Agilent E5022A/B and E5023A Hard Disk Read/Write Test System

Programming Manual

27th Edition

Software Revision
This manual applies to the system which has
the software revision B.04.00


Agilent Technologies

Part No. E5023-90091
March 2004

Printed in Japan

Notices

The information contained in this document is subject to change without notice.

Microsoft is a registered trademark of Microsoft Corporation.

Windows 95/2000 and Visual Basic are registered trademarks of Microsoft Corporation.

Acrobat Reader is are registered trademark of Adobe Corporation.

Manual Printing History

The manual's printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates that are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

November 1998	1st Edition
January 1999	2nd Edition
May 1999	3rd Edition
June 1999	4th Edition
August 1999	5th Edition
August 1999	6th Edition
November 1999	7th Edition
December 1999	8th Edition
February 2000	9th Edition
April 2000	10th Edition
June 2000	11th Edition
July 2000	12th Edition
July 2000	13th Edition
October 2000	14th Edition
November 2000	15th Edition
December 2000	16th Edition
March 2001	17th Edition
June 2001	18th Edition
August 2001	19th Edition
August 2001	20th Edition
October 2001	21st Edition
February 2002	22nd Edition (part number: E5023-90041)
July 2002	23rd Edition (part number: E5023-90051)
September 2002	24th Edition (part number: E5023-90061)
January 2003	25th Edition (part number: E5023-90071)
March 2003	26th Edition (part number: E5023-90081)
March 2004	27th Edition (part number: E5023-90091)

Assistance

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products.

For any assistance, contact your nearest Agilent Technologies Sales and Service Office. Addresses are provided at the back of this manual.

Sample Program

A sample program is installed in the PC. The sample program list is described in the programming manual.

The customer shall have the personal, non-transferable rights to use, copy, or modify SAMPLE PROGRAMS in this manual for the Customer's internal operations. The customer shall use the SAMPLE PROGRAMS solely and exclusively for their own purpose and shall not license, lease, market, or distribute the SAMPLE PROGRAMS or modification of any part thereof.

Agilent Technologies shall not be liable for the quality, performance, or behavior of the SAMPLE PROGRAMS. Agilent Technologies especially disclaims that the operation of the SAMPLE PROGRAMS shall be uninterrupted or error free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary rights by the SAMPLE PROGRAMS or their use. Agilent Technologies does not warrant that the SAMPLE PROGRAMS are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

Contents

Contents

Contents

Contents

1 Introduction

Introduction

Welcome to the Agilent Technologies E5022/E5023 Programming Manual. This manual provides the information on measurement functions of E5022/E5023 used for programming.

About This Manual

This Manual contains the following chapters:

- Chapter 1, Introduction : gives an overview of the product.
- Chapter 2, Programming : describes how to make a program and provides example programs in Visual Basic.
- Chapter 3, Function Reference : provides information on measurement functions for the Agilent E5022/E5023.
- Chapter 4, Function Reference for measurement using oscilloscope: provides information on measurement functions when oscilloscope is used. The oscilloscope is provided as a option.
- Chapter 5, Function Reference for measureement using the 4396B: provides information on measurement functions when the 4396B is used. The 4396B is not a part of the E5022/E5023 system, however, you can connect your own the 4396B with the E5022/E5023.
- Appendix A, Manual Changes : provides information on changes and updated versions of the manual.

Other Manuals for Programming

User-defined Sequence (UDS) Programming is described in the UDS Programming Manual. The UDS, individual test sequence such as write offset, erase, write etc. has been provided as a call function to allow the user to create his own measurement sequence.

The programming information for the E5039A/B/C is provided in the E5022/E5023 Programming Manual for the E5039A/B/C.

Agilent E5022/E5023 Programming Overview

The Agilent E5022/E5023 uses a PC to control the system. The operating system of the PC is Microsoft Windows 95/2000. The measurement library and applications are already installed in the PC at the factory.

The measurement library and applications that have been pre-installed are as follows :

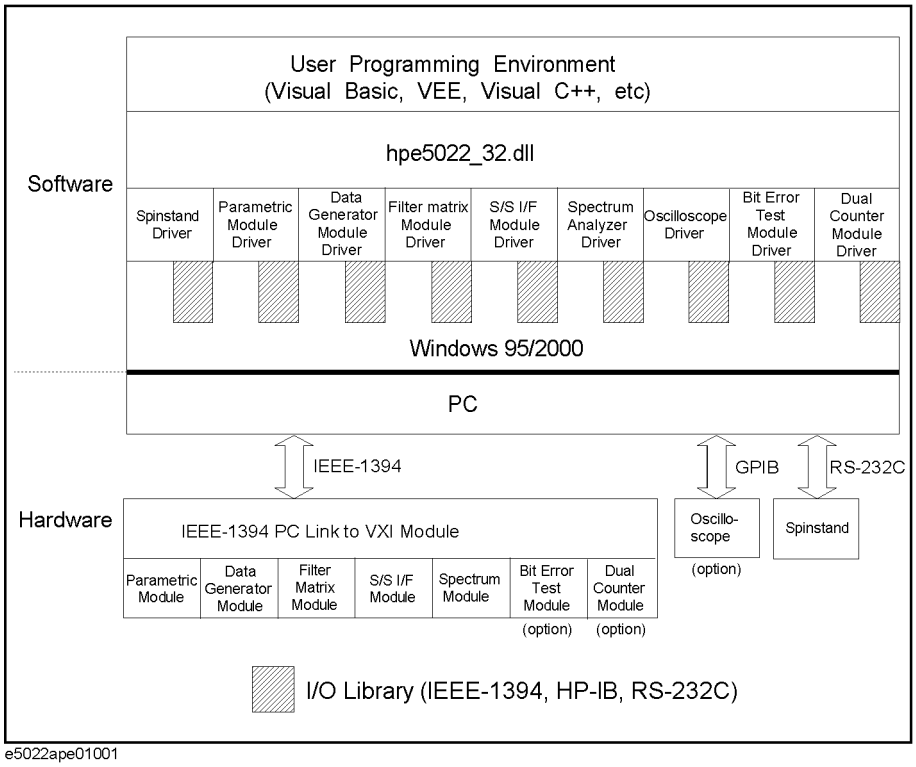
- VEE (Visual Engineering Environment)
- Agilent I/O Library

- Agilent E5022/E5023 Software (libraries and sample programs, an installer program)

The driver of each component (instruments and spinstand) controls the system hardware. The Dynamic Link Library (DLL) named “hpe5022_32.dll” controls each of the instrument driver. Users can access the DLL using applications such as Visual Basic, C and VEE.

The structure of the system software is shown in Figure 1-1.

Figure 1-1 System Software



e5022ape01001

Introduction

2 Programming

Programming Preparation

This section describes the preparations that you have to under take before you run the program.

DLLs, Include and Library files

All required DLLs, include and import library files have already been pre-installed at the factory.

Importing the library and including declaration file

You must import the library file and include the declaration file at the beginning of every program that contains Agilent Technologies E5022/E5023 function calls.

For Visual Basic, the file “hpe5022.bas” under the directory “c:\Program Files\Agilent\ E5022\include” should be added in the development environment as an object. When your system has E5039A/B/C, “hpe5022_BER.bas” under the same directory should be added. As path to the “c:\Program Files\Agilent\E5022\bin” directory is set when you install the Agilent Technologies E5022/E5023 software, the “hpe5022_32.dll” file is referred from any directories.

For C language, the file “hpe5022.h” under the directory “c:\Program Files\Agilent\E5022\include” should be included. The file named “hpe5022_32.lib” under the directory “c:\Program Files\Agilent\E5022\lib” should be linked.

For VEE, the file “hpe5022_32.dll” under the directory “c:\Program Files\Agilent\E5022\bin” should be imported. The file named “hpe5022_vee.h” under the directory “c:\Program Files\Agilent\E5022\include” should be included as a definition file.

Upgrading the System Software

Upgrading the system software allows you to upgrade the DLLs, include and library files. This allows you to install newly incorporated functions of the firmware into each module, such as E5035A through the VXI bus. The latest system software can be downloaded from the “http:\\dst.tm.agilent.com\”.

Before Running the Program

Before you run the program, the VXI mainframe and spinstand should be turned ON.

When trouble occurs during programming

When you develop a program, you might encounter a programming error due to bugs within the program. This section provides some tips on how to exit and debug the program.

The initialization of the system requires you to execute the “hpe5022_init” function. The closing requires you to perform the “hpe5022_close” function to close the programming.

