 0～9.
Set Syntax	:LTE:CONF:SSUB 1
Query Syntax	:LTE:CONF:SSUB?
Default	0
Return Type	Value (int) or character

[:SENSe]:LTE:CONFig:SUBFrame <I00>

(Read-write) Query or set subframe.

Applicable Mode	LTE Analysis
Parameter	Subframe number
	Range: 0～9.
Set Syntax	:LTE:CONF:SUBF 0
Query Syntax	:LTE:CONF:SUBF?
Default	0
Return Type	Value (int) or character

[:SENSe]:LTE:CONFig:UDLink <I00>

(Read-write) Query or set uplink/downlink configuration.

Applicable Mode	LTE Analysis
Parameter	Uplink and downlink numbers Range: 0～6.
Set Syntax	:LTE:CONF:UDL 1
Query Syntax	:LTE:CONF:UDL?
Default	0
Return Type	Value (int) or character

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:CBW <FF0>

(Read-write) Query or set carrier aggregation channel bandwidth.

Applicable Mode	LTE Analysis
Parameter	Carrier aggregation channel bandwidth (Hz) Range： 1.4MHz～20MHz.
Set Syntax	:LTE:CA:CARR1:CBW 1000
Query Syntax	:LTE:CA:CARR1:CBW?
Default	10MHz
Return Type	Value (double) or character

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:CELLid <I00>

(Read-write) Query or set carrier aggregation cell ID.

Applicable Mode	LTE Analysis
Parameter	Carrier aggregation cell ID Range: 0～503.
Set Syntax	:LTE:CA:CARR1:CELL 10
Query Syntax	:LTE:CA:CARR1:CELL?
Default	-1
Return Type	Value (int) or character

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:CELLid:AUTO <E>

(Read-write) Query or set carrier aggregation auto cell ID.

Applicable Mode	LTE Analysis
Parameter	Carrier aggregation auto cell ID ON/OFF OFF(0) Manual ON (1) Auto
Set Syntax	:LTE:CA:CARR1:CELL:AUTO OFF
Query Syntax	:LTE:CA:CARR1:CELL:AUTO?
Default	ON

Return Type String or value (int)

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:CENTer <FF0>

(Read-write) Query or set carrier aggregation center frequency.

Applicable Mode LTE Analysis

Parameter Carrier aggregation center frequency (Hz)

Range： 5Hz～9.099999995GHz.

Set Syntax :LTE:CA:CARR1:CENT 10000

Query Syntax :LTE:CA:CARR1:CENT?

Default 10MHz

Return Type Value (double) or character

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:CFI <E>

(Read-write) Query or set carrier aggregation CFI configuration.

Applicable Mode LTE Analysis

Parameter Carrier aggregation CFI configuration

AUTO (0) Auto

MAN1(1) Manual 1

MAN2(2) Manual 2

MAN3(3) Manual 3

MAN4(4) Manual 4

Set Syntax :LTE:CA:CARR1:CFI MAN1

Query Syntax :LTE:CA:CARR1:CFI?

Default AUTO

Return Type String or value (int)

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:CPRefix <E>

(Read-write) Query or set carrier aggregation cyclic prefix.

Applicable Mode LTE Analysis

Parameter Carrier aggregation cyclic prefix

NORMal(0) Normal

EXTend(1) Extend

Set Syntax :LTE:CA:CARR1:CPR EXT

Query Syntax :LTE:CA:CARR1:CPR?

Default NORMal

Return Type String or value (int)

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:PHICh:EXTend <E>

(Read-write) Query or set carrier aggregation PHICH mode.

Applicable Mode LTE Analysis

Parameter	PHICH configuration
	OFF(0) Normal
	ON(1) Extend
Set Syntax	:LTE:CA:CARR1:PHIC:EXT ON
Query Syntax	:LTE:CA:CARR1:PHIC:EXT?
Default	OFF
Return Type	String or value (int)

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5:PHICh:NG <E>

(Read-write) Query or set carrier aggregation PHICH number.

Applicable Mode	LTE Analysis
Parameter	PHICH configuration
	ONESixth(0) 1/6
	HALF(1) 1/2
	ONE(2) 1
	TWO(3) 2
Set Syntax	:LTE:CA:CARR1:PHIC:NG HALF
Query Syntax	:LTE:CA:CARR1:PHIC:NG?
Default	ONESixth
Return Type	String or value (int)

[:SENSe]:LTE[:MEAS]:CA:CARRiear[1]|2-5[:STATe] <E>

(Read-write) Query or set carrier aggregation ON/OFF.

Applicable Mode	LTE Analysis
Parameter	Carrier aggregation ON/OFF
	OFF(0) OFF
	ON (1) ON
Set Syntax	:LTE:CA:CARR1 ON
Query Syntax	:LTE:CA:CARR1?
Default	ON
Return Type	String or value (int)

[:SENSe]:LTE[:MEAS]:CC:CHANnel <E>

(Read-write)	Query or set control channel.
Applicable Mode	LTE Analysis
Parameter	Control channel number
	Options are PSS(0), SSS(1), PBCH(2), PCFICH(3), PHICH(4), PDCCH(5) and RS(6).
Set Syntax	:LTE:CC:CHAN PSS
Query Syntax	:LTE:CC:CHAN?

Default PCFICH

Return Type String or value (int)

[:SENSe]:LTE[:MEAS]:CC:CONS <E>

(Read-write) Query or set control channel constellation diagram.

Applicable Mode LTE Analysis

Parameter Control channel constellation diagram ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :LTE:CC:CONS OFF

Query Syntax :LTE:CC:CONS?

Default OFF

Return Type String or value (int)

[:SENSe]:LTE[:MEAS]:CC:RS:CHANnel <E>

(Read-write) Query or set control channel RS channel.

Applicable Mode LTE Analysis

Parameter Control channel RS channel number

Options are RS0(0), RS1(1), RS2(2), RS3(3) and RSALL(4).

Set Syntax :LTE:CC:RS:CHAN RS1

Query Syntax :LTE:CC:RS:CHAN?

Default -RS0

Return Type String or value (int)

[:SENSe]:LTE[:MEAS]:DAM:LIMit:DOWN <F00>

(Read-write) Query or set data allocation map color down limit

Applicable Mode LTE Analysis

Parameter Data Allocation Map color down limit

Range： -210～30.

Set Syntax :LTE:DAM:LIM:DOWN-10

Query Syntax :LTE:DAM:LIM:DOWN?

Default -100

Return Type String or value (float)

[:SENSe]:LTE[:MEAS]:DAM:LIMit:UP <F00>

(Read-write) Query or set data allocation map color up limit

Applicable Mode LTE Analysis

Parameter Data Allocation Map color up limit

Range： -209～30.

Set Syntax :LTE:DAM:LIM:UP-10

Query Syntax :LTE:DAM:LIM:UP?

Default -100

Return Type String or value (float)

[:SENSe]:LTE[:MEAS]:DATA:CA

(Read-write) Query carrier aggregation demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:CA?

Default None

Return Type String or value,returns the demodulation data of the five carriers sequentially, a total of 5*12 data,the data content in each carrier is:

Center Frequency
Synchronization symbol
Pss Frequency
Sss Frequency
Pbch Frequency
Rs Frequency
Pss EVM
Sss EVM
Pbch EVM
Rs EVM
Cell ID
Frequency error

[:SENSe]:LTE[:MEAS]:DATA:CC

(Read-write) Query control channel demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:CC?

Default None

Return Type String or value,returns EVM and power data for each channel,if the constellation diagram is displayed,constellation data is also returned,content is:

Synchronization symbol
Cell ID
Frequency error
Time error
EVM(rms)、EVM(peak) and power of each channel in turn. The order of channels is：pss, sss, pbch, pcfich, phich, pdcch, rs, rs0, rs1, rs2, rs3.
Subframe Power

IQ Offset
OFDM Symbol Power
Constellation Data.The IQ data is in order,and the total number is related to the channel displayed by the constellation.It can be calculated based on the amount of remaining data.A set of IQ data is 8 bytes 64 bits.

[:SENSe]:LTE[:MEAS]:DATA:CONS

(Read-write) Query Constellation demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:CONS?

Default None

Return Type String or value,content is:

Synchronization symbol
Frequency error
Time error
Qpsk EVM(rms)
qam16 EVM(rms)
qam16 EVM(rms)
Pdsch EVM(rms)
Pdsch EVM(peak)
Pdsch power
RS power
Constellation Data. The IQ data is in order. The total number can be calculated based on the amount of remaining data.A set of IQ data is 8 bytes 64 bits.

[:SENSe]:LTE[:MEAS]:DATA:DAM

(Read-write) Query Data Allocacation Map demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:DAM?

Default None

Return Type String or value,content is:

Synchronization symbol
OFDM Symbol Power
Data Utilization
Pdsch power
Rs power
RB power value of subframes 0-9.

The number of RB per subframe is transformed according to the channel bandwidth. The total quantity can be calculated according to the amount of remaining data. A RB power value is 4 bytes 32 bits.

[:SENSe]:LTE[:MEAS]:DATA:DC

(Read-write) Query Data Channel demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:DC?

Default None

Return Type String or value,content is:

Synchronization symbol
Cell ID
Frequency error
Time error
Pdsch EVM(rms)
Pdsch EVM(peak)
Currently rb power
IQ Offset
All RB power values of the current subframe. Quantity is transformed according to the channel bandwidth. The total quantity can be calculated according to the amount of remaining data. A RB power value is 4 bytes 32 bits.

[:SENSe]:LTE[:MEAS]:DATA:IDS

(Read-write) Query ID Scanner demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:IDS?

Default None

Return Type String or value,returns six sets of data,content is:

Cell ID
rsrp
rsrq
sinr
rssi

Pss power
Sss power

[:SENSe]:LTE[:MEAS]:DATA:PVT

(Read-write) Query Power vs Time demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:PVT?

Default None

Return Type String or value,content is:

Synchronization symbol
Cell ID
Frequency error
Time error
Subframe power
First Slot power
Second Slot power
Frame Avg Power
IQ Offset
IQ Imbalance
1200 Frame Power

[:SENSe]:LTE[:MEAS]:DATA:TA

(Read-write) Query Time Alignment demodulation data

Applicable Mode LTE Analysis

Parameter None

Set Syntax None

Query Syntax :LTE:DATA:TA?

Default None

Return Type String or value,content is:

Synchronization symbol
Cell ID
Frequency error
Time error
rs0 power
rs0 EVM(rms)
rs0 time deviation
rs1 power
rs1 EVM(rms)
rs1 time deviation
rs2 power
rs2 EVM(rms)
rs2 time deviation
rs3 power
rs3 EVM(rms)
rs3 time deviation

[:SENSe]:LTE[:MEAS]:DC:MARKer <E>

(Read-write) Query or set data channel mark ON/OFF.

Applicable Mode	LTE Analysis
Parameter	Data channel mark ON/OFF
	OFF(0) OFF
	ON (1) ON
Set Syntax	:LTE:DC:MARK OFF
Query Syntax	:LTE:DC:MARK?
Default	OFF
Return Type	String or value (int)

[:SENSe]:LTE[:MEAS]:DC:RB <I00>

(Read-write) Query or set data channel RB number.

Applicable Mode	LTE Analysis
Parameter	RB number
	Range: 0～ 99.
Set Syntax	:LTE:DC:RB 10
Query Syntax	:LTE:DC:RB?
Default	0
Return Type	Value (int) or character

[:SENSe]:LTE[:MEAS]:MODE <E>

(Read-write) Query or set measurement mode.

Applicable Mode	LTE Analysis
Parameter	Measurement mode
	CONS(0) Constellation diagram
	DC(1) Data channel
	CC(2) Control channel
	TA(3) Time alignment error
	CA(4) Carrier aggregation
	IDS(5) ID sweep
	PVT(6) Power VS Time
	DAM(7) Data distribution chart
Set Syntax	:LTE:MODE DC
Query Syntax	:LTE:MODE?
Default	CONS
Return Type	String or value (int)

[:SENSe]:MACPr:ADJ:BW[1]|2-3 <FF0>

(Read-write) Query or set multi-carrier adjacent channel power and adjacent channel bandwidth.

Applicable Mode	Spectrum Analyzer
Parameter	Adjacent channel carrier bandwidth (Hz)
	Range： 1kHz～200MHz.
Set Syntax	:MACP:ADJ:BW 300000
Query Syntax	:MACP:ADJ:BW?
Default	3MHz
Return Type	Value (double) or character

[:SENSe]:MACPr:ADJ:CH <I00>

(Read-write) Query or set multi-carrier adjacent channel power adjacent channel.

Applicable Mode	Spectrum Analyzer
Parameter	Adjacent channel number
	Range: 1～3.
Set Syntax	:MACP:ADJ:CH 1
Query Syntax	:MACP:ADJ:CH?
Default	1
Return Type	Value (int) or character

[:SENSe]:MACPr:ADJ:LIMit:LOWer[1]|2-3 <F00>

(Read-write) Query or set lower adjacent channel limit of multi-carrier adjacent channel power.

Applicable Mode	Spectrum Analyzer
Parameter	Lower adjacent channel limit (dB)
	Range：-200dB~200dB.
Set Syntax	:MACP:ADJ:LIM:LOW 10
Query Syntax	:MACP:ADJ:LIM:LOW?
Default	-200
Return Type	Value (float) or character

[:SENSe]:MACPr:ADJ:LIMit:STATe <E>

(Read-write) Query or set multi-carrier adjacent channel power - adjacent channel limit test state.

Applicable Mode	Spectrum Analyzer
Parameter	Limit test ON/OFF
	OFF(0) OFF
	ON (1) ON
Set Syntax	:MACP:ADJ:LIM:STAT ON

Query Syntax :MACP:ADJ:LIM:STAT?

Default OFF

Return Type String or value (int)

[:SENSe]:MACPr:ADJ:LIMit:UPPer [1]|2-3<F00>

(Read-write) Query or set upper adjacent channel limit of multi-carrier adjacent channel power.

Applicable Mode Spectrum Analyzer

Parameter Upper adjacent channel limit (dB)

Range：-200dB~200dB.

Set Syntax :MACP:ADJ:LIM:UPP 10

Query Syntax :MACP:ADJ:LIM:UPP?

Default -200

Return Type Value (float) or character

[:SENSe]:MACPr:ADJ:OFFSet [1]|2-3<FF0>

(Read-write) Query or set multi-carrier adjacent channel power - adjacent channel offset.

Applicable Mode Spectrum Analyzer

Parameter Adjacent channel offset (Hz)

Range： 0Hz～200MHz.

Set Syntax :MACP:ADJ:OFFS 300000

Query Syntax :MACP:ADJ:OFFS?

Default 0

Return Type Value (double) or character

[:SENSe]:MACPr:CARRier:BW[1]|2 <FF0>

(Read-write) Query or set carrier bandwidth.

Applicable Mode Spectrum Analyzer

Parameter Carrier bandwidth (Hz)

Range： 1kHz～200MHz.

Set Syntax :MACP:CARR:BW 100000

Query Syntax :MACP:CARR:BW?

Default 3MHz

Return Type Value (double) or character

[:SENSe]:MACPr:CARRier:FREQ[1]|2 <FF0>

(Read-write) Query or set carrier frequency.

Applicable Mode Spectrum Analyzer

Parameter Carrier frequency

Range： 100MHz～9.1GHz.

Set Syntax :MACP:CARR:FREQ 300000

Query Syntax :MACP:CARR:FREQ?

Default 4.55GHz

Return Type Value (double) or character

[:SENSe]:MACPr:CARRier:ID <I00>

(Read-write) Query or set multi-carrier adjacent channel power carrier ID.

Applicable Mode Spectrum Analyzer

Parameter Carrier ID

Optional 1 or 2.

Set Syntax :MACP:CARR:ID 1

Query Syntax :MACP:CARR:ID?

Default 1

Return Type Value (int) or character

[:SENSe]:MACPr:STATe <E>

(Read-write) Query or set multi-carrier adjacent channel power state.

Applicable Mode Spectrum Analyzer

Parameter Multi-carrier adjacent channel power ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :MACP:STAT 1

Query Syntax :MACP:STAT?

Default OFF

Return Type String or value (int)

[:SENSe]:MAP:ACPR:ADJBw <FF0>

(Read-write) Query or set adjacent channel bandwidth.

Applicable Mode Spectrum Analyzer

Parameter Adjacent channel bandwidth (Hz)

Range： 300Hz～20MHz.

Set Syntax :MAP:ACPR:ADJB 3000000

Query Syntax :MAP:ACPR:ADJB?

Default 3MHz

Return Type Value (double) or character

[:SENSe]:MAP:ACPR:GOOD <F00>

(Read-write) Query or set lower limit of adjacent channel power ratio "good".

Applicable Mode Spectrum Analyzer

Parameter Adjacent channel power ratio

Range：-200dBm~200dBm.

Set Syntax :MAP:ACPR:GOOD 0.0

Query Syntax :MAP:ACPR:GOOD?

Default -40.0 dB

Return Type Value (float) or character

[:SENSe]:MAP:ACPR:MAINbw <FF0>

(Read-write) Query or set main channel bandwidth.

Applicable Mode Spectrum Analyzer

Parameter Main channel bandwidth (Hz)

Range： 300Hz～20MHz.

Set Syntax :MAP:ACPR:MAIN 3000000

Query Syntax :MAP:ACPR:MAIN?

Default 3MHz

Return Type Value (double) or character

[:SENSe]:MAP:ACPR:OFFSet <F00>

(Read-write) Query or set adjacent channel limit.

Applicable Mode Spectrum Analyzer

Parameter Adjacent channel limit

Range：-200dB~200dB.

Set Syntax :MAP:ACPR:OFFS 10

Query Syntax :MAP:ACPR:OFFS?

Default 0.0 dB

Return Type Value (float) or character

[:SENSe]:MAP:ACPR:POOR <F00>

(Read-write) Query or set upper limit of adjacent channel power ratio “poor”.

Applicable Mode Spectrum Analyzer

Parameter Adjacent channel power ratio (dBm)

Range：-200dBm~200dBm.

Set Syntax :MAP:ACPR:POOR 0.0

Query Syntax :MAP:ACPR:POOR?

Default -40.0 dB

Return Type Value (float) or character

[:SENSe]:MAP:ACPR:SPACing <FF0>

(Read-write) Query or set channel spacing.

Applicable Mode Spectrum Analyzer

Parameter Channel spacing (Hz)

Range： 0Hz～45MHz.

Set Syntax :MAP:ACPR:SPAC 3000000

Query Syntax :MAP:ACPR:SPAC?

Default 3MHz

Return Type Value (double) or character

[:SENSe]:MAP:COLLect <E>

(Read-write) Query or set collection state.

Applicable Mode Spectrum Analyzer

Parameter Collection state

START(1) Start collection

STOP(0) Stop collection

Set Syntax :MAP:COLLect START

Query Syntax :MAP:COLLect?

Default STOP

Return Type String or value (int)

[:SENSe]:MAP:DELet

(Write only) Delete mark.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax :MAP:DEL

Query Syntax None

Default None

Return Type None

[:SENSe]:MAP:DIST <I00>

(Read-write) Query or set repetition distance.

Applicable Mode Spectrum Analyzer

Parameter Repeat distance (m)

Range：50~1000m.

Set Syntax :MAP:DIST 100

Query Syntax :MAP:DIST?

Default 100

Return Type Value (int) or character

[:SENSe]:MAP:MEAStype <E>

(Read-write) Query or set interference map test type.

Applicable Mode Spectrum Analyzer

Parameter Test type

RSSI(0) RSSI

ACPR(1) Adjacent channel power ratio

Set Syntax :MAP:MEAS 0

Query Syntax :MAP:MEAS?

Default RSSI

Return Type String or value (int)

[:SENSe]:MAP:REPeat <E>

(Read-write) Query or set repeat type.

Applicable Mode Spectrum Analyzer

Parameter Repeat type

TIME(0) Time

DIST(1) Distance

Set Syntax :MAP:REP 0

Query Syntax :MAP:REP?

Default TIME

Return Type String or value (int)

[:SENSe]:MAP:RSSI:EXCellent <F00>

(Read-write) Query or set lower limit value of RSSI "excellent".

Applicable Mode Spectrum Analyzer

Parameter RSSI value (dBm)

Range：-196~200dBm.

Set Syntax :MAP:RSSI:EXC 0.0

Query Syntax :MAP:RSSI:EXC?

Default 0.0

Return Type Value (float) or character

[:SENSe]:MAP:RSSI:FAIR <F00>

(Read-write) Query or set lower limit value of RSSI “fair”.

Applicable Mode Spectrum Analyzer

Parameter RSSI value (dBm)

Range：-199~197dBm.

Set Syntax :MAP:RSSI:FAIR -50

Query Syntax :MAP:RSSI:FAIR?

Default -60.0

Return Type Value (float) or character

[:SENSe]:MAP:RSSI:GOOD <F00>

(Read-write) Query or set lower limit value of RSSI "good".

Applicable Mode Spectrum Analyzer

Parameter RSSI value

Range：-198~198dBm.

Set Syntax :MAP:RSSI:GOOD -40

Query Syntax :MAP:RSSI:GOOD?

Default -40.0

Return Type Value (float) or character

[:SENSe]:MAP:RSSI:POOR <F00>

(Read-write) Query or set upper limit value of RSSI "poor".

Applicable Mode Spectrum Analyzer

Parameter RSSI value (dBm)

Range：-200~196dBm.

Set Syntax :MAP:RSSI:POOR -80

Query Syntax :MAP:RSSI:POOR?

Default -60.0

Return Type Value (float) or character

[:SENSe]:MAP:RSSI:VERYgood <F00>

(Read-write) Query or set lower limit value of RSSI "very good".

Applicable Mode Spectrum Analyzer

Parameter RSSI value (dBm)

Range：-197 ~199dBm.

Set Syntax :MAP:RSSI:VERY -20

Query Syntax :MAP:RSSI:VERY?

Default -20.0

Return Type Value (float) or character

[:SENSe]:MAP:STATe <E>

(Read-write) Query or set interference map.

Applicable Mode Spectrum Analyzer

Parameter Map ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :MAP:STAT 1

Query Syntax :MAP:STAT?

Default OFF

Return Type String or value (int)

[:SENSe]:MAP:TIME <F00>

(Read-write) Query or set repetition time.

Applicable Mode Spectrum Analyzer

Parameter Repeat time (s)

Range：5~600s.

Set Syntax :MAP:TIME 10

Query Syntax :MAP:TIME?

Default 10.0

Return Type Value (double) or character

[:SENSe]:MEASurement <E>

(Read-write) Query or set function measurement type, or set it directly through the function measurement ON/OFF. Only one function measurement can exist at a time.

Applicable Mode Spectrum Analyzer

Parameter Function measurement type

Set Parameter	Measurement type
NONE(0)	Close Measurement
FST(1)	Field Strength Meter
CHP(2)	Ch Power
OBW(3)	OBW
ACPR(4)	Adjacent channel power
DEMOD(5)	Audio demodulation
SEM(6)	Spurious emission mask
CNR(7)	Carrier-to-noise ratio measurement
IQCAP(8)	IQ capture
OUTMAP(9)	Outdoor map
INMAP(10)	Indoor map
SE(11)	Spurious emission mask
HARM(12)	Harmonic distortion
MACPR(13)	Multi-carrier adjacent channel power

Set Syntax :MEAS NONE

Query Syntax :MEAS?

Default NONE

Return Type Value (int) or character

[:SENSe]:MEASurement:AOFF

(Write only) Turn off function measurement and switch to normal spectrum measurement.

Applicable Mode	Spectrum Analyzer
Parameter	None
Set Syntax	:MEAS:AOFF
Query Syntax	None
Default	None
Return Type	None

[:SENSe]:MEASurement:DATA

(Read only) Query measurement results.

Applicable Mode	Spectrum Analyzer
Parameter	None
Set Syntax	None
Query Syntax	:MEAS:DATA?
Default	None
Return Type	String

[:SENSe]:NR5G[:MEAS] <E>

(Read-write) Query or set measurement mode.

Applicable Mode	5GNR
Parameter	Measurement mode
	BEAM(0) Beam measurement
	MPCI(1) Multi-PCI measurement
	OUTdoor(2) Path map outdoor
	Indoor(3) Path map indoor
Set Syntax	:NR5G MPCI
Query Syntax	:NR5G? 1
Default	BEAM
Return Type	String or value (int)

[:SENSe]:NR5G[:MEAS]:BAND:ARFCn <I00>

(Read-write) Query or set frequency raster.

Applicable Mode	5GNR
Parameter	Frequency raster
Set Syntax	:NR5G:BAND:ARFC 200000
Query Syntax	:NR5G:BAND:ARFC?
Default	None
Return Type	Value (int) or character

[:SENSe]:NR5G[:MEAS]:BAND:GSCN <I00>

(Read-write) Query or set synchronization raster.

Applicable Mode	5GNR
Parameter	Synchronization raster
Set Syntax	:NR5G:BAND:GSCN 100
Query Syntax	:NR5G:BAND:GSCN?
Default	None
Return Type	Value (int) or character

[:SENSe]:NR5G[:MEAS]:BAND:NAME <S>

(Read-write) Query or set band name.

Applicable Mode	5GNR
Parameter	Band name
Set Syntax	:NR5G:BAND:NAME "n1 UL"
Query Syntax	:NR5G:BAND:NAME?
Default	None
Return Type	String

[:SENSe]:NR5G[:MEAS]:BAND:SEARch

(Write only) Set automatic SSB search.

Applicable Mode	5GNR
Parameter	None
Set Syntax	:NR5G:BAND:SEAR
Query Syntax	None
Default	None
Return Type	None

[:SENSe]:NR5G[:MEAS]:BEAM <I00>

(Read-write) Query or set beam measurement beam number.

Applicable Mode	5GNR
Parameter	Beam index Range: 0～7.
Set Syntax	:NR5G:BEAM 0
Query Syntax	:NR5G:BEAM?
Default	0
Return Type	Value (int) or character

[:SENSe]:NR5G[:MEAS]:BEAM:VIEW <E>

(Read-write) Query or set beam measurement beam display mode.

Applicable Mode	5GNR

Parameter beam display mode
SING(0) Single beam
MULT(1) Multi beam
Set Syntax :NR5G:BEAM:VIEW SING
Query Syntax :NR5G:BEAM:VIEW?
Default SING
Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:CELLid <I00>

(Read-write) Query or set beam measurement cell ID.

Applicable Mode 5GNR
Parameter Cell ID
Range: 0～1007.
Set Syntax :NR5G:CELLid 1
Query Syntax :NR5G:CELLid?
Default 0
Return Type Value (int) or character

[:SENSe]:NR5G[:MEAS]:CELLid:AUTO <E>

(Read-write) Query or set beam measurement cell ID auto ON/OFF.

Applicable Mode 5GNR
Parameter Cell ID auto ON/OFF
OFF(0) Manual
ON (1) Auto
Set Syntax :NR5G:CELL:AUTO ON
Query Syntax :NR5G:CELL:AUTO?
Default ON
Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:CONS:CHANnel <E>

(Read-write) Query or set beam measurement constellation channel.

Applicable Mode 5GNR
Parameter Channel number
Optional PSS(0), SSS(1), PBCH(2) and DMRS(3).
Set Syntax :NR5G:CONS:CHAN PBCH
Query Syntax :NR5G:CONS:CHAN?
Default PSS
Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:CONS[:STATe] <E>

(Read-write) Query or set beam measurement constellation diagram ON/OFF.

Applicable Mode 5GNR

Parameter Constellation diagram ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :NR5G:CONS OFF

Query Syntax :NR5G:CONS?

Default OFF

Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:DATA:BEAM

(Read only) Query beam measurement results.

Applicable Mode 5GNR

Parameter None

Set Syntax None

Query Syntax :NR5G:DATA:BEAM?

Default None

Return Type String or block

The contents are as follows,return first

Reference source(0:inside 1:external 2:GPS)
Cell ID

Return the information of 8 Beams after,each beam's information is as follows:

Synchronization symbol
Demodulation symbol
Frequency error
Time error
RSRP
RSRQ
SINR
PSS power
SSS power
PBCH power
DMRS power
PSS EVM(peak)
SSS EVM(peak)
PBCH EVM(peak)
DMRS EVM(peak)
PSS EVM(rms)
SSS EVM(rms)
PBCH EVM(rms)
DMRS EVM(rms)

[:SENSe]:NR5G[:MEAS]:DATA:IQ

(Read only) Query constellation diagram IQ.

Applicable Mode 5GNR

Parameter None

Set Syntax None

Query Syntax :NR5G:DATA:IQ?

Default None

Return Type String or block,return IQ data , l , q order.The total quantity varies according to the channel. PSS channel:127、SSS channel:127、PBCH channel:432、DMRS channel:64.

[:SENSe]:NR5G[:MEAS]:DATA:MIB

(Read only) Query the radio channel MIB

Applicable Mode 5GNR

Parameter None

Set Syntax None

Query Syntax :NR5G:DATA:MIB?

Default None

Return Type String or block,return the following:

Cell ID
Beam
System frame number
Subcarrier spacing(0:15kHz 1:30kHz)
Subcarrier offset
DMRS location(0:location 2 1:location 3)
Control Channel SIB1
CORESET
Search space
Cell ban(0:prohibited 1:non-prohibited)
Same frequency access (0:allow 1:not allow)
Free

[:SENSe]:NR5G[:MEAS]:DATA:MPCI

(Read only) Query MPCI results.

Applicable Mode 5GNR

Parameter None

Set Syntax None

Query Syntax :NR5G:DATA:MPCI?

Default None

Return Type String or block

First return six group of pci measurement information,

the information of each group is as follows:

Valid symbol
Cell ID

Then return the information of eight beams,

the information of each beam is as follows:

Demodulation symbol
Beam
RSRP
RAPQ
SINR
DMRS Channel EVM(rms)
Time error

[:SENSe]:NR5G[:MEAS]:DATA:PVT

(Read only) Query Power vs Time measure results.

Applicable Mode 5GNR

Parameter None

Set Syntax None

Query Syntax :NR5G:DATA:PVT?

Default None

Return Type String or block,the content are as follows:

Reference source (0:inside 1:external 2:GPS)
Cell ID
Synchronization symbol
Demodulation symbol
Frequency error
Time error
Subframe Power
Frame Avg Power
Power value(800 total)

[:SENSe]:NR5G[:MEAS]:MAP:ADELete

(Write only) Delete all marks on the indoor map.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:ADEL

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:COLLection <E>

(Read-write) Query or set outdoor map collection ON/OFF.

Applicable Mode 5GNR

Parameter Collection ON/OFF

OFF(0) Stop collection

ON(1) Start collection

Set Syntax :NR5G:MAP:COLL ON

Query Syntax :NR5G:MAP:COLL?

Default OFF

Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:MAP:DELete

(Write only) Delete indoor map mark.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:DEL

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:LIMit:LOW <I00>

(Read-write) Query or set lower limit of outdoor map.

Applicable Mode 5GNR

Parameter Lower limit (dBm)

Range：174~49dBm.

Set Syntax :NR5G:MAP:LIM:LOW 0

Query Syntax :NR5G:MAP:LIM:LOW?

Default -100dBm

Return Type Value (int) or character

[:SENSe]:NR5G[:MEAS]:MAP:LIMit:UPP<I00>

(Read-write) Query or set upper limit of outdoor map.

Applicable Mode 5GNR

Parameter Upper limit (dBm)

Range：173~50dBm.

Set Syntax :NR5G:MAP:LIM:UPP 0

Query Syntax :NR5G:MAP:LIM:UPP?

Default -0dBm

Return Type Value (int) or character

[:SENSe]:NR5G[:MEAS]:MAP:LOAD <S>

(Write only) Indoor map data loading.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G[:MEAS]:MAP:LOAD " MAP "

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:MARK

(Write only) Set indoor map marks.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:MARK

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:MEAS <E>

(Read-write) Set indoor map measurement method.

Applicable Mode 5GNR

Parameter Measurement method

MPCI(0) PCI

BEAM(1) Beam

Set Syntax :NR5G:MAP:MEAS MPCI

Query Syntax :NR5G:MAP:MEAS?

Default BEAM

Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:MAP:MOVE:DOWN

(Write only) Move indoor map down.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:MOVE:DOWN

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:MOVE:LEFT

(Write only) Move indoor map to the left.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:MOVE:LEFT

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:MOVE:RIGHt

(Write only) Move indoor map to the right.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:MOVE:RIGH

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:MOVE:UP

(Write only) Move indoor map up.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:MOVE:UP

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:REPeat <E>

(Read-write) Query or set outdoor map repetition mode.

Applicable Mode 5GNR

Parameter Repeat mode

TIME(0) Time

DISTance(1) Distance

Set Syntax :NR5G:MAP:REP 0

Query Syntax :NR5G:MAP:REP?

Default TIME

Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:MAP:REPeat:DISTance <I00>

(Read-write) Query or set outdoor map repetition distance.

Applicable Mode 5GNR

Parameter Repeat distance (m)

Range：50~1000m.

Set Syntax :NR5G:MAP:REP:DIST 50

Query Syntax :NR5G:MAP:REP:DIST?

Default 100

Return Type Value (int) or character

[:SENSe]:NR5G[:MEAS]:MAP:REPeat:TIME <I00>

(Read-write) Query or set outdoor map repetition time.

Applicable Mode 5GNR

Parameter Repeat time (μs)

Range：5s～600s.

Set Syntax :NR5G:MAP:REP:TIME 20

Query Syntax :NR5G:MAP:REP:TIME?

Default 10

Return Type Value (int) or character

[:SENSe]:NR5G[:MEAS]:MAP:SAVE:BMP <S>

(Write only) Save map picture.

Applicable Mode 5GNR

Parameter File name

Set Syntax :NR5G:MAP:SAVE:BMP " Name "

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:SAVE:CSV <S>

(Write only) Save map CSV.

Applicable Mode 5GNR

Parameter File name

Set Syntax :NR5G:MAP:SAVE:CSV " Name "

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:SAVE:DATA <S>

(Write only) Save map data.

Applicable Mode 5GNR

Parameter File name

Set Syntax :NR5G:MAP:SAVE:DATA " Name "

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:ZOOM:IN

(Write only) Zoom in on the outdoor map.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:ZOOM:IN

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MAP:ZOOM:OUT

(Write only) Zoom out on the outdoor map.

Applicable Mode 5GNR

Parameter None

Set Syntax :NR5G:MAP:ZOOM:OUT

Query Syntax None

Default None

Return Type None

[:SENSe]:NR5G[:MEAS]:MPCI:BNUM <I00>

(Read-write) Query or set MPCI beam number.

Applicable Mode 5GNR

Parameter Number of beams

Range: 1～8.

Set Syntax :NR5G:MPCI:BNUM 2

Query Syntax :NR5G:MPCI:BNUM?

Default 3

Return Type Value (int) or character

[:SENSe]:NR5G[:MEAS]:MPCI:PAGE <I00>

(Read-write) Query or set current MPCI display page number.