If you fail to initialize:

If you can neither initialize nor close the system when you run the program, you should try quitting the applications such as Visual Basic or VEE. And run it again to initialize the system.

If this fails to work, reboot your PC.

If Visual Basic or C application often aborts:

If your application program often aborts, check if the array size of your program is correct. For example: the“hpe5022_stabilityData_Q” function has 3 parameters with the last parameter being in array form. If the size of the array is more or less than the correct number, you may encounter this trouble.

If an error occurs:

The program will stop when an error occurs. If the system is already running, it is recommended NOT to quit the program at once. Instead of quit, you had better to PAUSE the program and de-bug program then re-start your program.

When you quit the program and start it again, you may not be able to close nor reset the system even if the same handle number is used. This is because the resource is no longer in the PC memory.

Programming Concept

This section describes the programmimg concept before you develop a program.

Example Measurement Procedure

The example measurement procedure is provided in this section. The procedures listed below are categorized in seven steps.

Step 1. Initialization: This step sets the address for each module, initializes all modules, and gets the handle number The handle number is a given system number to control the system.

Step 2. Disk/Head Configuration and Drive On: This step specifies the spinstand configuration such as the spindle rpm, the data area and head position. After configuring the disk/head, you need to turn the drive-on in order to rotate the spindle and load the head.

Step 3. Auto Configuration (Setting Input Range): This step optimizes the input level control and sets the reference PW (pulse width) value.

Step 4. Common Measurement Parameters and Precompensation Setup: This step configures the bit error rate, data pattern and the precompensation setting that are used in every measurement.

Step 5. Measurement: This step is used to setup the measurements' own settings and measures the parameter, and queries the result.

Step 6. Aborting: This step is used to abort long measurements such as BER, Spectral SNR and Spectrum measurement. (This step is optional)

Step 7. Closing: This step turns the drive off and closes the system.

In actual measurement, the procedure listed above contains a group of functions that operate under each step, whose definition and operating parameters are described in chapter 3.

Two Sequence Types for Measurement

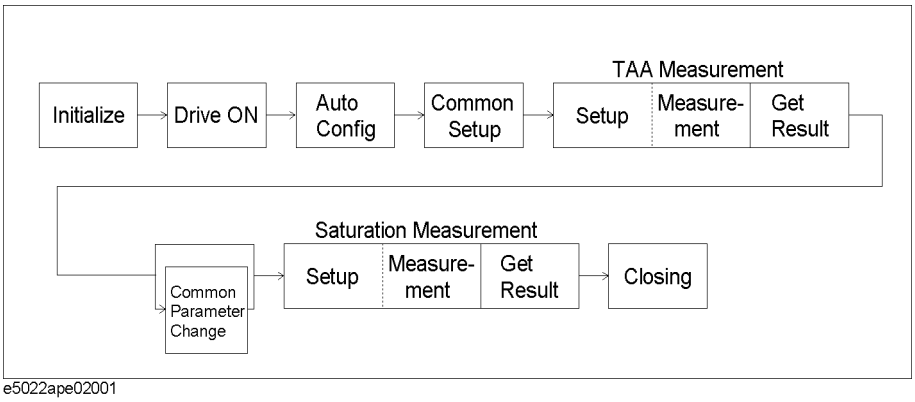
Since Agilent E5022/E5023 is designed for both laboratory and manufacturing users, two kinds of functions are provided for measurement sequence. One is for an easy of use and another is for a high through-put. The easy-of-use function is called as "At-once-measurement". While the through-put function is called as "Setup-then-measure measurement".

At-once-measurement

In "At-once-measurement", measurement is performed immediately after setup. The functions named "hpe5022_measure...." is in this category. For example, a TAA measurement, the "hpe5022_measureTaa" function is performed, then Agilent E5022/E5023 measures the TAA immediately, after that you can get the measurement result by the corresponding query function, such as the "hpe5022_taa_Q" function.

After executing one measurement, you can change the parameters which were set in step 4 (Measurement Common Parameters and Precompensation Setup).

Figure 2-1. At-once-measurement Sequence



e5022ape02001

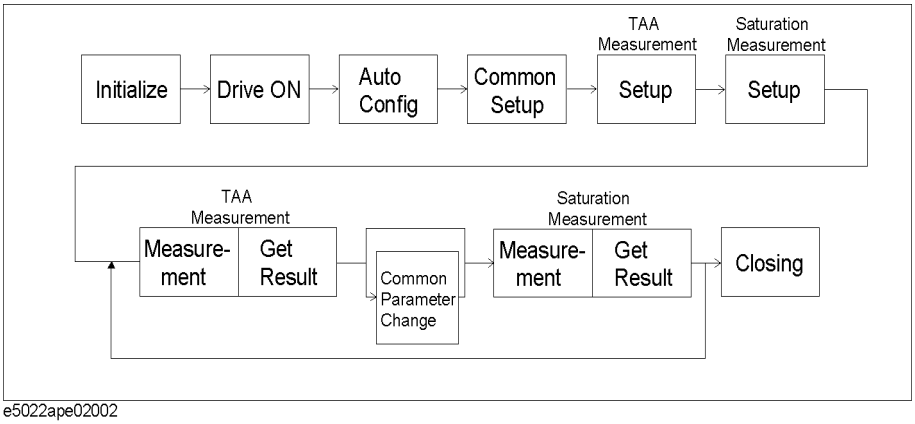
Setup-then-measure Measurement

"Setup-then-measure measurement" allows you to have a high through-put measurement when the same sequence measurement is performed. Twenty-five measurement setups can be performed in advance. However when software Revision 3.30 and later revision is installed approximately 50 sequences can be set up. The test numbers from 1 to 25 (for software Rev 3.30 and above approximately 50 can be set up) are assigned to the each measurement. After setting up all measurements, they are executed and the results are retrieved. The functions named "hpe5022_setup..." is in this category.

For example, TAA measurement, even if the "hpe5022_setupTaa" function is performed, the measurement is not executed. This function only setups the parameter and assigns the test number. The "hpe5022_measure" function actually performs the measurement. The measurement result is returned by the corresponding query function, such as the "hpe5022_taa_Q" function.

You can change the common parameters which is specified in step 4 between measurements.

Figure 2-2. Setup-then-measure Measurement Sequence



e5022ape02002

Example Program

This section describes the example programs for typical measurements. The program for each measurement is provided as an example of the “at-once measurement”. An example of the “setup-then-measure measurement” is provided at the last section. The sample program is saved at the directory named “c:\Program Files\Agilent\E5022\doc\vb\”. Sample1 is “at-once measurement” program, sample2 is a “setup-then-measure measurement” program.

Measurement Sequence is basically divided into six steps; Initializing, Drive On, Auto Configuration, Common Setup & Precompensation Setup, Measurement, aborting and Closing. The following sections provides the sample program and the description for each step.

Step 1. Initialization

The initialization step includes setting the address, initializing the system and to get the handler ID. This section explains an error handling subroutine for every function.

Example 2-1. Sample Program

```
Option Explicit
Dim Hpe5022 As Long
Dim Hpe5022abort As Boolean
Private Sub ErrorCheck(ErrorStatus As Long)
Dim ErrorMessage As String * 500
    '
    ' Check if Error Occurs
    '
    If ErrorStatus <> 0 Then
        '
        ' Get Error Message and Status Number
        '
        Call hpe5022_error_message(Hpe5022, ErrorStatus, ErrorMessage)
        '
        ' Display Error Message
        '
        Call MsgBox(ErrorMessage + " (" + Hex(ErrorStatus) + ") ", vbOKOnly, "Error")
    End If
End Sub
Private Sub CmdInitialize_Click()
    '
    ' Instrument Address Set (This exapmple for Option 102 and E5039A)
    '
    Const NumberOfInstrument As Integer = 6
    '
    Dim status, StatusArray(NumberOfInstrument - 1) As Long
    Dim rsrc As rsrcArrayType
    Dim ModuleNo As Integer
```

```
Dim i As Integer
Dim ModuleDescription As Variant
Dim ComponentName As instrNameType
Dim ComponentRev As revType
Dim ComponentSN As serialType
Dim errormessage As String * 100
Dim DLLRevision As String * 10
Dim Disp As String
Dim CompName As String
Dim CompRev As String
Dim CompSn As String

ModuleDescription = Array("E5035A ", "E5036A ", "E5037A ", "E5038A ", _
                          "E5040A ", "Spinstand ", "Oscilloscope ", "E5039A ", "E5041A")
'
' Instruments Address Set
'
rsrc.instr(0) = "VXI0::5::INSTR"  ' E5035A Spinstand I/F Module
rsrc.instr(1) = "VXI0::6::INSTR"  ' E5036A FilterMatrix Module
rsrc.instr(2) = "VXI0::7::INSTR"  ' E5037A Data Generator Module
rsrc.instr(3) = "VXI0::8::INSTR"  ' E5038A Parametric Module
rsrc.instr(4) = "VXI0::10::INSTR" ' E5040A Spectrum Measurement Module
rsrc.instr(5) = "ASRL1::INSTR"    ' E5010B Spinstand
'
' rsrc.instr(6) = "GPIB0::7::INSTR" ' Oscilloscope
' rsrc.instr(7) = "VXI0::9::INSTR"  ' E5039A BER Module
' rsrc.instr(8) = "VXI0::11::INSTR" ' E5041A Counter Module
'
' Initialize and Get Handle ID
'
status = hpe5022_init(NumberOfInstrument, rsrc, StatusArray(0), HpE5022)
'
' Error Handle
'
If status = 0 Then
    Call MsgBox("Initialize Succeed", vbOKOnly, "Message")
Else
    For ModuleNo = 0 To NumberOfInstrument - 1
        If StatusArray(ModuleNo) <> 0 Then
          Call hpe5022_error_message(HpE5022, StatusArray(ModuleNo), errormessage)
          Call MsgBox(ModuleDescription(ModuleNo) + " " + errormessage, vbOKOnly, "Error")
        End If
    Next ModuleNo
    End
End If
'
' E5022/E5023 System Firmware Query
'
ErrorCheck hpe5022_revision_query(HpE5022, DLLRevision)
```

```
    '
    ' E5022/E5023 Component Revision Query
    '
    With ComponentName
    For i = LBound(.instr) To UBound(.instr)
        .instr(i) = Space(hpe5022_MESSAGE_LENGTH_MAX)
    Next i
    End With
    With ComponentRev
    For i = LBound(.instr) To UBound(.instr)
        .instr(i) = Space(hpe5022_MESSAGE_LENGTH_MAX)
    Next i
    End With
    With ComponentSN
    For i = LBound(.instr) To UBound(.instr)
        .instr(i) = Space(hpe5022_MESSAGE_LENGTH_MAX)
    Next i
    End With
    '
    ErrorCheck hpe5022_subIdn_Q(HpE5022, ComponentName, ComponentRev, ComponentSN)
    '
    Disp = ""
    For ModuleNo = 0 To NumberOfInstrument - 1
        '
        ' Delete Space
        '
        CompName = Trim(ComponentName.instr(ModuleNo))
        CompRev = Trim(ComponentRev.instr(ModuleNo))
        CompSn = Trim(ComponentSN.instr(ModuleNo))
        '
        ' Delete a end character
        '
        Disp = Disp + Left(CompName, Len(CompName) - 1) + vbCrLf + _
               "Rev: " + Left(CompRev, Len(CompRev) - 1) + vbCrLf + _
               "S/N: " + Left(CompSn, Len(CompSn) - 1) + vbCrLf + vbCrLf
    Next ModuleNo
    '
    ' Display the Handle ID and Firmware Revision
    '
    lblID.Caption = HpE5022
    LblRevisionNo.Caption = DLLRevision
    LblResult.Caption = Disp
    '
End Sub
```

Description

The variable parameter named “HpE5022” should be defined as long in the general declaration. This is for the handle number.

The subroutine named “ErrorCheck” is an error handler procedure. When an error occurs on every function after

initialization, this subroutine returns the error message. This subroutine is used in every function call in the following examples.

The subroutine named “CmdInitialized_Click” is an initialization program. The instrument address depends on the option. See the “hpe5022_init” function.

Used Functions in this Step

“hpe5022_init” on page 91

“hpe5022_error_message” on page 107

Step2. Disk/Head Configuration and Drive ON

This step configures the drive such as data area of the disk, head position, and spindle rpm. After configuration, the drive is turned on to rotate the spindle and to load the head.

Example 2-2. Sample Program

```
Private Sub CmdDriveOn_Click()
Const SpindleRPM As Long = 5400 ' Spindle Revolution Speed [rpm]
    '
    Const PivotToCenter As Double = 0.058    ' Distance between Pivot and Center [Meter]
    Const PivotToGap As Double = 0.052  ' Distance between Pivot to Gap [Meter]
    Const SliderGap As Double = 0.00033  ' Distance of Slidier Gap [Meter]
    '
    Const RefBoss As Double = 0.0136 ' Distance between Reference Point and Boss Center [Meter]
    Const TipBoss As Double = 0.0146 ' Distance between Tip and Boss Center [Meter]
    Const BottomBoss As Double = 0.0015 ' Distance between Bottm and Boss Center [Meter]
    Const BossId As Double = 0.00147 'Innner Diameter of Boss [Meter]
    '
    Const CassetteID As String = "E5010-61125" ' HP Part Number of Cassette
    '
    Const InhibitMargin As Double = 0.001 ' Inhibit Area Magin [Meter]
    '
    Const IDRadius As Double = 0.023  ' Radius of the Inmost Track [Meter]
    Const TrackPitch As Double = 0.000003 ' Length of Track Pitch [Meter]
    Const NumOfTrack As Long = 5000 ' Number of Tracks
    Const HeadLoadRadius As Double = 0.039   ' Radius where the Head is loaded [Meter]
    Const InitialTrackNo As Long = 3000 ' Initial Track Number at Head Loading
    '
    ' Set Spindle RPM
    '
    ErrorCheck hpe5022_spindleSpeed(HpE5022, SpindleRPM)
    '
    ' Set Drive Configuration
    '(You can use the function of either "hpe5022_driveConfigRadiusSkew" or "hpe5022_driveConfigPivot".
    ' In this example, hpe5022_driveConfigPivot is used.)
    '
    ErrorCheck hpe5022_driveConfigPivot(HpE5022, PivotToCenter, PivotToGap)
    '
    ' Set the HP Part Number of Cassette
    '
    ErrorCheck hpe5022_hgaCassette(HpE5022, CassetteID)
    '
    ' Set HGA DImension
    '
    ErrorCheck hpe5022_hgaDimension(HpE5022, RefBoss, SliderGap, TipBoss, BottomBoss, BossId)
    '
    ' Set Inhibit Area Margin
    '
    ErrorCheck hpe5022_inhibitMargin(HpE5022, InhibitMargin)
```

```
    '
    ' Set Track Format
    '
    ErrorCheck hpe5022_dataArea(HpE5022, IDRadius, NumOfTrack, TrackPitch)
    '
    ' Set Head Road Radius and Initial Track Number
    '
    ErrorCheck hpe5022_headLoadConfig(HpE5022, hpe5022_LOAD_DYNAMIC, HeadLoadRadius, InitialTrackNo)
    '
    ' Drive On (Motor On, Head Load)
    '
    ErrorCheck hpe5022_driveState(HpE5022, VI_TRUE)
    '
    Call MsgBox("Spindle is running, Head is loaded", vbOKOnly, "Message")
    '
End Sub
```


Description

After turning the drive on, you cannot change the parameters which have been specified in this section. You must turn off the drive if you want to change the parameter such as spindle rpm.

Used Functions in this Step

“hpe5022_spindleSpeed” on page 213

“hpe5022_driveConfigPivot” on page 154

“hpe5022_dataArea” on page 180

“hpe5022_headLoadConfig” on page 192

“hpe5022_driveState” on page 209

Step 3. Auto Configuration (Setting Input Range)

The auto configuration optimizes the input range and sets the reference pulse width (PW) for the parametric module. See Chapter 4 of the Operation Manual for more details.