Applicable Mode 5GNR

Parameter Current page number

Range: 1～6.

Set Syntax :NR5G:MPCI:PAGE 3

Query Syntax :NR5G:MPCI:PAGE?

Default 1

Return Type Value (int) or character

[:SENSe]:NR5G[:MEAS]:MPCI:VIEW <E>

(Read-write) Query or set MPCI display mode.

Applicable Mode 5GNR

Parameter Display type

CHARt(0) bar chart

TABLe(1) table

Set Syntax :NR5G:MPCI:VIEW CHAR

Query Syntax :NR5G:MPCI:VIEW?

Default CHARt

Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:SSB:OFFSet <FF0>

(Read-write) Query or set SSB offset.

Applicable Mode 5GNR

Parameter SSB offset (Hz)

Range： -1.92GHz～7.18GHz.

Set Syntax :NR5G:SSB:OFFS 1000000

Query Syntax :NR5G:SSB:OFFS?

Default 218.65MHz

Return Type Value (double) or character

[:SENSe]:NR5G[:MEAS]:SSB:SCSCase <E>

(Read-write) Query or set carrier spacing.

Applicable Mode 5GNR

Parameter Carrier spacing

A(0) Mode A 15kHz

B(1) Mode B 30kHz

C(2) Mode C 30kHz

Set Syntax :NR5G:SSB:SCSC B

Query Syntax :NR5G:SSB:SCSC?

Default C

Return Type String or value (int)

[:SENSe]:NR5G[:MEAS]:SUBFrame <I00>

(Read-write) Query or set Power vs Time mode subframe No..

Applicable Mode 5GNR

Parameter Subframe No.

Range： 0～9.

Set Syntax :NR5G:SUBF3

Query Syntax :NR5G:SUBF?

Default 0

Return Type String or value (float)

[:SENSe]:NR5G[:MEAS]:TRIGger <E>

(Read-write) Query or set Power vs Time mode Trigger..

Applicable Mode	5GNR
Parameter	Trigger
	EXT (0) External Trigger GPS (1) GPS Trigger
Set Syntax	:NR5G:TRIGger GPS
Query Syntax	:NR5G:TRIGger?
Default	EXT
Return Type	String or value (int)

[:SENSe]:OBW:METHod <E>

(Read-write) Query or set occupied bandwidth function measurement method.

Applicable Mode	Spectrum Analyzer
Parameter	OBW measurement method
	PPOW(0) power percentage
	XDB(1) XdB
Set Syntax	:OBW:METH XDB
Query Syntax	:OBW:METH?
Default	PPOW
Return Type	String or value (int)

[:SENSe]:OBW:OBW

(Read only) query OBW value(**valid after OBW is on and a sweep is completed**).

Applicable Mode	Spectrum Analyzer
Parameter	None
Set Syntax	None
Query Syntax	:OBW:OBW?
Default	None
Return Type	Value (double) or character

[:SENSe]:OBW:PPOW <F00>

(Read-write) Query or set OBW percentage value.

Applicable Mode	Spectrum Analyzer
Parameter	OBW percentage value (no unit)
	Range：10.00%-99.99%.
Set Syntax	:OBW:PPOW 90
Query Syntax	:OBW:PPOW?
Default	99%
Return Type	Value (float) or character

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

(Read-write) Query or set OBW function measurement ON/OFF**(other measurement functions will be disabled if this function is enabled)** . [:SENSe]:MEASurement command is also available.

Applicable Mode	Spectrum Analyzer
Parameter	OBW ON/OFF
	OFF(0) OBW OFF
	ON(1) OBW ON
Set Syntax	:OBW ON
Query Syntax	:OBW?
Default	OFF
Return Type	String or value (int)

[:SENSe]:OBW:XDB <F00>

(Read-write) Query or set occupied bandwidth XdB value.

Applicable Mode	Spectrum Analyzer
Parameter	OBW bandwidth XdB(dB)
	Range：-100.0dB~ -0.1dB.
Set Syntax	:OBW:XDB -3
Query Syntax	:OBW:XDB?
Default	-3dB
Return Type	Value (float) or character

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

(Read-write) Query or set limit ON/OFF.

Applicable Mode	Orientation Analysis
Parameter	Limit ON/OFF
	OFF(0) OFF
	ON (1) ON
Set Syntax	:POW:LIM:STAT OFF
Query Syntax	:POW:LIM:STAT?
Default	OFF
Return Type	String or value (int)

[:SENSe]:POWer[:RF]:ATTenuation <I00>

(Read-write) query or set attenuation value.

Applicable Mode	all modes
Parameter	Attenuation value (dB)
	It can be set to 0, 5, 10, 15, 20, 25 and 30
Set Syntax	:POW:ATT 20

Query Syntax :POW:ATT?

Default 10

Return Type Value (int) or character

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

(Read-write) query or set attenuation auto On/Off.

Applicable Mode Spectrum Analysis, Interference Analysis, Orientation Analysis, Real-time Spectrum, 5GNR,LTE Analysis

Parameter Set attenuation to auto On/Off. When attenuation is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

OFF(0) Manual

ON (1) Auto

Set Syntax :POW:ATT:AUTO ON

Query Syntax :POW:ATT:AUTO?

Default ON

Return Type String or value (int)

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

(Read-write) query or set pre-amplifier to On/Off.

Applicable Mode all modes

Parameter Preamplifier ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :POW:GAIN OFF

Query Syntax :POW:GAIN?

Default ON

Return Type String or value (int)

[:SENSe]:ROSC:SOUR<E>

(Read-write) Query or set 10MHz frequency reference source mode.

Applicable Mode all modes

Parameter Frequency reference type

OFF(0)	Frequency reference is internal and reference output is of
INTernal(1)	Frequency reference is internal and reference output is on
EXTernal(2)	Frequency reference is external, reference output is off

Set Syntax :ROSC:SOUR EXT

Query Syntax :ROSC:SOUR?

Default OFF

Return Type String or value (int)

[:SENSe]:RTSA:MEASurement <E>

(Read-write) Query or set measurement mode.

Applicable Mode Real-time Spectrum

Parameter Measurement mode

DENSITY(0) Afterglow

SPECTROGRAM(1) Waterfall

Set Syntax :RTSA:MEAS DENSITY

Query Syntax :RTSA:MEAS?

Default DENSITY

Return Type String or value (int)

[:SENSe]:RTSA:MEASurement:DENSity:BPLevel <F00>

(Read-write) Query or set lower limit of fluorescence probability.

Applicable Mode Real-time Spectrum

Parameter Lower limit of fluorescence probability

Range：0%~100%.

Set Syntax :RTSA:MEAS:DENS:BPL 20

Query Syntax :RTSA:MEAS:DENS:BPL?

Default 0

Return Type Value (float) or character

[:SENSe]:RTSA:MEASurement:DENSity:RPLevel <F00>

(Read-write) Query or set upper limit of fluorescence probability.

Applicable Mode Real-time Spectrum

Parameter Upper limit of fluorescence probability

Range：0.05%~100%.

Set Syntax :RTSA:MEAS:DENS:RPL 80

Query Syntax :RTSA:MEAS:DENS:RPL?

Default 100

Return Type Value (float) or character

[:SENSe]:RTSA:MEASurement:DENSity[:STATe] <E>

(Read-write) Query or set fluorescent display.

Applicable Mode Real-time Spectrum

Parameter Afterglow spectrum ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :RTSA:MEAS:DENS ON

Query Syntax :RTSA:MEAS:DENS?

Default ON

Return Type String or value (int)

[:SENSe]:RTSA:MEASurement:PRESet

(Write only) Reset measurement menu.

Applicable Mode Real-time Spectrum

Parameter None

Set Syntax :RTSA:MEAS:PRES

Query Syntax None

Default None

Return Type None

[:SENSe]:RTSA:MEASurement:SPECtrogram:BPLevel <F00>

(Read-write) Query or set lower limit of waterfall plot.

Applicable Mode Real-time Spectrum

Parameter Lower limit of color scale (dBm)

Range：-210~30dBm.

Set Syntax :RTSA:MEAS:SPEC:BPL -80

Query Syntax :RTSA:MEAS:SPEC:BPL?

Default -100dBm

Return Type Value (float) or character

[:SENSe]:RTSA:MEASurement:SPECtrogram:RPLevel <F00>

(Read-write) Query or set upper limit of waterfall plot.

Applicable Mode Real-time Spectrum

Parameter Upper limit of color scale

Range：-210~30dBm.

Set Syntax :RTSA:MEAS:SPEC:BPL -10

Query Syntax :RTSA:MEAS:SPEC:RPL?

Default -0dBm

Return Type Value (float) or character

[:SENSe]:RTSA:TRACe:CURS:POS <I00>

(Read-write) Query or set time marker position on the waterfall plot.

Applicable Mode Real-time Spectrum

Parameter Time marker position

Range: 0～317.

Set Syntax :RTSA:TRAC:CURS:POS 10

Query Syntax :RTSA:TRAC:CURS:POS?

Default 0

Return Type Value (int) or character

[:SENSe]:RTSA:TRACe:CURS[:STATe] <E>

(Read-write) Query or set time marker on the waterfall plot ON/OFF.

Applicable Mode Real-time Spectrum

Parameter Time marker ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :RTSA:TRAC:CURS ON

Query Syntax :RTSA:TRAC:CURS?

Default OFF

Return Type String or value (int)

[:SENSe]:RTSA:TRACe:CURS:TIME <F00>

(Read-write) Query or set time marker value on the waterfall plot (ms).

Applicable Mode Real-time Spectrum

Parameter Time marker value (ms)

Range：0s～6.30s.

Set Syntax :RTSA:TRAC:CURS:TIME 100

Query Syntax :RTSA:TRAC:CURS:TIME?

Default 0

Return Type Value (double) or character

[:SENSe]:RTSA:TRACe:PRESet:ALL

(Write only) Reset trace menu settings.

Applicable Mode Real-time Spectrum

Parameter None

Set Syntax :RTSA:TRAC:PRES:ALL

Query Syntax None

Default None

Return Type None

[:SENSe]:SE:EMESsions:ADD

(Write only) Spurious Emessions Add Seg.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax :SE:EMES:ADD

Query Syntax None

Default None

Return Type None

[:SENSe]:SE:EMESsions:ADD:SEGMent<FF0>,<FF0>,<FF0>,<FF0>,<FF0>,<FF0>

(Write only) Spurious Emessions Edit Add Seg.

Applicable Mode Spectrum Analyzer

Parameter Start Freq（0～9.1GHz）
Stop Freq（0～9.1GHz）
Res BW（1Hz～20MHz）
Video BW（1Hz～20MHz）
Start Limit（-174dBm～50dBm）
Stop Limit（-174dBm～50dBm）

Set Syntax :SE:EMES:ADD:SEGM 1e+9,2e+9,1e+6,1e+6,-50,-30

Query Syntax None

Default 8e+8,1e+9,1e+6,1e+6,-10,-10

Return Type None

[:SENSe]:SE:EMESsions:CLEar

(Write only) Spurious Emessions Edit Delete All.

Applicable Mode Spectrum Analyzer

Parameter None

Set Syntax :SE:EMES:CLE

Query Syntax None

Default None

Return Type None

[:SENSe]:SE:EMESsions:DELete <I00>

(Write only) Spurious Emessions Edit Delete Seg.

Applicable Mode Spectrum Analyzer

Parameter No.
Range：1～10.

Set Syntax :SE:EMES:DEL

Query Syntax None

Default 1

Return Type None

[:SENSe]:SE:EMESsions:SEGMent <I00>,<FF0>,<FF0>,<FF0>,<FF0>,<FF0>,<FF0>

(Read-write) Query or set Spurious Emessions Edit Add Seg.

Applicable Mode Spectrum Analyzer

Parameter No.(1～10)

Start Freq（0～9.1GHz）
Stop Freq（0～9.1GHz）
Res BW（1Hz～20MHz）
Video BW（1Hz～20MHz）
Start Limit（-174dBm～50dBm）
Stop Limit（-174dBm～50dBm）

Set Syntax :SE:EMES:SEGM 1,1e+9,2e+9,1e+6, 1e+6,-50,-30

Query Syntax :SE:EMES:SEGM?

Default None

Return Type Value or character

[:SENSe]:SE:ID <I00>

(Read-write) Query or set current segment number of SEM.

Applicable Mode Spectrum Analyzer

Parameter Current segment number

Range: 1～5.

Set Syntax :SE:ID 1

Query Syntax :SE:ID?

Default 5

Return Type Value (int) or character

[:SENSe]:SE:MODE <E>

(Read-write) Query or set spurious emission mask sweep mode.

Applicable Mode Spectrum Analyzer

Parameter Sweep mode

ONCE(0) Single segment

ALL(1) All segments

Set Syntax :SE:MODE 1

Query Syntax :SE:MODE?

Default ONCE

Return Type String or value (int)

[:SENSe]:SE:STATe <E>

(Read-write) Query or set spurious emission mask state.

Applicable Mode Spectrum Analyzer

Parameter Spurious emission mask ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :SE:STAT 1

Query Syntax :SE:STAT?

Default OFF

Return Type String or value (int)

[:SENSe]:SE:TYPE <E>

(Read-write) Query or set spurious emission mask sweep type.

Applicable Mode Spectrum Analyzer

Parameter Sweep type

SINGLE(0) Single

CONT(1) Continuous

Set Syntax :SE:TYPE SINGLE

Query Syntax :SE:TYPE?

Default SINGLE

Return Type String or value (int)

[:SENSe]:SWEep:FAST <E>

(Read-write) Query or set fast sweep ON/OFF.

Applicable Mode Spectrum Analyzer

Parameter Sweep mode

OFF(0) Default

ON(1) Fast

Set Syntax :SWE:FAST ON

Query Syntax :SWE:FAST?

Default OFF

Return Type String or value (int)

[:SENSe]:SWEep:MODE <E>

(Read-write) Query or set sweep mode.

Applicable Mode Spectrum Analyzer

Parameter Sweep mode

LIN(0) Linear sweep

LIST(1) List sweep

Set Syntax :SWE:MODE LIN

Query Syntax : SWE:MODE?

Default LIN

Return Type String or value (int)

[:SENSe]:SWEep:POINts <I00>

(Read-write) Query or set sweep points.

Applicable Mode Spectrum Analysis, Interference Analysis

Parameter Sweep points

Available values are 201, 501, 1001, 2001 and 4001.

Set Syntax :SWE:POIN 501

Query Syntax	:SWE:POIN
Default	1001
Return Type	Value (int) or character

[:SENSe]:SWEep:TIME <F00>

(Read-write) Query or set sweep time in linear sweep mode. The sweep time is the time required for the local oscillator to tune through the selected frequency spacing. Sweep time directly affects the time required to complete a test, which does not include the dead time between the completion of a sweep and the start of the next sweep. Sweep time usually varies with Span, RBW and VBW. Sweep time cannot be set when RBW<=1kHz in the Spectrum Analyzer mode.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter	Sweep time in linear sweep mode (in ms)
Set Syntax	:SWE:TIME 100
Query Syntax	:SWE:TIME?
Default	98.739ms
Return Type	Value (double) or character

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

(Read only) Query or Set sweep time to auto On/Off. When set to On, the instrument will adopt the sweep speed as high as possible; or you can manually increase the sweep time to meet some specific measurement requirements. Sweep time set manually must be > = automatic sweep time.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter	Auto ON/OFF of sweep time in linear sweep mode OFF(0) manual sweep time ON(1) auto sweep time
Set Syntax	:SWE:TIME:AUTO ON
Query Syntax	:SWE:TIME:AUTO?
Default	ON
Return Type	String or value (int)

[:SENSe]:SWEep:TRIGger <E>

(Read-write) Query or set trigger type.

Applicable Mode	Spectrum Analyzer
Parameter	Trigger type FREE(0) Free trigger VIDEO(1) Video trigger EXTR(2) External trigger
Set Syntax	:SWE:TRIG FREE
Query Syntax	: SWE: TRIG?
Default	FREE
Return Type	String or value (int)

[:SENSe]:SWEep:TRIGger:EXTRa:DELay <F00>

(Read-write) Query or set external trigger delay.

Applicable Mode	Spectrum Analyzer
Parameter	External trigger delay (μs) Range：1μs～500ms.
Set Syntax	:SWE:TRIG:EXTR:DEL 1
Query Syntax	:SWE:TRIG:EXTR:DEL?
Default	1μs
Return Type	Value (double) or character

[:SENSe]:SWEep:TRIGger:EXTRa:SLOP <E>

(Read-write) Query or set external trigger polarity.

Applicable Mode	Spectrum Analyzer
Parameter	External trigger polarity POS(0) positive NEG(1) negative
Set Syntax	:SWE:TRIG:EXTR:SLOP POS
Query Syntax	:SWE:TRIG:EXTR:SLOP?
Default	POS
Return Type	String or value (int)

[:SENSe]:SWEep:TRIGger:VIDeo:AMPLitude <F00>

(Read-write) Query or set video trigger level.

Applicable Mode	Spectrum Analyzer
Parameter	Video trigger level (dBm) Range：-150dBm～30dBm.
Set Syntax	:SWE:TRIG:VID:AMPL 10
Query Syntax	:SWE:TRIG:VID:AMPL?
Default	-25dBm
Return Type	Value (float) or character

[:SENSe]:TAListen:AVOLume <I00>

(Read-write) Query or set audio demodulation measurement volume.

Applicable Mode	Spectrum Analyzer
Parameter	Demodulation volume (no unit) Range: 0～100.
Set Syntax	:TAL:AVOL 80
Query Syntax	:TAL:AVOL?
Default	95

Return Type Value (int) or character

[:SENSe]:TAListen:DMODe <E>

(Read-write) Query or set audio demodulation measurement demodulation mode. The intermittent mode is that the data is scanned after one screen and then demodulated intermittently for a period of time according to the demodulation time, then the data is scanned again after one screen and then demodulated intermittently for a period of time according to the demodulation time, and so on and so forth; the continuous mode is that the data is continuously demodulated after one screen is scanned and the data is not scanned anymore.

Applicable Mode	Spectrum Analyzer
Parameter	Demodulation mode
	INTer(0) Intermittent
	CONT(1) Continuous
Set Syntax	:TAL:DMOD CONT
Query Syntax	:TAL:DMOD?
Default	INTer
Return Type	Value (int) or character

[:SENSe]:TAListen:DSTate <E>

(Read-write) Query or set audio demodulation measurement ON/OFF**(other measurement functions will be disabled if this function is enabled)**. [:SENSe]:MEASurement command is also available.

Applicable Mode	Spectrum Analyzer
Parameter	Demodulation ON/OFF
	OFF(0) demodulation OFF
	ON(1) demodulation ON
Set Syntax	:TAL:DST ON
Query Syntax	:TAL:DST?
Default	OFF
Return Type	String or value (int)

[:SENSe]:TAListen:DTYPe <E>

(Read-write) Query or set audio demodulation measurement demodulation type.

Applicable Mode	Spectrum Analyzer
Parameter	Demodulation type
	FM(0) Frequency modulation
	AM(1) Amplitude modulation
	USB(2) Upper sideband
	LSB(3) Lower sideband
Set Syntax	:TAL:DTYP FM
Query Syntax	:TAL:DTYP?
Default	FM

Return Type String or value (int)

[:SENSe]:TAListen:LTIMe <FF0>

(Read-write) Query or set audio demodulation measurement demodulation time, which works when the demodulation mode is intermittent mode, and is the time in demodulation state after scanning.

Applicable Mode Spectrum Analyzer

Parameter Demodulation time (μs)

Range：1μs～400s.

Set Syntax :TAL:LTIM 100

Query Syntax :TAL:LTIM?

Default 100ms

Return Type Value (double) or character

:SYSTem:BATTery:STATe

(Read only) Query battery state.

Applicable Mode all modes

Parameter None

Set Syntax None

Query Syntax :SYST:BATT:STAT?

Default None

Return Type Value (int) or character

:SYSTem:BATTery:VOLume

(Read only) Query battery volume.

Applicable Mode all modes

Parameter None

Set Syntax None

Query Syntax :SYST:BATT:VOL?

Default None

Return Type Value (double) or character

:SYSTem:GNSS:GAUSs <E>

(Read-write) Query or set Gaussian coordinate ON/OFF.

Applicable Mode all modes

Parameter Gaussian coordinate ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :SYST:GNSS:GAUS OFF

Query Syntax :SYST:GNSS:GAUS?

Default OFF

Return Type String or value (int)

:SYSTem:GNSS:IDENtifier <E>

(Read-write) Select GNSS positioning system.

Applicable Mode all modes

Parameter Positioning system

GPS(0)	GPS
SBAS(1)	SBAS
BDS(3)	Beidou
QZSS(5)	QZSS
GLON(6)	GLONASS

Set Syntax :SYST:GNSS:IDEN 3

Query Syntax :SYST:GNSS:IDEN?

Default GPS

Return Type String or value (int)

:SYSTem:GPS <E>

(Read-write) Query or set GPS ON/OFF. When set to ON, the data collected by the GPS, such as longitude, latitude, and altitude, will be displayed on the screen. This command is an interleaved command. Use ***OPC?**to query whether the command is completed before sending other commands.

Applicable Mode all modes

Parameter GPS ON/OFF

OFF(0)	GPS OFF
ON(1)	GPS ON

Set Syntax :SYST:GPS ON; *OPC?

Query Syntax :SYST:GPS?

Default OFF

Return Type Value (bool) or character

:SYSTem:GPS:DATA

(Read only) return current GPS data in the following format: "<Longitude>,<Latitude>,<Altitude>,<Time UTC>".

Applicable Mode all modes

Parameter None

Set Syntax None

Query Syntax :SYST:GPS:DATA?

Default None

Return Type String

:SYSTem:GPS:RECeive[:STATe]

(Read only) Query GPS receiver status.

Applicable Mode	all modes
Parameter	None
Set Syntax	None
Query Syntax	:SYST:GPS:REC?
Default	None
Return Type	Value (int) or character (0 means the receiver is OFF, 1 means the receiver is ON)

:SYSTem:GPS:RST

(Write only) GPS cold start, sometimes, for example, when in places where the signal is extremely poor, there is no way to receive the GPS signal for a long time, and sometimes there is no way to change the location. At this time, choose cold start to run the new star search positioning will make the starting speed faster. In this case, you can choose cold start to let the module search for galaxy positioning again..

Applicable Mode	all modes
Parameter	None
Set Syntax	: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
Set Syntax	None
Query Syntax	:SYST:GPS:STAT?
Default	None
Return Type	Value (int) or character 0 means not positioned 1 means non-differential positioning 2 means differential positioning 3 mean invalid PPS 4 means estimating

:SYSTem:INFO

(Read only) Query system information.

Applicable Mode	all modes
Parameter	None
Set Syntax	None
Query Syntax	:SYST:INFO?
Default	None

Return Type String

:SYSTem:PWR:SHUTdown <I00>

(Read-write) Query or set shutdown time.

Applicable Mode all modes

Parameter Shutdown time

Range：0~240 min.

Set Syntax :SYST:PWR:SHUT 20

Query Syntax :SYST:PWR:SHUT?

Default 10 min

Return Type Value (int) or character

:SYSTem:PWR:SHUTdown:STATe <E>

(Read-write) Query or set shutdown time switch. When it is turned on, the instrument will automatically shut down when the timer reaches the set shutdown time**(Doing anything will cause the shutdown timer to re-time)**.

Applicable Mode all modes

Parameter Automatic shutdown ON/OFF

OFF(0) OFF

ON (1) ON

Set Syntax :SYST:PWR:SHUT:STAT OFF

Query Syntax :SYST:PWR:SHUT:STAT?

Default OFF

Return Type String or value (int)

:SYSTem:PWR:SLEep <I00>

(Read-write) Query or set sleep time.

Applicable Mode all modes

Parameter Sleep time

Range：0~240 min.

Set Syntax :SYST:PWR:SLE 20

Query Syntax :SYST:PWR:SLE?

Default 5 min

Return Type Value (int) or character

:SYSTem:PWR:SLEep:STATe <E>

(Read-write) Query or set sleep time ON/OFF. When it is turned on, the instrument will automatically sleep and close the screen display when it counts to the set sleep time**(Doing anything will cause the sleep timer to re-time)**.

Applicable Mode all modes

Parameter Auto sleep ON/OFF

OFF(0) means OFF

	ON(1) means ON
Set Syntax	:SYST:PWR:SLE:STAT OFF
Query Syntax	:SYST:PWR:SLE:STAT?
Default	OFF
Return Type	String or value (int)

:SYSTem:TEMP

(Read only) Query the temperature.

Applicable Mode	all modes
Parameter	None
Set Syntax	None
Query Syntax	:SYST:TEMP?
Default	None
Return Type	Value (double) or character

:SYSTem:TIME <I00>,<I00>,<I00>,<I00>,<I00>

(Read-write) Query or set time.

Applicable Mode	all modes
Parameter	Time: set year, month, day, hour, minute and second respectively.
Set Syntax	:SYST:TIME 2015,12,10,8,15,38
Query Syntax	:SYST:TIME?
Default	None
Return Type	String

:TRAC[1]|2-4:DATA

(Read only) query trace data.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter<n>	Trace number
	It can be set to 1, 2 and 3, respectively indicating the Trace 1, 2 and 3. n is 1 if not specified.
Parameter	None
Set Syntax	None
Query Syntax	:TRAC1:DATA?
Default	None
Return Type	String or block
	The string format is xx xx,xx. xx,…xx. xx \ n, where xx.xx refers to float data, "," between values, refers to the end of the interval, and "\n" refers to the end.
	The format of data block is "#NXXXX data". Where XXXX represents the size of binary data. N stands for the number of digits of XXXX. For example \3512 shows that the size digits of binary data are 3. Read the three digits after "3" to get 512,

followed by 512 bytes of binary data. The data type of each point of the trace is float type, accounting for 4 bytes.

:TRAC[1]|2-4:TYPE <E>

(Read-write) query or set trace state. The trace number, which can be set to 1,2 and 3, respectively representing Trace 1, Trace 2 and Trace 3.

n is 1 if not specified.

Applicable Mode	Spectrum Analysis, Interference Analysis, Real-time Spectrum
Parameter	Trace status
	CLRW(0) indicates refresh trace
	MAXH(1) indicates maximum hold
	MINH(2) indicates minimum hold
	VIEW(3) indicates to hold trace
	BLANk(4) indicates to hide trace
Set Syntax	:TRAC2:TYPE CLRW
Query Syntax	:TRAC2:TYPE?
Default	CLRW
Return Type	String or value (int)

Section IV Programing instances

This chapter introduces the use of different I/O libraries and programming languages to illustrate the control of the spectrum analyzer. The control of the device is realized through the LAN (when the USB port is selected for communication, the USB driver needs to be installed first. For the specific steps, refer to the instructions on USB driver installation in Section 1 of Chapter 2. After the USB driver is successfully installed, the specific implementation steps are the same as the LAN port communication).

1. Example of C/C++

The PC shall at least meet the following configurations:
Windows XP operating system
VC6.0 integrated development environment
VISA Library from NI
Network card;

2． Running C/C++ design program

To run an example program written in C/C++,the corresponding library file needs to be included in the project in VC6.0:
If the VISA library is used, the following steps are required:
Add visa.h to the header file
Add visatype.h to the header file
Add visa32.lib to the project

3. Network design example

In order to use the following example correctly, please first confirm the IP address of the 4024CA spectrum analyzer.

1) Set frequency and query using socket and C++

The specific implementation code is (the code can be ported, and this section provides only an implementation example):
Establish a dialog based MFC project, and add the following code to 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 successful!"));
            }
            else
            {
                AfxMessageBox(_T("Socket creation failed!"));
```