Example 2-3. Sample Program

```
Private Sub CmdAutoConfig_Click()
Const WriteCurrent As Double = 0.035  ' Write Current [Ampere]
    Const SenseCurrent As Double = 0.008 ' Sense Current [Ampere]
    Const EraseCurrent As Double = 0.035 ' Erase Current [Ampere]
    '
    Dim ConfigureInputRange As Integer
    Dim taa As Double, pwLength As Double, InputLevel As Double, PWreference As Double
    '
    ' Set Write/Sense/Erase Parameters
    '
    ErrorCheck hpe5022_writeCurrent(HpE5022, WriteCurrent)
    ErrorCheck hpe5022_senseCurrent(HpE5022, SenseCurrent)
    ErrorCheck hpe5022_eraseCurrent(HpE5022, EraseCurrent)
    '
    ' Set Erase Type for DC Erase
    '
    ErrorCheck hpe5022_eraseType(HpE5022, hpe5022_ERASE_DC_NEG)
    LblResult.Caption = "Write Current: " + Format(WriteCurrent, "0.00E-##A") + vbCrLf _
                      + "Sense Current: " + Format(SenseCurrent, "0.00E-##A") + vbCrLf
    '
    ' Perform Auto Configuration
    '
    ConfigureInputRange = 1 ' 1: Auto Configuration, 2: Input typical TAA and PW length of an Isolated Pulse
    '                         3: Input actual input level and PW length
    '
    Select Case ConfigureInputRange
    Case 1
        ErrorCheck hpe5022_measurementMode(HpE5022, hpe5022_MEAS_AUTO)
        ErrorCheck hpe5022_autoConfig(HpE5022, taa, pwLength)
        Call MsgBox("Auto Config Done" + vbCrLf + "TAA:" + Format(taa, "0.00E-##") + " V ," + _
                    "  PW:" + Format(pwLength, "0.00E-##") + " sec.", vbOKOnly, "Message")
    Case 2
        taa = 0.04
        pwLength = 0.000002
        ErrorCheck hpe5022_measurementMode(HpE5022, hpe5022_MEAS_AUTO)
        ErrorCheck hpe5022_isolatedPulseReference(HpE5022, taa, pwLength)
        Call MsgBox("Input Range Setting Done", vbOKOnly, "Message")
    Case 3
        InputLevel = 0.04
        PWreference = 0.00002
        ErrorCheck hpe5022_measurementMode(HpE5022, hpe5022_MEAS_MAN)
        ErrorCheck hpe5022_inputRange(HpE5022, InputLevel)
        ErrorCheck hpe5022_pwReference(HpE5022, PWreference)
        Call MsgBox("Input Range Setting Done", vbOKOnly, "Message")
    End Select
    '
```

```
End Sub
```

Description

Generally, as long as you continue to measure the same kind of head and without changing the sense/write current greatly, it is not necessary to perform this again. If you change the head under test and seems that the input signal level has changed significantly, then you should perform this again.

As the write, sense and erase currents are related with the auto configuration, you should specify these parameters within its limits before performing the auto configuration.

Used Functions in this Step

“hpe5022_writeCurrent” on page 323

“hpe5022_senseCurrent” on page 336

“hpe5022_eraseCurrent” on page 358

“hpe5022_eraseType” on page 355

“hpe5022_measurementMode” on page 254

“hpe5022_autoConfig” on page 257

Step 4. Measurement Common Parameter and Precompensation Setup

This setup specifies the parameters related with most measurements such as read/write track offset and precompensation.

Example 2-4. Sample Program

```
Private Sub CmdCommonSetup_Click()
    Const PeriodOfHFPattern As Integer = 2 ' Transition Period for HF Pattern
    Const PeriodOfLFPattern As Integer = 4 ' Transition Period for LF Pattern
    Const PeriodOfIsolatedPattern As Integer = 12 ' Transition Period for Isolated Pulse Pattern
    Const PeriodOfRepetitivePattern As Integer = 10 ' Transition Period for Repetitive Pattern
    Const FactorOfPRBS As Long = 137 ' Factor of Pseudo Random Pattern (x^7+x^3+x^1: 137=2^7+2^3+1)
    '
    Const UserDefDataLen As Integer = 16 ' User Defined Data Pattern Length
    Const UserDefPattern As String = "0100100010010001" ' Uer Defined Data Pattern (Binary Case)
    '
    Const ChannelBitRate As Double = 276000000# ' Channel Bit Rate [bps]
    '
    Const WriteTrackOffset As Double = 0 ' Write Track Offset [Meter]
    Const ReadTrackOffset As Double = -0.00000041 ' Read Track Offset [Meter]
    '
    Const TrackNo As Long = 3000 ' Track Number for Measurement
    Const rpm As Long = 6000 ' Spindle Speed [rpm]
    '
    Const EraseFrequency As Double = 100000000# ' Erase Frequency for AC Erase [Flux]
    '
    Const timeout As Double = 1 'Waiting time [sec]
    '
    ' Select Filter in the Filter Matrix Module
    '
    ErrorCheck hpe5022_selectFilter(HpE5022, hpe5022_FILTER0)
    '
    ' Set Test Data Patterns' Transition Period
    '
    ErrorCheck hpe5022_hfPattern(HpE5022, PeriodOfHFPattern)
    ErrorCheck hpe5022_lfPattern(HpE5022, PeriodOfLFPattern)
    ErrorCheck hpe5022_isolatedPulsePattern(HpE5022, PeriodOfIsolatedPattern)
    ErrorCheck hpe5022_repetitivePattern(HpE5022, PeriodOfRepetitivePattern)
    ErrorCheck hpe5022_prbsPattern(HpE5022, FactorOfPRBS)
    '
    ' Set User Defined Test Pattern
    '
    ErrorCheck hpe5022_userPattern(HpE5022, hpe5022_USER_DATA_BIN, UserDefDataLen, UserDefPattern)    '
    '
```

```
    ' Set Channel Bit Rate
    '
    ErrorCheck hpe5022_channelBitRate(HpE5022, ChannelBitRate)
    '
    ' Set Read Track Offset
    '
    ErrorCheck hpe5022_readTrackOffset(HpE5022, ReadTrackOffset)
    '
    ' Set Write Track Offset
    '
    ErrorCheck hpe5022_writeTrackOffset(HpE5022, WriteTrackOffset)
    '
    ' Set Track Number
    '
    ErrorCheck hpe5022_track(HpE5022, TrackNo)
    '
    ' Set Erase Frequency for AC Erase
    '
    ErrorCheck hpe5022_eraseAcFrequency(HpE5022, EraseFrequency)
    '
    ' Change Spindle Speed
    '
    ErrorCheck hpe5022_spindleSpeed(HpE5022, rpm)
    '
    'Set the time to wait
    '
    ErrorCheck hpe5022_wai(HpE5022, timeout)
    '
    Call MsgBox("Common Setup Done", vbOKOnly, "Message")
End Sub
```

Description

The parameters in this section can be changed any time before and after each measurement. For example, when you want to change the channel bit rate after executing the measurement, you can perform the “hpe5022_channelBitRate” function after measurement. Off course, these parameters can be set before the drive on and the auto configuration.

Used Functions in this Step

“hpe5022_selectFilter” on page 149

“hpe5022_hfPattern” on page 122

“hpe5022_lfPattern” on page 124

“hpe5022_isolatedPulsePattern” on page 126

“hpe5022_prbsPattern” on page 128

“hpe5022_repetitivePattern” on page 131
“hpe5022_userPattern” on page 140
“hpe5022_channelBitRate” on page 116
“hpe5022_readTrackOffset” on page 219
“hpe5022_writeTrackOffset” on page 222
“hpe5022_track” on page 229
“hpe5022_eraseAcFrequency” on page 360

Precompensation Setup

The precompensation function allows you to delay a write timing for a specified pattern.