```
                sockClient.Close();
            }
        }
        flag = sockClient.Connect(name,5025); //* name Is the IP address of the 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) Implementation of setup and query commands using VISA library and C++
The following files need to be included

```
#include<visa.h>
#include <afxsock.h>
#include<visa.h>
extern char ResourceStr[50];
ViSession DftRM;
ViSession vi;
//* Starting the Instrument
ViStatus AV4024_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::5025::SOCKET"); //* x.x.x.x Indicates the IP address of the 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;
}
//* SetCenterFreq
ViStatus AV4024_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;
}
//* QueryCenterFreq
ViStatus AV4024_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 Description of Secondary Development Library Functions

For the convenience of users, we have encapsulated the SCPI commands into a dynamic link library. Users can easily Query or set up the 4024CA by calling the dynamic link library, which is suitable for users to build an automatic test system.

Section I Driver installation

If the instrument is to be controlled via a direct network cable, the specific installation procedure is as follows:

1. Install VISA library and IVI library (either Agilent or Ni can be installed. It is better not to install them at the same time. There may be conflicts).

To install the NI library, first install the NI VISA library (folder NI VISA1750, the computer has been installed without reinstallation), and then install the IVI library IviSharedComponents_261.exe.

If Agilent library is to be installed, just install the Agilent IO library program IOLibSuite_18_1_24715.exe.

2. Install the IVI program control function library CySAnIviComDriver.msi. for 4024CA. After installation, you can see the program control function library header file CySAn.h in C:\Program Files\IVI Foundation\IVI\Include. The static link library file CySAn.lib is in C:\Program Files\IVI Foundation\IVI\Lib\msc, and the dynamic link library file CySAn.dll is in C:\Program Files\IVI Foundation\IVI\Bin, which can directly call the CySAn.h file in Function control 4024CA spectrum instrument

If the instrument is controlled via usb, the usb driver 4024USBSETUP.inf needs to be installed in addition to the above steps.

Section II Function Description

Instructions for using dynamic link library.

The dynamic library includes three files, CySAn.h, CySAn.h.dll and CySAn.h.lib. Users can use the functions in CySAn.h to control the instrument by adding the above three files to the project in the programming environment.

Instrument connection - turn on the instrument

ViStatus CySAn_init(ViRsrc ResourceName, ViBoolean IdQuery, ViBoolean Reset, ViSession* Vi)

Function Usage:

Starting the Instrument

This function is the first function to be called when accessing the instrument driver. It completes the following initialization operations:

According to the interface and logical address information specified by the parameter ResourceName, open the handle of the module and connect with the spectrum analyzer to establish a data channel.

Return Vi to identify the module in a subsequent call to the instrument driver function.

Parameter list:

ResourceName

Instrument resource character

The TCP connection resource string is "TCPIP::172.141.11.202::5025::SOCKET", and the underlined part is the default IP address of the instrument. If the IP address of the instrument is changed, the actual IP address should be used.

USB connection resource string is " USB0::0x0525::0xA4A7::NI-VISA-10001::1::RAW"

IdQuery

ID query parameter. If set to VI_TRUE, the function queries the instrument ID and checks whether it conforms to this driver. If yes, the instrument is opened successfully. If no, it fails to open.

Reset

This parameter is set to VI_TRUE, then this function resets the instrument and has the same effect as sending the *RST command.

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Instrument connection - turn off the instrument

ViStatus CySAn_close(ViSession Vi)

Function Usage:

Turn off the instrument; after controlling the instrument, it is required to recall this function to turn off the instrument.

Parameter list:

Vi

Instrument handle

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Reset

ViStatus _VI_FUNC CySAn_reset(ViSession Vi)

Function Usage:

Restore the current operating mode of the instrument to a known default state, called the default state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Common functions for all measurement modes

Mode - Query available instrument modes

ViStatus _VI_FUNC CySAn_GetInstCatalog(ViSession Vi, ViInt32* Val)

Function Usage:

Query available instrument operating modes.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Available instrument operating modes.

Bit 0 is the spectrum analysis test bit, which is 1 (required);

Bit 1 is the interference analysis test bit, with 1 being settable (optional) and 0 being non-settable;

Bit 2 is the directional analysis test bit, with 1 being settable (optional) and 0 being non-settable;

Bit 3 is the GSM/EDGE test bit, with 1 being settable (optional) and 0 being non-settable;

Bit 4 is the 5GNR test bit, with 1 being settable (optional) and 0 being non-settable;

Bit 5 is the LTE analysis test bit, 1 is settable (optional), 0 is not settable.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Mode - Set instrument operating mode

ViStatus _VI_FUNC CySAn_SetInstMode(ViSession Vi, ViInt32 InstMode)

Function Usage:

Set instrument operating mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

InstMode

Instrument mode.

SA(1) Spectrum analysis

IA(2) Interference analysis

COMPASS(3) Orientation Analysis

RTSA(4) Real-time spectrum analysis

GSM(5) GSM/EDGE

NR(6) 5GNR

LTE(7) LTE analysis

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Mode - Query instrument operating mode

ViStatus _VI_FUNC CySAn_GetInstMode(ViSession Vi, ViInt32* InstMode)

Function Usage:

Query instrument operating mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

InstMode

Instrument mode.

SA(1) Spectrum analysis

IA(2) Interference analysis

COMPASS(3) Orientation Analysis

RTSA(4) Real-time spectrum analysis

GSM(5) GSM/EDGE

NR(6) 5GNR

LTE(7) LTE analysis

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Data - Set data format

ViStatus _VI_FUNC CySAn_SetDataFormat(ViSession Vi, ViInt32 DataFormat)

Function Usage:

Set data format.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DataFormat

Data type

ASC (0) indicates character format.

HEX(1) indicates numeric format.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Data - Query data format

ViStatus _VI_FUNC CySAn_GetDataFormat(ViSession Vi, ViInt32* DataFormat)

Function Usage:

Query data format.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DataFormat

Data type.

ASC (0) indicates character format.

HEX(1) indicates numeric format.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Download file

ViStatus _VI_FUNC CySAn_GetFileData(ViSession Vi, ViConstString FileName, ViInt32 FileType, ViInt32 ValBufferSize, ViChar Val[])

Function Usage:

Download files.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

FileType

File type.

ValBufferSize

Data buffer size.

Val[]

Data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete data file

ViStatus _VI_FUNC CySAn_DeleteDataFile(ViSession Vi, ViConstString FileNane)

Function Usage:

Delete data files in the current mode**(the command is invalid if the file does not exist and is valid only for the current storage location)**。

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete all data file

ViStatus _VI_FUNC CySAn_DeleteDataFileAll(ViSession Vi)

Function Usage:

Delete all data files in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load data file

ViStatus _VI_FUNC CySAn_LoadDataFile(ViSession Vi, ViConstString FileName)

Function Usage:

Call data files in the current mode**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store data file

ViStatus _VI_FUNC CySAn_StoreDataFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store data files in current mode**(the file will overwrite an existing file and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store CSV data file

ViStatus _VI_FUNC CySAn_StoreCSVDataFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store CSV data files in current mode**(the file will overwrite an existing file and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete state file

ViStatus _VI_FUNC CySAn_DeleteStateFile(ViSession Vi, ViConstString FileName)

Function Usage:

Delete the state files in the current mode (**the command is invalid if the file does not exist and is valid only for the current storage location**).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete all state files

ViStatus _VI_FUNC CySAn_DeleteAllStateFile(ViSession Vi)

Function Usage:

Delete all state files in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load state file

ViStatus _VI_FUNC CySAn_LoadStateFile(ViSession Vi, ViConstString FileName)

Function Usage:

Recall state files in the current mode (**the command is invalid if the file does not exist** and

is valid only for the current storage location).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store state file

ViStatus _VI_FUNC CySAn_StoreStateFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store the current state as a state file**(the file will overwrite an existing file and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Set storage location

ViStatus _VI_FUNC CySAn_SetStorageLocation(ViSession Vi, ViInt32 StorageLocation)

Function Usage:

Set File Location.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

StorageLocation

Storage location.

INT(0) Internal

SD(1) SD card

USB(2) USB

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Query storage location

ViStatus _VI_FUNC CySAn_GetStorageLocation(ViSession Vi, ViInt32*

StorageLocation)

Function Usage:

Query the current storage location.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

StorageLocation

Storage location.

INT(0) Internal

SD(1) SD card

USB(2) USB

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store screen copy

ViStatus _VI_FUNC CySAn_StoreScreen(ViSession Vi, ViConstString FileName)

Function Usage:

Copy the screen, that is, to store the current screenshot as a file (**the file will overwrite an existing file and is valid only for the current storage location**).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Upload antenna factor file

ViStatus _VI_FUNC CySAn_UploadAntennaFile(ViSession Vi, ViConstString Data)

Function Usage:

Upload antenna factor file.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Data

Antenna factor data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Upload limit line file

ViStatus _VI_FUNC CySAn_UploadAntennaFile(ViSession Vi, ViConstString Data)

Function Usage:

Upload limit line file.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Data

Limit line data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Upload list file

ViStatus _VI_FUNC CySAn_UploadListFile(ViSession Vi, ViConstString Data)

Function Usage:

Upload list file.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Data

List data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set frequency reference

ViStatus _VI_FUNC CySAn_SetRefType(ViSession Vi, ViInt32 RefType)

Function Usage:

Set 10MHz frequency reference source mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RefType

Frequency reference type

OFF(0) Frequency reference is internal, reference output is OFF

INTernal(1) Frequency reference is internal, reference output is ON

EXTernal(2) Frequency reference is external, reference output is off

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query frequency reference

ViStatus _VI_FUNC CySAn_GetRefType(ViSession Vi, ViInt32* RefType)

Function Usage:

Query 10MHz frequency reference source mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RefType

Frequency reference type

OFF(0) Frequency reference is internal, reference output is OFF

INTernal(1) Frequency reference is internal, reference output is ON

EXTernal(2) Frequency reference is external, reference output is off

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set GNSS positioning system type

ViStatus _VI_FUNC CySAn_SetGnssType(ViSession Vi, ViInt32 GnssType)

Function Usage:

Set GNSS positioning system type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GnssType

Positioning system type.

GPS(0) GPS

SBAS(1) SBAS

BDS(3) Beidou

QZSS(5) QZSS

GLON(6) GLONASS

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query GNSS positioning system type

ViStatus _VI_FUNC CySAn_GetGnssType(ViSession Vi, ViInt32* GnssType)

Function Usage:

Query GNSS positioning system type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GnssType

Positioning system type.

GPS(0) GPS

SBAS(1) SBAS

BDS(3) Beidou

QZSS(5) QZSS

GLON(6) GLONASS

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - GPS - Set GPS ON/OFF

ViStatus _VI_FUNC CySAn_SetGPSOn(ViSession Vi, ViBoolean GPSOn)

Function Usage:

Set GPS to On/Off. When set to on, the data collected by the GPS chip, such as longitude, latitude, and altitude, will be displayed on the screen.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GPSOn

GPS ON/OFF: OFF(0) means GPS off, and ON(1) means GPS on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - GPS - Query GPS ON/OFF

ViStatus _VI_FUNC CySAn_GetGPSOn(ViSession Vi, ViBoolean* GPSOn)

Function Usage:

Query GPS On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GPSOn

GPS ON/OFF: OFF(0) means GPS off, and ON(1) means GPS on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - GPS - Query GPS state

ViStatus _VI_FUNC CySAn_GetGPSState(ViSession Vi, ViInt32* Val)

Function Usage:

Query GPS state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

GPS state

0 means not positioned

1 means non-differential positioning

2 means differential positioning

3 means invalid PPS

4 means estimating

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - GPS - Query GPS receiver state

ViStatus _VI_FUNC CySAn_GetGPSReceiveState(ViSession Vi, ViBoolean* Val)

Function Usage:

Query GPS receiver state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

GPS receiver state.

0 means that there is no data in the receive

1 means that there is data in the receiver

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - GPS - Cold start

ViStatus _VI_FUNC CySAn_ResetGPS(ViSession Vi)

Function Usage:

It is for GPS cold start. Sometimes, for example, when in places where the signal is extremely poor, there is no way to receive the GPS signal for a long time, and sometimes there is no way to change the location. At this time, choose cold start to run the new star search positioning will make the starting speed faster. In this case, you can choose cold start to let the module search for galaxy positioning again.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - GPS - Query GPS Data

ViStatus _VI_FUNC CySAn_GetGPSData(ViSession Vi, ViInt32 ValBufferSize, ViChar Val[])

Function Usage:

Query data collected by the GPS, and return current GPS data in the following format: “<Longitude>,<Latitude>,<Altitude>,<Time UTC>”.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Returned GPS data.

Example: return “38 28’11.22” N,122 42’13.23” W,152,06/28/2010 23:35:38\n” where there are data

Return “--,--,--,--\n” when there are no data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Set auto shutdown ON/OFF

ViStatus _VI_FUNC CySAn_SetShutDownOn(ViSession Vi, ViBoolean ShutDownOn)

Function Usage:

Set the automatic shutdown ON/OFF. When it is turned on, the instrument will automatically shut down when the timer reaches the set shutdown time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ShutDownOn

Automatic shutdown ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Query auto shutdown ON/OFF

ViStatus _VI_FUNC CySAn_GetShutDownOn(ViSession Vi, ViBoolean* ShutDownOn)

Function Usage:

Query auto shutdown ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ShutDownOn

Automatic shutdown ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Set shutdown time

ViStatus _VI_FUNC CySAn_SetShutDownTime(ViSession Vi, ViReal64 ShutDownTime)

Function Usage:

Set the shutdown time. When the automatic shutdown ON/OFF is set to ON, the instrument will automatically shut down when the timer reaches the set shutdown time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ShutDownTime

Shutdown time (minutes), in the range of 0-240 minutes.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Query shutdown time

ViStatus _VI_FUNC CySAn_GetShutDownTime(ViSession Vi, ViReal64* ShutDownTime)

Function Usage:

Query shutdown time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ShutDownTime

Shutdown time (minutes).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Set auto sleep ON/OFF

ViStatus _VI_FUNC CySAn_SetSleepOn(ViSession Vi, ViBoolean SleepOn)

Function Usage:

Set the automatic sleep ON/OFF. When it is set to ON, the instrument will automatically sleep and close the screen display when it counts to the set sleep time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SleepOn

Automatic sleep ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Query auto sleep ON/OFF

ViStatus _VI_FUNC CySAn_GetSleepOn(ViSession Vi, ViBoolean* SleepOn)

Function Usage:

Query auto sleep ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SleepOn

Automatic sleep ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Set sleep time

ViStatus _VI_FUNC CySAn_SetSleepTime(ViSession Vi, ViReal64 SleepTime)

Function Usage:

Set the automatic sleep time. When automatic sleep is set to ON, the instrument will automatically sleep and close the screen display when it counts to the set sleep time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SleepTime

Sleep time (minutes), in the range of 0~240 minutes.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Shutdown - Query sleep time

ViStatus _VI_FUNC CySAn_GetSleepTime(ViSession Vi, ViReal64* SleepTime)

Function Usage:

Query sleep time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SleepTime

Sleep time (minutes).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set time format

ViStatus _VI_FUNC CySAn_SetTimeFormat(ViSession Vi, ViInt32 Format)

Function Usage:

Set time format.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Format

YMD(0) Year/Month/Day

MDY(1) Month/Day/Year

DMY(2) Day/Month/Year

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set time

ViStatus _VI_FUNC CySAn_SetSystemTime(ViSession Vi, ViInt32 Year, ViInt32 Month, ViInt32 Day, ViInt32 Time, ViInt32 Minute)

Function Usage:

Set time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Year, Month, Day, Time, Minute

Year, Month, Day, Time, Minute

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query time

ViStatus _VI_FUNC CySAn_GetSystemTime(ViSession Vi, ViInt32 ValBufferSize, ViChar Val[])

Function Usage:

Query time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Time

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set title

ViStatus _VI_FUNC CySAn_SetTitle(ViSession Vi, ViConstString Title)

Function Usage:

Set title.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Title

Title name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set title ON/OFF

ViStatus _VI_FUNC CySAn_SetTitleOn(ViSession Vi, ViBoolean TitleOn)

Function Usage:

Set title ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TitleOn

Title ON/OFF: OFF(0) means title off, and ON(1) means title on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query title ON/OFF

ViStatus _VI_FUNC CySAn_GetTitleOn(ViSession Vi, ViBoolean* TitleOn)

Function Usage:

Query title ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TitleOn

Title ON/OFF: OFF(0) means title off, and ON(1) means title on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set display mode

ViStatus _VI_FUNC CySAn_SetDisplayMode(ViSession Vi, ViInt32 DisplayMode)

Function Usage:

Set display mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DisplayMode

Display mode.

DEFA(0) Default mode

OUT(1) Outdoor mode

NIGHT(2) Night vision mode

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query display mode

ViStatus _VI_FUNC CySAn_GetDisplayMode(ViSession Vi, ViInt32* DisplayMode)

Function Usage:

Query display mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DisplayMode

Display mode.

DEFA(0) Default mode

OUT(1) Outdoor mode

NIGHT(2) Night vision mode

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set auto brightness adjustment ON/OFF

ViStatus _VI_FUNC CySAn_SetAutoBrightness(ViSession Vi, ViBoolean AutoBrightness)

Function Usage:

Set auto brightness adjustment ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoBrightness

Automatic brightness adjustment ON/OFF: OFF (0) means automatic adjustment is off, and ON (1) means automatic adjustment is on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query automatic brightness adjustment ON/OFF

ViStatus _VI_FUNC CySAn_GetAutoBrightness(ViSession Vi, ViBoolean* AutoBrightness)

Function Usage:

Query automatic brightness adjustment ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoBrightness

Automatic brightness adjustment ON/OFF: OFF (0) means automatic adjustment is off, and ON (1) means automatic adjustment is on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set brightness level

ViStatus _VI_FUNC CySAn_SetBrightness(ViSession Vi, ViInt32 Brightness)

Function Usage:

Set brightness level.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Brightness

Brightness level, in the range of 1~5.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query brightness level

ViStatus _VI_FUNC CySAn_GetBrightness(ViSession Vi, ViInt32* Brightness)

Function Usage:

Query brightness level.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Brightness

Brightness level.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query battery state

ViStatus _VI_FUNC CySAn_GetBatteryState(ViSession Vi, ViInt32* Val)

Function Usage:

Query battery state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Battery state.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query battery volume

ViStatus _VI_FUNC CySAn_GetBatteryVolume(ViSession Vi, ViReal64* Val)

Function Usage:

Query battery volume.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Battery volume.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Set Gaussian coordinate ON/OFF

ViStatus _VI_FUNC CySAn_SetGaussOn(ViSession Vi, ViBoolean GaussOn)

Function Usage:

Set Gaussian coordinate ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GaussOn

Gaussian coordinate ON/OFF: OFF(0) means Gaussian off, and ON(1) means Gaussian on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query Gaussian coordinate ON/OFF

ViStatus _VI_FUNC CySAn_GetGaussOn(ViSession Vi, ViBoolean* GaussOn)

Function Usage:

Query Gaussian coordinate ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GaussOn

Gaussian coordinate ON/OFF: OFF(0) means Gaussian off, and ON(1) means Gaussian on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Query system information

ViStatus _VI_FUNC CySAn_GetSystemInfo(ViSession Vi, ViInt32 ValBufferSize, ViChar Val[])

Function Usage:

Query system information

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

System information.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query package information

ViStatus _VI_FUNC CySAn_GetParamInfo(ViSession Vi, ViInt32 ValBufferSize, ViChar Val[])

Function Usage:

Query package information

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

System information.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

System - Query bias plate temperature

ViStatus _VI_FUNC CySAn_GetBiasTemp(ViSession Vi, ViReal64* Val)

Function Usage:

Query bias plate temperature

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Bias plate temperature.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Trigger sweep once

ViStatus _VI_FUNC CySAn_InitiateTrace(ViSession Vi)

Function Usage:

Trigger a single sweep (valid only in Swp Once.).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set sweep type

ViStatus _VI_FUNC CySAn_SetSweepModeContinuous(ViSession Vi, ViBoolean SweepModeContinuous)

Function Usage:

Set sweep type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepModeContinuous

Sweep type: OFF (0) means sweep once, and ON (1) means continuous sweep.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query sweep type

ViStatus _VI_FUNC CySAn_GetSweepModeContinuous(ViSession Vi, ViBoolean* SweepModeContinuous)

Function Usage:

Query sweep type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepModeContinuous

Sweep type: OFF (0) means sweep once, and ON (1) means continuous sweep.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set center frequency

ViStatus _VI_FUNC CySAn_SetCenterFreq(ViSession Vi, ViReal64 Center)

Function Usage:

Set center frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Center

Frequency value. Allowable range of center frequency in each mode:

SA	0Hz~9.1GHz
IA	5Hz~9.1GHz
COMPASS	1MHz~9.1GHz
RTSA	1831Hz~9.09999817GHz
GSM/EDGE	0Hz~9.1GHz
5GNR	0Hz~9.1GHz
LTE	5Hz~9.1GHz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query center frequency

ViStatus _VI_FUNC CySAn_GetCenterFreq(ViSession Vi, ViReal64* Center)

Function Usage:

Query center frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Center

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set step frequency

ViStatus _VI_FUNC CySAn_SetStepFreq(ViSession Vi, ViReal64 Step)

Function Usage:

Set the step value of the center frequency in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Step

Frequency value, in the range of 1Hz~5GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query step frequency

ViStatus _VI_FUNC CySAn_GetStepFreq(ViSession Vi, ViReal64* Step)

Function Usage:

Query the step value of center frequency in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Step

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set automatic step frequency ON/OFF

ViStatus _VI_FUNC CySAn_SetAutoStep(ViSession Vi, ViBoolean AutoStep)

Function Usage:

Set the automatic step frequency ON/OFF. When it is set to automatic, the step frequency value of the instrument is 1MHz. When it is set to manual, it can be set to 1Hz-5GHz.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoStep

Automatic step frequency ON/OFF: OFF (0) means automatic is off, and ON (1) means automatic is on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query automatic step frequency ON/OFF

ViStatus _VI_FUNC CySAn_GetAutoStep(ViSession Vi, ViBoolean* AutoStep)

Function Usage:

Query automatic step frequency ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoStep

Automatic step frequency ON/OFF: OFF (0) means automatic is off, and ON (1) means automatic is on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set attenuation value

ViStatus _VI_FUNC CySAn_SetAttenuation(ViSession Vi, ViReal64 Attenuation)

Function Usage:

Set the attenuation value, which can only be set to 0, 5, 10, 15, 20, 25 and 30. If set to other values, the adjacent attenuation values will be set.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Attenuation

The attenuation value can be set to 0, 5, 10, 15, 20, 25 and 30.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query attenuation value

ViStatus _VI_FUNC CySAn_GetAttenuation(ViSession Vi, ViReal64* Attenuation)

Function Usage:

Query attenuation value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Attenuation

The attenuation value can be set to 0, 5, 10, 15, 20, 25 and 30.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set preamplifier ON/OFF

ViStatus _VI_FUNC CySAn_SetIfGainAtten(ViSession Vi, ViInt32 IfGainAtten)

Function Usage:

Set pre-amplifier to On/Off. When set to On, it can improve the measurement accuracy of

low-power signals, but it is better to turn it off when measuring high-power signals, otherwise it may lead to AD overloading of measurement.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfGainAtten

Preamplifier ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query pre-amplifier ON/OFF

ViStatus _VI_FUNC CySAn_GetIfGainAtten(ViSession Vi, ViInt32* IfGainAtten)

Function Usage:

Query pre-amplifier On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfGainAtten

Preamplifier ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Spectrum analysis mode function

Frequency - Set span

ViStatus _VI_FUNC CySAn_SetSpan(ViSession Vi, ViReal64 Span)

Function Usage:

Set span in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Span

Frequency value.

SA 0Hz~9.1GHz

IA 0Hz~9.1GHz

RTSA 3.662kHz~120MHz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query span

ViStatus _VI_FUNC CySAn_GetSpan(ViSession Vi, ViReal64* Span)

Function Usage:

Query span in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Span

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Full span

ViStatus _VI_FUNC CySAn_SetFullSpan(ViSession Vi)

Function Usage:

Set to full span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Zero span

ViStatus _VI_FUNC CySAn_SetZeroSpan(ViSession Vi)

Function Usage:

Set to zero Span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set previous span

ViStatus _VI_FUNC CySAn_SetPreviousSpan(ViSession Vi)

Function Usage:

Set to previous span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set start frequency

ViStatus _VI_FUNC CySAn_SetStartFreq(ViSession Vi, ViReal64 Start)

Function Usage:

Set start frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Start

Frequency value.

The spectrum analysis frequency can be set in the range of 0Hz-9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query start frequency

ViStatus _VI_FUNC CySAn_GetStartFreq(ViSession Vi, ViReal64* Start)

Function Usage:

Query start frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Start

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set stop frequency

ViStatus _VI_FUNC CySAn_SetStopFreq(ViSession Vi, ViReal64 Stop)

Function Usage:

Set stop frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Stop

Frequency value.

The spectrum analysis frequency can be set in the range of 0Hz~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query stop frequency

ViStatus _VI_FUNC CySAn_GetStopFreq(ViSession Vi, ViReal64* Stop)

Function Usage:

Query stop frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Stop

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set signal tracking ON/OFF

ViStatus _VI_FUNC CySAn_SetSignalTrackOn(ViSession Vi, ViBoolean SignalTrackOn)

Function Usage:

Query sweep time auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SignalTrackOn

Signal tracking ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query signal tracking ON/OFF

ViStatus _VI_FUNC CySAn_GetSignalTrackOn(ViSession Vi, ViBoolean* SignalTrackOn)

Function Usage:

Query sweep time auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SignalTrackOn

Signal tracking ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set signal standard name

ViStatus _VI_FUNC CySAn_SetSignalStandard(ViSession Vi, ViConstString SignalStandard)

Function Usage:

Set signal standard name in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SignalStandard

Signal standard name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query signal standard name

ViStatus _VI_FUNC CySAn_GetSignalStandard(ViSession Vi, ViInt32 SignalStandardBufferSize, ViChar SignalStandard[])

Function Usage:

Query signal standard name in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SignalStandardBufferSize

Signal standard buffer size.

Standard

Signal standard name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set signal standard channel number

ViStatus _VI_FUNC CySAn_SetChannelNum(ViSession Vi, ViInt32 ChannelNum)

Function Usage:

Set channel number in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

channelNum

Channel number.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query signal standard channel number

ViStatus _VI_FUNC CySAn_GetChannelNum(ViSession Vi, ViInt32* ChannelNum)

Function Usage:

Set channel number in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

channelNum

Channel number.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set zero span IF output ON/OFF

ViStatus _VI_FUNC CySAn_SetIfOutOn(ViSession Vi, ViBoolean IfOut)

Function Usage:

Set zero span IF output ON/OFF

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfOut

IF output ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query zero span IF output ON/OFF

ViStatus _VI_FUNC CySAn_GetIfOutOn(ViSession Vi, ViBoolean* IfOut)

Function Usage:

Query zero span IF output ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfOut

IF output ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set zero span IF output IF selection

ViStatus _VI_FUNC CySAn_SetIfOutSelect(ViSession Vi, ViBoolean IfOutSelect)

Function Usage:

Set zero span IF output IF selection.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfOutSelect

IF selection: 3IF(0) means three IF outputs, and 4IF(1) means four IF outputs.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query zero span IF output IF selection

ViStatus _VI_FUNC CySAn_GetIfOutSelect(ViSession Vi, ViBoolean* IfOutSelect)

Function Usage:

Query zero span IF output IF selection.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfOutSelect

IF selection

IF selection: 3IF(0) means three IF outputs, and 4IF(1) means four IF outputs.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set reference level

ViStatus _VI_FUNC CySAn_SetReference(ViSession Vi, ViReal64 Reference)

Function Usage:

Set reference level value. The reference level value is related to the current amplitude unit, the setting range corresponds to dBm, and conversion is required.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value, in the range of -150dBm~+30dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query reference level

ViStatus _VI_FUNC CySAn_GetReference(ViSession Vi, ViReal64* Reference)

Function Usage:

Query reference level value (reference value). The reference level value is related to the current amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value, in the range of -150dBm~+30dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set reference position

ViStatus _VI_FUNC CySAn_SetReferPosition(ViSession Vi, ViInt32 ReferPosition)

Function Usage:

Set reference position.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ReferPosition

Reference position, in the range of -10~10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query reference position

ViStatus _VI_FUNC CySAn_GetReferPosition(ViSession Vi, ViInt32* ReferPosition)

Function Usage:

Query reference position.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ReferPosition

Reference position, in the range of -10~10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_SetAttenuationAuto(ViSession Vi, ViBoolean AttenuationAuto)

Function Usage:

Set attenuation to auto On/Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_GetAttenuationAuto(ViSession Vi, ViBoolean* AttenuationAuto)

Function Usage:

Query attenuation auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set scale/div.

ViStatus _VI_FUNC CySAn_SetYScalse(ViSession Vi, ViReal32 YScalse)

Function Usage:

Set scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. value, in the range of 0.01dB~100dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query scale/div.

ViStatus _VI_FUNC CySAn_GetYScalse(ViSession Vi, ViReal32* YScalse)

Function Usage:

Query scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. value, in the range of 0.01dB~100dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set scale type

ViStatus _VI_FUNC CySAn_SetVideoType(ViSession Vi, ViInt32 VideoType)

Function Usage:

Set scale type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoType

Scale type: LOG(0) means logarithmic type and LIN(1) means linear type.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query scale type

ViStatus _VI_FUNC CySAn_GetVideoType(ViSession Vi, ViInt32* VideoType)

Function Usage:

Query scale type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoType

Scale type: LOG(0) means logarithmic type and LIN(1) means linear type.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set unit

ViStatus _VI_FUNC CySAn_SetAmplitudeUnits(ViSession Vi, ViInt32 AmplitudeUnits)

Function Usage:

Set amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AmplitudeUnits

Amp Unit

Instrument mode	Unit
Spectrum	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA

Analyzer	0,1,2,3,4,5,6, 7,8,9,10
IA	DBM 0
Orientation analysis	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query unit

ViStatus _VI_FUNC CySAn_GetAmplitudeUnits(ViSession Vi, ViInt32* AmplitudeUnits)

Function Usage:

Query amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AmplitudeUnits

Amp Unit

Instrument mode	Unit
Spectrum Analyzer	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10
IA	DBM 0
Orientation analysis	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set resolution bandwidth

ViStatus _VI_FUNC CySAn_SetResolutionBandwidth(ViSession Vi, ViReal64 ResolutionBandwidth)

Function Usage:

Set resolution bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidth

Frequency value (in Hz), in the range of 1 Hz~20 MHz in 1-3-10 steps.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query resolution bandwidth

ViStatus _VI_FUNC CySAn_GetResolutionBandwidth(ViSession Vi, ViReal64* ResolutionBandwidth)

Function Usage:

Query resolution bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidth

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set video bandwidth

ViStatus _VI_FUNC CySAn_SetVideoBandwidth(ViSession Vi, ViReal64 VideoBandwidth)

Function Usage:

Set video bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidth

Frequency value (in Hz), in the range of 1 Hz~20 MHz in 1-3-10 steps.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query video bandwidth

ViStatus _VI_FUNC CySAn_GetVideoBandwidth(ViSession Vi, ViReal64* VideoBandwidth)

Function Usage:

Query video bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidth

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set auto resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_SetResolutionBandwidthAuto(ViSession Vi, ViBoolean ResolutionBandwidthAuto)

Function Usage:

Set RBW to be auto On/Off. When set to Auto, RBW will adjust RBW by span according to the ratio of SPAN/RBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidthAuto

Automatic resolution bandwidth ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query auto resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_GetResolutionBandwidthAuto(ViSession Vi, ViBoolean* ResolutionBandwidthAuto)

Function Usage:

Query auto On/Off status of RBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidthAuto

Automatic resolution bandwidth ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set video resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_SetVideoBandwidthAuto(ViSession Vi, ViBoolean VideoBandwidthAuto)

Function Usage:

Set auto on/off of VBW. When set to Auto, VBW will adjust RBW by resolution bandwidth according to the ratio of RBW/VBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidthAuto

Automatic video bandwidth ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query video resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_GetVideoBandwidthAuto(ViSession Vi, ViBoolean* VideoBandwidthAuto)

Function Usage:

Query auto On/Off status of VBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidthAuto

Automatic video bandwidth ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set SPAN/RBW

ViStatus _VI_FUNC CySAn_SetSpanRBWRatio(ViSession Vi, ViInt32 SpanRBWRatio)

Function Usage:

Set the value of SPAN/RBW in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpanRBWRatio

SPAN/RBW value, range 1~500.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query SPAN/RBW

ViStatus _VI_FUNC CySAn_GetSpanRBWRatio(ViSession Vi, ViInt32* SpanRBWRatio)

Function Usage:

Query the SPAN/RBW value in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpanRBWRatio

SPAN/RBW value, range 1~500.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set RBW/VBW

ViStatus _VI_FUNC CySAn_SetRBWVBWRatio(ViSession Vi, ViInt32 RBWVBWRatio)

Function Usage:

Set RBW/VBW value in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RBWVBWRatio

RBW/VBW value, range 1~100.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query RBW/VBW

ViStatus _VI_FUNC CySAn_GetRBWVBWRatio(ViSession Vi, ViInt32* RBWVBWRatio)

Function Usage:

Query RBW/VBW value in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RBWVBWRatio

RBW/VBW value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Mkr - SetMkrState

ViStatus _VI_FUNC CySAn_SetMarkerType(ViSession Vi, ViInt32 Index, ViInt32 Type)

Function Usage:

Set marker state in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Type

Marker state.

OFF(0) Marker OFF

NORM(1) Normal marker ON

DELTa(2) Delta marker ON

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Mkr - QueryMkrState

ViStatus _VI_FUNC CySAn_GetMarkerType(ViSession Vi, ViInt32 Index, ViInt32* Type)

Function Usage:

Query marker state in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Type

Marker state.

OFF(0) Marker OFF

NORM(1) Normal marker ON

DELTa(2) Delta marker ON

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Activate marker

ViStatus _VI_FUNC CySAn_SetMarkerActive(ViSession Vi, ViInt32 Index)

Function Usage:

Activate marker in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Marker function (marker->)

ViStatus _VI_FUNC CySAn_SetInstrumentToMarker(ViSession Vi, ViInt32 Index, ViInt32 InstrumentSetting)

Function Usage:

Set marker function in the current mode (which is Mkr -> in the Spectrum Analyzer mode).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

InstrumentSetting

Instrument mode	InstrumentSetting	Function
Non-zero span	STARt(0)	Marker -> Start frequency (set marker frequency to start frequency)
	STOP(1)	Marker -> Stop frequency (set marker frequency to stop frequency)
	CENTer (2)	Marker -> Center frequency (set marker frequency to center frequency)
	STEP(3)	Marker -> Step frequency (set marker frequency to step frequency)
Span Zero	STARt(0)	Marker -> Start frequency (set marker index to minimum index)
	STOP(1)	Marker -> Stop frequency (set marker index to maximum index)
	CENTer (2)	Marker -> Center frequency (set marker index to center index)
	STEP(3)	Marker -> Step frequency (set marker frequency to step frequency)

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set all markers OFF

ViStatus _VI_FUNC CySAn_SetAllMarkerOff(ViSession Vi)

Function Usage:

Turn off all markers in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set marker X value

ViStatus _VI_FUNC CySAn_SetMarkerx(ViSession Vi, ViInt32 Index, ViReal64 MarkerX)

Function Usage:

Set marker X value in the current mode. X may be negative when the marker is a delta marker.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

MarkerX

Marker X value.

Instrument mode	Parameter unit
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Real-time spectrum	Hz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker X value

ViStatus _VI_FUNC CySAn_GetMarkerx(ViSession Vi, ViInt32 Index, ViReal64* MarkerX)

Function Usage:

Set marker X value in the current mode. X may be negative when the marker is a delta marker.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

MarkerX

Marker X value.

Instrument mode	Parameter unit
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Real-time spectrum	Hz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker Y value

ViStatus _VI_FUNC CySAn_GetMarkerY(ViSession Vi, ViInt32 Index, ViReal64* Val)

Function Usage:

Query marker Y value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Val

Marker Y value in dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Search

ViStatus _VI_FUNC CySAn_SetMarkerSearch(ViSession Vi, ViInt32 Type, ViInt32 Index)

Function Usage:

Move the marker in the current mode to the position of Maximum, Minimum, Peak, Sub Peak, Left Peak, and Right Peak.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Type

Search Type

PEAK - peak value

MAXimum - max value

MINimum - min value

PNEXt Sub Peak

PLEFt Left Peak

PRIGht Right Peak

Index

Marker index, optional 1~6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set marker counter ON/OFF

ViStatus _VI_FUNC CySAn_SetFreqCounterEnabled(ViSession Vi, ViInt32 Index, ViBoolean Enabled)

Function Usage:

Set marker counter to On/Off in the current mode, and the set marker will be switched to the Normal state first.

Note: Only one marker counter can be turned on currently.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Enabled

Counter ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker counter ON/OFF

ViStatus _VI_FUNC CySAn_GetFreqCounterEnabled(ViSession Vi, ViInt32 Index, ViBoolean* Enabled)

Function Usage:

Query the marker On/Off state in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Enabled

Counter ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker counter frequency

ViStatus _VI_FUNC CySAn_QueryFrequencyCounter(ViSession Vi, ViInt32 Index, ViReal64* Val)

Function Usage:

Query counter frequency (**Invalid when counter is not on or not counting yet**).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Val

Counter frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set noise marker ON/OFF

ViStatus _VI_FUNC CySAn_SetNoiseMarkerOn(ViSession Vi, ViInt32 Index, ViBoolean NoiseMarker)

Function Usage:

Set the noise marker to On/Off in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

NoiseMarker

Noise marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query noise marker ON/OFF

ViStatus _VI_FUNC CySAn_GetNoiseMarkerOn(ViSession Vi, ViInt32 Index, ViBoolean* NoiseMarker)

Function Usage:

Query the noise marker On/Off status in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

NoiseMarker

Noise marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set peak tracking ON/OFF

ViStatus _VI_FUNC CySAn_SetPeakTrackOn(ViSession Vi, ViBoolean PeakTrack)

Function Usage:

Set peak tracking ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PeakTrack

Peak tracking ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query peak tracking ON/OFF

ViStatus _VI_FUNC CySAn_GetPeakTrackOn(ViSession Vi, ViBoolean* PeakTrack)

Function Usage:

Query the peak tracking ON/OFF in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PeakTrack

Peak tracking ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set sweep mode

ViStatus _VI_FUNC CySAn_SetSweepMode(ViSession Vi, ViBoolean SweepMode)

Function Usage:

Set sweep mode: linear sweep or list sweep.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepMode

Sweep mode: LIN(0) for linear sweep, and LIST(1) for list sweep.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query sweep mode

ViStatus _VI_FUNC CySAn_GetSweepMode(ViSession Vi, ViBoolean* SweepMode)

Function Usage:

Query sweep mode: linear sweep or list sweep. The user can edit the list segment to observe the signal of multiple sweep segments.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepMode

Sweep mode: LIN(0) for linear sweep, and LIST(1) for list sweep.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set fast sweep ON/OFF

ViStatus _VI_FUNC CySAn_SetFastSweep(ViSession Vi, ViBoolean FastSweep)

Function Usage:

Set fast sweep ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FastSweep

Sweep mode: OFF(0) for default, and ON(1) for fast.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query fast sweep ON/OFF

ViStatus _VI_FUNC CySAn_GetFastSweep(ViSession Vi, ViBoolean* FastSweep)

Function Usage:

Query fast sweep ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FastSweep

Sweep mode: OFF(0) for default, and ON(1) for fast.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set sweep points

ViStatus _VI_FUNC CySAn_SetSweepPoint(ViSession Vi, ViInt32 SweepPoint)

Function Usage:

Set number of sweep points

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepPoint

The number of sweep points, which can be set to 201, 501, 1001, 2001 and 4001.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query sweep points

ViStatus _VI_FUNC CySAn_GetSweepPoint(ViSession Vi, ViInt32* SweepPoint)

Function Usage:

QuerySwpPoints.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepPoint

The number of sweep points, which can be set to 201, 501, 1001, 2001 and 4001.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set sweep time

ViStatus _VI_FUNC CySAn_SetSweepTime(ViSession Vi, ViReal64 SweepTime)

Function Usage:

Set sweep time in the current mode. sweep time is the time required for the local oscillator tuning through the selected frequency interval. Sweep time directly affects the time required to complete a test, which does not include the dead time between the completion of a sweep and the start of the next sweep. Sweep time usually varies with Span, RBW and VBW. Sweep time cannot be set when RBW<=1kHz in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTime

Time (ms).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query sweep time

ViStatus _VI_FUNC CySAn_GetSweepTime(ViSession Vi, ViReal64* SweepTime)

Function Usage:

Query the sweep time in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTime

Time (ms).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set auto sweep time ON/OFF

ViStatus _VI_FUNC CySAn_SetSweepTimeAuto(ViSession Vi, ViBoolean SweepTimeAuto)

Function Usage:

Set sweep time to auto On/Off. When set to On, the instrument will adopt the sweep speed as high as possible; or you can manually increase the sweep time to meet some specific measurement requirements. Sweep time set manually must be > = automatic sweep time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTimeAuto

Automatic ON/OFF of sweep time in linear sweep mode.

OFF(0) manual sweep time

ON(1) auto sweep time

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query auto sweep time ON/OFF

ViStatus _VI_FUNC CySAn_GetSweepTimeAuto(ViSession Vi, ViBoolean* SweepTimeAuto)

Function Usage:

Query sweep time auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTimeAuto

Automatic ON/OFF of sweep time in linear sweep mode.

OFF(0) manual sweep time

ON(1) auto sweep time

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - List Editor - Add default segment to list

ViStatus _VI_FUNC CySAn_AddDefaultListSeg(ViSession Vi)

Function Usage:

Add a default sweep segment to the list editor in the current mode.

Start Freq	1GHz
Stop Freq	2GHz
RBW	1MHz
VBW	30kHz
Sweep Points	201
ON/OFF	Off

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - List Edit - Delete segment from list

ViStatus _VI_FUNC CySAn_DeleteListSeg(ViSession Vi, ViInt32 Index)

Function Usage:

Delete segment from the list editor in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Segment index

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - List Editor - Clear List

ViStatus _VI_FUNC CySAn_DeleteAllListSeg(ViSession Vi)

Function Usage:

Delete all segments from the list editor in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - List Editor - Add segment

ViStatus _VI_FUNC CySAn_AddListSeg(ViSession Vi, ViReal64 StartFreq, ViReal64 StopFreq, ViInt32 SweepPoints, ViReal64 RBW, ViReal64 VBW, ViBoolean OnOff)

Function Usage:

Add a segment to the list editor in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

StartFreq

Start frequency (0~9.1GHz).

StopFreq

Stop frequency (0~9.1GHz).

SweepPoints

Sweep points (201~1001).

RBW

Resolution bandwidth (1~40MHz).

VBW

Video bandwidth (0~10MHz).

OnOff

ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - List Editor - Edit segment

ViStatus _VI_FUNC CySAn_EditListSeg(ViSession Vi, ViInt32 Index, ViReal64 StartFreq, ViReal64 StopFreq, ViReal64 SweepPoints, ViReal64 RBW, ViReal64 VBW, ViBoolean OnOff)

Function Usage:

Add a segment to the list editor in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Segment index

StartFreq

Start frequency (0~9.1GHz).

StopFreq

Stop frequency (0~9.1GHz).

SweepPoints

Sweep points (201~1001).

RBW

Resolution bandwidth (1~40MHz).

VBW

Video bandwidth (1~40MHz).

OnOff

ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set trigger mode

ViStatus _VI_FUNC CySAn_SetTrigSource(ViSession Vi, ViInt32 TrigSource)

Function Usage:

Set trigger mode: free trigger, video trigger and external trigger.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TrigSource

Trigger type.

FREE(0) Free trigger

VIDEO(1) Video trigger

EXTR(2) External trigger

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query trigger mode

ViStatus _VI_FUNC CySAn_GetTrigSource(ViSession Vi, ViInt32* TrigSource)

Function Usage:

Query trigger mode: free trigger, video trigger and external trigger.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TrigSource

Trigger type.

FREE(0) Free trigger

VIDEO(1) Video trigger

EXTR(2) External trigger

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set video trigger level

ViStatus _VI_FUNC CySAn_SetVideoTriggerLevel(ViSession Vi, ViReal64 Level)

Function Usage:

Set video trigger level.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Level

Video trigger level (dBm), in the range of -150dBm ~30dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query video trigger level

ViStatus _VI_FUNC CySAn_GetVideoTriggerLevel(ViSession Vi, ViReal64* Level)

Function Usage:

Query video trigger level.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Level

Video trigger level (dBm), in the range of -150dBm~30dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set external trigger slope

ViStatus _VI_FUNC CySAn_SetExternalTriggerSlope(ViSession Vi, ViInt32 Slope)

Function Usage:

Set external trigger slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Slope

External trigger slope: POS(0) means positive, and NEG(1) means negative.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query external trigger slope

ViStatus _VI_FUNC CySAn_GetExternalTriggerSlope(ViSession Vi, ViInt32* Slope)

Function Usage:

Query external trigger slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Slope

External trigger slope: POS(0) means positive, and NEG(1) means negative.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set external trigger delay

ViStatus _VI_FUNC CySAn_SetTriggerDelay(ViSession Vi, ViReal64 Delay)

Function Usage:

Set external trigger delay.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Delay

Trigger delay (μs), in the range of 1μs~500ms.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query external trigger delay

ViStatus _VI_FUNC CySAn_GetTriggerDelay(ViSession Vi, ViReal64* Delay)

Function Usage:

Query external trigger delay.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Delay

Trigger delay (μs).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate delay

ViStatus _VI_FUNC CySAn_SetGateDelay(ViSession Vi, ViReal64 GateDelay)

Function Usage:

Set time gate delay.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateDelay

Delay time, in the range of 0s~20s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate delay

ViStatus _VI_FUNC CySAn_GetGateDelay(ViSession Vi, ViReal64* GateDelay)

Function Usage:

Query time gate delay.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateDelay

Delay time, in the range of 0s~20s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate length

ViStatus _VI_FUNC CySAn_SetGateLength(ViSession Vi, ViReal64 GateLength)

Function Usage:

Set time gate length.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateLength

Time gate length, in the range of 1 μs~20s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate length

ViStatus _VI_FUNC CySAn_GetGateLength(ViSession Vi, ViReal64* GateLength)

Function Usage:

Query time gate length.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateLength

Time gate length, in the range of 1 μs~20s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate edge slope

ViStatus _VI_FUNC CySAn_SetGateSlope(ViSession Vi, ViBoolean GateSlope)

Function Usage:

Set time gate edge slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateSlope

Edge slope: FALLING(0) for negative slope, and RISING(1) means positive slope.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate edge slope

ViStatus _VI_FUNC CySAn_GetGateSlope(ViSession Vi, ViBoolean* GateSlope)

Function Usage:

Query time gate edge slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateSlope

Edge slope: FALLING(0) for negative slope, and RISING(1) means positive slope.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate state

ViStatus _VI_FUNC CySAn_SetGateState(ViSession Vi, ViBoolean GateState)

Function Usage:

Set time gate state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateState

Time gate state: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate state

ViStatus _VI_FUNC CySAn_GetGateState(ViSession Vi, ViBoolean* GateState)

Function Usage:

Query time gate state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateState

Time gate state: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate trigger source

ViStatus _VI_FUNC CySAn_SetGateTrigger(ViSession Vi, ViBoolean GateTrigger)

Function Usage:

Set time gate trigger source.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateTrigger

Trigger source: EXT(0) for external, and GPS(1) for GPS.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate trigger source

ViStatus _VI_FUNC CySAn_GetGateTrigger(ViSession Vi, ViBoolean* GateTrigger)

Function Usage:

Query time gate trigger source.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateTrigger

Trigger source: EXT(0) for external, and GPS(1) for GPS.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate view ON/OFF

ViStatus _VI_FUNC CySAn_SetGateViewOn(ViSession Vi, ViBoolean GateViewOn)

Function Usage:

Set time gate view ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateViewOn

View ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate view ON/OFF

ViStatus _VI_FUNC CySAn_GetGateViewOn(ViSession Vi, ViBoolean* GateViewOn)

Function Usage:

Query time gate view ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GateViewOn

View ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate view time

ViStatus _VI_FUNC CySAn_SetViewTime(ViSession Vi, ViReal64 ViewTime)

Function Usage:

Set time gate view time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ViewTime

View time, in the range of 3.472μs~6000s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate view time

ViStatus _VI_FUNC CySAn_GetViewTime(ViSession Vi, ViReal64* ViewTime)

Function Usage:

Query time gate view time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ViewTime

View time, in the range of 3.472μs~6000s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate spectrum RBW

ViStatus _VI_FUNC CySAn_SetSpectrumRBW(ViSession Vi, ViReal64 SpectrumRBW)

Function Usage:

Set time gate spectrum RBW

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumRBW

Time gate spectrum RBW, in the range of 1.5kHz~750kHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate spectrum RBW

ViStatus _VI_FUNC CySAn_GetSpectrumRBW(ViSession Vi, ViReal64* SpectrumRBW)

Function Usage:

Query time gate spectrum RBW

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumRBW

Time gate spectrum RBW, in the range of 1.5kHz~750kHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate spectrum interval

ViStatus _VI_FUNC CySAn_SetSpectrumInteval(ViSession Vi, ViInt32 SpectrumInteval)

Function Usage:

Set time gate spectrum interval

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumInteval

Spectrum interval, the value can be 10(0)、20(1)、50(2)、100(3)、200(4)、500(5)、1000(6).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate spectrum interval

ViStatus _VI_FUNC CySAn_GetSpectrumInteval(ViSession Vi, ViInt32* SpectrumInteval)

Function Usage:

Query time gate spectrum interval

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumInteval

Spectrum interval, the value can be 10(0)、20(1)、50(2)、100(3)、200(4)、500(5)、1000(6).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate spectrum ON/OFF

ViStatus _VI_FUNC CySAn_SetSpectrumOn(ViSession Vi, ViBoolean SpectrumOn)

Function Usage:

Set time gate spectrum ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumOn

Spectrum ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate spectrum ON/OFF

ViStatus _VI_FUNC CySAn_GetSpectrumOn(ViSession Vi, ViBoolean* SpectrumOn)

Function Usage:

Query time gate spectrum ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumOn

Spectrum ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate spectrum RBW automatic ON/OFF

ViStatus _VI_FUNC CySAn_SetSpectrumRBWAuto(ViSession Vi, ViBoolean SpectrumRBWAuto)

Function Usage:

Set time gate spectrum RBW automatic ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumRBWAuto

Spectrum RBW automatic ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate spectrum RBW automatic ON/OFF

ViStatus _VI_FUNC CySAn_GetSpectrumRBWAuto(ViSession Vi, ViBoolean* SpectrumRBWAuto)

Function Usage:

Query time gate spectrum RBW automatic ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumRBWAuto

Spectrum RBW automatic ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Set time gate spectrum span

ViStatus _VI_FUNC CySAn_SetSpectrumSpan(ViSession Vi, ViReal64 SpectrumSpan)

Function Usage:

Set time gate spectrum span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumSpan

Time gate spectrum span: in the range of 1MHz~200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Time Gate - Query time gate spectrum span

ViStatus _VI_FUNC CySAn_GetSpectrumSpan(ViSession Vi, ViReal64* SpectrumSpan)

Function Usage:

Get time gate spectrum span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrumSpan

Time gate spectrum span: in the range of 1MHz~200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Set average ON/OFF

ViStatus _VI_FUNC CySAn_SetAverageEnabled(ViSession Vi, ViBoolean AverageEnabled)

Function Usage:

Set averaging to On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageEnabled

Average ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Query average ON/OFF

ViStatus _VI_FUNC CySAn_GetAverageEnabled(ViSession Vi, ViBoolean* AverageEnabled)

Function Usage:

Query average On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageEnabled

Average ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Set average count

ViStatus _VI_FUNC CySAn_SetAverageCount(ViSession Vi, ViInt32 AverageCount)

Function Usage:

Setting averaging times.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageCount

Average count, in the range of 1~1000.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Query average count

ViStatus _VI_FUNC CySAn_GetAverageCount(ViSession Vi, ViInt32* AverageCount)

Function Usage:

Query averaging count.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageCount

Average count, in the range of 1~1000.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Clear average

ViStatus _VI_FUNC CySAn_ClearAverage(ViSession Vi)

Function Usage:

Count current averaging from 0.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Query current average count

ViStatus _VI_FUNC CySAn_GetCurrentAverage(ViSession Vi, ViInt32* Val)

Function Usage:

Query current average count.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Averaged count.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Set detector type

ViStatus _VI_FUNC CySAn_SetDetectorType(ViSession Vi, ViInt32 DetectorType)

Function Usage:

Set detection type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorType

Detection type

Spectrum Analysis, Interference Analysis	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Sample
	NORMal(3)	Normal (Rosenfeld)
	AVERage(4)	Average
	RMS(5)	Rms
Orientation analysis	AVERage(0)	Average
	POSitive(1)	Peak
	SAMPle(2)	Real-time
Real-time spectrum	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Smp
	AVERage(4)	Average

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Query detector type

ViStatus _VI_FUNC CySAn_GetDetectorType(ViSession Vi, ViInt32* DetectorType)

Function Usage:

Query detection type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorType

Detection type

Spectrum Analysis, Interference Analysis	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Sample
	NORMal(3)	Normal (Rosenfeld)
	AVERage(4)	Average
	RMS(5)	Rms
Orientation analysis	AVERage(0)	Average
	POSitive(1)	Peak
	SAMPle(2)	Real-time
Real-time spectrum	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Smp
	AVERage(4)	Average

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Set auto detector ON/OFF

ViStatus _VI_FUNC CySAn_SetDetectorTypeAuto(ViSession Vi, ViBoolean DetectorTypeAuto)

Function Usage:

Set detection to auto On/Off. In auto detection mode, the instrument will automatically select detection type according to different measurements..

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorTypeAuto

Automatic detector ON/OFF: OFF (0) for manual, and (1) for automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection- Query auto detector ON/OFF

ViStatus _VI_FUNC CySAn_GetDetectorTypeAuto(ViSession Vi, ViBoolean* DetectorTypeAuto)

Function Usage:

Query detection auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorTypeAuto

Automatic detector ON/OFF: OFF (0) for manual, and (1) for automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Set trace status

ViStatus _VI_FUNC CySAn_SetTraceType(ViSession Vi, ViConstString TraceName, ViInt32 TraceType)

Function Usage:

Set trace status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

TraceType

Trace status.

CLRW(0) indicates refresh trace

MAXH(1) indicates maximum hold

MINH(2) indicates minimum hold

VIEW(3) indicates to hold trace

BLANk(4) indicates to hide trace

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Query trace status

ViStatus _VI_FUNC CySAn_GetTraceType(ViSession Vi, ViConstString Trace, ViInt32* TraceType)

Function Usage:

Query trace status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

TraceType

Trace status.

CLRW(0) indicates refresh trace

MAXH(1) indicates maximum hold

MINH(2) indicates minimum hold

VIEW(3) indicates to hold trace

BLANk(4) indicates to hide trace

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Data - Query trace data

ViStatus _VI_FUNC CySAn_FetchTraceY(ViSession Vi, ViConstString TraceName, ViInt32 ArrayLength, ViInt32* ActualPoints, ViReal64 Amplitude[])

Function Usage:

Query trace data in the spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

ArrayLength

Length of trace data received.

ActualPoints

Points

Amplitude[]

Array of trace data storage.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set audio alarm ON/OFF

ViStatus _VI_FUNC CySAn_SetLimitBeep(ViSession Vi, ViBoolean Beep)

Function Usage:

Set limit alarm to on/off. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every

sweep.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Beep

Audio alarm ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query audio alarm ON/OFF

ViStatus _VI_FUNC CySAn_GetLimitBeep(ViSession Vi, ViBoolean* Beep)

Function Usage:

Query limit alarm on/off status. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every sweep.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Beep

Audio alarm ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set lower limit display ON/OFF

ViStatus _VI_FUNC CySAn_SetLowLimitDisplayOn(ViSession Vi, ViBoolean Display)

Function Usage:

Set lower limit display ON/OFF

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Lower limit display ON/OFF: OFF (0) for display off, and ON (1) for display on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query lower limit display ON/OFF

ViStatus _VI_FUNC CySAn_GetUpLimitDisplayOn(ViSession Vi, ViBoolean* Display)

Function Usage:

Query lower limit display ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Lower limit display ON/OFF: OFF (0) for display off, and ON (1) for display on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set upper limit display ON/OFF

ViStatus _VI_FUNC CySAn_SetUpLimitDisplayOn(ViSession Vi, ViBoolean Display)

Function Usage:

Set upper limit display ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Upper limit display ON/OFF: OFF (0) for display off, and ON (1) for display on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query upper limit display ON/OFF

ViStatus _VI_FUNC CySAn_GetUpLimitDisplayOn(ViSession Vi, ViBoolean* Display)

Function Usage:

Query upper limit display ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Upper limit display ON/OFF: OFF (0) for display off, and ON (1) for display on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set lower limit test ON/OFF

ViStatus _VI_FUNC CySAn_SetLowLimitTestOn(ViSession Vi, ViBoolean Test)

Function Usage:

Set lower limit test to On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Lower limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query lower limit test ON/OFF

ViStatus _VI_FUNC CySAn_GetLowLimitTestOn(ViSession Vi, ViBoolean* Test)

Function Usage:

Query lower limit test On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Lower limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set upper limit test ON/OFF

ViStatus _VI_FUNC CySAn_SetUpLimitTestOn(ViSession Vi, ViBoolean Test)

Function Usage:

Set upper limit test On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Upper limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query upper limit test ON/OFF

ViStatus _VI_FUNC CySAn_GetUpLimitTestOn(ViSession Vi, ViBoolean* Test)

Function Usage:

Query upper limit test On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Upper limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set lower limit margin

ViStatus _VI_FUNC CySAn_SetLowLimitMargin(ViSession Vi, ViReal64 Margin)

Function Usage:

Set lower limit margin value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Margin

Margin value (0dB~40dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query lower limit margin

ViStatus _VI_FUNC CySAn_GetLowLimitMargin(ViSession Vi, ViReal64* Margin)

Function Usage:

Query lower limit margin value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Margin

Margin value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set upper limit margin

ViStatus _VI_FUNC CySAn_SetUpLimitMargin(ViSession Vi, ViReal64 Margin)

Function Usage:

Set upper limit margin value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Margin

Margin value (-40dB~0dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query upper limit margin

ViStatus _VI_FUNC CySAn_GetUpLimitMargin(ViSession Vi, ViReal64* Margin)

Function Usage:

Query upper limit margin value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Margin

Margin value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Add lower limit default point

ViStatus _VI_FUNC CySAn_AddLowLimitPoint(ViSession Vi)

Function Usage:

Add lower limit default point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete lower limit current point

ViStatus _VI_FUNC CySAn_DeleteLowLimitCurrentPoint(ViSession Vi)

Function Usage:

Delete lower limit current point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete all lower limit points

ViStatus _VI_FUNC CySAn_DeleteLowLimitAllPoint(ViSession Vi)

Function Usage:

Delete all lower limit edit points.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Edit lower limit point

ViStatus _VI_FUNC CySAn_EditLowLimitPoint(ViSession Vi, ViInt32 Index, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Set lower limit edit point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Limit point index.

Freq

Frequency value (Hz) (0~9.1GHz).

Amp

Amplitude value (dBm) (-174~50 dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Add upper limit default point

ViStatus _VI_FUNC CySAn_AddUpLimitPoint(ViSession Vi)

Function Usage:

Add upper limit default point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete upper limit current point

ViStatus _VI_FUNC CySAn_DeleteUpLimitCurrentPoint(ViSession Vi)

Function Usage:

Delete upper limit current point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete all upper limit points

ViStatus _VI_FUNC CySAn_DeleteUpLimitAllPoint(ViSession Vi)

Function Usage:

Delete all upper limit edit points.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Edit upper limit point

ViStatus _VI_FUNC CySAn_EditUpLimitPoint(ViSession Vi, ViInt32 Index, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Set upper limit edit point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Limit point index.

Freq

Frequency value (Hz) (0~9.1GHz).

Amp

Amplitude value (dBm) (-174~50 dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set function measurement

ViStatus _VI_FUNC CySAn_SetSAMeasureType(ViSession Vi, ViInt32 SAMeasureType)

Function Usage:

Set the function measurement type, or set it directly through the function measurement ON/OFF. Only one function measurement can exist at a time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SAMeasureType

Function measurement type.

Set Parameter	Measurement type
NONE(0)	Close Measurement
FST(1)	Field Strength Meter

CHP(2)	Ch Power
OBW(3)	OBW
ACPR(4)	Adjacent channel power
DEMOD(5)	Audio demodulation
SEM(6)	Spurious emission mask
CNR(7)	Carrier-to-noise ratio measurement
IQCAP(8)	IQ capture
OUTMAP(9)	Outdoor map
INMAP(10)	Indoor map
SE(11)	Spurious emission mask
HARM(12)	Harmonic distortion
MACPR(13)	Multi-carrier adjacent channel power

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query function measurement

ViStatus _VI_FUNC CySAn_GetSAMeasureType(ViSession Vi, ViInt32* SAMeasureType)

Function Usage:

Query function measurement type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SAMeasureType

Function measurement type.

Set Parameter	Measurement type
NONE(0)	Close Measurement
FST(1)	Field Strength Meter
CHP(2)	Ch Power
OBW(3)	OBW
ACPR(4)	Adjacent channel power
DEMOD(5)	Audio demodulation

SEM(6)	Spurious emission mask
CNR(7)	Carrier-to-noise ratio measurement
IQCAP(8)	IQ capture
OUTMAP(9)	Outdoor map
INMAP(10)	Indoor map
SE(11)	Spurious emission mask
HARM(12)	Harmonic distortion
MACPR(13)	Multi-carrier adjacent channel power

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Turn off measurement

ViStatus _VI_FUNC CySAn_MeasureOff(ViSession Vi)

Function Usage:

Turn off current function measurement and switch to normal spectrum measurement.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query measurement results

ViStatus _VI_FUNC CySAn_GetSAMeasureData(ViSession Vi, ViInt32 ValBufferSize, ViChar Val[])

Function Usage:

Query measurement results.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Measurement results.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Turn off antenna factor

ViStatus _VI_FUNC CySAn_ClearAntenna(ViSession Vi)

Function Usage:

Turn off antenna factor loading and set it to no-antenna factor state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Set field strength ON/OFF

ViStatus _VI_FUNC CySAn_SetFstOn(ViSession Vi, ViBoolean FstOn)

Function Usage:

Set the field strength function measurement ON/OFF, or turn on by the function **CySAn_SetInstMode()(other measurement functions will be disabled if this function is enabled)** .

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FstOn

Field strength ON/OFF: OFF (0) for field strength off, and ON (1) for field strength on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Query field strength ON/OFF

ViStatus _VI_FUNC CySAn_GetFstOn(ViSession Vi, ViBoolean* FstOn)

Function Usage:

Query field strength function measurement On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FstOn

Field strength ON/OFF: OFF (0) for field strength off, and ON (1) for field strength on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Add default point

ViStatus _VI_FUNC CySAn_AddAntennaDefaultPoint(ViSession Vi)

Function Usage:

Edit antenna factor and add default point. Frequency: 1GHz Antenna factor value: 0dB

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Delete point

ViStatus _VI_FUNC CySAn_DeletePoint(ViSession Vi, ViInt32 Index)

Function Usage:

Edit antenna factor and delete point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Point index.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Edit point

ViStatus _VI_FUNC CySAn_EditAntennaPoint(ViSession Vi, ViInt32 Index, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Edit antenna factor and edit point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Point index.

Freq

Frequency (0~9.1GHz).

Amp

Antenna factor value (-200~200dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Add point

ViStatus _VI_FUNC CySAn_AddAntennaPoint(ViSession Vi, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Edit antenna factor and add point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Freq

Frequency (0~9.1GHz).

Amp

Antenna factor value (-200~200dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Channel power - Set channel power ON/OFF

ViStatus _VI_FUNC CySAn_SetChannelPowerOn(ViSession Vi, ViBoolean ChannelPowerOn)

Function Usage:

Set channel power ON/OFF, or turn on by the function**CySAn_SetInstMode()(other measurement functions will be disabled if this function is enabled)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelPowerOn

Channel power ON/OFF: OFF (0) for channel power off, and ON (1) for channel power on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Channel power - Query channel power ON/OFF

ViStatus _VI_FUNC CySAn_GetChannelPowerOn(ViSession Vi, ViBoolean* ChannelPowerOn)

Function Usage:

Query channel power On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelPowerOn

Channel power ON/OFF: OFF (0) for channel power off, and ON (1) for channel power on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Channel power - Set channel power bandwidth

ViStatus _VI_FUNC CySAn_SetChannelBW(ViSession Vi, ViReal64 ChannelBW)

Function Usage:

Set channel power BW value of the channel power function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelBW

Channel power bandwidth (Hz), in the range of 5 Hz~9.1 GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Channel power - Query channel power bandwidth

ViStatus _VI_FUNC CySAn_GetChannelBW(ViSession Vi, ViReal64* ChannelBW)

Function Usage:

Query channel power BW value of the channel power function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelBW

Channel power bandwidth (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Channel power - Query channel power value

ViStatus _VI_FUNC CySAn_GetChannelPower(ViSession Vi, ViReal32* Val)

Function Usage:

Query the channel power value of the channel power function measurement in the Spectrum Analyzer mode (**which is valid when channel power is on and after a valid sweep**).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Power value (dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Channel power - Query channel power density

ViStatus _VI_FUNC CySAn_GetChannelPowerDensity(ViSession Vi, ViReal32* Val)

Function Usage:

Query the channel power density value of the channel power function measurement in the Spectrum Analyzer mode **(which is valid when channel power is on and after a valid sweep).**

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Power density value (dBm/Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Set occupied bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_SetOBWOn(ViSession Vi, ViBoolean OBWOn)

Function Usage:

Set occupied bandwidth function measurement ON/OFF, or turn on by the function **CySAn_SetInstMode()(other measurement functions will be disabled if this function is enabled)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

OBWOn

Occupied bandwidth ON/OFF: OFF (0) for OBW off, and ON (1) for OBW on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Query occupied bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_GetOBWOn(ViSession Vi, ViBoolean* OBWOn)

Function Usage:

Query OBW On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

OBWOn

Occupied bandwidth ON/OFF: OFF (0) for OBW off, and ON (1) for OBW on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Set measurement method

ViStatus _VI_FUNC CySAn_SetOBWMethod(ViSession Vi, ViBoolean Method)

Function Usage:

Set OBW measurement method. The percentage measurement method is to obtain the bandwidth of x% of the total power of the whole span, and the XdB measurement method is to obtain the bandwidths smaller than xdB on both sides of the max power value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Method

Measurement method of occupied bandwidth: PPOW(0) for percentage and XDB (1) for XdB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Query measurement method

ViStatus _VI_FUNC CySAn_GetOBWMethod(ViSession Vi, ViBoolean* Method)

Function Usage:

Query OBW measurement method.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Method

Measurement method of occupied bandwidth: PPOW(0) for percentage and XDB (1) for XdB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Set percentage

ViStatus _VI_FUNC CySAn_SetOBWPercent(ViSession Vi, ViReal64 Percent)

Function Usage:

OBWSetPercent

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Percent

Percentage value, in the range of 10.00% ~99.99%.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Query percentage

ViStatus _VI_FUNC CySAn_GetOBWPercent(ViSession Vi, ViReal64* Percent)

Function Usage:

Query OBW percentage value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Percent

Percentage value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Set XdB

ViStatus _VI_FUNC CySAn_SetOBWXdb(ViSession Vi, ViReal64 Xdb)

Function Usage:

Set the XdB value of OBW, which is valid in the XdB measurement method.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Xdb

XdB value (dB), in the range of -100.0dB~ -0.1dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Query XdB

ViStatus _VI_FUNC CySAn_GetOBWXdb(ViSession Vi, ViReal64* Xdb)

Function Usage:

Query XdB value of OBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Xdb

XdB value (dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Occupied bandwidth - Query occupied bandwidth

ViStatus _VI_FUNC CySAn_GetOBWResult(ViSession Vi, ViReal64* Val)

Function Usage:

Query OBW value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

OBW value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Set demodulation ON/OFF

ViStatus _VI_FUNC CySAn_SetTuneListenOn(ViSession Vi, ViBoolean TuneListenOn)

Function Usage:

Set AF/FM function measurement ON/OFF, or turn on by the function**CySAn_SetInstMode()(other measurement functions will be disabled if this function is enabled)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TuneListenOn

Demodulation ON/OFF: OFF (0) for demodulation off, and ON (1) for demodulation on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Query demodulation ON/OFF

ViStatus _VI_FUNC CySAn_GetTuneListenOn(ViSession Vi, ViBoolean* TuneListenOn)

Function Usage:

Query AM/FM ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TuneListenOn

Demodulation ON/OFF: OFF (0) for demodulation off, and ON (1) for demodulation on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Set demodulation mode

ViStatus _VI_FUNC CySAn_SetListenMode(ViSession Vi, ViBoolean ListenMode)

Function Usage:

Set demodulation mode. The intermittent mode is that the data is scanned after one screen and then demodulated intermittently for a period of time according to the demodulation time, then the data is scanned again after one screen and then demodulated intermittently for a period of time according to the demodulation time, and so on and so forth; the continuous mode is that the data is continuously demodulated after one screen is scanned and the data is not scanned anymore.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ListenMode

Demodulation mode: INTer(0) for intermittent mode, and CONT(1) for continuous mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Query demodulation mode

ViStatus _VI_FUNC CySAn_GetListenMode(ViSession Vi, ViBoolean* ListenMode)

Function Usage:

Query demodulation mode. The intermittent mode is that the data is scanned after one screen and then demodulated intermittently for a period of time according to the demodulation time, then the data is scanned again after one screen and then demodulated intermittently for a period of time according to the demodulation time, and so on and so forth; the continuous mode is that the data is continuously demodulated after one screen is scanned and the data is not scanned anymore.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ListenMode

Demodulation mode: INTer(0) for intermittent mode, and CONT(1) for continuous mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Set demodulation type

ViStatus _VI_FUNC CySAn_SetTuneListenDemodType(ViSession Vi, ViInt32 DemodType)

Function Usage:

Set demodulation type, selectable from FM, AM, upper sideband and lower sideband.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodType

Demodulation type

FM(0) Frequency modulation

AM(1) Amplitude modulation

USB(2) Upper sideband

LSB(3) Lower sideband

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Query demodulation type

ViStatus _VI_FUNC CySAn_GetTuneListenDemodType(ViSession Vi, ViInt32* DemodType)

Function Usage:

Query demodulation type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodType

Demodulation type

FM(0) Frequency modulation