Example 2-5. Sample Program

```
Private Sub CmdPrecompensation_Click()
Dim preBit As Integer, postBit As Integer, DelayNo As Integer
    Dim delay(3) As Double
    Dim BinDisplay As Variant
    Dim Disp As String
    BinDisplay = Array("000", "001", "010", "011", "100", "101", "110", "111")
    '
    ' Specifies the Delay Time. ( You can specifies delay time up to three.)
    '
    delay(1) = -0.000000001 ' Delay Time No.1
    delay(2) = 0.000000002 ' Delay Time No.2
    '
    ' Clear Precompenstaion pattern
    '
    ErrorCheck hpe5022_clearPrecomp(HpE5022)
    '
    ' Asign Delay Time to Precompensation Number
    '
    ErrorCheck hpe5022_precompDelay(HpE5022, hpe5022_PREC_1, delay(1))
    ErrorCheck hpe5022_precompDelay(HpE5022, hpe5022_PREC_2, delay(2))
    '
    ' Set Precompensation Pattern and the Delay Time
    '
    '   For Delay Time 1
    '      The delay time of the pattern of "*11x***" is set at the Value of Delay(1)
    '      (ie.  x:target bit,  *:wild card(0 or 1))
    '
    ErrorCheck hpe5022_precompPattern(HpE5022, hpe5022_PREC_PAT_11, hpe5022_PREC_PAT_X, hpe5022_PREC_1)
    '
    '   For Dealy Time 2
    '      The delay time of the following pattern is set at the Value of Delay(2)
    '
    '      "001x011", "001x110", "101x110" (ie. x: Target bit)
    '      (These patterns had better not duplicate with the pattern for the other delay time.
    '       In this case, the "001x011", "001x110", "101x110" had better not be included of  "*11x***"..
    '       If there is duplecated, the later setting is valid. The setting is overwraped.)
    '
    ErrorCheck hpe5022_precompPattern(HpE5022, hpe5022_PREC_PAT_001, hpe5022_PREC_PAT_011, hpe5022_PREC_2)
    ErrorCheck hpe5022_precompPattern(HpE5022, hpe5022_PREC_PAT_001, hpe5022_PREC_PAT_110, hpe5022_PREC_2)
    ErrorCheck hpe5022_precompPattern(HpE5022, hpe5022_PREC_PAT_101, hpe5022_PREC_PAT_110, hpe5022_PREC_2)
    '
    ' Precompensation ON for a Test Data Pattern
    ' (The NLTS 5th harmonic Test Data Pattern has a precompensation in this Example)
    '
    ErrorCheck hpe5022_precompState(HpE5022, hpe5022_PAT_NLTS_5TH, VI_TRUE)
    '
    '  Confirm the Delay Value for All of Patterns
    '
```

```
    Disp$ = ""
    For postBit = hpe5022_PREC_PAT_000 To hpe5022_PREC_PAT_111
        For preBit = hpe5022_PREC_PAT_000 To hpe5022_PREC_PAT_111
            ErrorCheck hpe5022_precompPattern_Q(HpE5022, preBit, postBit, DelayNo)
            If DelayNo <> hpe5022_PREC_OFF Then
                Disp$ = Disp$ + BinDisplay(preBit - hpe5022_PREC_PAT_000) + "x" _
                              + BinDisplay(postBit - hpe5022_PREC_PAT_000) _
                              + " : Delay Time = " + Format(delay(DelayNo), "0.00E-##") + " sec" + vbCrLf
            End If
        Next preBit
    Next postBit
    LblResult.Caption = Disp$
End Sub
```

Description

Agilent E5022/E5023 allows you to have three kinds of delay time for a specified data pattern. Delay time is defined as three bits previous and three bits behind the target bit. You can select which data pattern you want to have a delay to write the data. The “hpe5022_precompDelay” assigns the delay time to the precompensation numbers 1 to 3. The “hpe5022_precompPattern” function defines the data patterns which have a precompensation. The “hpe5022_precompState” function selects which test data patterns have a delay.

The data pattern definition is overlapped until the “hpe5022_clearPrecomp” is performed, it is recommended to confirm which pattern has a delay after setting it up.

Used Functions in this Step

“hpe5022_clearPrecomp” on page 386

“hpe5022_precompDelay” on page 374

“hpe5022_precompPattern” on page 377

“hpe5022_precompState” on page 383

“hpe5022_precompPattern_Q” on page 381

Step 5. Measurement

The Agilent E5022/E5023 provides the following measurements:

- Parametric (measure TAA, PW, Baseline and Resolution at one time)
- TAA
- PW
- Baseline
- Resolution
- Narrow Band TAA (TAA by a spectrum analyzer)
- Parametric Measurement by a Oscilloscope (Waveform Analysis)
- NLTS by 5th Harmonic Method
- NLTS by Dipulse Extraction Method
- NLTS by Time Correlation Method
- Overwrite
- Partial Erasure
- Track Profile
- MicroTrack Profile
- Write Current Sweep
- Sense Current Sweep
- Channel Bit Rate Sweep
- RollOff
- Precompensation Delay Sweep
- Stability
- DC-R

This section will show the sample programs for these measurements.

Parametric Measurement

Parametric Measurement can measure TAA, PW, Baseline and Resolution at one time. See Chapter 4 of the Operation Manual for measurement definition.

Example 2-6. Sample Program

```
Private Sub CmdParametic_Click()
Const NumOfRevolutionForAverage As Integer = 10 ' Number of Revolutions for Averaging
    Const PwRatio As Double = 0.5 ' Threshold Level Ratio for PW Measurement
    Dim HFTaa1 As Double, LFTaa As Double, IsolatedTAA As Double, pw As Double, baseline As Double
    Dim resolution As Double, HFTaa2 As Double, HFTaaPos As Double, HFTaaNeg As Double, HFTaaAsym As Double
    Dim HFTAAData() As Double
    Dim DataSize As Long
    '
    ' Set PW ratio and its Reference Level
    '
    ErrorCheck hpe5022_pwThreshold(HpE5022, PwRatio, hpe5022_PW_TAA_TRACKED)
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measureParametric(HpE5022, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_parametric_Q(HpE5022, HFTaa1, LFTaa, IsolatedTAA, pw, baseline, resolution)
    ErrorCheck hpe5022_parametricHfTaa_Q(HpE5022, HFTaa2, HFTaaPos, HFTaaNeg, HFTaaAsym)
    '
    ErrorCheck hpe5022_parametricHfTaaDataSize_Q(HpE5022, DataSize)
    ReDim HFTAAData(DataSize - 1)
    ErrorCheck hpe5022_parametricHfTaaData_Q(HpE5022, hpe5022_DATA_TAA_POS, HFTAAData(0))
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "HF TAA: " + Format(HFTaa1, "0.00E-##V") + vbCrLf _
                      + "LF TAA: " + Format(LFTaa, "0.00E-##V") + vbCrLf _
                      + "Isolated Pulse TAA: " + Format(IsolatedTAA, "0.00E-##V") + vbCrLf _
                      + "PW: " + Format(pw, "0.00E-##") + "sec" + vbCrLf _
                      + "Baseline: " + Format(baseline, "0.00E-##V") + vbCrLf _
                      + "Resolution: " + Format(resolution, "0.0%")
    '
End Sub
```

Description

For PW measurement, the default ratio is PW50. Refer to “hpe5022_pwThreshold” function to change the ratio. In Parametric measurement, the HF, LF and Isolated data patterns are used automatically. You don’t need to setup the data patterns by “hpe5022_selectPattern”function. The HF and LF data patterns are used in TAA and Resolution measurements. While Isolated pattern is used for TAA, PW and Baseline measurements.

After executing the “hpe5022_measureParametric” function, any query functions related with parametric measurements can be executed to get the measurement results.

As the Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “HFTAAData(0)” in line of “hpe5022_parametricHfTaaData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_pwThreshold” on page 428

“hpe5022_measureParametric” on page 466

“hpe5022_parametric_Q” on page 471

“hpe5022_parametricHfTaa_Q” on page 473

“hpe5022_parametricHfTaaDataSize_Q” on page 475

“hpe5022_parametricHfTaaData_Q” on page 476

TAA Measurement

The TAA Measurement can measure the TAA, TAA positive, TAA negative, TAA asymmetry and statistic data for each of these values. See Chapter 4 of the Operation Manual for the measurement definition.

Example 2-7. Sample Program

```
Private Sub CmdTAA_Click()
Const NumOfRevolutionForAverage As Integer = 10
    '
    Dim taa As Double, taaPos As Double, taaNeg As Double, taaAsym As Double, PosMean As Double
    Dim PosMin As Double, PosMax As Double, PosStdDev As Double, NegMean As Double, NegMin As Double
    Dim NegMax As Double, NegStdDev As Double, TAAData() As Double
    Dim DataSize As Long
    Dim Disp As String
    Dim i As Integer
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measureTaa(HpE5022, hpe5022_SEQ_ER_WR_M, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_taa_Q(HpE5022, taa, taaPos, taaNeg, taaAsym)
    ErrorCheck hpe5022_taaStatistic_Q(HpE5022, hpe5022_DATA_TAA_POS, PosMean, PosMin, PosMax, PosStdDev)
    ErrorCheck hpe5022_taaStatistic_Q(HpE5022, hpe5022_DATA_TAA_NEG, NegMean, NegMin, NegMax, NegStdDev)
    '
    ErrorCheck hpe5022_taaDataSize_Q(HpE5022, DataSize)
    ReDim TAAData(DataSize - 1)
    ErrorCheck hpe5022_taaData_Q(HpE5022, hpe5022_DATA_TAA_POS, TAAData(0))
    '
    ' Display Mesurement Result
    '
    Disp$ = "TAA :" + vbCrLf _
                    + "TAA: " + Format(taa, "0.00E-##V") + vbCrLf _
                    + "TAA+: " + Format(taaPos, "0.00E-##V") + vbCrLf _
                    + "TAA-: " + Format(taaNeg, "0.00E-##V") + vbCrLf _
                    + "TAA Asym: " + Format(taaAsym, "0.0%") + vbCrLf _
                    + "TAA Positive Statistic: " + vbCrLf _
                    + "Mean: " + Format(PosMean, "0.00E-##V") + vbCrLf _
                    + "Min: " + Format(PosMin, "0.00E-##V") + vbCrLf _
                    + "Max: " + Format(PosMax, "0.00E-##V") + vbCrLf _
                    + "Standard Deviation: " + Format(PosStdDev, "0.00E-##") _
                    + "TAA Negative Statistic: " + vbCrLf _
                    + "Mean: " + Format(NegMean, "0.00E-##V") + vbCrLf _
                    + "Min: " + Format(NegMin, "0.00E-##V") + vbCrLf _
                    + "Max: " + Format(NegMax, "0.00E-##V") + vbCrLf _
                    + "Standard Deviation: " + Format(NegStdDev, "0.00E-##") + vbCrLf
    Disp$ = Disp$ + vbCrLf + "TAA DATA" + vbCrLf
    For i = 0 To DataSize - 1
```

```
        Disp$ = Disp$ + Format(TAAData(i), "0.00E-##V") + vbCrLf
    Next i
    LblResult.Caption = Disp$
End Sub
```

Description

The TAA measurement requires you to select the data pattern. If the required data pattern has already been set before sequence, you do not need to set in this section.

After executing the “hpe5022_measureTaa” function, any query functions related with TAA measurement can be executed to get the measurement result.

As the Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first “TAAData(0)” in line of “hpe5022_taaData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_measureTaa” on page 409

“hpe5022_taa_Q” on page 414

“hpe5022_taaStatistic_Q” on page 416

“hpe5022_taaDataSize_Q” on page 418

“hpe5022_taaData_Q” on page 419

PW Measurement

The PW Measurement can measure PW, PW positive, PW negative, PW asymmetry and also the statistic data for each of these values. See Chapter 4 of the Operation Manual for measurement definition.

Example 2-8. Sample Program

```
Private Sub CmdPW_Click()
Const NumOfRevolutionForAverage As Integer = 10 ' Number of Revolutions for Averaging
    Const PwRatio As Double = 0.5 ' Threshold Level Ratio for PW Measurement
    Dim pw As Double, pwPos As Double, pwNeg As Double, pwAsym As Double, mean As Double
    Dim min As Double, max As Double, stdDev As Double, PWData() As Double
    Dim DataSize As Long
    Dim Disp As String
    Dim i As Integer
    '
    ' Set the PW ratio and its reference level
    '
    ErrorCheck hpe5022_pwThreshold(HpE5022, PwRatio, hpe5022_PW_TAA_TRACKED)
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_ISO)
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measurePw(HpE5022, hpe5022_SEQ_ER_WR_M, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_pw_Q(HpE5022, pw, pwPos, pwNeg, pwAsym)
    ErrorCheck hpe5022_pwStatistic_Q(HpE5022, hpe5022_DATA_PW_POS, mean, min, max, stdDev)
    '
    ErrorCheck hpe5022_pwDataSize_Q(HpE5022, DataSize)
    ReDim PWData(DataSize - 1)
    ErrorCheck hpe5022_pwData_Q(HpE5022, hpe5022_DATA_PW_POS, PWData(0))
    '
    ' Display Mesurement Result
    '
    Disp$ = "PW Measurement:" + vbCrLf _
          + "PW: " + Format(pw, "0.00E-##") + " sec" + vbCrLf _
          + "PW+: " + Format(pwPos, "0.00E-##") + " sec" + vbCrLf _
          + "PW-: " + Format(pwNeg, "0.00E-##") + " sec" + vbCrLf _
          + "PW Asym: " + Format(pwAsym, "0.0%") + vbCrLf _
          + "PW+ Statistic: " + vbCrLf _
          + "Mean: " + Format(mean, "0.00E-##") + " sec" + vbCrLf _
          + "Min: " + Format(min, "0.00E-##") + " sec" + vbCrLf _
          + "Max: " + Format(max, "0.00E-##") + " sec" + vbCrLf _
          + "Standard Deviation: " + Format(stdDev, "0.00E-##") + vbCrLf
    Disp$ = Disp$ + vbCrLf + "PW POS DATA" + vbCrLf
```

```
    For i = 0 To DataSize - 1
        Disp$ = Disp$ + Format(PWData(i), "0.00E-##") + vbCrLf
    Next i
    LblResult.Caption = Disp$
End Sub
```

Description

The PW measurement requires the user to select data pattern. If data pattern has already been set before sequence, then it is not necessarry to set up the data pattern in this section.

After executing the “hpe5022_measurePw” function, any query functions related with the PW measurement can be executed to get the measurement result.

As the Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “PWData(0)” in line of “hpe5022_parametricPwData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_pwThreshold” on page 428

“hpe5022_selectPattern” on page 119

“hpe5022_measurePw” on page 431

“hpe5022_pw_Q” on page 435

“hpe5022_pwStatistic_Q” on page 440

“hpe5022_pwDataSize_Q” on page 437

“hpe5022_pwData_Q” on page 438

Baseline Measurement

The Resolution Measurement can measure Baseline, Baseline positive, Baseline negative, separation and also the statistic data for each of these values. See Chapter 4 of the Operation Manual for the measurement definition.

Example 2-9. Sample Program

```
Private Sub CmdBaseline_Click()
Const NumOfRevolutionForAverage As Integer = 10
    '
    Dim baseline As Double, BLPos As Double, BLNeg As Double, Separation As Double
    Dim mean As Double, min As Double, max As Double, stdDev As Double, BLData() As Double
    Dim DataSize As Long
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_ISO)
    '
    ' Measurement (Measurement Sequence: Erase -> Write -> Measurement)
    '
    ErrorCheck hpe5022_measureBaseline(HpE5022, hpe5022_SEQ_ER_WR_M, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_baseline_Q(HpE5022, baseline, BLPos, BLNeg, Separation)
    ErrorCheck hpe5022_baselineStatistic_Q(HpE5022, hpe5022_DATA_BL_POS, mean, min, max, stdDev)
    '
    ErrorCheck hpe5022_baselineDataSize_Q(HpE5022, DataSize)
    ReDim BLData(DataSize - 1)
    ErrorCheck hpe5022_baselineData_Q(HpE5022, hpe5022_DATA_BL_POS, BLData(0))
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "Baseline :" + vbCrLf _
                    + "Baseline: " + Format(baseline, "0.00E-##V") + vbCrLf _
                    + "Baseline Pos: " + Format(BLPos, "0.00E-##V") + vbCrLf _
                    + "Baseline Neg: " + Format(BLNeg, "0.00E-##V") + vbCrLf _
                    + "Separation: " + Format(Separation, "0.00E-##V") + vbCrLf _
                    + "Baseline Positive Statistic: " + vbCrLf _
                    + "Mean: " + Format(mean, "0.00E-##V") + vbCrLf _
                    + "Min.: " + Format(min, "0.00E-##V") + vbCrLf _
                    + "Max.: " + Format(max, "0.00E-##V") + vbCrLf _
                    + "Standard Deviation: " + Format(stdDev, "0.00E-##")
End Sub
```

Description

The baseline measurement requires you to select the data pattern. If the required data pattern has already been set before sequence, you do not need to set in this section.

After executing the “hpe5022_measureBaseline” function, any query functions related with the baseline measurement can be executed to get the measurement result.

As the Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “BLData(0)” in line of “hpe5022_baselineData_Q” passes the first element of the

array by reference.

Functions used in this measurement

“hpe5022_selectPattern” on page 119

“hpe5022_measureBaseline” on page 447

“hpe5022_baseline_Q” on page 452

“hpe5022_baselineStatistic_Q” on page 457

“hpe5022_baselineDataSize_Q” on page 454

“hpe5022_baselineData_Q” on page 455

Resolution Measurement

The Resolution Measurement can measure the resolution and its statistic data. See Chapter 4 of the Operation Manual for measurement definition.