AM(1) Amplitude modulation

USB(2) Upper sideband

LSB(3) Lower sideband

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Set demodulation time

ViStatus _VI_FUNC CySAn_SetListenTime(ViSession Vi, ViReal64 ListenTime)

Function Usage:

Set the demodulation time, which works when the demodulation mode is intermittent mode, and is the time in demodulation state after scanning.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ListenTime

Demodulation time (ms), in the range of 1μs~400s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Query demodulation time

ViStatus _VI_FUNC CySAn_GetListenTime(ViSession Vi, ViReal64* ListenTime)

Function Usage:

Query demodulation time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ListenTime

Demodulation time (ms).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Set volume

ViStatus _VI_FUNC CySAn_SetVolume(ViSession Vi, ViInt32 Volume)

Function Usage:

Set speaker volume for demodulation.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Volume

Demodulation volume (no unit), in the range of 0~100.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Audio demodulation - Query volume

ViStatus _VI_FUNC CySAn_GetVolume(ViSession Vi, ViInt32* Volume)

Function Usage:

Query speaker volume for demodulation.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Volume

Demodulation volume (no unit), in the range of 0~100.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Set adjacent channel power ratio ON/OFF

ViStatus _VI_FUNC CySAn_SetACPowerOn(ViSession Vi, ViBoolean ACPowerOn)

Function Usage:

Set adjacent channel power ratio ON/OFF, or turn on by the function**CySAn_SetInstMode()(other measurement functions will be disabled if this function is enabled)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ACPowerOn

Adjacent channel power ratio ON/OFF: OFF (0) for ACPR off, and ON (1) for ACPR on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query adjacent channel power ratio ON/OFF

ViStatus _VI_FUNC CySAn_GetACPowerOn(ViSession Vi, ViBoolean* ACPowerOn)

Query ACPR On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ACPowerOn

Adjacent channel power ratio ON/OFF: OFF (0) for ACPR off, and ON (1) for ACPR on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Set main channel bandwidth

ViStatus _VI_FUNC CySAn_SetMainChBW(ViSession Vi, ViReal64 MainChBW)

Function Usage:

Set the main channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MainChBW

Frequency value (in Hz), in the range of 300Hz~20 MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query main channel bandwidth

ViStatus _VI_FUNC CySAn_GetMainChBW(ViSession Vi, ViReal64* MainChBW)

Function Usage:

Query the main channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MainChBW

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Set adjacent channel bandwidth

ViStatus _VI_FUNC CySAn_SetAdjChBW(ViSession Vi, ViReal64 AdjChBW)

Function Usage:

Set the adjacent channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChBW

Frequency value (in Hz), in the range of 300Hz~20 MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query adjacent channel bandwidth

ViStatus _VI_FUNC CySAn_GetAdjChBW(ViSession Vi, ViReal64* AdjChBW)

Function Usage:

Query the adjacent channel bandwidth of the ACPR function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChBW

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Set channel spacing

ViStatus _VI_FUNC CySAn_SetSpace(ViSession Vi, ViReal64 Space)

Function Usage:

Set the channel spacing width of the ACPR function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Space

Frequency value (in Hz), in the range of 0Hz~45MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query channel spacing

ViStatus _VI_FUNC CySAn_GetSpace(ViSession Vi, ViReal64* Space)

Function Usage:

Query the channel spacing width of the ACPR function measurement in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Space

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Set limit test ON/OFF

ViStatus _VI_FUNC CySAn_SetLimitState(ViSession Vi, ViBoolean LimitState)

Function Usage:

Set adjacent channel power ratio limit test ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

LimitState

limit test ON/OFF: OFF (0) for limit test off, and ON (1) for limit test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query limit test ON/OFF

ViStatus _VI_FUNC CySAn_GetLimitState(ViSession Vi, ViBoolean* LimitState)

Function Usage:

Query adjacent channel power ratio limit test ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

LimitState

Adjacent channel power ratio ON/OFF: OFF (0) for ACPR off, and ON (1) for ACPR on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Set lower adjacent channel limit

ViStatus _VI_FUNC CySAn_SetLowACPRLimit(ViSession Vi, ViReal32 LowACPRLimit)

Function Usage:

Set lower adjacent channel limit value of adjacent channel power ratio.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

LowACPRLimit

Adjacent channel limit (dB), in the range of -200dB~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Measure - Adjacent channel power ratio - Query lower adjacent channel limit

ViStatus _VI_FUNC CySAn_GetLowACPRLimit(ViSession Vi, ViReal32* LowACPRLimit)

Function Usage:

Query lower adjacent channel limit value of adjacent channel power ratio.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

LowACPRLimit

Adjacent channel limit (dB), in the range of -200dB ~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Set upper adjacent channel limit

ViStatus _VI_FUNC CySAn_SetUpACPRLimit(ViSession Vi, ViReal32 UpACPRLimit)

Function Usage:

Set upper adjacent channel limit value of adjacent channel power ratio.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

UpACPRLimit

Adjacent channel limit (dB), in the range of -200dB~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query upper adjacent channel limit

ViStatus _VI_FUNC CySAn_GetUpACPRLimit(ViSession Vi, ViReal32* UpACPRLimit)

Function Usage:

Query upper adjacent channel limit value of adjacent channel power ratio.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

UpACPRLimit

Adjacent channel limit (dB), in the range of -200dB~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query upper adjacent channel power ratio

ViStatus _VI_FUNC CySAn_GetUpACPR(ViSession Vi, ViReal32* Val)

Function Usage:

Query upper adjacent channel power ratio of adjacent channel power ratio.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Adjacent channel power ratio (dBc).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Adjacent channel power ratio - Query lower adjacent channel power

ViStatus _VI_FUNC CySAn_GetLowACPR(ViSession Vi, ViReal32* Val)

Function Usage:

Query lower adjacent channel power ratio of adjacent channel power ratio.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Adjacent channel power ratio (dBc).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Set SEM ON/OFF

ViStatus _VI_FUNC CySAn_SetSEOn(ViSession Vi, ViBoolean SEOn)

Function Usage:

Set spurious emission mask ON/OFF, or turn on by the function **CySAn_SetInstMode()(other measurement functions will be disabled if this function is enabled)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SEOn

Spurious emission mask ON/OFF: OFF (0) for SEM off, and ON (1) for SEM on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Query SEM ON/OFF

ViStatus _VI_FUNC CySAn_GetSEOn(ViSession Vi, ViBoolean* SEOn)

Function Usage:

Query spurious emission mask ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SEOn

Spurious emission mask ON/OFF: OFF (0) for SEM off, and ON (1) for SEM on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Set current segment number of SEM

ViStatus _VI_FUNC CySAn_SetSESegID(ViSession Vi, ViInt32 SegID)

Function Usage:

Set current segment number of SEM.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegID

Current segment number, in the range of 1~5.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Query current segment number of SEM

ViStatus _VI_FUNC CySAn_GetSESegID(ViSession Vi, ViInt32* SegID)

Function Usage:

Query current segment number of SEM.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegID

Current segment number, in the range of 1~5.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Set SEM sweep mode

ViStatus _VI_FUNC CySAn_SetSESweepMode(ViSession Vi, ViBoolean SweepMode)

Function Usage:

Set SEM sweep mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepMode

Sweep mode: ONCE(0) for single segment, and ALL(1) for full segment.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Query SEM sweep mode

ViStatus _VI_FUNC CySAn_GetSESweepMode(ViSession Vi, ViBoolean* SweepMode)

Function Usage:

Query SEM sweep mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepMode

Sweep mode: ONCE(0) for single segment, and ALL(1) for full segment.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Set SEM sweep type

ViStatus _VI_FUNC CySAn_SetSESweepType(ViSession Vi, ViBoolean SweepType)

Function Usage:

Set SEM sweep type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepType

Sweep type: SINGLE(0) for sweep once, and CONT (1) for continuous sweep.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Query SEM sweep type

ViStatus _VI_FUNC CySAn_GetSESweepType(ViSession Vi, ViBoolean* SweepType)

Function Usage:

Query SEM sweep type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepType

Sweep type: SINGLE(0) for sweep once, and CONT (1) for continuous sweep.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Set reference channel bandwidth

ViStatus _VI_FUNC CySAn_SetSEMChannelBW(ViSession Vi, ViReal64 ChannelBW)

Function Usage:

Set SEM reference channel bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelBW

Frequency value (in Hz), in the range of 1kHz~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Query reference channel bandwidth

ViStatus _VI_FUNC CySAn_GetSEMChannelBW(ViSession Vi, ViReal64* ChannelBW)

Function Usage:

Query SEM reference channel bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelBW

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Set reference power type

ViStatus _VI_FUNC CySAn_SetRefPowerType(ViSession Vi, ViInt32 RefPowerType)

Function Usage:

Set SEM reference power type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RefPowerType

Reference power type: PEAK(0) for peak and CHANnel(1) for channel.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Measure - Spurious emission mask - Query reference power type

ViStatus _VI_FUNC CySAn_GetRefPowerType(ViSession Vi, ViInt32* RefPowerType)

Function Usage:

Query SEM reference power type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RefPowerType

Reference power type: PEAK(0) for peak and CHANnel(1) for channel.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Set SEM peak marker ON/OFF

ViStatus _VI_FUNC CySAn_SetPeakMarkerOn(ViSession Vi, ViBoolean PeakMarkerOn)

Function Usage:

Set SEM peak marker ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PeakMarkerOn

Peak marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Query SEM peak marker ON/OFF

ViStatus _VI_FUNC CySAn_GetPeakMarkerOn(ViSession Vi, ViBoolean* PeakMarkerOn)

Function Usage:

Query SEM peak marker ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PeakMarkerOn

Peak marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Query whether SEM passes the test

ViStatus _VI_FUNC CySAn_GetSEMFail(ViSession Vi, ViBoolean* Val)

Function Usage:

Query whether SEM passes the test.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

0 means pass, and 1 means fail.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Add default emission seg

ViStatus _VI_FUNC CySAn_AddDefaultEmesSeg(ViSession Vi)

Function Usage:

Add SEM default emission seg.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask - Add emission edit seg

ViStatus _VI_FUNC CySAn_AddEmesSeg(ViSession Vi, ViReal64 StartFreq, ViReal64 StopFreq, ViReal64 RBW, ViReal64 VBW, ViReal64 StartLimit, ViReal64 StopLimit)

Function Usage:

Add SEM emission edit seg.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

StartFreq

StartFreq(Hz),in the range of 0～9.1GHz。

StopFreq

StopFreq(Hz) ,in the range of 0～9.1GHz。

RBW

RBW(Hz) ,in the range of 1Hz～20MHz。

VBW

VBW(Hz) ,in the range of 1Hz～20MHz。

StartLimit

StartLimit(dBm) ,in the range of -174dBm～50dBm。

StopLimit

StopLimit(dBm) ,in the range of -174dBm～50dBm。

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask – Clear the list of emission seg

ViStatus _VI_FUNC CySAn_ClearEmesSeg(ViSession Vi)

Function Usage:

Clear the list of EMS emission seg.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask – Delete emission seg

ViStatus _VI_FUNC CySAn_DeleteEmesSeg(ViSession Vi, ViInt32 Index)

Function Usage:

Delete EMS emission seg.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Delete item index

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spurious emission mask – Edit emission seg

ViStatus _VI_FUNC CySAn_EditEmesSeg(ViSession Vi, ViInt32 Index, ViReal64 StartFreq, ViReal64 StopFreq, ViReal64 RBW, ViReal64 VBW, ViReal64 StartLimit, ViReal64 StopLimit)

Function Usage:

Edit SEM emission seg.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Edit seg index

StartFreq

StartFreq(Hz),in the range of 0～9.1GHz。

StopFreq

StopFreq(Hz) ,in the range of 0～9.1GHz。

RBW

RBW(Hz) ,in the range of 1Hz～20MHz。

VBW

VBW(Hz) ,in the range of 1Hz～20MHz。

StartLimit

StartLimit(dBm) ,in the range of -174dBm～50dBm。

StopLimit

StopLimit(dBm) ,in the range of -174dBm～50dBm。

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Set CNR ON/OFF

ViStatus _VI_FUNC CySAn_SetCNRatioOn(ViSession Vi, ViBoolean CNRatioOn)

Function Usage:

Set CNR ON/OFF, or turn on by the function**CySAn_SetInstMode()(other measurement functions will be disabled if this function is enabled)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CNRatioOn

CNR measurement ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Query CNR ON/OFF

ViStatus _VI_FUNC CySAn_GetCNRatioOn(ViSession Vi, ViBoolean* CNRatioOn)

Function Usage:

Query CNR ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CNRatioOn

CNR measurement ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Measure - Carrier-to-noise ratio - Set CNR carrier bandwidth

ViStatus _VI_FUNC CySAn_SetCNRCarrierBW(ViSession Vi, ViReal64 CarrierBW)

Function Usage:

Set CNR carrier bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CarrierBW

Frequency value (in Hz), in the range of 300Hz~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Query CNR carrier bandwidth

ViStatus _VI_FUNC CySAn_GetCNRCarrierBW(ViSession Vi, ViReal64* CarrierBW)

Function Usage:

Query CNR carrier bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CarrierBW

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Set CNR noise bandwidth

ViStatus _VI_FUNC CySAn_SetCNRNoiseBW(ViSession Vi, ViReal64 NoiseBW)

Function Usage:

Set CNR noise bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

NoiseBW

Frequency value (in Hz), in the range of 300Hz~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Query CNR noise bandwidth

ViStatus _VI_FUNC CySAn_GetCNRNoiseBW(ViSession Vi, ViReal64* NoiseBW)

Function Usage:

Query CNR noise bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

NoiseBW

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Set CNR frequency offset

ViStatus _VI_FUNC CySAn_SetCNSpace(ViSession Vi, ViReal64 CNSpace)

Function Usage:

Set CNR frequency offset.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CNSpace

Frequency value (in Hz), in the range of 0Hz ~100MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Query CNR frequency offset

ViStatus _VI_FUNC CySAn_GetCNSpace(ViSession Vi, ViReal64* CNSpace)

Function Usage:

Query CNR frequency offset.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CNSpace

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Carrier-to-noise ratio - Query CNR measurement results

ViStatus _VI_FUNC CySAn_GetCNRResult(ViSession Vi, ViReal32* Val)

Function Usage:

Query CNR measurement results.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Measurement results (dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set IQ capture ON/OFF

ViStatus _VI_FUNC CySAn_SetIQCaptureOn(ViSession Vi, ViBoolean IQCaptureOn)

Function Usage:

Set IQ capture ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IQCaptureOn

IQ capture measurement ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query IQ capture ON/OFF

ViStatus _VI_FUNC CySAn_GetIQCaptureOn(ViSession Vi, ViBoolean* IQCaptureOn)

Function Usage:

Query IQ capture ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IQCaptureOn

IQ capture measurement ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Start capture

ViStatus _VI_FUNC CySAn_StartCapture(ViSession Vi)

Function Usage:

Set IQ capture - Start capture.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Stop capture

ViStatus _VI_FUNC CySAn_StopCapture(ViSession Vi)

Function Usage:

Set IQ capture - Stop capture.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set capture time

ViStatus _VI_FUNC CySAn_SetCaptureTime(ViSession Vi, ViReal64 CaptureTime)

Function Usage:

Set IQ capture - capture time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CaptureTime

Capture time (us), in the range of 1μs~98.739ms.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query capture time

ViStatus _VI_FUNC CySAn_GetCaptureTime(ViSession Vi, ViReal64* CaptureTime)

Function Usage:

Query IQ capture - capture time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CaptureTime

IQ capture time (μs).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set IQ capture mode

ViStatus _VI_FUNC CySAn_SetIQCaptureMode(ViSession Vi, ViBoolean CaptureMode)

Function Usage:

Set IQ capture mode: single capture and continuous capture.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CaptureMode

IQ capture mode: SING(0) for single capture, and CONT (1) for continuous capture.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query IQ capture mode

ViStatus _VI_FUNC CySAn_GetIQCaptureMode(ViSession Vi, ViBoolean* CaptureMode)

Function Usage:

Query IQ capture mode: single capture and continuous capture.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CaptureMode

IQ capture mode: SING(0) for single capture, and CONT (1) for continuous capture.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set sampling rate

ViStatus _VI_FUNC CySAn_SetSampleRate(ViSession Vi, ViReal64 SampleRate)

Function Usage:

Set IQ capture sampling rate.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SampleRate

Sampling rate.

Capture sampling rate	Capture bandwidth
184.32MHz	120.00MHz
92.16MHz	60.00MHz
46.08MHz	30.00MHz
23.04MHz	15.00MHz
11.52MHz	7.50MHz

5.76MHz	3.75MHz
2.88MHz	1.875MHz
1.44MHz	937.50kHz
720.00kHz	468.75kHz
360.00kHz	234.375kHz
180.00kHz	117.188kHz
90.00kHz	58.594kHz
45.00kHz	29.297kHz
22.50kHz	14.648kHz
11.25kHz	7.324kHz
5.625kHz	3.662kHz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query sampling rate

ViStatus _VI_FUNC CySAn_GetSampleRate(ViSession Vi, ViReal64* SampleRate)

Function Usage:

Query IQ capture sampling rate.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SampleRate

Sampling rate.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set IQ capture save name

ViStatus _VI_FUNC CySAn_SetCaptureSaveFileName(ViSession Vi, ViConstString CaptureSaveFileName)

Function Usage:

Set IQ capture file storage name.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CaptureSaveFileName

File save name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query IQ capture save name

ViStatus _VI_FUNC CySAn_GetCaptureSaveFileName(ViSession Vi, ViInt32 CaptureSaveFileNameBufferSize, ViChar CaptureSaveFileName[])

Function Usage:

Set IQ capture file storage name.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CaptureSaveFileName[]

File save name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set trigger mode

ViStatus _VI_FUNC CySAn_SetTriggerType(ViSession Vi, ViBoolean TriggerType)

Function Usage:

Set trigger mode: free trigger and external trigger.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TriggerType

Trigger type: FREE(0) means free trigger, and EXTR(1) means external trigger.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query trigger mode

ViStatus _VI_FUNC CySAn_GetTriggerType(ViSession Vi, ViBoolean* TriggerType)

Function Usage:

Query trigger mode: free trigger and external trigger.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TriggerType

Trigger type: FREE(0) means free trigger, and EXTR(1) means external trigger.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Measure - IQ capture - Set external trigger slope

ViStatus _VI_FUNC CySAn_SetTriggerSlope(ViSession Vi, ViBoolean TriggerSlope)

Function Usage:

Set external trigger slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TriggerSlope

Trigger slope: POS(0) means positive, and NEG(1) means negative.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query external trigger slope

ViStatus _VI_FUNC CySAn_GetTriggerSlope(ViSession Vi, ViBoolean* TriggerSlope)

Function Usage:

Query external trigger slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TriggerSlope

Trigger slope: POS(0) means positive, and NEG(1) means negative.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set external trigger delay

ViStatus _VI_FUNC CySAn_SetExternalTriggerDelay(ViSession Vi, ViReal64 ExternalTriggerDelay)

Function Usage:

Set IQ capture external trigger delay.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ExternalTriggerDelay

Trigger delay (μs), in the range of 1μs~500ms.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query external trigger delay

ViStatus _VI_FUNC CySAn_GetExternalTriggerDelay(ViSession Vi, ViReal64*

ExternalTriggerDelay)

Function Usage:

Query IQ capture external trigger delay.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ExternalTriggerDelay

Trigger delay (μs), in the range of 1μs~500ms.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Set external trigger slope

ViStatus _VI_FUNC CySAn_SetTriggerSlope(ViSession Vi, ViBoolean TriggerSlope)

Function Usage:

Set IQ capture external trigger slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TriggerSlope

Trigger slope: POS(0) means positive, and NEG(1) means negative.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - IQ capture - Query external trigger slope

ViStatus _VI_FUNC CySAn_GetTriggerSlope(ViSession Vi, ViBoolean* TriggerSlope)

Function Usage:

Query IQ capture external trigger slope.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TriggerSlope

Trigger slope: POS(0) means positive, and NEG(1) means negative.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spectrum emission mask - Set SEM ON/OFF

ViStatus _VI_FUNC CySAn_SetSEMOn(ViSession Vi, ViBoolean SEMOn)

Function Usage:

Set spectrum emission mask ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SEMOn

Spectrum emission mask ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Spectrum emission mask - Query SEM ON/OFF

ViStatus _VI_FUNC CySAn_GetSEMOn(ViSession Vi, ViBoolean* SEMOn)

Function Usage:

Set spectrum emission mask ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SEMOn

Spectrum emission mask ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Set fundamental frequency

ViStatus _VI_FUNC CySAn_SetFundamentalFrequency(ViSession Vi, ViReal64 FundamentalFrequency)

Function Usage:

Set harmonic distortion fundamental frequency.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FundamentalFrequency

Fundamental frequency, in the range of 0~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Query fundamental frequency

ViStatus _VI_FUNC CySAn_GetFundamentalFrequency(ViSession Vi, ViReal64* FundamentalFrequency)

Function Usage:

Query harmonic distortion fundamental frequency.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FundamentalFrequency

Fundamental frequency, in the range of 0~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Set harmonic number

ViStatus _VI_FUNC CySAn_SetHarmonicNum(ViSession Vi, ViInt32 HarmonicNum)

Function Usage:

Query harmonic distortion harmonic number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicNum

harmonic number, in the range of 1~10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Query harmonic number

ViStatus _VI_FUNC CySAn_GetHarmonicNum(ViSession Vi, ViInt32* HarmonicNum)

Function Usage:

Query harmonic distortion harmonic number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicNum

harmonic number, in the range of 1 ~10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Set resolution bandwidth

ViStatus _VI_FUNC CySAn_SetHarmonicRBW(ViSession Vi, ViReal64 HarmonicRBW)

Function Usage:

Set harmonic distortion resolution bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicRBW

Resolution bandwidth, in the range of 1Hz~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Query resolution bandwidth

ViStatus _VI_FUNC CySAn_GetHarmonicRBW(ViSession Vi, ViReal64* HarmonicRBW)

Function Usage:

Query harmonic distortion resolution bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicRBW

Resolution bandwidth, in the range of 1Hz~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Set harmonic distortion status

ViStatus _VI_FUNC CySAn_SetHarmonicDistortionOn(ViSession Vi, ViBoolean HarmonicDistortionOn)

Function Usage:

Set harmonic distortion status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicDistortionOn

Harmonic distortion ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Query harmonic distortion status

ViStatus _VI_FUNC CySAn_GetHarmonicDistortionOn(ViSession Vi, ViBoolean* HarmonicDistortionOn)

Function Usage:

Query harmonic distortion status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicDistortionOn

Harmonic distortion ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Set harmonic distortion video bandwidth

ViStatus _VI_FUNC CySAn_SetHarmonicVBW(ViSession Vi, ViReal64 HarmonicVBW)

Function Usage:

Set harmonic distortion video bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicVBW

Video bandwidth, in the range of 1Hz~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Harmonic distortion - Query harmonic distortion video bandwidth

ViStatus _VI_FUNC CySAn_SetHarmonicVBW(ViSession Vi, ViReal64 HarmonicVBW)

Function Usage:

Query harmonic distortion video bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

HarmonicVBW

Video bandwidth, in the range of 1Hz~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set adjacent channel bandwidth

ViStatus _VI_FUNC CySAn_SetAdjChannelBW(ViSession Vi, ViReal64 AdjChannelBW)

Function Usage:

Set multi-carrier adjacent channel power - adjacent channel carrier bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChannelBW

Adjacent channel carrier bandwidth, in the range of 1kHz~200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query adjacent channel bandwidth

ViStatus _VI_FUNC CySAn_GetAdjChannelBW(ViSession Vi, ViReal64* AdjChannelBW)

Function Usage:

Query multi-carrier adjacent channel power - adjacent channel carrier bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChannelBW

Adjacent channel carrier bandwidth, in the range of 1kHz~200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set adjacent number

ViStatus _VI_FUNC CySAn_SetAdjChannel(ViSession Vi, ViInt32 AdjChannel)

Function Usage:

Set multi-carrier adjacent channel power - adjacent channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChannel

Adjacent channel number, in the range of 1~3.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query adjacent number

ViStatus _VI_FUNC CySAn_GetAdjChannel(ViSession Vi, ViInt32* AdjChannel)

Function Usage:

Query multi-carrier adjacent channel power - adjacent channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChannel

Adjacent channel number, in the range of 1~3.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set lower adjacent channel limit

ViStatus _VI_FUNC CySAn_SetAdjChLowLimit(ViSession Vi, ViReal64 AdjChLowLimit)

Function Usage:

Set multi-carrier adjacent channel power - lower adjacent channel limit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChLowLimit

Lower adjacent channel limit, in the range of -200dB~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query lower adjacent channel limit

ViStatus _VI_FUNC CySAn_GetAdjChLowLimit(ViSession Vi, ViReal64* AdjChLowLimit)

Function Usage:

Query multi-carrier adjacent channel power - lower adjacent channel limit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChLowLimit

Lower adjacent channel limit, in the range of -200dB~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set adjacent channel limit test state

ViStatus _VI_FUNC CySAn_SetAdjChLimitOn(ViSession Vi, ViBoolean AdjChLimitOn)

Function Usage:

Set multi-carrier adjacent channel power - adjacent channel limit test state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChLimitOn

Limit test ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query adjacent channel limit test state

ViStatus _VI_FUNC CySAn_GetAdjChLimitOn(ViSession Vi, ViBoolean* AdjChLimitOn)

Function Usage:

Query multi-carrier adjacent channel power - adjacent channel limit test state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChLimitOn

Limit test ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set upper adjacent channel limit

ViStatus _VI_FUNC CySAn_SetAdjChUpLimit(ViSession Vi, ViReal64 AdjChUpLimit)

Function Usage:

Set multi-carrier adjacent channel power - upper adjacent channel limit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChUpLimit

Upper adjacent channel limit, in the range of -200dB ~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query upper adjacent channel limit

ViStatus _VI_FUNC CySAn_GetAdjChUpLimit(ViSession Vi, ViReal64* AdjChUpLimit)

Function Usage:

Query multi-carrier adjacent channel power - upper adjacent channel limit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChUpLimit

Upper adjacent channel limit, in the range of -200dB ~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set adjacent channel offset

ViStatus _VI_FUNC CySAn_SetAdjChOffsetFreq(ViSession Vi, ViReal64 AdjChOffsetFreq)

Function Usage:

Set multi-carrier adjacent channel power - adjacent channel offset.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChOffsetFreq

Adjacent channel offset, in the range of 0Hz ~ 200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query adjacent channel offset

ViStatus _VI_FUNC CySAn_GetAdjChOffsetFreq(ViSession Vi, ViReal64* AdjChOffsetFreq)

Function Usage:

Query multi-carrier adjacent channel power - adjacent channel offset.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChOffsetFreq

Adjacent channel offset, in the range of 0Hz ~ 200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set carrier bandwidth

ViStatus _VI_FUNC CySAn_SetCarrierBW(ViSession Vi, ViReal64 CarrierBW)

Function Usage:

Set multi-carrier adjacent channel power - carrier bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CarrierBW

Carrier bandwidth, in the range of 1kHz ~ 200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query carrier bandwidth

ViStatus _VI_FUNC CySAn_GetCarrierBW(ViSession Vi, ViReal64* CarrierBW)

Function Usage:

Query multi-carrier adjacent channel power - carrier bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CarrierBW

Carrier bandwidth, in the range of 1kHz ~ 200MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set carrier frequency

ViStatus _VI_FUNC CySAn_SetCarrierFreq(ViSession Vi, ViReal64 CarrierFreq)

Function Usage:

Set multi-carrier adjacent channel power - carrier frequency.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CarrierFreq

Carrier frequency, in the range of 100MHz~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query carrier frequency

ViStatus _VI_FUNC CySAn_GetCarrierFreq(ViSession Vi, ViReal64* CarrierFreq)

Function Usage:

Query multi-carrier adjacent channel power - carrier frequency.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CarrierFreq

Carrier frequency, in the range of 100MHz~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set carrier ID

ViStatus _VI_FUNC CySAn_SetCurrentCarrier(ViSession Vi, ViInt32 CurrentCarrier)

Function Usage:

Set multi-carrier adjacent channel power - carrier ID.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CurrentCarrier

carrier ID, optional 1 or 2.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query carrier ID

ViStatus _VI_FUNC CySAn_GetCurrentCarrier(ViSession Vi, ViInt32* CurrentCarrier)

Function Usage:

Query multi-carrier adjacent channel power - carrier ID.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CurrentCarrier

carrier ID, optional 1 or 2.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Set adjacent channel power state

ViStatus _VI_FUNC CySAn_SetMACPROn(ViSession Vi, ViBoolean MACPROn)

Function Usage:

Set multi-carrier adjacent channel power state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MACPROn

Multi-carrier adjacent channel power ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Multi-carrier adjacent channel power - Query adjacent channel power state

ViStatus _VI_FUNC CySAn_GetMACPROn(ViSession Vi, ViBoolean* MACPROn)

Function Usage:

Query multi-carrier adjacent channel power state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MACPROn

Multi-carrier adjacent channel power ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set adjacent channel bandwidth

ViStatus _VI_FUNC CySAn_SetMAPAdjChBW(ViSession Vi, ViReal64 AdjChBW)

Function Usage:

Set adjacent channel bandwidth on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChBW

Adjacent channel bandwidth, in the range of 300Hz~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query adjacent channel bandwidth

ViStatus _VI_FUNC CySAn_SetMAPAdjChBW(ViSession Vi, ViReal64 AdjChBW)

Function Usage:

Query adjacent channel bandwidth on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChBW

Adjacent channel bandwidth, in the range of 300Hz ~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set lower limit of adjacent channel power ratio "good”

ViStatus _VI_FUNC CySAn_SetMAPACPRGood(ViSession Vi, ViReal64 ACPRGood)

Function Usage:

Set lower limit of adjacent channel power ratio "good”.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ACPRGood

Adjacent channel power ratio, in the range of -200dBm ~ 200dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query lower limit of adjacent channel power ratio "good”

ViStatus _VI_FUNC CySAn_GetMAPACPRGood(ViSession Vi, ViReal64* ACPRGood)

Function Usage:

Query lower limit of adjacent channel power ratio "good”.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ACPRGood

Adjacent channel power ratio, in the range of -200dBm~ 200dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set main channel bandwidth

ViStatus _VI_FUNC CySAn_SetMAPMainChBW(ViSession Vi, ViReal64 MainChBW)

Function Usage:

Set main channel bandwidth on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MainChBW

Main channel bandwidth, in the range of 300Hz ~20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query main channel bandwidth

ViStatus _VI_FUNC CySAn_GetMAPMainChBW(ViSession Vi, ViReal64* MainChBW)

Function Usage:

Query main channel bandwidth on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MainChBW

Main channel bandwidth, in the range of 300Hz~ 20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set adjacent channel limit

ViStatus _VI_FUNC CySAn_SetMAPAdjChOffset(ViSession Vi, ViReal64 AdjChOffset)

Function Usage:

Set adjacent channel limit on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChOffset

Adjacent channel limit, in the range of -200dB ~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query adjacent channel limit

ViStatus _VI_FUNC CySAn_GetMAPAdjChOffset(ViSession Vi, ViReal64* AdjChOffset)

Function Usage:

Query adjacent channel limit on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AdjChOffset

Adjacent channel limit, in the range of -200dB ~200dB.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set upper limit of adjacent channel power ratio "poor”

ViStatus _VI_FUNC CySAn_SetMAPACPRPoor(ViSession Vi, ViReal64 ACPRPoor)

Function Usage:

Set upper limit of adjacent channel power ratio "poor” on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ACPRPoor

Adjacent channel power ratio, in the range of -200dBm ~200dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query upper limit of adjacent channel power ratio "poor”

ViStatus _VI_FUNC CySAn_GetMAPACPRPoor(ViSession Vi, ViReal64* ACPRPoor)

Function Usage:

Query upper limit of adjacent channel power ratio "poor” on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ACPRPoor

Adjacent channel power ratio, in the range of -200dBm~ 200dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Measure - Map - Set channel spacing

ViStatus _VI_FUNC CySAn_SetMAPChSpacing(ViSession Vi, ViReal64 ChSpacing)

Function Usage:

Set channel spacing on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChSpacing

Channel spacing, in the range of 0Hz~ 45MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query channel spacing

ViStatus _VI_FUNC CySAn_GetMAPChSpacing(ViSession Vi, ViReal64* ChSpacing)

Function Usage:

Query channel spacing on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChSpacing

Channel spacing, in the range of 0Hz ~45MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set interference map test type

ViStatus _VI_FUNC CySAn_SetSAMapMeasureType(ViSession Vi, ViBoolean MapMeasureType)

Function Usage:

Set interference map test type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapMeasureType

Test type: RSSI(0) means RSSI and ACPR(1) means adjacent channel power ratio.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query interference map test type

ViStatus _VI_FUNC CySAn_GetSAMapMeasureType(ViSession Vi, ViBoolean*

MapMeasureType)

Function Usage:

Query interference map test type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapMeasureType

Test type: RSSI(0) means RSSI and ACPR(1) means adjacent channel power ratio.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set lower limit of RSSI "excellent”

ViStatus _VI_FUNC CySAn_SetMapRSSIExcellent(ViSession Vi, ViReal64 RSSIExcellent)

Function Usage:

Set lower limit of RSSI "excellent” on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIExcellent

RSSI value, in the range of -196~200dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query lower limit of RSSI "excellent”

ViStatus _VI_FUNC CySAn_GetMapRSSIExcellent(ViSession Vi, ViReal64* RSSIExcellent)

Function Usage:

Query lower limit of RSSI "excellent” on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIExcellent

RSSI value, in the range of -196~200dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set lower limit of RSSI "fair”

ViStatus _VI_FUNC CySAn_SetMapRSSIFair(ViSession Vi, ViReal64 RSSIFair)

Function Usage:

Set lower limit of RSSI "fair" on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIFair

RSSI value, in the range of -199 ~197dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query lower limit of RSSI "fair"

ViStatus _VI_FUNC CySAn_GetMapRSSIFair(ViSession Vi, ViReal64* RSSIFair)

Function Usage:

Query lower limit of RSSI "fair" on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIFair

RSSI value, in the range of -199 ~197dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set lower limit of RSSI "good"

ViStatus _VI_FUNC CySAn_SetMapRSSIGood(ViSession Vi, ViReal64 RSSIGood)

Function Usage:

Set lower limit of RSSI "good" on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIGood

RSSI value, in the range of -198~ 198dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query lower limit of RSSI "good"

ViStatus _VI_FUNC CySAn_GetMapRSSIGood(ViSession Vi, ViReal64* RSSIGood)

Function Usage:

Query lower limit of RSSI "good" on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIGood

RSSI value, in the range of -198 ~ 198dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set upper limit of RSSI "poor"

ViStatus _VI_FUNC CySAn_SetMapRSSIPoor(ViSession Vi, ViReal64 RSSIPoor)

Function Usage:

Set lower limit of RSSI "poor" on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIPoor

RSSI value, in the range of -200 ~196dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query upper limit of RSSI "poor"

ViStatus _VI_FUNC CySAn_GetMapRSSIPoor(ViSession Vi, ViReal64* RSSIPoor)

Function Usage:

Query upper limit of RSSI "poor" on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIPoor

RSSI value, in the range of -200~196dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set lower limit of RSSI "very good"

ViStatus _VI_FUNC CySAn_SetMapRSSIVeryGood(ViSession Vi, ViReal64 RSSIVeryGood)

Function Usage:

Set lower limit of RSSI "very good" on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIVeryGood

RSSI value, in the range of -197~199dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Query lower limit of RSSI "very good”

ViStatus _VI_FUNC CySAn_GetMapRSSIVeryGood(ViSession Vi, ViReal64* RSSIVeryGood)

Function Usage:

Query lower limit of RSSI "very good” on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RSSIVeryGood

RSSI value, in the range of -197 ~199dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set map ON/OFF

ViStatus _VI_FUNC CySAn_SetMapOn(ViSession Vi, ViBoolean MapOn)

Function Usage:

Set map ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapOn

Map ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Map - Set map ON/OFF

ViStatus _VI_FUNC CySAn_GetMapOn(ViSession Vi, ViBoolean* MapOn)

Function Usage:

Set map ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapOn

Map ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Set collection ON/OFF

ViStatus _VI_FUNC CySAn_SetMapCollectOn(ViSession Vi, ViBoolean MapCollectOn)

Function Usage:

Set outdoor map collection ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapCollectOn

Collection ON/OFF: START(1) for start collection, and STOP(0) for stop collection.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Query collection ON/OFF

ViStatus _VI_FUNC CySAn_GetMapCollectOn(ViSession Vi, ViBoolean* MapCollectOn)

Function Usage:

Set outdoor map collection ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapCollectOn

Collection ON/OFF: START(1) for start collection, and STOP(0) for stop collection.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Set repeat type

ViStatus _VI_FUNC CySAn_SetMapRepeatType(ViSession Vi, ViBoolean RepeatType)

Function Usage:

Set outdoor map repeat type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatType

Repeat type: TIME(0) means time and DIST(1) means distance.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Query repeat type

ViStatus _VI_FUNC CySAn_GetMapRepeatType(ViSession Vi, ViBoolean*

RepeatType)

Function Usage:

Query outdoor map repeat type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatType

Repeat type: TIME(0) means time and DIST(1) means distance.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Set repeat distance

ViStatus _VI_FUNC CySAn_SetMapDistance(ViSession Vi, ViReal64 Distance)

Function Usage:

Set outdoor map repeat distance.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Distance

Repeat distance, in the range of 50~1000m.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Query repeat distance

ViStatus _VI_FUNC CySAn_GetMapDistance(ViSession Vi, ViReal64* Distance)

Function Usage:

Query outdoor map repeat distance.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Distance

Repeat distance, in the range of 50~1000m.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Set repeat time

ViStatus _VI_FUNC CySAn_SetMapRepeatTime(ViSession Vi, ViReal64 RepeatTime)

Function Usage:

Set outdoor map repeat time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatTime

Repeat time, in the range of 5 ~ 600s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Query repeat time

ViStatus _VI_FUNC CySAn_SetMapRepeatTime(ViSession Vi, ViReal64 RepeatTime)

Function Usage:

Query outdoor map repeat time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatTime

Repeat time, in the range of 5 ~ 600s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Indoor map - Delete mark

ViStatus _VI_FUNC CySAn_DeleteAllMapData(ViSession Vi)

Function Usage:

Delete all marks on the indoor map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Outdoor map - Set collection ON/OFF

ViStatus _VI_FUNC CySAn_SetMapCollectOn(ViSession Vi, ViBoolean MapCollectOn)

Function Usage:

Set outdoor map collection ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapCollectOn

Collection ON/OFF: START(1) for start collection, and STOP(0) for stop collection.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load antenna factor

ViStatus _VI_FUNC CySAn_LoadAntennaFile(ViSession Vi, ViConstString FileName)

Function Usage:

Load the antenna factor in the field strength function measurement in the spectrum analysis mode**(the command is invalid if the file does not exist and is valid only for the current storage location)**, so that the antenna factor can be weighted when the corresponding measurement function is turned on.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store antenna factor

ViStatus _VI_FUNC CySAn_StoreAntennaFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store the antenna factor edited in the field strength function measurement in the spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete antenna factor

ViStatus _VI_FUNC CySAn_DeleteAntenna(ViSession Vi, ViConstString FileName)

Function Usage:

Delete antenna factor file.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete all antenna factors

ViStatus _VI_FUNC CySAn_DeleteAntennaAll(ViSession Vi)

Function Usage:

Delete all antenna factor files.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load SEM file

ViStatus _VI_FUNC CySAn_LoadSemFile(ViSession Vi, ViConstString FileName)

Function Usage:

Load the limit line as the mask in the SEM measurement.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store list to file

ViStatus _VI_FUNC CySAn_StoreListFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store list as list file**(the file will overwrite an existing file and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

chStr

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load list file

ViStatus _VI_FUNC CySAn_LoadListFile(ViSession Vi, ViConstString FileName)

Function Usage:

Load list file**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete list file

ViStatus _VI_FUNC CySAn_DeleteLimitFile(ViSession Vi, ViConstString FileName)

Function Usage:

Delete list file**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete all list files

ViStatus _VI_FUNC CySAn_DeleteLimitFileAll(ViSession Vi)

Function Usage:

Delete all list files.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete limit file

ViStatus _VI_FUNC CySAn_DeleteLimitFile(ViSession Vi, ViConstString FileName)

Function Usage:

Delete limit file**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store limit file

ViStatus _VI_FUNC CySAn_StoreLimitFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store limit line as limit file**(the file will overwrite an existing file and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load limit line

ViStatus _VI_FUNC CySAn_LoadLimitFile(ViSession Vi, ViConstString FileName)

Function Usage:

Load limit file to limit line**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete all limit files

ViStatus _VI_FUNC CySAn_DeleteDataFileAll(ViSession Vi)

Function Usage:

Delete all limit files.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Interference analysis mode function

Frequency - Set span

ViStatus _VI_FUNC CySAn_SetSpan(ViSession Vi, ViReal64 Span)

Function Usage:

Set span in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Span

Frequency value.

SA 0Hz~9.1GHz

IA 0Hz~9.1GHz

RTSA 3.662kHz~120MHz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query span

ViStatus _VI_FUNC CySAn_GetSpan(ViSession Vi, ViReal64* Span)

Function Usage:

Query span in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Span

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Full span

ViStatus _VI_FUNC CySAn_SetFullSpan(ViSession Vi)

Function Usage:

Set to full span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Frequency - Zero span

ViStatus _VI_FUNC CySAn_SetZeroSpan(ViSession Vi)

Function Usage:

Set to zero Span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set previous span

ViStatus _VI_FUNC CySAn_SetPreviousSpan(ViSession Vi)

Function Usage:

Set to previous span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set start frequency

ViStatus _VI_FUNC CySAn_SetStartFreq(ViSession Vi, ViReal64 Start)

Function Usage:

Set start frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Start

Frequency value.

The spectrum analysis frequency can be set in the range of 0Hz~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query start frequency

ViStatus _VI_FUNC CySAn_GetStartFreq(ViSession Vi, ViReal64* Start)

Function Usage:

Query start frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Start

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set stop frequency

ViStatus _VI_FUNC CySAn_SetStopFreq(ViSession Vi, ViReal64 Stop)

Function Usage:

Set stop frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Stop

Frequency value.

The spectrum analysis frequency can be set in the range of 0Hz~9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query stop frequency

ViStatus _VI_FUNC CySAn_GetStopFreq(ViSession Vi, ViReal64* Stop)

Function Usage:

Query stop frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Stop

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set signal standard name

ViStatus _VI_FUNC CySAn_SetSignalStandard(ViSession Vi, ViConstString SignalStandard)

Function Usage:

Set signal standard name in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SignalStandard

Signal standard name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query signal standard name

ViStatus _VI_FUNC CySAn_GetSignalStandard(ViSession Vi, ViInt32 SignalStandardBufferSize, ViChar SignalStandard[])

Function Usage:

Query signal standard name in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SignalStandardBufferSize

Signal standard buffer size.

Standard

Signal standard name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set signal standard channel number

ViStatus _VI_FUNC CySAn_SetChannelNum(ViSession Vi, ViInt32 ChannelNum)

Function Usage:

Set channel number in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

channelNum

Channel number.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query signal standard channel number

ViStatus _VI_FUNC CySAn_GetChannelNum(ViSession Vi, ViInt32* ChannelNum)

Function Usage:

Set channel number in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

channelNum

Channel number.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set reference level

ViStatus _VI_FUNC CySAn_SetReference(ViSession Vi, ViReal64 Reference)

Function Usage:

Set reference level value. The reference level value is related to the current amplitude unit, the setting range corresponds to dBm, and conversion is required.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value (-150dBm ~ +30dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query reference level

ViStatus _VI_FUNC CySAn_GetReference(ViSession Vi, ViReal64* Reference)

Function Usage:

Query reference level value (reference value). The reference level value is related to the current amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value (-150dBm ~ +30dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set reference position

ViStatus _VI_FUNC CySAn_SetReferPosition(ViSession Vi, ViInt32 ReferPosition)

Function Usage:

Set reference position.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ReferPosition

Reference position, in the range of -10~10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query reference position

ViStatus _VI_FUNC CySAn_GetReferPosition(ViSession Vi, ViInt32* ReferPosition)

Function Usage:

Query reference position.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ReferPosition

Reference position, in the range of -10 ~10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_SetAttenuationAuto(ViSession Vi, ViBoolean AttenuationAuto)

Function Usage:

Set attenuation to auto On/Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_GetAttenuationAuto(ViSession Vi, ViBoolean* AttenuationAuto)

Function Usage:

Query attenuation auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set scale/div.

ViStatus _VI_FUNC CySAn_SetYScalse(ViSession Vi, ViReal32 YScalse)

Function Usage:

Set scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. (0.01dB~100dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query scale/div.

ViStatus _VI_FUNC CySAn_GetYScalse(ViSession Vi, ViReal32* YScalse)

Function Usage:

Query scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. (0.01dB~100dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set scale type

ViStatus _VI_FUNC CySAn_SetVideoType(ViSession Vi, ViInt32 VideoType)

Function Usage:

Set scale type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoType

Scale type: LOG(0) means logarithmic type and LIN(1) means linear type.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query scale type

ViStatus _VI_FUNC CySAn_GetVideoType(ViSession Vi, ViInt32* VideoType)

Function Usage:

Query scale type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoType

Scale type: LOG(0) means logarithmic type and LIN(1) means linear type.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set unit

ViStatus _VI_FUNC CySAn_SetAmplitudeUnits(ViSession Vi, ViInt32 AmplitudeUnits)

Function Usage:

Set amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AmplitudeUnits

Amp Unit

Instrument mode	Unit
Spectrum Analyzer	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10
IA	DBM 0
Orientation analysis	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query unit

ViStatus _VI_FUNC CySAn_GetAmplitudeUnits(ViSession Vi, ViInt32* AmplitudeUnits)

Function Usage:

Query amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AmplitudeUnits

Amp Unit

Instrument mode	Unit
Spectrum Analyzer	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10
IA	DBM 0
Orientation analysis	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set resolution bandwidth

ViStatus _VI_FUNC CySAn_SetResolutionBandwidth(ViSession Vi, ViReal64 ResolutionBandwidth)

Function Usage:

Set resolution bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidth

Frequency value (in Hz), in the range of 1 Hz ~20 MHz in 1-3-10 steps.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query resolution bandwidth

ViStatus _VI_FUNC CySAn_GetResolutionBandwidth(ViSession Vi, ViReal64* ResolutionBandwidth)

Function Usage:

Query resolution bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidth

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set video bandwidth

ViStatus _VI_FUNC CySAn_SetVideoBandwidth(ViSession Vi, ViReal64 VideoBandwidth)

Function Usage:

Set video bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidth

Frequency value (in Hz), in the range of 1 Hz ~20 MHz in 1-3-10 steps.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query video bandwidth

ViStatus _VI_FUNC CySAn_GetVideoBandwidth(ViSession Vi, ViReal64* VideoBandwidth)

Function Usage:

Query video bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidth

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set auto resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_SetResolutionBandwidthAuto(ViSession Vi, ViBoolean ResolutionBandwidthAuto)

Function Usage:

Set RBW to be auto On/Off. When set to Auto, RBW will adjust RBW by span according to the ratio of SPAN/RBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidthAuto

Automatic ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query auto resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_GetResolutionBandwidthAuto(ViSession Vi, ViBoolean* ResolutionBandwidthAuto)

Function Usage:

Query auto On/Off status of RBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidthAuto

Automatic ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set video resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_SetVideoBandwidthAuto(ViSession Vi, ViBoolean VideoBandwidthAuto)

Function Usage:

Set auto on/off of VBW. When set to Auto, VBW will adjust RBW by resolution bandwidth according to the ratio of RBW/VBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidthAuto

Automatic ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query video resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_GetVideoBandwidthAuto(ViSession Vi, ViBoolean* VideoBandwidthAuto)

Function Usage:

Query auto On/Off status of VBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

VideoBandwidthAuto

Automatic ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set SPAN/RBW

ViStatus _VI_FUNC CySAn_SetSpanRBWRatio(ViSession Vi, ViInt32 SpanRBWRatio)

Function Usage:

Set the value of SPAN/RBW in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpanRBWRatio

SPAN/RBW value, range 1~500.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query SPAN/RBW

ViStatus _VI_FUNC CySAn_GetSpanRBWRatio(ViSession Vi, ViInt32* SpanRBWRatio)

Function Usage:

Query the SPAN/RBW value in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpanRBWRatio

SPAN/RBW value, range 1~500.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set RBW/VBW

ViStatus _VI_FUNC CySAn_SetRBWVBWRatio(ViSession Vi, ViInt32 RBWVBWRatio)

Function Usage:

Set RBW/VBW value in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RBWVBWRatio

RBW/VBW value, range 1~100.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query RBW/VBW

ViStatus _VI_FUNC CySAn_GetRBWVBWRatio(ViSession Vi, ViInt32* RBWVBWRatio)

Function Usage:

Query RBW/VBW value in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RBWVBWRatio

RBW/VBW value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Mkr - SetMkrState

ViStatus _VI_FUNC CySAn_SetMarkerType(ViSession Vi, ViInt32 Index, ViInt32 Type)

Function Usage:

Set marker state in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Type

Marker state.

OFF(0)	Marker OFF
NORM(1)	Normal marker ON
DELTa(2)	Delta marker ON

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Mkr - QueryMkrState

ViStatus _VI_FUNC CySAn_GetMarkerType(ViSession Vi, ViInt32 Index, ViInt32* Type)

Function Usage:

Query marker state in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Type

Marker state.

OFF(0)	Marker OFF
NORM(1)	Normal marker ON
DELTa(2)	Delta marker ON

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Activate marker

ViStatus _VI_FUNC CySAn_SetMarkerActive(ViSession Vi, ViInt32 Index)

Function Usage:

Activate marker in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Marker function (marker->)

ViStatus _VI_FUNC CySAn_SetInstrumentToMarker(ViSession Vi, ViInt32 Index, ViInt32 InstrumentSetting)

Function Usage:

Set marker function in the current mode (which is Mkr -> in the Spectrum Analyzer mode).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

InstrumentSetting

Instrument mode	InstrumentSetting	Function
Non-zero span	STARt(0)	Marker -> Start frequency (set marker frequency to start frequency)
	STOP(1)	Marker -> Stop frequency (set marker frequency to stop frequency)
	CENTer (2)	Marker -> Center frequency (set marker frequency to center frequency)
	STEP(3)	Marker -> Step frequency (set marker frequency to step frequency)
Span Zero	STARt(0)	Marker -> Start frequency (set marker index to minimum index)
	STOP(1)	Marker -> Stop frequency (set marker index to

		maximum index)
	CENTer (2)	Marker -> Center frequency (set marker index to center index)
	STEP(3)	Marker -> Step frequency (set marker frequency to step frequency)

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set all markers OFF

ViStatus _VI_FUNC CySAn_SetAllMarkerOff(ViSession Vi)

Function Usage:

Turn off all markers in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set marker X value

ViStatus _VI_FUNC CySAn_SetMarkerx(ViSession Vi, ViInt32 Index, ViReal64 MarkerX)

Function Usage:

Set marker X value in the current mode. X may be negative when the marker is a delta marker.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

MarkerX

Marker X value.

Instrument mode	Parameter unit
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Real-time spectrum	Hz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker X value

ViStatus _VI_FUNC CySAn_GetMarkerx(ViSession Vi, ViInt32 Index, ViReal64* MarkerX)

Function Usage:

Set marker X value in the current mode. X may be negative when the marker is a delta marker.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

MarkerX

Marker X value.

Instrument mode	Parameter unit
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Real-time spectrum	Hz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker Y value

ViStatus _VI_FUNC CySAn_GetMarkerY(ViSession Vi, ViInt32 Index, ViReal64* Val)

Function Usage:

Query marker Y value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Val

Marker Y value in dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Search

ViStatus _VI_FUNC CySAn_SetMarkerSearch(ViSession Vi, ViInt32 Type, ViInt32 Index)

Function Usage:

Move the marker in the current mode to the position of Maximum, Minimum, Peak, Sub Peak, Left Peak, and Right Peak.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Type

Search Type

PEAK - peak value

MAXimum - max value

MINimum - min value

PNEXt Sub Peak

PLEFt Left Peak

PRIGht Right Peak

Index

Marker index, optional 1~6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set noise marker ON/OFF

ViStatus _VI_FUNC CySAn_SetNoiseMarkerOn(ViSession Vi, ViInt32 Index, ViBoolean NoiseMarker)

Function Usage:

Set the noise marker to On/Off in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

NoiseMarker

Noise marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query noise marker ON/OFF

ViStatus _VI_FUNC CySAn_GetNoiseMarkerOn(ViSession Vi, ViInt32 Index, ViBoolean* NoiseMarker)

Function Usage:

Query the noise marker On/Off status in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

NoiseMarker

Noise marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query sweep points

ViStatus _VI_FUNC CySAn_GetSweepPoint(ViSession Vi, ViInt32* SweepPoint)

Function Usage:

QuerySwpPoints.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepPoint

The number of sweep points, which can be set to 201, 501, 1001, 2001 and 4001.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set sweep time

ViStatus _VI_FUNC CySAn_SetSweepTime(ViSession Vi, ViReal64 SweepTime)

Function Usage:

Set sweep time in the current mode. sweep time is the time required for the local oscillator tuning through the selected frequency interval. Sweep time directly affects the time required to complete a test, which does not include the dead time between the completion of a sweep and the start of the next sweep. Sweep time usually varies with Span, RBW and VBW. Sweep time cannot be set when RBW<=1kHz in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTime

Time (ms).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query sweep time

ViStatus _VI_FUNC CySAn_GetSweepTime(ViSession Vi, ViReal64* SweepTime)

Function Usage:

Query the sweep time in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTime

Time (ms).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set auto sweep time ON/OFF

ViStatus _VI_FUNC CySAn_SetSweepTimeAuto(ViSession Vi, ViBoolean SweepTimeAuto)

Function Usage:

Set sweep time to auto On/Off. When set to On, the instrument will adopt the sweep speed as high as possible; or you can manually increase the sweep time to meet some specific measurement requirements. Sweep time set manually must be > = automatic sweep time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTimeAuto

Automatic sweep time ON/OFF in linear sweep mode: OFF(0) means manual sweep time, and ON(1) means auto scan time.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query auto sweep time ON/OFF

ViStatus _VI_FUNC CySAn_GetSweepTimeAuto(ViSession Vi, ViBoolean* SweepTimeAuto)

Function Usage:

Query sweep time auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTimeAuto

Automatic sweep time ON/OFF in linear sweep mode: OFF(0) means manual sweep time, and ON(1) means auto scan time.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set sweep points

ViStatus _VI_FUNC CySAn_SetSweepPoint(ViSession Vi, ViInt32 SweepPoint)

Function Usage:

Set number of sweep points

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepPoint

The number of sweep points, which can be set to 201, 501, 1001, 2001 and 4001.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Data - Query trace data

ViStatus _VI_FUNC CySAn_FetchTraceY(ViSession Vi, ViConstString TraceName, ViInt32 ArrayLength, ViInt32* ActualPoints, ViReal64 Amplitude[])

Function Usage:

Query trace data in the spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

ArrayLength

Length of trace data received.

ActualPoints

Points.

Amplitude[]

Array of trace data storage.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Set trace status

ViStatus _VI_FUNC CySAn_SetTraceType(ViSession Vi, ViConstString TraceName, ViInt32 TraceType)

Function Usage:

Set trace status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

TraceType

Trace status.

CLRW(0) indicates refresh trace

MAXH(1) indicates maximum hold

MINH(2) indicates minimum hold

VIEW(3) indicates to hold trace

BLANk(4) indicates to hide trace

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Query trace status

ViStatus _VI_FUNC CySAn_GetTraceType(ViSession Vi, ViConstString Trace, ViInt32* TraceType)

Function Usage:

Query trace status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

TraceType

Trace status.

CLRW(0) indicates refresh trace

MAXH(1) indicates maximum hold

MINH(2) indicates minimum hold

VIEW(3) indicates to hold trace

BLANk(4) indicates to hide trace

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set measurement mode

ViStatus _VI_FUNC CySAn_SetMeasureMode(ViSession Vi, ViInt32 MeasureMode)

Function Usage:

Set measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureMode

Measurement mode

NORMAL(0) Spectrum measurement

SG(1) Waterfall plot

RSSI(2) RSSI

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query measurement mode

ViStatus _VI_FUNC CySAn_GetMeasureMode(ViSession Vi, ViInt32* MeasureMode)

Function Usage:

Query measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureMode

Measurement mode.

NORMAL(0) Spectrum measurement

SG(1) Waterfall plot

RSSI(2) RSSI

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Set average ON/OFF

ViStatus _VI_FUNC CySAn_SetAverageEnabled(ViSession Vi, ViBoolean AverageEnabled)

Function Usage:

Set averaging to On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageEnabled

Average ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Query average ON/OFF

ViStatus _VI_FUNC CySAn_GetAverageEnabled(ViSession Vi, ViBoolean*

AverageEnabled)

Function Usage:

Query average On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageEnabled

Average ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Set average count

ViStatus _VI_FUNC CySAn_SetAverageCount(ViSession Vi, ViInt32 AverageCount)

Function Usage:

Setting averaging times.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageCount

Average count, in the range of 1~1000.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Query average count

ViStatus _VI_FUNC CySAn_GetAverageCount(ViSession Vi, ViInt32* AverageCount)

Function Usage:

Query averaging count.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AverageCount

Averaging times.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Clear average

ViStatus _VI_FUNC CySAn_ClearAverage(ViSession Vi)

Function Usage:

Count current averaging from 0.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Average - Query current average count

ViStatus _VI_FUNC CySAn_GetCurrentAverage(ViSession Vi, ViInt32* Val)

Function Usage:

Query current average count.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Averaged count.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Set detector type

ViStatus _VI_FUNC CySAn_SetDetectorType(ViSession Vi, ViInt32 DetectorType)

Function Usage:

Set detection type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorType

Detection type

Spectrum Analysis, Interference Analysis	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Sample
	NORMal(3)	Normal (Rosenfeld)
	AVERage(4)	Average
	RMS(5)	Rms
Orientation analysis	AVERage(0)	Average
	POSitive(1)	Peak
	SAMPle(2)	Real-time

Real-time spectrum	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Smp
	AVERage(4)	Average

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Query detector type

ViStatus _VI_FUNC CySAn_GetDetectorType(ViSession Vi, ViInt32* DetectorType)

Function Usage:

Query detection type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorType

Detection type

Spectrum Analysis, Interference Analysis	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Sample
	NORMal(3)	Normal (Rosenfeld)
	AVERage(4)	Average
	RMS(5)	Rms
Orientation analysis	AVERage(0)	Average
	POSitive(1)	Peak
	SAMPle(2)	Real-time
Real-time spectrum	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Smp
	AVERage(4)	Average

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Set auto detector ON/OFF

ViStatus _VI_FUNC CySAn_SetDetectorTypeAuto(ViSession Vi, ViBoolean DetectorTypeAuto)

Function Usage:

Set detection to auto On/Off. In auto detection mode, the instrument will automatically select detection type according to different measurements..

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorTypeAuto

Automatic detector ON/OFF: OFF (0) for manual, and (1) for automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection- Query auto detector ON/OFF

ViStatus _VI_FUNC CySAn_GetDetectorTypeAuto(ViSession Vi, ViBoolean* DetectorTypeAuto)

Function Usage:

Query detection auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorTypeAuto

Automatic detector ON/OFF: OFF (0) for manual, and (1) for automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Set span time

ViStatus _VI_FUNC CySAn_SetSpanTime(ViSession Vi, ViInt32 SpanTime)

Function Usage:

Set span time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpanTime

Span time, in the range of 1~2880000 min.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Query span time

ViStatus _VI_FUNC CySAn_GetSpanTime(ViSession Vi, ViInt32* SpanTime)

Function Usage:

Query span time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpanTime

Span time, in the range of 1~2880000 min.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Set autoSave ON/OFF

ViStatus _VI_FUNC CySAn_SetAutoSave(ViSession Vi, ViBoolean AutoSave)

Function Usage:

Set autostore ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoSave

AutoSave ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Query AutoSave ON/OFF

ViStatus _VI_FUNC CySAn_GetAutoSave(ViSession Vi, ViBoolean* AutoSave)

Function Usage:

Query AutoSave ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoSave

AutoSave ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Set time cursor

ViStatus _VI_FUNC CySAn_SetTimeCursor(ViSession Vi, ViInt32 TimeCursor)

Function Usage:

Set measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TimeCursor

Time cursor (int), in the range of 1~290.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Set sweep interval

ViStatus _VI_FUNC CySAn_SetSweepInterval(ViSession Vi, ViReal64 SweepInterval)

Function Usage:

Set sweep interval.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepInterval

Sweep interval (ms), in the range of 0s~1000s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Query sweep interval

ViStatus _VI_FUNC CySAn_GetSweepInterval(ViSession Vi, ViReal64* SweepInterval)

Function Usage:

Query sweep interval.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepInterval

Sweep interval (ms), in the range of 0s~1000s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

AutoSave - Restart measurement

ViStatus _VI_FUNC CySAn_Restart(ViSession Vi)

Function Usage:

Restart measurement.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Orientation analysis mode function

Frequency - Query storage position data

ViStatus _VI_FUNC CySAn_GetPositionData(ViSession Vi, ViInt32 Index, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query storage position data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Data index.

ValBufferSize

Data buffer size.

Val[]

Data.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Delete edit points

ViStatus _VI_FUNC CySAn_DeletePositionData(ViSession Vi, ViInt32 Index)

Function Usage:

Delete edit points.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Data index.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Clear edit points

ViStatus _VI_FUNC CySAn_DeleteAllPositionData(ViSession Vi)

Function Usage:

Clear edit points.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Positioning save

ViStatus _VI_FUNC CySAn_SavePositionData(ViSession Vi)

Function Usage:

Positioning save.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set positioning selection

ViStatus _VI_FUNC CySAn_SetSelectPosition(ViSession Vi, ViInt32 SelectPosition)

Function Usage:

Positioning selection.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SelectPosition

Positioning index, in the range of 1 ~ 6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query positioning selection

ViStatus _VI_FUNC CySAn_GetSelectPosition(ViSession Vi, ViInt32* SelectPosition)

Function Usage:

Query positioning selection.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SelectPosition

Positioning index, in the range of 1 ~ 6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set scale/div.

ViStatus _VI_FUNC CySAn_SetYScalse(ViSession Vi, ViReal32 YScalse)

Function Usage:

Set scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. (0.01dB~100dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query scale/div.

ViStatus _VI_FUNC CySAn_GetYScalse(ViSession Vi, ViReal32* YScalse)

Function Usage:

Query scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. (0.01dB~100dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set reference level

ViStatus _VI_FUNC CySAn_SetReference(ViSession Vi, ViReal64 Reference)

Function Usage:

Set reference level value. The reference level value is related to the current amplitude unit, the setting range corresponds to dBm, and conversion is required.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value (-150dBm ~ +30dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query reference level

ViStatus _VI_FUNC CySAn_GetReference(ViSession Vi, ViReal64* Reference)

Function Usage:

Query reference level value (reference value). The reference level value is related to the current amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value (-150dBm ~ +30dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set unit

ViStatus _VI_FUNC CySAn_SetAmplitudeUnits(ViSession Vi, ViInt32 AmplitudeUnits)

Function Usage:

Set amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AmplitudeUnits

Amp Unit

Instrument mode	Unit
Spectrum Analyzer	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10
IA	DBM 0
Orientation analysis	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query unit

ViStatus _VI_FUNC CySAn_GetAmplitudeUnits(ViSession Vi, ViInt32* AmplitudeUnits)

Function Usage:

Query amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AmplitudeUnits

Amp Unit

Instrument mode	Unit
Spectrum Analyzer	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10
IA	DBM 0
Orientation analysis	DBM,DBMV,DBUV,V,W,A,DBW,DBV,DBA,DBMA,DBUA 0,1,2,3,4,5,6, 7,8,9,10

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set limit ON/OFF

ViStatus _VI_FUNC CySAn_SetLAPowerLimit(ViSession Vi, ViReal64 PowerLimit)

Function Usage:

Set limit ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PowerLimit

Limit ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query limit ON/OFF

ViStatus _VI_FUNC CySAn_GetLAPowerLimit(ViSession Vi, ViReal64* PowerLimit)

Function Usage:

Query limit ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PowerLimit

Limit ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_SetAttenuationAuto(ViSession Vi, ViBoolean AttenuationAuto)

Function Usage:

Set attenuation to auto On/Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_GetAttenuationAuto(ViSession Vi, ViBoolean* AttenuationAuto)

Function Usage:

Query attenuation auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set audio alarm ON/OFF

ViStatus _VI_FUNC CySAn_SetLimitBeep(ViSession Vi, ViBoolean Beep)

Function Usage:

Set limit alarm to on/off. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every sweep.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Beep

Audio alarm ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query audio alarm ON/OFF

ViStatus _VI_FUNC CySAn_GetLimitBeep(ViSession Vi, ViBoolean* Beep)

Function Usage:

Query limit alarm on/off status. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every sweep.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Beep

Audio alarm ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set bandwidth

ViStatus _VI_FUNC CySAn_SetIfBandwidth(ViSession Vi, ViReal64 IfBandwidth)

Function Usage:

Set bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfBandwidth

Bandwidth, in the range of 150Hz ~ 150kHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query bandwidth

ViStatus _VI_FUNC CySAn_GetIfBandwidth(ViSession Vi, ViReal64* IfBandwidth)

Function Usage:

Query bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IfBandwidth

Bandwidth, in the range of 150Hz ~ 150kHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set maximum field strength angle

ViStatus _VI_FUNC CySAn_SetMaxFstAngle(ViSession Vi, ViReal64 MaxFstAngle)

Function Usage:

Set maximum field strength angle.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MaxFstAngle

Maximum field strength angle, in the range of 0～359.9°.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query maximum field strength angle

ViStatus _VI_FUNC CySAn_GetMaxFstAngle(ViSession Vi, ViReal64* MaxFstAngle)

Function Usage:

Query maximum field strength angle.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MaxFstAngle

Maximum field strength angle, in the range of 0～359.9°.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set automatic maximum field strength angle ON/OFF

ViStatus _VI_FUNC CySAn_SetMaxFstAngleAuto(ViSession Vi, ViBoolean MaxFstAngleAuto)

Function Usage:

Set automatic maximum field strength angle ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MaxFstAngleAuto

Automatic maximum field strength angle ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query automatic maximum field strength angle ON/OFF

ViStatus _VI_FUNC CySAn_GetMaxFstAngleAuto(ViSession Vi, ViBoolean* MaxFstAngleAuto)

Function Usage:

Query automatic maximum field strength angle ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MaxFstAngleAuto

Automatic maximum field strength angle ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set maximum field strength angle range

ViStatus _VI_FUNC CySAn_SetAngleRange(ViSession Vi, ViInt32 AngleRange)

Function Usage:

Set maximum field strength angle range.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AngleRange

Angle range (even), in the range of 2.0 °~ 180.0 °.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query maximum field strength angle range

ViStatus _VI_FUNC CySAn_GetAngleRange(ViSession Vi, ViInt32* AngleRange)

Function Usage:

Query maximum field strength angle range.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AngleRange

Angle range (even), in the range of 2.0 °~ 180.0 °.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Search

ViStatus _VI_FUNC CySAn_SetMarkerSearch(ViSession Vi, ViInt32 Type, ViInt32 Index)

Function Usage:

Move the marker in the current mode to the position of Peak, Sub Peak, Left Peak, and Right Peak.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Type

Search Type

PEAK - peak value

PNEXt Sub Peak

PLEFt Left Peak

PRIGht Right Peak

Index

Marker index, optional 1~6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep/Antenna- Set demodulation mode

ViStatus _VI_FUNC CySAn_SetDemodType(ViSession Vi, ViInt32 DemodType)

Function Usage:

Set demodulation mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodType

Demodulation mode.

CW(0) OFF

FM(1) Frequency modulation

AM(2) Amplitude modulation

USB(3) Upper sideband

LSB(4) Lower sideband

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep/Antenna- Query demodulation mode

ViStatus _VI_FUNC CySAn_SetDemodType(ViSession Vi, ViInt32 DemodType)

Function Usage:

Query demodulation mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodType

Demodulation mode.

CW(0) OFF

FM(1) Frequency modulation

AM(2) Amplitude modulation

USB(3) Upper sideband

LSB(4) Lower sideband

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Demodulation - Set audio speaker ON/OFF

ViStatus _VI_FUNC CySAn_SetSpeakerOn(ViSession Vi, ViBoolean SpeakerOn)

Function Usage:

Set audio speaker ON/OFF

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpeakerOn

Audio speaker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Demodulation - Query audio speaker ON/OFF

ViStatus _VI_FUNC CySAn_GetSpeakerOn(ViSession Vi, ViBoolean* SpeakerOn)

Function Usage:

Query audio speaker ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpeakerOn

Audio speaker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Demodulation - Set demodulation volume

ViStatus _VI_FUNC CySAn_SetDemodVolume(ViSession Vi, ViInt32 DemodVolume)

Function Usage:

Set demodulation volume.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodVolume

Demodulation volume (0~100).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Demodulation - Query demodulation volume

ViStatus _VI_FUNC CySAn_SetDemodVolume(ViSession Vi, ViInt32 DemodVolume)

Function Usage:

Query demodulation volume.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodVolume

Demodulation volume (0~100).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Mark current position on the map

ViStatus _VI_FUNC CySAn_MarkerCurrent(ViSession Vi)

Function Usage:

Mark current position on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Delete all mark positions on the map

ViStatus _VI_FUNC CySAn_MarkerDeleteAll(ViSession Vi)

Function Usage:

Delete all mark positions on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Delete current mark position on the map

ViStatus _VI_FUNC CySAn_MarkerDeleteCurrent(ViSession Vi)

Function Usage:

Delete current mark position on the map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Delete positioning data point

ViStatus _VI_FUNC CySAn_DeleteMapPositionData(ViSession Vi, ViInt32 Index)

Function Usage:

Delete positioning data point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Positioning point index.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Delete all positioning data

ViStatus _VI_FUNC CySAn_MarkerDeleteAll(ViSession Vi)

Function Usage:

Delete all positioning data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Load positioning data

ViStatus _VI_FUNC CySAn_LoadMapPositionData(ViSession Vi, ViInt32 Index, ViInt32 Position)

Function Usage:

Load positioning data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Positioning data index.

Position

Positioning data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Set map positioning mode

ViStatus _VI_FUNC CySAn_SetMapMeasureType(ViSession Vi, ViInt32 MapMeasureType)

Function Usage:

Set map positioning mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapMeasureType

Positioning mode: REAL(0) means real-time mode, and LOAD(1) means loading mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Query map positioning mode

ViStatus _VI_FUNC CySAn_GetMapMeasureType(ViSession Vi, ViInt32* MapMeasureType)

Function Usage:

Query map positioning mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapMeasureType

Positioning mode: REAL(0) means real-time mode, and LOAD(1) means loading mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Map - Set map zoom mode

ViStatus _VI_FUNC CySAn_SetZoomInOut(ViSession Vi, ViBoolean In)

Function Usage:

Set map zoom mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

In

Zoom mode: OUT(0) means zoom out, and IN(1) means zoom in.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query current horizontal scan amplitude value

ViStatus _VI_FUNC CySAn_GetHorscanCurrentAmplitude(ViSession Vi, ViReal32* Val)

Function Usage:

Query horizontal scan amplitude value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Horizontal scan amplitude value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query array of horizontal scan amplitude values

ViStatus _VI_FUNC CySAn_GetHorscanAmplitude(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query array of horizontal scan amplitude values.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Data.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query maximum horizontal scan amplitude

ViStatus _VI_FUNC CySAn_GetHorscanAmplitudeMax(ViSession Vi, ViReal32* Val)

Function Usage:

Query array of horizontal scan amplitude values.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Maximum amplitude.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query current horizontal scan field strength value

ViStatus _VI_FUNC CySAn_GetHorscanCurrentFst(ViSession Vi, ViReal32* Val)