Example 2-10. Sample Program

```
Private Sub CmdResolution_Click()
Const NumOfRevolutionForAverage As Integer = 10 ' Number of Revolutions for Averaging
    Dim resolution As Double, mean As Double, min As Double, max As Double, stdDev As Double
    Dim ResolutionData() As Double
    Dim DataSize As Long
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measureResolution(HpE5022, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_resolution_Q(HpE5022, resolution)
    ErrorCheck hpe5022_resolutionStatistic_Q(HpE5022, mean, min, max, stdDev)
    '
    ErrorCheck hpe5022_resolutionDataSize_Q(HpE5022, DataSize)
    ReDim ResolutionData(DataSize - 1)
    ErrorCheck hpe5022_resolutionData_Q(HpE5022, ResolutionData(0))
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "Resolution :" + vbCrLf _
                    + "Resolution: " + Format(resolution, "0.0%") + vbCrLf _
                    + "Resolution Statistic: " + vbCrLf _
                    + "Mean: " + Format(mean, " 0.0%") + vbCrLf _
                    + "Min.: " + Format(min, "0.0%") + vbCrLf _
                    + "Max.: " + Format(max, "0.0%") + vbCrLf _
                    + "Standard Deviation: " + Format(stdDev, "0.00E-##")

End Sub
```

Description

The HF and the LF data patterns are used in this measurement automatically.

Functions used in this measurement

“hpe5022_measureResolution” on page 501

“hpe5022_resolution_Q” on page 506

“hpe5022_resolutionStatistic_Q” on page 509

Narrow Band TAA Measurement

Narrow Band TAA measurement can measure the TAA. Since this TAA is measured not by a parametric module but by a spectrum analyzer, you can measure low signal amplitude level. However, the measurement speed will be slower than the one using a parametric module. See Chapter 4 of the Operation Manual for its measurement definition.

Example 2-11. Sample Program

```
Private Sub CmdNarrowBandTAA_Click()
Const NumOfRevolutionForAverage As Integer = 10
    Dim NarrowBandTAA As Double, mean As Double, min As Double, max As Double
    Dim stdDev As Double, TAAData() As Double
    Dim DataSize As Long
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    ErrorCheck hpe5022_narrowBandTaaFrequencyMode(HpE5022, hpe5022_NB_TAA_FREQ_AUTO)
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measureNarrowBandTaa(HpE5022, hpe5022_SEQ_ER_WR_M, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_narrowBandTaa_Q(HpE5022, NarrowBandTAA)
    ErrorCheck hpe5022_narrowBandTaaStatistic_Q(HpE5022, mean, min, max, stdDev)
    '
    ErrorCheck hpe5022_narrowBandTaaDataSize_Q(HpE5022, DataSize)
    ReDim TAAData(DataSize - 1)
    ErrorCheck hpe5022_narrowBandTaaData_Q(HpE5022, TAAData(0))
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "Narrow Band TAA: HF " + vbCrLf _
                    + "TAA:  " + Format(NarrowBandTAA, "0.00E-##") + " V" + vbCrLf _
                    + "Mean: " + Format(mean, "0.00E-##") + " V" + vbCrLf _
                    + "Min:  " + Format(min, "0.00E-##") + " V" + vbCrLf _
                    + "Max:  " + Format(max, "0.00E-##") + " V" + vbCrLf _
                    + "Std Dev: " + Format(stdDev, "0.00E-##")
End Sub
```

Description

The Narrow Band TAA measurement requires you to select the data pattern. If the required data pattern has already been set before sequence, you do not need to set in this section. As the “hpe5022_narrowBandTaaFrequencyMode” is set at auto mode on the default setting, you may skip it if you want to set at auto mode.

Functions used in this measurement

“hpe5022_selectPattern” on page 119

“hpe5022_narrowBandTaaFrequencyMode” on page 564

“hpe5022_measureNarrowBandTaa” on page 572

“hpe5022_narrowBandTaa_Q” on page 577

“hpe5022_narrowBandTaaStatistic_Q” on page 580

Waveform Analysis Measurement

The waveform analysis measurement can measure TAA, PW, and Baseline measurements by using an oscilloscope. See Chapter 4 of the Operation Manual for measurement definition

Example 2-12. Sample Program

```
Private Sub CmdWaveformAnalysis_Click()
Const OverSampleRate As Long = 4 ' OverSampling Rate for digitizing
    Const WaveAverage As Long = 20 'Averaging Factor of Data Acquisition
    Const DelayTime As Double = 0.0002 ' Delay Time from the Index Pulse from the Spinstand [Second]
    Const PWThreshold As Double = 0.5 ' Threshold level for PW measurement
    '
    Dim taa As Double, taaPos As Double, taaNeg As Double, taaAsym As Double, pw As Double, pwPos As Double
    Dim pwNeg As Double, pwAsym As Double, baseline As Double, BaseLinePos As Double, BaseLineNeg As Double
    Dim BaseSeparation As Double, TimePOStoNEG As Double, TimeNEGtoPOS As Double, TimeAsym As Double
    Dim WaveFormData() As Double
    Dim DataSize As Long
    '
    ErrorCheck hpe5022_waveOverSampleRate(HpE5022, OverSampleRate)
    ErrorCheck hpe5022_waveAverage(HpE5022, WaveAverage)
    ErrorCheck hpe5022_waveDelayTime(HpE5022, DelayTime)
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_ISO)
    '
    ' Data Acquisition
    '
    ErrorCheck hpe5022_measureWave(HpE5022, hpe5022_SEQ_ER_WR_M)
    '
    ' Query Averaged Waveform Data
    '
    ErrorCheck hpe5022_waveAverageDataSize_Q(HpE5022, DataSize)
    ReDim WaveFormData(DataSize - 1)
    ErrorCheck hpe5022_waveAverageData_Q(HpE5022, WaveFormData(0))
    '
    ' Calcurate Parameters
    '
    ErrorCheck hpe5022_waveTaa_Q(HpE5022, taa, taaPos, taaNeg, taaAsym)
    ErrorCheck hpe5022_wavePw_Q(HpE5022, PWThreshold, pw, pwPos, pwNeg, pwAsym)
    ErrorCheck hpe5022_waveBaseline_Q(HpE5022, baseline, BaseLinePos, BaseLineNeg, BaseSeparation)
    ErrorCheck hpe5022_waveTimeAsymmetry_Q(HpE5022, TimeAsym, TimePOStoNEG, TimeNEGtoPOS)
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "Wave Form Analysis;" + vbCrLf _
                    + "TAA:       " + Format(taa, "0.00E-##") + " V" + vbCrLf _
                    + "TAA+:      " + Format(taaPos, "0.00E-##") + " V" + vbCrLf _
                    + "TAA-:      " + Format(taaNeg, "0.00E-##") + " V" + vbCrLf _
```

```
+ "TAA Asym:  " + Format(taaAsym, "0.0%") + vbCrLf + vbCrLf _
+ "PW:        " + Format(pw, "0.00E-##") + " V" + vbCrLf _
+ "PW+:       " + Format(pwPos, "0.00E-##") + " sec" + vbCrLf _
+ "PW-:       " + Format(pwNeg, "0.00E-##") + " sec" + vbCrLf _
+ "PW Asym:   " + Format(pwAsym, "0.0%") + vbCrLf + vbCrLf _
+ "Baseline:  " + Format(baseline, "0.00E-##") + " V" + vbCrLf _
+ "Baseline+: " + Format(BaseLinePos, "0.00E-##") + " V" + vbCrLf _
+ "Baseline-: " + Format(BaseLineNeg, "0.00E-##") + " V" + vbCrLf _
+ "Baseline Separation: " + Format(BaseSeparation, "0.00E-##") + " V" + vbCrLf + vbCrLf _
+ "Time Asymmetry:      " + Format(TimeAsym, "0.00E-##") + " sec."
End Sub
```

Description

For PW measurement, the threshold level is different from the PW measurement using the parametric module. The “hpe5022_pwThreshold” function is not valid for this measurement.

The waveform measurement requires you to select the data pattern. If the required data pattern has already been set before sequence, you do not need to set in this section.

After executing the “hpe5022_measureWave” function, any query functions related with the waveform measurement can be executed to get the measurement result.

As the Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “WaveFormData(0)” in line of “hpe5022_waveAverageData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_waveOverSampleRate” on page 1019

“hpe5022_waveAverage” on page 1022

“hpe5022_waveDelayTime” on page 1025

“hpe5022_measureWave” on page 1028

“hpe5022_waveTaa_Q” on page 1041

“hpe5022_wavePw_Q” on page 1043

“hpe5022_waveBaseline_Q” on page 1045

“hpe5022_waveTimeAsymmetry_Q” on page 1047

“hpe5022_selectPattern” on page 119

NLTS by 5th Harmonic Method Measurement

The NLTS by 5th harmonic method is used to calculate NLTS by measuring the 5th harmonic level for two kinds of pattern. Spectrum analyzer is used for this measurement.