Function Usage:

Query current horizontal scan field strength value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Horizontal scan field strength value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query array of horizontal scan field strength values

ViStatus _VI_FUNC CySAn_GetHorscanFst(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query array of horizontal scan field strength values.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Data.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query maximum horizontal scan field strength value

ViStatus _VI_FUNC CySAn_GetHorscanFstMax(ViSession Vi, ViReal32* Val)

Function Usage:

Query maximum horizontal scan field strength value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Maximum field strength value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query map positioning field strength value

ViStatus _VI_FUNC CySAn_GetMapAmplitude(ViSession Vi, ViReal32* Val)

Function Usage:

Query map positioning field strength value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Field strength value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query amplitude value

ViStatus _VI_FUNC CySAn_GetPointScanAmplitude(ViSession Vi, ViReal32* Val)

Function Usage:

Query amplitude value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Amplitude value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query maximum amplitude value

ViStatus _VI_FUNC CySAn_GetPointScanAmplitude(ViSession Vi, ViReal32* Val)

Function Usage:

Query maximum amplitude value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Maximum amplitude value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query field strength value

ViStatus _VI_FUNC CySAn_GetPointScanFst(ViSession Vi, ViReal32* Val)

Function Usage:

Query field strength value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Field strength value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query maximum field strength value

ViStatus _VI_FUNC CySAn_GetPointScanFstMax(ViSession Vi, ViReal32* Val)

Function Usage:

Query maximum field strength value.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Maximum field strength value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query heading angle

ViStatus _VI_FUNC CySAn_GetLAHeading(ViSession Vi, ViReal32* Val)

Function Usage:

Query heading.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Heading angle.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query maximum point heading angle

ViStatus _VI_FUNC CySAn_GetLAMaxHeading(ViSession Vi, ViReal32* Val)

Function Usage:

Query maximum point heading angle.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Heading angle.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set measurement mode

ViStatus _VI_FUNC CySAn_SetLAMeasureType(ViSession Vi, ViInt32 MeasureType)

Function Usage:

Set measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode.

POTScan(0) Position scan

HORScan(1) Horizontal scan

MAP(2) Map positioning

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query measurement mode

ViStatus _VI_FUNC CySAn_SetLAMeasureType(ViSession Vi, ViInt32 MeasureType)

Function Usage:

Query measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode.

POTScan(0) Position scan

HORScan(1) Horizontal scan

MAP(2) Map positioning

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query pitch angle

ViStatus _VI_FUNC CySAn_GetLAPictch(ViSession Vi, ViReal32* Val)

Function Usage:

Query pitch angle.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Pitch angle.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query maximum pitch angle

ViStatus _VI_FUNC CySAn_GetLAMaxPictch(ViSession Vi, ViReal32* Val)

Function Usage:

Query maximum pitch angle.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Maximum pitch angle.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query roll angle

ViStatus _VI_FUNC CySAn_GetLARoll(ViSession Vi, ViReal32* Val)

Function Usage:

Query roll angle.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Roll angle.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query maximum roll angle

ViStatus _VI_FUNC CySAn_GetLAMaxRoll(ViSession Vi, ViReal32* Val)

Function Usage:

Query maximum roll angle.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Val

Maximum roll angle.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Add default point

ViStatus _VI_FUNC CySAn_AddAntennaDefaultPoint(ViSession Vi)

Function Usage:

Edit antenna factor and add default point. Frequency: 1GHz Antenna factor value: 0dB

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Delete point

ViStatus _VI_FUNC CySAn_DeletePoint(ViSession Vi, ViInt32 Index)

Function Usage:

Edit antenna factor and delete point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Point index.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Edit point

ViStatus _VI_FUNC CySAn_EditAntennaPoint(ViSession Vi, ViInt32 Index, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Edit antenna factor and edit point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Point index.

Freq

Frequency (0~9.1GHz).

Amp

Antenna factor value (-200~ 200dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Edit antenna factor - Add point

ViStatus _VI_FUNC CySAn_AddAntennaPoint(ViSession Vi, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Edit antenna factor and add point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Freq

Frequency (0~9.1GHz).

Amp

Antenna factor value (-200 ~ 200dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Field strength - Turn off antenna factor

ViStatus _VI_FUNC CySAn_ClearAntenna(ViSession Vi)

Function Usage:

Turn off antenna factor loading and set it to no-antenna factor state.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete antenna factor

ViStatus _VI_FUNC CySAn_DeleteAntenna(ViSession Vi, ViConstString FileName)

Function Usage:

Delete antenna factor file.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete all antenna factors

ViStatus _VI_FUNC CySAn_DeleteAntennaAll(ViSession Vi)

Function Usage:

Delete all antenna factor files.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load antenna factor

ViStatus _VI_FUNC CySAn_LoadAntennaFile(ViSession Vi, ViConstString FileName)

Function Usage:

Load the antenna factor in the field strength function measurement in the spectrum analysis mode**(the command is invalid if the file does not exist and is valid only for the current storage location)**, so that the antenna factor can be weighted when the corresponding measurement function is turned on.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store antenna factor

ViStatus _VI_FUNC CySAn_StoreAntennaFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store the antenna factor edited in the field strength function measurement in the spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Set detector type

ViStatus _VI_FUNC CySAn_SetDetectorType(ViSession Vi, ViInt32 DetectorType)

Function Usage:

Set detection type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorType

Detection type

Spectrum Analysis, Interference Analysis	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Sample
	NORMal(3)	Normal (Rosenfeld)
	AVERage(4)	Average
	RMS(5)	Rms
Orientation analysis	AVERage(0)	Average
	POSitive(1)	Peak
	SAMPle(2)	Real-time
Real-time spectrum	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Smp
	AVERage(4)	Average

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Query detector type

ViStatus _VI_FUNC CySAn_GetDetectorType(ViSession Vi, ViInt32* DetectorType)

Function Usage:

Query detection type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorType

Detection type

Spectrum Analysis, Interference Analysis	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Sample
	NORMal(3)	Normal (Rosenfeld)
	AVERage(4)	Average
	RMS(5)	Rms
Orientation analysis	AVERage(0)	Average
	POSitive(1)	Peak
	SAMPle(2)	Real-time
Real-time spectrum	POSitive(0)	Positive peak value
	NEGative(1)	Negative peak value
	SAMPle(2)	Smp
	AVERage(4)	Average

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Real time spectrum mode function

Frequency - Set span

ViStatus _VI_FUNC CySAn_SetSpan(ViSession Vi, ViReal64 Span)

Function Usage:

Set span in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Span

Frequency value.

SA 0Hz~9.1GHz

IA 0Hz~9.1GHz

RTSA 3.662kHz~120MHz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query span

ViStatus _VI_FUNC CySAn_GetSpan(ViSession Vi, ViReal64* Span)

Function Usage:

Query span in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Span

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Maximum span

ViStatus _VI_FUNC CySAn_SetFullSpan(ViSession Vi)

Function Usage:

Set maximum span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Minimal span

ViStatus _VI_FUNC CySAn_SetMinSpan(ViSession Vi)

Function Usage:

Set minimum span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set previous span

ViStatus _VI_FUNC CySAn_SetPreviousSpan(ViSession Vi)

Function Usage:

Set to previous span.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set start frequency

ViStatus _VI_FUNC CySAn_SetStartFreq(ViSession Vi, ViReal64 Start)

Function Usage:

Set start frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Start

Frequency value.

The spectrum analysis frequency can be set in the range of 0Hz-9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query start frequency

ViStatus _VI_FUNC CySAn_GetStartFreq(ViSession Vi, ViReal64* Start)

Function Usage:

Query start frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Start

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set stop frequency

ViStatus _VI_FUNC CySAn_SetStopFreq(ViSession Vi, ViReal64 Stop)

Function Usage:

Set stop frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Stop

Frequency value.

The spectrum analysis frequency can be set in the range of 0Hz-9.1GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query stop frequency

ViStatus _VI_FUNC CySAn_GetStopFreq(ViSession Vi, ViReal64* Stop)

Function Usage:

Query stop frequency value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Stop

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set scale/div.

ViStatus _VI_FUNC CySAn_SetYScalse(ViSession Vi, ViReal32 YScalse)

Function Usage:

Set scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. (0.01dB~100dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query scale/div.

ViStatus _VI_FUNC CySAn_GetYScalse(ViSession Vi, ViReal32* YScalse)

Function Usage:

Query scale.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

YScalse

Scale/div. (0.01dB~100dB).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set reference level

ViStatus _VI_FUNC CySAn_SetReference(ViSession Vi, ViReal64 Reference)

Function Usage:

Set reference level value. The reference level value is related to the current amplitude unit, the setting range corresponds to dBm, and conversion is required.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value (-150dBm ~ +30dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query reference level

ViStatus _VI_FUNC CySAn_GetReference(ViSession Vi, ViReal64* Reference)

Function Usage:

Query reference level value (reference value). The reference level value is related to the current amplitude unit.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Reference

Reference level value (-150dBm ~ +30dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set reference position

ViStatus _VI_FUNC CySAn_SetReferPosition(ViSession Vi, ViInt32 ReferPosition)

Function Usage:

Set reference position.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ReferPosition

Reference position, in the range of -10 ~ 10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query reference position

ViStatus _VI_FUNC CySAn_GetReferPosition(ViSession Vi, ViInt32* ReferPosition)

Function Usage:

Query reference position.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ReferPosition

Reference position, in the range of -10 ~ 10.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_SetAttenuationAuto(ViSession Vi, ViBoolean AttenuationAuto)

Function Usage:

Set attenuation to auto On/Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_GetAttenuationAuto(ViSession Vi, ViBoolean* AttenuationAuto)

Function Usage:

Query attenuation auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set resolution bandwidth

ViStatus _VI_FUNC CySAn_SetResolutionBandwidth(ViSession Vi, ViReal64 ResolutionBandwidth)

Function Usage:

Set resolution bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidth

Frequency value (in Hz), in the range of 1 Hz ~ 20 MHz in 1-3-10 steps.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query resolution bandwidth

ViStatus _VI_FUNC CySAn_GetResolutionBandwidth(ViSession Vi, ViReal64* ResolutionBandwidth)

Function Usage:

Query resolution bandwidth of the linear sweep in spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidth

Frequency value (Hz).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Set auto resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_SetResolutionBandwidthAuto(ViSession Vi, ViBoolean ResolutionBandwidthAuto)

Function Usage:

Set RBW to be auto On/Off. When set to Auto, RBW will adjust RBW by span according to the ratio of SPAN/RBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidthAuto

Automatic ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Bandwidth - Query auto resolution bandwidth ON/OFF

ViStatus _VI_FUNC CySAn_GetResolutionBandwidthAuto(ViSession Vi, ViBoolean* ResolutionBandwidthAuto)

Function Usage:

Query auto On/Off status of RBW.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ResolutionBandwidthAuto

Automatic ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Activate marker

ViStatus _VI_FUNC CySAn_SetMarkerActive(ViSession Vi, ViInt32 Index)

Function Usage:

Activate marker in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set all markers OFF

ViStatus _VI_FUNC CySAn_SetAllMarkerOff(ViSession Vi)

Function Usage:

Turn off all markers in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Search

ViStatus _VI_FUNC CySAn_SetMarkerSearch(ViSession Vi, ViInt32 Type, ViInt32 Index)

Function Usage:

Move the marker in the current mode to the position of Maximum, Minimum, Peak, Sub

Peak, Left Peak, and Right Peak.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Type

Search Type

PEAK - peak value

MAXimum - max value

MINimum - min value

PNEXt Sub Peak

PLEFt Left Peak

PRIGht Right Peak

Index

Marker index, optional 1~6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Marker function (marker->)

ViStatus _VI_FUNC CySAn_SetInstrumentToMarker(ViSession Vi, ViInt32 Index, ViInt32 InstrumentSetting)

Function Usage:

Set marker function in the current mode (which is Mkr -> in the Spectrum Analyzer mode).

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

InstrumentSetting

Instrument mode	InstrumentSetting	Function
Non-zero span	STARt(0)	Marker -> Start frequency (set marker frequency to start frequency)
	STOP(1)	Marker -> Stop frequency (set marker frequency to stop frequency)
	CENTer (2)	Marker -> Center frequency (set marker frequency to center frequency)
	STEP(3)	Marker -> Step frequency (set marker frequency to step frequency)

Span Zero	STARt(0)	Marker -> Start frequency (set marker index to minimum index)
	STOP(1)	Marker -> Stop frequency (set marker index to maximum index)
	CENTer (2)	Marker -> Center frequency (set marker index to center index)
	STEP(3)	Marker -> Step frequency (set marker frequency to step frequency)

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Mkr - SetMkrState

ViStatus _VI_FUNC CySAn_SetMarkerType(ViSession Vi, ViInt32 Index, ViInt32 Type)

Function Usage:

Set marker state in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Type

Marker state.

OFF(0) Marker OFF

NORM(1) Normal marker ON

DELTa(2) Delta marker ON

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Mkr - QueryMkrState

ViStatus _VI_FUNC CySAn_GetMarkerType(ViSession Vi, ViInt32 Index, ViInt32* Type)

Function Usage:

Query marker state in current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Type

Marker state.

OFF(0) Marker OFF

NORM(1) Normal marker ON

DELTa(2) Delta marker ON

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Set marker X value

ViStatus _VI_FUNC CySAn_SetMarkerx(ViSession Vi, ViInt32 Index, ViReal64 MarkerX)

Function Usage:

Set marker X value in the current mode. X may be negative when the marker is a delta marker.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

MarkerX

Marker X value.

Instrument mode	Parameter unit
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Real-time spectrum	Hz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker X value

ViStatus _VI_FUNC CySAn_GetMarkerx(ViSession Vi, ViInt32 Index, ViReal64* MarkerX)

Function Usage:

Set marker X value in the current mode. X may be negative when the marker is a delta marker.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

MarkerX

Marker X value.

Instrument mode	Parameter unit
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Interference analysis (non-zero span)	Hz
Interference analysis (zero span)	ns
Real-time spectrum	Hz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Marker - Query marker Y value

ViStatus _VI_FUNC CySAn_GetMarkerY(ViSession Vi, ViInt32 Index, ViReal64* Val)

Function Usage:

Query marker Y value in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Marker index, optional 1~6.

Val

Marker Y value in dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set sweep time

ViStatus _VI_FUNC CySAn_SetSweepTime(ViSession Vi, ViReal64 SweepTime)

Function Usage:

Set sweep time in the current mode. sweep time is the time required for the local oscillator tuning through the selected frequency interval. Sweep time directly affects the time required to complete a test, which does not include the dead time between the completion of a sweep and the start of the next sweep. Sweep time usually varies with Span, RBW and VBW. Sweep time cannot be set when RBW<=1kHz in the Spectrum Analyzer mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTime

Time (ms).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query sweep time

ViStatus _VI_FUNC CySAn_GetSweepTime(ViSession Vi, ViReal64* SweepTime)

Function Usage:

Query the sweep time in the current mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTime

Time (ms).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Set auto sweep time ON/OFF

ViStatus _VI_FUNC CySAn_SetSweepTimeAuto(ViSession Vi, ViBoolean SweepTimeAuto)

Function Usage:

Set sweep time to auto On/Off. When set to On, the instrument will adopt the sweep speed as high as possible; or you can manually increase the sweep time to meet some specific measurement requirements. Sweep time set manually must be > = automatic sweep time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTimeAuto

Automatic sweep time ON/OFF in linear sweep mode: OFF(0) means manual sweep time, and ON(1) means auto scan time.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Sweep - Query auto sweep time ON/OFF

ViStatus _VI_FUNC CySAn_GetSweepTimeAuto(ViSession Vi, ViBoolean* SweepTimeAuto)

Function Usage:

Query sweep time auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SweepTimeAuto

Automatic sweep time ON/OFF in linear sweep mode: OFF(0) means manual sweep time, and ON(1) means auto scan time.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Set time marker position on the waterfall plot

ViStatus _VI_FUNC CySAn_SetTimeMarkerNum(ViSession Vi, ViInt32 TimeMarkerNum)

Function Usage:

Set time marker position on the waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TimeMarkerNum

Time marker position, in the range of 0~317.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Query time marker position on the waterfall plot

ViStatus _VI_FUNC CySAn_GetTimeMarkerNum(ViSession Vi, ViInt32* TimeMarkerNum)

Function Usage:

Query time marker position on the waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TimeMarkerNum

Time marker position, in the range of 0~317.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Set time marker ON/OFF on the waterfall plot

ViStatus _VI_FUNC CySAn_SetTimeMarkerOn(ViSession Vi, ViBoolean TimeMarkerOn)

Function Usage:

Set time marker ON/OFF on the waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TimeMarkerOn

Time marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Query time marker ON/OFF on the waterfall plot

ViStatus _VI_FUNC CySAn_GetTimeMarkerOn(ViSession Vi, ViBoolean* TimeMarkerOn)

Function Usage:

Query time marker ON/OFF on the waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TimeMarkerOn

Time marker ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Set time marker value on the waterfall plot

ViStatus _VI_FUNC CySAn_SetTimeMarkerVal(ViSession Vi, ViReal64 TimeMarkerVal)

Function Usage:

Set time marker value on the waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TimeMarkerVal

Time marker value, in the range of 0ns~6.34s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Query time marker value on the waterfall plot

ViStatus _VI_FUNC CySAn_GetTimeMarkerVal(ViSession Vi, ViReal64* TimeMarkerVal)

Function Usage:

Query time marker value on the waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TimeMarkerVal

Time marker value, in the range of 0ns~6.34s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Set trace status

ViStatus _VI_FUNC CySAn_SetTraceType(ViSession Vi, ViConstString TraceName, ViInt32 TraceType)

Function Usage:

Set trace status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

TraceType

Trace status.

CLRW(0) indicates refresh trace

MAXH(1) indicates maximum hold

MINH(2) indicates minimum hold

VIEW(3) indicates to hold trace

BLANk(4) indicates to hide trace

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Query trace status

ViStatus _VI_FUNC CySAn_GetTraceType(ViSession Vi, ViConstString Trace, ViInt32* TraceType)

Function Usage:

Query trace status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

TraceType

Trace status.

CLRW(0) indicates refresh trace

MAXH(1) indicates maximum hold

MINH(2) indicates minimum hold

VIEW(3) indicates to hold trace

BLANk(4) indicates to hide trace

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Trace - Reset default trace menu

ViStatus _VI_FUNC CySAn_TraceReset(ViSession Vi)

Function Usage:

Reset default trace menu.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Data - Query trace data

ViStatus _VI_FUNC CySAn_FetchTraceY(ViSession Vi, ViConstString TraceName, ViInt32 ArrayLength, ViInt32* ActualPoints, ViReal64 Amplitude[])

Function Usage:

Query trace data in the spectrum analysis mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

TraceName

Trace name.

ArrayLength

Length of trace data received.

ActualPoints

Points.

Amplitude[]

Array of trace data storage.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set audio alarm ON/OFF

ViStatus _VI_FUNC CySAn_SetLimitBeep(ViSession Vi, ViBoolean Beep)

Function Usage:

Set limit alarm to on/off. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every sweep.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Beep

Audio alarm ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query audio alarm ON/OFF

ViStatus _VI_FUNC CySAn_GetLimitBeep(ViSession Vi, ViBoolean* Beep)

Function Usage:

Query limit alarm on/off status. If the audio alarm is turned on, when the limit test switch is turned on and the test fails, the buzzer of the instrument will give a short audio alarm "Beep" after every sweep.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Beep

Audio alarm ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set lower limit display ON/OFF

ViStatus _VI_FUNC CySAn_SetLowLimitDisplayOn(ViSession Vi, ViBoolean Display)

Function Usage:

Set lower limit display ON/OFF

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Lower limit display ON/OFF: OFF (0) for display off, and ON (1) for display on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query lower limit display ON/OFF

ViStatus _VI_FUNC CySAn_GetUpLimitDisplayOn(ViSession Vi, ViBoolean* Display)

Function Usage:

Query lower limit display ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Lower limit display ON/OFF: OFF (0) for display off, and ON (1) for display on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set upper limit display ON/OFF

ViStatus _VI_FUNC CySAn_SetUpLimitDisplayOn(ViSession Vi, ViBoolean Display)

Function Usage:

Set upper limit display ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Upper limit display ON/OFF.

OFF(0) Display off

ON(1) Display on

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query upper limit display ON/OFF

ViStatus _VI_FUNC CySAn_GetUpLimitDisplayOn(ViSession Vi, ViBoolean* Display)

Function Usage:

Query upper limit display ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Display

Upper limit display ON/OFF: OFF (0) for display off, and ON (1) for display on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set lower limit test ON/OFF

ViStatus _VI_FUNC CySAn_SetLowLimitTestOn(ViSession Vi, ViBoolean Test)

Function Usage:

Set lower limit test to On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Lower limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query lower limit test ON/OFF

ViStatus _VI_FUNC CySAn_GetLowLimitTestOn(ViSession Vi, ViBoolean* Test)

Function Usage:

Query lower limit test On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Lower limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Set upper limit test ON/OFF

ViStatus _VI_FUNC CySAn_SetUpLimitTestOn(ViSession Vi, ViBoolean Test)

Function Usage:

Set upper limit test On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Upper limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Query upper limit test ON/OFF

ViStatus _VI_FUNC CySAn_GetUpLimitTestOn(ViSession Vi, ViBoolean* Test)

Function Usage:

Query upper limit test On/Off.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Test

Upper limit test ON/OFF: OFF (0) for test off, and ON (1) for test on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Add lower limit default point

ViStatus _VI_FUNC CySAn_AddLowLimitPoint(ViSession Vi)

Function Usage:

Add lower limit default point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete lower limit current point

ViStatus _VI_FUNC CySAn_DeleteLowLimitCurrentPoint(ViSession Vi)

Function Usage:

Delete lower limit current point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete all lower limit points

ViStatus _VI_FUNC CySAn_DeleteLowLimitAllPoint(ViSession Vi)

Function Usage:

Delete all lower limit edit points.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Edit lower limit point

ViStatus _VI_FUNC CySAn_EditLowLimitPoint(ViSession Vi, ViInt32 Index, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Set lower limit edit point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Limit point index.

Freq

Frequency value (Hz) (0~9.1GHz).

Amp

Amplitude value (dBm) (-174 ~50 dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Add upper limit default point

ViStatus _VI_FUNC CySAn_AddUpLimitPoint(ViSession Vi)

Function Usage:

Add upper limit default point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete upper limit current point

ViStatus _VI_FUNC CySAn_DeleteUpLimitCurrentPoint(ViSession Vi)

Function Usage:

Delete upper limit current point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Delete all upper limit points

ViStatus _VI_FUNC CySAn_DeleteUpLimitAllPoint(ViSession Vi)

Function Usage:

Delete all upper limit edit points.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Limit - Edit upper limit point

ViStatus _VI_FUNC CySAn_EditUpLimitPoint(ViSession Vi, ViInt32 Index, ViReal64 Freq, ViReal64 Amp)

Function Usage:

Set upper limit edit point.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Limit point index.

Freq

Frequency value (Hz) (0~9.1GHz).

Amp

Amplitude value (dBm) (-174~ 50 dBm).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set measurement mode

ViStatus _VI_FUNC CySAn_SetRTSAMeasureType(ViSession Vi, ViInt32 MeasureType)

Function Usage:

Set real-time spectrum measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode: DENSITY(0) means afterglow mode, and SPECTROGRAM(1) means waterfall mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query measurement mode

ViStatus _VI_FUNC CySAn_GetRTSAMeasureType(ViSession Vi, ViInt32* MeasureType)

Function Usage:

Query real-time spectrum measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode: DENSITY(0) means afterglow mode, and SPECTROGRAM(1) means waterfall mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set lower limit of fluorescence probability

ViStatus _VI_FUNC CySAn_SetDPXBPLevel(ViSession Vi, ViReal64 DPXBPLevel)

Function Usage:

Set lower limit of fluorescence probability.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DPXBPLevel

Lower limit of fluorescence probability, in the range of 0% ~ 100%.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query lower limit of fluorescence probability

ViStatus _VI_FUNC CySAn_GetDPXBPLevel(ViSession Vi, ViReal64* DPXBPLevel)

Function Usage:

Query lower limit of fluorescence probability.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DPXBPLevel

Lower limit of fluorescence probability, in the range of 0% ~ 100%.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set upper limit of fluorescence probability

ViStatus _VI_FUNC CySAn_SetDPXRPLevel(ViSession Vi, ViReal64 DPXRPLevel)

Function Usage:

Set upper limit of fluorescence probability.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DPXRPLevel

Upper limit of fluorescence probability, in the range of 0.05% ~ 100%.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query upper limit of fluorescence probability

ViStatus _VI_FUNC CySAn_GetDPXRPLevel(ViSession Vi, ViReal64* DPXRPLevel)

Function Usage:

Query upper limit of fluorescence probability.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DPXRPLevel

Upper limit of fluorescence probability, in the range of 0.05% ~ 100%.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set fluorescent display

ViStatus _VI_FUNC CySAn_SetDPXShownOn(ViSession Vi, ViBoolean DPXShownOn)

Function Usage:

Set fluorescent display.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DPXShownOn

Afterglow spectrum ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query fluorescent display

ViStatus _VI_FUNC CySAn_GetDPXShownOn(ViSession Vi, ViBoolean* DPXShownOn)

Function Usage:

Query fluorescent display.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DPXShownOn

Afterglow spectrum ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure- Reset default measurement menu

ViStatus _VI_FUNC CySAn_Preset(ViSession Vi)

Function Usage:

Reset default measurement menu.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set lower limit of waterfall plot

ViStatus _VI_FUNC CySAn_SetSpectrogramBPLevel(ViSession Vi, ViReal64 SpectrogramBPLevel)

Function Usage:

Set lower limit of waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrogramBPLevel

Lower limit of color scale, in the range of -210 ~30 dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query lower limit of waterfall plot

ViStatus _VI_FUNC CySAn_GetSpectrogramBPLevel(ViSession Vi, ViReal64* SpectrogramBPLevel)

Function Usage:

Query lower limit of waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrogramBPLevel

Lower limit of color scale, in the range of -210 ~ 30 dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set upper limit of waterfall plot

ViStatus _VI_FUNC CySAn_SetSpectrogramRPLevel(ViSession Vi, ViReal64 SpectrogramRPLevel)

Function Usage:

Set upper limit of waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrogramRPLevel

Upper limit of color scale, in the range of -210 ~ 30 dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query upper limit of waterfall plot

ViStatus _VI_FUNC CySAn_GetSpectrogramRPLevel(ViSession Vi, ViReal64* SpectrogramRPLevel)

Function Usage:

Query upper limit of waterfall plot.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpectrogramRPLevel

Upper limit of color scale, in the range of -210 ~30 dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection - Set auto detector ON/OFF

ViStatus _VI_FUNC CySAn_SetDetectorTypeAuto(ViSession Vi, ViBoolean DetectorTypeAuto)

Function Usage:

Set detection to auto On/Off. In auto detection mode, the instrument will automatically select detection type according to different measurements..

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorTypeAuto

Automatic detector ON/OFF: OFF (0) for manual, and (1) for automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Detection- Query auto detector ON/OFF

ViStatus _VI_FUNC CySAn_GetDetectorTypeAuto(ViSession Vi, ViBoolean* DetectorTypeAuto)

Function Usage:

Query detection auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DetectorTypeAuto

Automatic detector ON/OFF: OFF (0) for manual, and (1) for automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete limit file

ViStatus _VI_FUNC CySAn_DeleteLimitFile(ViSession Vi, ViConstString FileName)

Function Usage:

Delete limit file**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Delete all limit files

ViStatus _VI_FUNC CySAn_DeleteDataFileAll(ViSession Vi)

Function Usage:

Delete all limit files.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Load limit line

ViStatus _VI_FUNC CySAn_LoadLimitFile(ViSession Vi, ViConstString FileName)

Function Usage:

Load limit file to limit line**(the command is invalid if the file does not exist and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Store/Load - Store limit file

ViStatus _VI_FUNC CySAn_StoreLimitFile(ViSession Vi, ViConstString FileName)

Function Usage:

Store limit line as limit file**(the file will overwrite an existing file and is valid only for the current storage location)**.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

GSM/EDGE mode function

Frequency - Set band channel number

ViStatus _VI_FUNC CySAn_SetGSMChannelNum(ViSession Vi, ViInt32 GSMChannelNum)

Function Usage:

Set band channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GSMChannelNum

Band channel number.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query band channel number

ViStatus _VI_FUNC CySAn_GetGSMChannelNum(ViSession Vi, ViInt32* GSMChannelNum)

Function Usage:

Query band channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GSMChannelNum

Band channel number.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set band name

ViStatus _VI_FUNC CySAn_SetGSMBand(ViSession Vi, ViConstString GSMBand)

Function Usage:

Set band name.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GSMBand

Band name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query band name

ViStatus _VI_FUNC CySAn_GetGSMBand(ViSession Vi, ViInt32 GSMBandBufferSize, ViChar GSMBand[])

Function Usage:

Query band name.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GSMBandBufferSize

Band data buffer size.

GSMBand[]

Band data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set measurement mode

ViStatus _VI_FUNC CySAn_SetGSMMeasureType(ViSession Vi, ViInt32 MeasureType)

Function Usage:

Set measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode.

DEMod(0) Constellation diagram

SUMMary(1) Demodulation summary

PFail(2) Test pass

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query measurement mode

ViStatus _VI_FUNC CySAn_GetGSMMeasureType(ViSession Vi, ViInt32* MeasureType)

Function Usage:

Query measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode.

DEMod(0) Constellation diagram

SUMMary(1) Demodulation summary

PFail(2) Test pass

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Acquire demodulation data

ViStatus _VI_FUNC CySAn_GetGSMDemodData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Acquire demodulation data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Demodulation data.

Returned data are respectively:

Modulation method(1:8PSK 0:GMSK)
BSIC
NCC
BCC
C/I
Frequency error(Hz)
Frequency error(ppm)
Phase error(rms)
Phase error(peak)
EVM(rms)
EVM(peak)

Origin Offset
Amplitude error

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Acquire constellation data

ViStatus _VI_FUNC CySAn_GetGSMIQData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Acquire demodulated constellation data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Demodulation data.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set I/Q display mode

ViStatus _VI_FUNC CySAn_SetGSMDemodView(ViSession Vi, ViBoolean DemodView)

Function Usage:

Set I/Q display mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodView

Display mode: CONStant(0) means scalar mode, and VECTor(1) means vector mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query I/Q display mode

ViStatus _VI_FUNC CySAn_GetGSMDemodView(ViSession Vi, ViBoolean* DemodView)

Function Usage:

Query I/Q display mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodView

Display mode: CONStant(0) means scalar mode, and VECTor(1) means vector mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Pass test - Configure test items

ViStatus _VI_FUNC CySAn_ConfigPassFail(ViSession Vi, ViInt32 Index, ViReal64 MaxVal, ViReal64 MinVal, ViBoolean OnOff)

Function Usage:

Configure test items.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Test item index , in the range of 1~10.

MaxVal

Max

MinVal

Min

OnOff

Test item ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Pass test - Query test items

ViStatus _VI_FUNC CySAn_GetPassFailConfig(ViSession Vi, ViInt32 Index, ViReal64 MaxVal, ViReal64 MinVal, ViBoolean OnOff)

Function Usage:

Query test items.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Index

Test item index , in the range of 1~10.

MaxVal

Max

MinVal

Min

OnOff

Test item ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Pass test - Get pass test data

ViStatus _VI_FUNC CySAn_GetGSMPassFailData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Get pass test data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Pass test data, Returned data are respectively:

Channel power
C/I
Frequency error(Hz)
Frequency error(ppm)
Phase error(rms)
Phase error(peak)
EVM(rms)
EVM(peak)
Origin Offset
Amplitude error

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Pass test - Load test items

ViStatus _VI_FUNC CySAn_LoadGSMPassFailFile(ViSession Vi, ViConstString FileName)

Function Usage:

Load test items.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

Test item name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Pass test - Reset test items

ViStatus _VI_FUNC CySAn_ResetGSMPassFail(ViSession Vi)

Function Usage:

Reset test items.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Pass test - Save test items

ViStatus _VI_FUNC CySAn_SaveGSMPassFailFile(ViSession Vi, ViConstString FileName)

Function Usage:

Save test items.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

Test item name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Get summary data

ViStatus _VI_FUNC CySAn_GetGSMSummaryData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Get summary data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Summary data. Returned data are respectively:

Channel power
Modulation method (1:8PSK 0:GMSK)
BSIC
NCC
BCC
C/I
Frequency error(Hz)
Frequency error(ppm)
Phase error(rms)
Phase error(peak)
EVM(rms)
EVM(peak)
Origin Offset
Amplitude error

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

5GNR mode function

Frequency - Set frequency raster

ViStatus _VI_FUNC CySAn_SetNRARFCN(ViSession Vi, ViInt32 ARFCN)

Function Usage:

Set frequency raster.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ARFCN

Frequency raster

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query frequency raster

ViStatus _VI_FUNC CySAn_GetNRARFCN(ViSession Vi, ViInt32* ARFCN)

Function Usage:

Query frequency raster.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ARFCN

Frequency raster

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Frequency - Set synchronization raster

ViStatus _VI_FUNC CySAn_SetNRGSCN(ViSession Vi, ViInt32 GSCN)

Function Usage:

Set synchronization raster.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GSCN

Synchronization raster.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query synchronization raster

ViStatus _VI_FUNC CySAn_GetNRGSCN(ViSession Vi, ViInt32* GSCN)

Function Usage:

Query synchronization raster.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

GSCN

Synchronization raster.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set band

ViStatus _VI_FUNC CySAn_SetNRBand(ViSession Vi, ViConstString Band)

Function Usage:

Set band.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Band

Band number.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query band

ViStatus _VI_FUNC CySAn_GetNRBand(ViSession Vi, ViInt32 BandBufferSize, ViChar Band[])

Function Usage:

Set band.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BandBufferSize

Data buffer size.

Band[]

Band data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set SSB offset

ViStatus _VI_FUNC CySAn_SetNRSSBOffset(ViSession Vi, ViReal64 SSBOffset)

Function Usage:

Set SSB offset.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SSBOffset

SSB offset, in the range of -1.92GHz ~ 7.18GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query SSB offset

ViStatus _VI_FUNC CySAn_GetNRSSBOffset(ViSession Vi, ViReal64* SSBOffset)

Function Usage:

Query SSB offset.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SSBOffset

SSB offset, in the range of -1.92GHz~ 7.18GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set automatic SSB search

ViStatus _VI_FUNC CySAn_SSBSearch(ViSession Vi)

Function Usage:

Set automatic SSB search.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set carrier spacing

ViStatus _VI_FUNC CySAn_SetNRSCSCase(ViSession Vi, ViInt32 SCSCase)

Function Usage:

Set carrier spacing.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SCSCase

Carrier spacing.

A(0) Mode A 15kHz

B(1) Mode B 30kHz

C(2) Mode C 30kHz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query carrier spacing

ViStatus _VI_FUNC CySAn_GetNRSCSCase(ViSession Vi, ViInt32* SCSCase)

Function Usage:

Query carrier spacing.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SCSCase

Carrier spacing.

A(0) Mode A 15kHz

B(1) Mode B 30kHz

C(2) Mode C 30kHz

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_SetAttenuationAuto(ViSession Vi, ViBoolean AttenuationAuto)

Function Usage:

Set attenuation to auto On/Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_GetAttenuationAuto(ViSession Vi, ViBoolean* AttenuationAuto)

Function Usage:

Query attenuation auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set measurement mode

ViStatus _VI_FUNC CySAn_SetNRMeasureType(ViSession Vi, ViInt32 NRMeasureType)

Function Usage:

Set measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

NRMeasureType

Measurement mode.

BEAM(0) Beam measurement

MPCI(1) Multi-PCI measurement

OUTdoor(2) Path map outdoor

Indoor(3) Path map indoor

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Measure - Query measurement mode

ViStatus _VI_FUNC CySAn_GetNRMeasureType(ViSession Vi, ViInt32* NRMeasureType)

Function Usage:

Query measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

NRMeasureType

Measurement mode.

BEAM(0) Beam measurement

MPCI(1) Multi-PCI measurement

OUTdoor(2) Path map outdoor

Indoor(3) Path map indoor

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query 5GNR power vs time data

ViStatus _VI_FUNC CySAn_GetNRPVTData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query 5GNR power vs time data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Data, content is

Reference source(0:internal 1:external 2:GPS)
Cell ID
Synchronization symbol
Demodulation symbol
Frequency error
Time error
Subframe power
Frame Avg Power
Power value(A total of 800)

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set power vs time mode's subframe number

ViStatus _VI_FUNC CySAn_SetSubframe(ViSession Vi, ViInt32 Subframe)

Function Usage:

Set power vs time mode's subframe number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Subframe

Subframe number,in the range of 0~9.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query power vs time mode's subframe number

ViStatus _VI_FUNC CySAn_GetSubframe(ViSession Vi, ViInt32*Subframe)

Function Usage:

Query power vs time mode's subframe number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Subframe

Subframe number,in the range of 0~9.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set power vs time mode's trigger mode

ViStatus _VI_FUNC CySAn_SetTrigger(ViSession Vi, ViBoolean Trigger)

Function Usage:

Set power vs time mode's trigger mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Trigger

Trigger mode, EXT(0) means external trigger, GPS(1) means GPS trigger.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query power vs time mode's trigger mode

ViStatus _VI_FUNC CySAn_GetTrigger(ViSession Vi, ViBoolean Trigger)

Function Usage:

Get power vs time mode's trigger mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Trigger

Trigger mode, EXT(0) means external trigger, GPS(1) means GPS trigger.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set current beam index

ViStatus _VI_FUNC CySAn_SetBeamIndex(ViSession Vi, ViInt32 BeamIndex)

Function Usage:

Set current beam index.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BeamIndex

Current beam index, in the range of 0 ~ 7.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query current beam index

ViStatus _VI_FUNC CySAn_GetBeamIndex(ViSession Vi, ViInt32* BeamIndex)

Function Usage:

Query current beam index.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BeamIndex

Current beam index, in the range of 0 ~ 7.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set display type

ViStatus _VI_FUNC CySAn_SetNRBeamView(ViSession Vi, ViBoolean BeamView)

Function Usage:

Set beam measurement display type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BeamView

Beam display mode: SING(0) means single beam, and MULT(1) means multi beam.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query display type

ViStatus _VI_FUNC CySAn_GetNRBeamView(ViSession Vi, ViBoolean* BeamView)

Function Usage:

Query beam measurement display type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BeamView

Beam display mode: SING(0) means single beam, and MULT(1) means multi beam.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set cell ID

ViStatus _VI_FUNC CySAn_SetNRCellID(ViSession Vi, ViInt32 CellID)

Function Usage:

Set beam measurement cell ID.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CellID

Cell ID, in the range of 0~1007.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query cell ID

ViStatus _VI_FUNC CySAn_GetNRCellID(ViSession Vi, ViInt32* CellID)

Function Usage:

Query beam measurement cell ID.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CellID

Cell ID, in the range of 0~1007.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set automatic cell ID ON/OFF

ViStatus _VI_FUNC CySAn_SetNRAutoCellID(ViSession Vi, ViBoolean AutoCellID)

Function Usage:

Set automatic beam measurement cell ID ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoCellID

Automatic cell ID ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query automatic cell ID ON/OFF

ViStatus _VI_FUNC CySAn_GetNRAutoCellID(ViSession Vi, ViBoolean* AutoCellID)

Function Usage:

Query automatic beam measurement cell ID ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoCellID

Automatic cell ID ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set automatic cell ID ON/OFF

ViStatus _VI_FUNC CySAn_SetNRAutoCellID(ViSession Vi, ViBoolean AutoCellID)

Function Usage:

Set automatic beam measurement cell ID ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoCellID

Automatic cell ID ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set constellation channel

ViStatus _VI_FUNC CySAn_SetNRConstellationChannel(ViSession Vi, ViInt32 ConstellationChannel)

Function Usage:

Set beam measurement constellation channel.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ConstellationChannel

Channel number: optional PSS(0), SSS(1), PBCH(2) and DMRS(3).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query constellation channel

ViStatus _VI_FUNC CySAn_GetNRConstellationChannel(ViSession Vi, ViInt32* ConstellationChannel)

Function Usage:

Query beam measurement constellation channel.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ConstellationChannel

Channel number: optional PSS(0), SSS(1), PBCH(2) and DMRS(3).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set constellation ON/OFF

ViStatus _VI_FUNC CySAn_SetNRConstellationOn(ViSession Vi, ViBoolean ConstellationOn)

Function Usage:

Set beam measurement constellation ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ConstellationOn

Constellation diagram ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query constellation ON/OFF

ViStatus _VI_FUNC CySAn_GetNRConstellationOn(ViSession Vi, ViBoolean* ConstellationOn)

Function Usage:

Query beam measurement constellation ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ConstellationOn

Constellation diagram ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set MIB ON/OFF

ViStatus _VI_FUNC CySAn_SetMIBOn(ViSession Vi, ViBoolean MIBOn)

Function Usage:

Set broadcast channel MIB ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MIBOn

MIB ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query MIB ON/OFF

ViStatus _VI_FUNC CySAn_GetMIBOn(ViSession Vi, ViBoolean MIBOn)

Function Usage:

Query broadcast channel MIB ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MIBOn

MIB ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query beam measurement results

ViStatus _VI_FUNC CySAn_GetNRBeamData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query beam measurement results.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Measurement data

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query constellation IQ

ViStatus _VI_FUNC CySAn_GetNRIQData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query beam measurement constellation IQ.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Constellation IQ data.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query broadcast channel MIB

ViStatus _VI_FUNC CySAn_GetMIBData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query beam measurement broadcast channel MIB.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

Broadcast channel MIB data.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - MPCI measurement - Query MPCI measurement results

ViStatus _VI_FUNC CySAn_GetNRMPCIData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query MPCI measurement results.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Data buffer size.

Val[]

MPCI Measurement data.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - MPCI measurement - Set MPCI beam number

ViStatus _VI_FUNC CySAn_SetMPCIBeamNum(ViSession Vi, ViInt32 BeamNum)

Function Usage:

Set MPCI beam number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BeamNum

Beam number, in the range of 1 ~ 8.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - MPCI measurement - Query MPCI beam number

ViStatus _VI_FUNC CySAn_GetMPCIBeamNum(ViSession Vi, ViInt32* BeamNum)

Function Usage:

Query MPCI beam number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BeamNum

Beam number, in the range of 1 ~ 8.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - MPCI measurement - Set current MPCI display page number

ViStatus _VI_FUNC CySAn_SetMPCIPage(ViSession Vi, ViInt32 MPCIPage)

Function Usage:

Set current MPCI display page number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MPCIPage

Current page number, in the range of 1 ~ 6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - MPCI measurement - Query current MPCI display page number

ViStatus _VI_FUNC CySAn_GetMPCIPage(ViSession Vi, ViInt32* MPCIPage)

Function Usage:

Query current MPCI display page number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MPCIPage

Current page number, in the range of 1 ~ 6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Set MPCI display type

ViStatus _VI_FUNC CySAn_SetMPCIView(ViSession Vi, ViBoolean MPCIView)

Function Usage:

Set MPCI display type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MPCIView

Display type: CHARt(0) means bar chart, and TABLe(1) means table.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Beam measurement - Query MPCI display type

ViStatus _VI_FUNC CySAn_GetMPCIView(ViSession Vi, ViBoolean* MPCIView)

Function Usage:

Query MPCI display type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MPCIView

Display type: CHARt(0) means bar chart, and TABLe(1) means table.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map indoor - Delete mark

ViStatus _VI_FUNC CySAn_NRMapDelete(ViSession Vi)

Function Usage:

Delete marks on the indoor map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map indoor - Delete all marks

ViStatus _VI_FUNC CySAn_NRMapDeleteAll(ViSession Vi)

Function Usage:

Delete all marks on the indoor map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map indoor - Mark

ViStatus _VI_FUNC CySAn_NRIndoorMark(ViSession Vi)

Function Usage:

Set indoor map marks.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map indoor - Move down

ViStatus _VI_FUNC CySAn_NRIndoorMoveDown(ViSession Vi)

Function Usage:

Move indoor path map down.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map indoor - Move left

ViStatus _VI_FUNC CySAn_NRIndoorMoveLeft(ViSession Vi)

Function Usage:

Move indoor path map to the left.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map indoor - Move right

ViStatus _VI_FUNC CySAn_NRIndoorMoveRight(ViSession Vi)

Function Usage:

Move indoor path map to the right.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map indoor - Move up

ViStatus _VI_FUNC CySAn_NRIndoorMoveUp(ViSession Vi)

Function Usage:

Move indoor path map up.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Set collection ON/OFF

ViStatus _VI_FUNC CySAn_SetStartCollect(ViSession Vi, ViBoolean StartCollect)

Function Usage:

Set outdoor map collection ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

StartCollect

Collection ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Query collection ON/OFF

ViStatus _VI_FUNC CySAn_SetStartCollect(ViSession Vi, ViBoolean StartCollect)

Function Usage:

Query outdoor map collection ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

StartCollect

Collection ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Set repeat type

ViStatus _VI_FUNC CySAn_SetRepeatedType(ViSession Vi, ViBoolean OutdoorRepeatedType)

Function Usage:

Set outdoor path map repeat type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

OutdoorRepeatedType

Repeat type: TIME(0) means time and DISTance(1) means distance.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Query repeat type

ViStatus _VI_FUNC CySAn_GetRepeatedType(ViSession Vi, ViBoolean* OutdoorRepeatedType)

Function Usage:

Query outdoor path map repeat type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

OutdoorRepeatedType

Repeat type: TIME(0) means time and DISTance(1) means distance.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Set repeat distance

ViStatus _VI_FUNC CySAn_SetNRRepeatDistance(ViSession Vi, ViReal64 RepeatDistance)

Function Usage:

Set outdoor path map repeat distance.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatDistance

Repeat distance, in the range of 50~1000m.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Query repeat distance

ViStatus _VI_FUNC CySAn_GetNRRepeatDistance(ViSession Vi, ViReal64* RepeatDistance)

Function Usage:

Query outdoor path map repeat distance.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatDistance

Repeat distance, in the range of 50~1000m.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Set repeat time

ViStatus _VI_FUNC CySAn_SetNRRepeatTime(ViSession Vi, ViReal64 RepeatTime)

Function Usage:

Set outdoor path map repeat time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatTime

Repeat time, in the range of 5~ 600s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Query repeat time

ViStatus _VI_FUNC CySAn_GetNRRepeatTime(ViSession Vi, ViReal64* RepeatTime)

Function Usage:

Set outdoor path map repeat time.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RepeatTime

Repeat time, in the range of 5 ~600s.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Zoom-in path map

ViStatus _VI_FUNC CySAn_NRZoomIn(ViSession Vi)

Function Usage:

Zoom-in outdoor path map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map outdoor - Zoom-out path map

ViStatus _VI_FUNC CySAn_NRZoomOut(ViSession Vi)

Function Usage:

Zoom-out outdoor path map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Set lower limit

ViStatus _VI_FUNC CySAn_SetNRMapLowLimit(ViSession Vi, ViReal32 IndoorLowLimit)

Function Usage:

Set lower limit of the path map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IndoorLowLimit

lower limit, in the range of 174~49dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Query lower limit

ViStatus _VI_FUNC CySAn_GetNRMapLowLimit(ViSession Vi, ViReal32* IndoorLowLimit)

Function Usage:

Query lower limit of the path map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IndoorLowLimit

lower limit, in the range of 174~49dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Set upper limit

ViStatus _VI_FUNC CySAn_SetNRMapUpLimit(ViSession Vi, ViReal32 IndoorUpLimit)

Function Usage:

Set upper limit of the path map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IndoorUpLimit

Upper limit, in the range of 173 ~ 50dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Query upper limit

ViStatus _VI_FUNC CySAn_GetNRMapUpLimit(ViSession Vi, ViReal32* IndoorUpLimit)

Function Usage:

Query upper limit of the path map.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

IndoorUpLimit

Upper limit, in the range of 173 ~50dBm.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Load map data

ViStatus _VI_FUNC CySAn_NRMapLoad(ViSession Vi, ViConstString Name)

Function Usage:

Load map data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Name

Map data name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Set measurement type

ViStatus _VI_FUNC CySAn_SetNRMapMeasureType(ViSession Vi, ViInt32

MapMeasureType)

Function Usage:

Set measurement type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapMeasureType

Measurement type: MPCI(0) means PCI and BEAM(1) means beam.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Query measurement type

ViStatus _VI_FUNC CySAn_GetNRMapMeasureType(ViSession Vi, ViInt32* MapMeasureType)

Function Usage:

Query measurement type.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MapMeasureType

Measurement type: MPCI(0) means PCI and BEAM(1) means beam.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Save BMP file

ViStatus _VI_FUNC CySAn_NRMapSaveBmp(ViSession Vi, ViConstString FileName)

Function Usage:

Save map BMP file.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Save CSV file

ViStatus _VI_FUNC CySAn_NRMapSaveCSV(ViSession Vi, ViConstString FileName)

Function Usage:

Save map CSV file.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Path map - Save collection data

ViStatus _VI_FUNC CySAn_NRMapSaveData(ViSession Vi, ViConstString FileName)

Function Usage:

Save map collection data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

FileName

File name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

LTE analysis mode function

Frequency - Set EARFCN

ViStatus _VI_FUNC CySAn_SetLTEEARFCN(ViSession Vi, ViInt32 EARFCN)

Function Usage:

Set EARFCN.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

EARFCN

EARFCN value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query EARFCN

ViStatus _VI_FUNC CySAn_GetLTEEARFCN(ViSession Vi, ViInt32* EARFCN)

Function Usage:

Query EARFCN.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

EARFCN

EARFCN value.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set band

ViStatus _VI_FUNC CySAn_SetLTEBand(ViSession Vi, ViConstString Band)

Function Usage:

Set band.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

Band

Band name.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query band

ViStatus _VI_FUNC CySAn_GetLTEBand(ViSession Vi, ViInt32 BandBufferSize, ViChar Band[])

Function Usage:

Query band.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

BandBufferSize

Band buffer size.

Band[]

Band data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Set channel bandwidth

ViStatus _VI_FUNC CySAn_SetLTEChannelBW(ViSession Vi, ViReal64 ChannelBW)

Function Usage:

Set channel bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelBW

Channel bandwidth

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Frequency - Query channel bandwidth

ViStatus _VI_FUNC CySAn_GetLTEChannelBW(ViSession Vi, ViReal64* ChannelBW)

Function Usage:

Query channel bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ChannelBW

Channel bandwidth.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Set auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_SetAttenuationAuto(ViSession Vi, ViBoolean AttenuationAuto)

Function Usage:

Set attenuation to auto On/Off. When attenuation Auto is turned on, the instrument will set corresponding attenuation according to the reference value automatically.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Amplitude - Query auto attenuation ON/OFF

ViStatus _VI_FUNC CySAn_GetAttenuationAuto(ViSession Vi, ViBoolean* AttenuationAuto)

Function Usage:

Query attenuation auto On/Off status.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AttenuationAuto

Automatic attenuation ON/OFF: OFF (0) means manual, and (1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set cell ID

ViStatus _VI_FUNC CySAn_SetLTECellID(ViSession Vi, ViInt32 CellID)

Function Usage:

Set channel bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CellID

Cell ID.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query cell ID

ViStatus _VI_FUNC CySAn_GetLTECellID(ViSession Vi, ViInt32* CellID)

Function Usage:

Query cell ID.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CellID

Cell ID.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set automatic cell ID ON/OFF

ViStatus _VI_FUNC CySAn_SetLTEAutoCellID(ViSession Vi, ViBoolean AutoCellID)

Function Usage:

Set automatic cell ID ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoCellID

Automatic cell ID ON/OFF: OFF (0) means manual off, and ON (1) means automatic on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query automatic cell ID ON/OFF

ViStatus _VI_FUNC CySAn_GetLTEAutoCellID(ViSession Vi, ViBoolean* AutoCellID)

Function Usage:

Query automatic cell ID ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

AutoCellID

Automatic cell ID ON/OFF: OFF (0) means manual off, and ON (1) means automatic on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set CFI configuration

ViStatus _VI_FUNC CySAn_SetLTECFI(ViSession Vi, ViInt32 CFI)

Function Usage:

Set CFI configuration.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CFI

CFI configuration

AUTO (0) Auto

MAN1(1) Manual 1

MAN2(2) Manual 2

MAN3(3) Manual 3

MAN4(4) Manual 4

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query CFI configuration

ViStatus _VI_FUNC CySAn_GetLTECFI(ViSession Vi, ViInt32* CFI)

Function Usage:

Query CFI configuration.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CFI

CFI configuration

AUTO (0) Auto

MAN1(1) Manual 1

MAN2(2) Manual 2

MAN3(3) Manual 3

MAN4(4) Manual 4

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set cyclic prefix

ViStatus _VI_FUNC CySAn_SetLTECPMode(ViSession Vi, ViBoolean CPMode)

Function Usage:

Set cyclic prefix.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CPMode

Cyclic prefix: NORMal(0) means normal and EXTend(1) means extended.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query cyclic prefix

ViStatus _VI_FUNC CySAn_GetLTECPMode(ViSession Vi, ViBoolean* CPMode)

Function Usage:

Query cyclic prefix.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

CPMode

Cyclic prefix: NORMal(0) means normal and EXTend(1) means extended.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set measurement mode

ViStatus _VI_FUNC CySAn_SetLTEDemodMode(ViSession Vi, ViBoolean DemodMode)

Function Usage:

Set measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodMode

Measurement mode: FDD(0) means FDD, and TDD(1) means TDD.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query measurement mode

ViStatus _VI_FUNC CySAn_GetLTEDemodMode(ViSession Vi, ViBoolean* DemodMode)

Function Usage:

Query measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DemodMode

Measurement mode: FDD(0) means FDD, and TDD(1) means TDD.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set PHICH configuration Mode

ViStatus _VI_FUNC CySAn_SetPHICHExtend(ViSession Vi, ViBoolean PHICHExtend)

Function Usage:

Set PHICH configuration mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PHICHExtend

PHICH configuration: OFF(0) means normal, and ON(1) means extended.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query PHICH configuration mode

ViStatus _VI_FUNC CySAn_GetPHICHExtend(ViSession Vi, ViBoolean* PHICHExtend)

Function Usage:

Query PHICH configuration mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PHICHExtend

PHICH configuration: OFF(0) means normal, and ON(1) means extended.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set PHICH number

ViStatus _VI_FUNC CySAn_SetLTENGType(ViSession Vi, ViInt32 NGType)

Function Usage:

Set PHICH number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

NGType

PHICH number.

ONESixth(0) 1/6

HALF(1) 1/2

ONE(2) 1

TWO(3) 2

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query PHICH number

ViStatus _VI_FUNC CySAn_GetLTENGType(ViSession Vi, ViInt32* NGType)

Function Usage:

Query PHICH number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

NGType

PHICH number.

ONESixth(0) 1/6

HALF(1) 1/2

ONE(2) 1

TWO(3) 2

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set port number

ViStatus _VI_FUNC CySAn_SetLTEPortConfig(ViSession Vi, ViInt32 PortConfig)

Function Usage:

Set port number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PortConfig

Port number.

AUTO (0) Auto

ONE(1) 1

TWO(2) 2

FOUR(3) 3

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query port number

ViStatus _VI_FUNC CySAn_GetLTEPortConfig(ViSession Vi, ViInt32* PortConfig)

Function Usage:

Query port number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

PortConfig

Port number.

AUTO (0) Auto

ONE(1) 1

TWO(2) 2

FOUR(3) 3

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set special subframe

ViStatus _VI_FUNC CySAn_SetSpecialSubFrame(ViSession Vi, ViInt32 SpecialSubFrame)

Function Usage:

Set special subframe.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpecialSubFrame

Special subframe number, in the range of 0 ~ 9.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query special subframe

ViStatus _VI_FUNC CySAn_GetSpecialSubFrame(ViSession Vi, ViInt32* SpecialSubFrame)

Function Usage:

Query special subframe.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SpecialSubFrame

Special subframe number, in the range of 0 ~ 9.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set subframe

ViStatus _VI_FUNC CySAn_SetLTESubFrame(ViSession Vi, ViInt32 SubFrame)

Function Usage:

Set subframe.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SubFrame

Subframe number, in the range of 0 ~ 9.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query subframe

ViStatus _VI_FUNC CySAn_GetLTESubFrame(ViSession Vi, ViInt32* SubFrame)

Function Usage:

Query subframe.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SubFrame

Subframe number, in the range of 0 ~ 9.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Set uplink/downlink configuration

ViStatus _VI_FUNC CySAn_SetLTEUpDownLink(ViSession Vi, ViInt32 UpDownLink)

Function Usage:

Set uplink/downlink configuration.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

UpDownLink

Uplink/downlink number, in the range of 0 ~ 6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measurement setup - Query uplink/downlink configuration

ViStatus _VI_FUNC CySAn_GetLTEUpDownLink(ViSession Vi, ViInt32* UpDownLink)

Function Usage:

Query uplink/downlink configuration.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

UpDownLink

Uplink/downlink number, in the range of 0 ~ 6.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation channel bandwidth

ViStatus _VI_FUNC CySAn_SetCaChannelbw(ViSession Vi, ViInt32 SegNum, ViReal64 CAChannelBW)

Function Usage:

Set carrier aggregation channel bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAChannelBW

Carrier aggregation channel bandwidth, in the range of 1.4MHz ~ 20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregation channel bandwidth

ViStatus _VI_FUNC CySAn_GetCaChannelbw(ViSession Vi, ViInt32 SegNum, ViReal64* CAChannelBW)

Function Usage:

Query carrier aggregation channel bandwidth.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAChannelBW

Carrier aggregation channel bandwidth, in the range of 1.4MHz ~ 20MHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation cell ID

ViStatus _VI_FUNC CySAn_SetCaCellid(ViSession Vi, ViInt32 SegNum, ViInt32 CACellID)

Function Usage:

Set carrier aggregation cell ID.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACellID

Carrier aggregation cell ID, in the range of 0-503.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregation cell ID

ViStatus _VI_FUNC CySAn_GetCaCellid(ViSession Vi, ViInt32 SegNum, ViInt32* CACellID)

Function Usage:

Query carrier aggregation cell ID.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACellID

Carrier aggregation cell ID, in the range of 0~503.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set automatic carrier aggregation cell ID ON/OFF

ViStatus _VI_FUNC CySAn_SetCaAutoCellid(ViSession Vi, ViInt32 SegNum, ViBoolean CAAutoCellID)

Function Usage:

Set automatic carrier aggregation cell ID ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAAutoCellID

Automatic carrier aggregation cell ID ON/OF: OFF (0) means manual, and ON(1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query automatic carrier aggregation cell ID ON/OFF

ViStatus _VI_FUNC CySAn_GetCaAutoCellid(ViSession Vi, ViInt32 SegNum, ViBoolean* CAAutoCellID)

Function Usage:

Query automatic carrier aggregation cell ID ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAAutoCellID

Automatic carrier aggregation cell ID ON/OF: OFF (0) means manual, and ON(1) means automatic.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation center frequency

ViStatus _VI_FUNC CySAn_SetCaCenterFreq(ViSession Vi, ViInt32 SegNum, ViReal64 CACenterFreq)

Function Usage:

Set carrier aggregation center frequency.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACenterFreq

Carrier aggregation center frequency, in the range of 5Hz~9.099999995GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregation center frequency

ViStatus _VI_FUNC CySAn_SetCaCenterFreq(ViSession Vi, ViInt32 SegNum, ViReal64 CACenterFreq)

Function Usage:

Query carrier aggregation center frequency.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACenterFreq

Carrier aggregation center frequency, in the range of 5Hz~9.099999995GHz.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation CFI configuration

ViStatus _VI_FUNC CySAn_SetCacfi(ViSession Vi, ViInt32 SegNum, ViInt32 CACFI)

Function Usage:

Set carrier aggregation CFI configuration.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACFI

Carrier aggregation CFI configuration

AUTO (0) Auto

MAN1(1) Manual 1

MAN2(2) Manual 2

MAN3(3) Manual 3

MAN4(4) Manual 4

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregation CFI configuration

ViStatus _VI_FUNC CySAn_GetCacfi(ViSession Vi, ViInt32 SegNum, ViInt32* CACFI)

Function Usage:

Query carrier aggregation CFI configuration.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACFI

Carrier aggregation CFI configuration

AUTO (0) Auto

MAN1(1) Manual 1

MAN2(2) Manual 2

MAN3(3) Manual 3

MAN4(4) Manual 4

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation cyclic prefix

ViStatus _VI_FUNC CySAn_SetCacpMode(ViSession Vi, ViInt32 SegNum, ViBoolean CACPMode)

Function Usage:

Set carrier aggregation cyclic prefix.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACPMode

Carrier aggregation cyclic prefix: NORMal(0) means normal mode and EXTend(1) means extended mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus

means failure.

Measure - Query carrier aggregation cyclic prefix

ViStatus _VI_FUNC CySAn_GetCacpMode(ViSession Vi, ViInt32 SegNum, ViBoolean* CACPMode)

Function Usage:

Query carrier aggregation cyclic prefix.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CACPMode

Carrier aggregation cyclic prefix: NORMal(0) means normal mode and EXTend(1) means extended mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation PHICH mode

ViStatus _VI_FUNC CySAn_SetCAPhichExtend(ViSession Vi, ViInt32 SegNum, ViBoolean CAPhichExtend)

Function Usage:

Set carrier aggregation PHICH mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAPhichExtend

PHICH configuration: OFF(0) means normal mode, and ON(1) means extended mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregation PHICH mode

ViStatus _VI_FUNC CySAn_GetCAPhichExtend(ViSession Vi, ViInt32 SegNum, ViBoolean* CAPhichExtend)

Function Usage:

Query carrier aggregation PHICH mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAPhichExtend

PHICH configuration: OFF(0) means normal mode, and ON(1) means extended mode.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation PHICH number

ViStatus _VI_FUNC CySAn_SetCaNg(ViSession Vi, ViInt32 SegNum, ViInt32 CANg)

Function Usage:

Set carrier aggregation PHICH number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CANg

PHICH configuration

ONESixth(0) 1/6

HALF(1) 1/2

ONE(2) 1

TWO(3) 2

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregation PHICH number

ViStatus _VI_FUNC CySAn_GetCaNg(ViSession Vi, ViInt32 SegNum, ViInt32* CANg)

Function Usage:

Query carrier aggregation PHICH number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CANg

PHICH configuration

ONESixth(0) 1/6

HALF(1) 1/2

ONE(2) 1

TWO(3) 2

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set carrier aggregation ON/OFF

ViStatus _VI_FUNC CySAn_SetCaOn(ViSession Vi, ViInt32 SegNum, ViBoolean CAOn)

Function Usage:

Set carrier aggregation ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAOn

Carrier aggregation ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregation ON/OFF

ViStatus _VI_FUNC CySAn_GetCaOn(ViSession Vi, ViInt32 SegNum, ViBoolean* CAOn)

Function Usage:

Query carrier aggregation ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

CAOn

Carrier aggregation ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set control channel number

ViStatus _VI_FUNC CySAn_SetLTESelectChannel(ViSession Vi, ViInt32 SelectChannel)

Function Usage:

Set control channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

SelectChannel

Control channel number. Options are PSS(0), SSS(1), PBCH(2), PCFICH(3), PHICH(4), PDCCH(5) and RS(6).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query control channel number

ViStatus _VI_FUNC CySAn_SetLTESelectChannel(ViSession Vi, ViInt32 SelectChannel)

Function Usage:

Query control channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SegNum

Channel number.

SelectChannel

Control channel number. Options are PSS(0), SSS(1), PBCH(2), PCFICH(3), PHICH(4), PDCCH(5) and RS(6).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set control channel constellation diagram

ViStatus _VI_FUNC CySAn_SetControlChConstellation(ViSession Vi, ViBoolean ControlChConstellation)

Function Usage:

Set control channel constellation diagram.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ControlChConstellation

Channel channel constellation diagram ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query control channel constellation diagram

ViStatus _VI_FUNC CySAn_GetControlChConstellation(ViSession Vi, ViBoolean*

ControlChConstellation)

Function Usage:

Query control channel constellation diagram.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ControlChConstellation

Channel channel constellation diagram ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set control channel RS channel

ViStatus _VI_FUNC CySAn_SetSelectRSChannel(ViSession Vi, ViInt32 SelectRSChannel)

Function Usage:

Set control channel RS channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SelectRSChannel

Control channel RS channel number. Options are RS0(0), RS1(1), RS2(2), RS3(3) and RSALL(4).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query control channel RS channel

ViStatus _VI_FUNC CySAn_GetSelectRSChannel(ViSession Vi, ViInt32* SelectRSChannel)

Function Usage:

Query control channel RS channel number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

SelectRSChannel

Control channel RS channel number. Options are RS0(0), RS1(1), RS2(2), RS3(3) and RSALL(4).

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query carrier aggregated demodulated data

ViStatus _VI_FUNC CySAn_GetCAData(ViSession Vi, ViInt32 ValBufferSize, ViReal32

Val[], ViInt32* ValActualSize)

Function Usage:

Query carrier aggregated demodulated data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Returns the demodulation data of the five carriers sequentially, a total of 5*12 data,the data content in each carrier is**:**

Center Frequency
Synchronization symbol
Pss Frequency
Sss Frequency
Pbch Frequency
Rs Frequency
Pss EVM
Sss EVM
Pbch EVM
Rs EVM
Cell ID
Frequency error

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query constellation demodulation data

ViStatus _VI_FUNC CySAn_GetCONData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query constellation demodulated data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Data, the content is:

Synchronization symbol
Frequency error
Time error
Qpsk EVM(rms)

qam16 EVM(rms)
qam16 EVM(rms)
Pdsch EVM(rms)
Pdsch EVM(peak)
Pdsch power
RS power
Constellation Data. The IQ data is in order. The total number can be calculated based on the amount of remaining data.A set of IQ data is 8 bytes 64 bits.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query data allocacation map demodulation data

ViStatus _VI_FUNC CySAn_GetDAMData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query constellation demodulated data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Data, the content is:

Synchronization symbol
OFDM Symbol Power
Data Utilization
Pdsch power
Rs power
RB power value of subframes 0-9. The number of RB per subframe is transformed according to the channel bandwidth. The total quantity can be calculated according to the amount of remaining data. A RB power value is 4 bytes 32 bits.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query data channel demodulation data

ViStatus _VI_FUNC CySAn_GetDCData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query data channel demodulation data

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Data, the content is:

Synchronization symbol
Cell ID
Frequency error
Time error
Pdsch EVM(rms)
Pdsch EVM(peak)
Currently rb power
IQ Offset
All RB power values of the current subframe. Quantity is transformed according to the channel bandwidth. The total quantity can be calculated according to the amount of remaining data. A RB power value is 4 bytes 32 bits.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query ID Scanner demodulation data

ViStatus _VI_FUNC CySAn_GetIDSData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query data channel demodulation data

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Data, the content is:

Cell ID
rsrp
rsrq
sinr
rssi
Pss power
Sss power

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query power vs time demodulation data

ViStatus _VI_FUNC CySAn_GetPVTData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query power vs time demodulation data

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Data, the content is:

Synchronization symbol
Cell ID
Frequency error
Time error
Subframe power
First Slot power
Second Slot power
Frame Avg Power
IQ Offset
IQ Imbalance
1200 Frame Power

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query time alignment demodulation data

ViStatus _VI_FUNC CySAn_GetTAData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query time alignment demodulation data

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Data, the content is:

Synchronization symbol
Cell ID
Frequency error
Time error
rs0 power
rs0 EVM(rms)
rs0 time deviation
rs1 power
rs1 EVM(rms)
rs1 time deviation
rs2 power
rs2 EVM(rms)
rs2 time deviation
rs3 power
rs3 EVM(rms)
rs3 time deviation

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query control channel demodulation data

ViStatus _VI_FUNC CySAn_GetCCData(ViSession Vi, ViInt32 ValBufferSize, ViReal32 Val[], ViInt32* ValActualSize)

Function Usage:

Query control channel demodulation data.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

ValBufferSize

Buffer size of data.

Val[]

Returns EVM and power data for each channel,if the constellation diagram is displayed,constellation data is also returned,content is:

Synchronization symbol

Cell ID
Frequency error
Time error
EVM(rms)、EVM(peak) and power of each channel in turn. The order of channels is：pss, sss, pbch, pcfich, phich, pdcch, rs, rs0, rs1, rs2, rs3.
Subframe Power
IQ Offset
OFDM Symbol Power
Constellation Data.The IQ data is in order,and the total number is related to the channel displayed by the constellation.It can be calculated based on the amount of remaining data.A set of IQ data is 8 bytes 64 bits.

ValActualSize

Actual size of data.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set data channel mark ON/OFF

ViStatus _VI_FUNC CySAn_SetDataChMarker(ViSession Vi, ViBoolean DataChMarker)

Function Usage:

Set data channel mark ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DataChMarker

Data channel mark ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query data channel mark ON/OFF

ViStatus _VI_FUNC CySAn_GetDataChMarker(ViSession Vi, ViBoolean* DataChMarker)

Function Usage:

Query data channel mark ON/OFF.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DataChMarker

Data channel mark ON/OFF: OFF (0) means off, and ON (1) means on.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set data channel RB number

ViStatus _VI_FUNC CySAn_SetRBIndex(ViSession Vi, ViInt32 RBIndex)

Function Usage:

Set data channel RB number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RBIndex

RB number, in the range of 0~99.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query data channel RB number

ViStatus _VI_FUNC CySAn_GetRBIndex(ViSession Vi, ViInt32* RBIndex)

Function Usage:

Query data channel RB number.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

RBIndex

RB number, in the range of 0~99.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure – Set data allocacation map color limit up

ViStatus _VI_FUNC CySAn_SetDAMLimitUp(ViSession Vi, ViReal32 DAMLimitUp)

Function Usage:

Set data allocation map color limit up.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DAMLimitUp

Color limit up, in the range of -209~30.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure – Query data allocacation map color limit up

ViStatus _VI_FUNC CySAn_GetDAMLimitUp(ViSession Vi, ViReal32* DAMLimitUp)

Function Usage:

Query data allocation map color limit up.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DAMLimitUp

Color limit up, in the range of -209~30.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure – Set data allocacation map color limit down

ViStatus _VI_FUNC CySAn_SetDAMLimitDown(ViSession Vi, ViReal32 DAMLimitUp)

Function Usage:

Set data allocation map color limit down.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DAMLimitDown

Color limit down, in the range of -210~29.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure – Query data allocacation map color limit down

ViStatus _VI_FUNC CySAn_GetDAMLimitDown(ViSession Vi, ViReal32* DAMLimitUp)

Function Usage:

Query data allocation map color limit down.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

DAMLimitDown

Color limit down, in the range of -210~29.

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Set measurement mode

ViStatus _VI_FUNC CySAn_SetLTEMeasureType(ViSession Vi, ViInt32 MeasureType)

Function Usage:

Set measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode.

CONS(0)	Constellation diagram
DC(1)	Data channel
CC(2)	Control channel
TA(3)	Time alignment error
CA(4)	Carrier aggregation
IDS(5)	ID sweep
PVT(6)	Power VS Time
DAM(7)	Data distribution chart

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.

Measure - Query measurement mode

ViStatus _VI_FUNC CySAn_GetLTEMeasureType(ViSession Vi, ViInt32* MeasureType)

Function Usage:

Query measurement mode.

Parameter list:

Vi

Instrument handle returned by the function, communicating with the instrument.

MeasureType

Measurement mode.

CONS(0)	Constellation diagram
DC(1)	Data channel
CC(2)	Control channel
TA(3)	Time alignment error
CA(4)	Carrier aggregation
IDS(5)	ID sweep
PVT(6)	Power VS Time
DAM(7)	Data distribution chart

Returned Value:

Returned value indicates the execution result of the function: 0 means success, and minus means failure.