Example 2-13. Sample Program

```
Private Sub CmdNLTS5th_Click()
Const NumOfRevolutionForAverage As Integer = 10 ' Number of Revolutions for Averaging
    '
    Dim nlts As Double, NLTSData() As Double
    Dim DataSize As Long
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measureNlts5th(HpE5022, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_nlts5th_Q(HpE5022, nlts)
    '
    ErrorCheck hpe5022_nlts5thDataSize_Q(HpE5022, DataSize)
    ReDim NLTSData(DataSize - 1)
    ErrorCheck hpe5022_nlts5thData_Q(HpE5022, NLTSData(0))
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "NLTS 5th: " + Format(nlts, "0.0%")
    '
End Sub
```

Description

In the NLTS by 5th Harmonic method measurement, the required patterns (See the “hpe5022_measureNlts5th” function) are used automatically. You don’t need to setup by the “hpe5022_selectPattern” function.

Functions used in this measurement

“hpe5022_measureNlts5th” on page 810

“hpe5022_nlts5th_Q” on page 815

NLTS by Dipulse Extraction Method Measurement

The NLTS by Dipulse Extraction method is to calculate NLTS by measuring the amplitude at zero point and echoes from the dipulse extraction data. An oscilloscope is used for this measurement.

Example 2-14. Sample Program

```
Private Sub CmdNltsDipulse_Click()
Const OverSampleRate As Long = 4 ' Over Sampling Rate for Digitizing
    Const WaveAverage As Long = 20 ' Averaging Factor of Data Acquisition
    Const DelayTime As Double = 0.0002 ' Delay Time from the Index Pulse from the Spinstand [Second]
    '
    ' Echo Location (When PRBS = x^7+x^3+1, the PRBS factor is set in CmdCommonSetup_Click()
    '
    Const InitialMagnitude As Double = -30.5
    Const AdjacentTransition As Double = 25.5
    Const SecondAdjTransition As Double = 45.5
    '
    Dim nlts As Double, HardEasyTransShift As Double, SecondAdjacent As Double, ACSNR As Double
    '
    ' Set the Oversampling Rate, Averaging Factor and Delay Time
    '
    ErrorCheck hpe5022_waveOverSampleRate(HpE5022, OverSampleRate)
    ErrorCheck hpe5022_waveAverage(HpE5022, WaveAverage)
    ErrorCheck hpe5022_waveDelayTime(HpE5022, DelayTime)
    '
    ' Select Test Data Pattern (PRBS must be used)
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_PRBS)
    '
    ' Data Acquisition
    '
    ErrorCheck hpe5022_measureWave(HpE5022, hpe5022_SEQ_ER_WR_M)
    '
    ' Caliculate NLTS Dipulse Extraction
    '
    ErrorCheck hpe5022_waveDipulseExtractionNlts_Q(HpE5022, AdjacentTransition, nlts)
    ErrorCheck hpe5022_waveDipulseExtractionNlts_Q(HpE5022, InitialMagnitude, HardEasyTransShift)
    ErrorCheck hpe5022_waveDipulseExtractionNlts_Q(HpE5022, SecondAdjTransition, SecondAdjacent)
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "NLTS Dipulse Extraction: " + vbCrLf _
                    + "NLTS: " + Format(nlts, "0.0%") + vbCrLf _
                    + "Hard/Easy Transition Shift: " + Format(HardEasyTransShift, "0.0%") + vbCrLf _
                    + "2nd Adjacent:" + Format(SecondAdjacent, "0.0%")

End Sub
```

Description

In the NLTS by Dipulse Extraction method measurement, the pseudo random pattern should be selected. The “hpe5022_waveDipulseExtraction_Q” function requires an echo location to calculate the parameters.

When the pseudo random pattern of x^7+x^3+1 is used, the echo at a location of 25.5 bit periods corresponds to the NLTS, the second small echo at 45.5 bit period reflects the influence of the second adjacent transition shift. The echo at -30.5 bit periods (left side of main peak) corresponds to the Hard Transition shift.

When the pattern $x^6+x^4+x^3+x+1$ is used, the echo locations are -18.5, -22.5 and 7.5, respectively.

Functions used in this measurement

“hpe5022_waveOverSampleRate” on page 1019

“hpe5022_waveAverage” on page 1022

“hpe5022_waveDelayTime” on page 1025

“hpe5022_measureWave” on page 1028

“hpe5022_waveDipulseExtraction_Q” on page 1051

“hpe5022_selectPattern” on page 119

NLTS by Time Correlation Method Measurement

The NLTS by Time Correlation program sequence is similar to the NLTS by Dipulse Extraction method. The sample program is provided in the furnished sample program.

Overwrite Measurement

The overwrite measurement measures the overwrite ratio of a new data signal with respect to the residual signal from data previously recorded on the disk.

Example 2-15. Sample Program

```
Private Sub CmdOverWrite_Click()
Const PeriodOfOWHFDataPattern As Integer = 3 ' Transition Period of HF Pattern for Overwrite Measurement
    Const PeriodOfOWLFDataPattern As Integer = 8 ' Transition Period of LF Pattern for Overwrite Measurement
    Const NumOfRevolutionForAverage As Integer = 10 ' Number of Revolutions for Averaging
    '
    Dim overwrite As Double, OverWriteData() As Double
    Dim DataSize As Long
    Dim Disp As String
    '
    ' Set Transition Period of Overwrite Test Data Pattern
    '
    ErrorCheck hpe5022_overwriteHfPattern(HpE5022, PeriodOfOWHFDataPattern)
    ErrorCheck hpe5022_overwriteLfPattern(HpE5022, PeriodOfOWLFDataPattern)
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measureOverwrite(HpE5022, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_overwrite_Q(HpE5022, overwrite)
    '
    ErrorCheck hpe5022_overwriteDataSize_Q(HpE5022, DataSize)
    ReDim OverWriteData(DataSize - 1)
    ErrorCheck hpe5022_overwriteData_Q(HpE5022, OverWriteData(0))
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "OverWrite: " + Format(overwrite, "0.0") + " dB"
End Sub
```

Description

The data patterns specified by the “hpe5022_overwriteHfPattern” and “hpe5022_overwriteLfPattern” functions are used automatically in this measurement. It is not necessary to select patterns set by the “hpe5022_selectPattern” function.

Functions used in this measurement

“hpe5022_overwriteHfPattern” on page 136

“hpe5022_overwriteLfPattern” on page 138

“hpe5022_measureOverwrite” on page 910

“hpe5022_overwrite_Q” on page 915

Partial Erasure Measurement

The partial erasure is the amplitude loss of the play-back voltage when two transitions are written at a small separation.

Example 2-16. Sample Program

```
Private Sub CmdPartialErase_Click()
Const NumOfRevolutionForAverage As Integer = 10 ' Number of Revolutions for Averaging
    Dim PartualErase As Double, mean As Double, min As Double, max As Double, stdDev As Double
    Dim PEData() As Double
    Dim DataSize As Long
    Dim Disp As String
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measurePartialErasure(HpE5022, NumOfRevolutionForAverage)
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_partialErasure_Q(HpE5022, PartualErase)
    ErrorCheck hpe5022_partialErasureStatistic_Q(HpE5022, mean, min, max, stdDev)
    '
    ErrorCheck hpe5022_partialErasureDataSize_Q(HpE5022, DataSize)
    ReDim PEData(DataSize - 1)
    ErrorCheck hpe5022_partialErasureData_Q(HpE5022, PEData(0))
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "Partiual Erase: " + vbCrLf _
                    + "Partial Erase: " + Format(PartualErase, "0.0%") + "  " + vbCrLf _
                    + "Mean: " + Format(mean, "0.0%") + "  " + vbCrLf _
                    + "Min: " + Format(min, "0.0%") + "  " + vbCrLf _
                    + "Max: " + Format(max, "0.0%") + "  " + vbCrLf _
                    + "Std Dev: " + Format(stdDev, "0.00")
End Sub
```

Description

The required data patterns (3T and 1T data patterns) for this measurement are used automatically. It is not necessary to select those pattern by the “hpe5022_selectPattern” function.

Functions used in this measurement

“hpe5022_measurePartialErasure” on page 934

“hpe5022_partialErasure_Q” on page 939

“hpe5022_partialErasureStatistic_Q” on page 942

Track Profile Measurement

The Track Profile measurement can observe read offset characteristics of the TAA or the Narrow Band TAA. See Chapter 4 of the Operation Manual for its measurement definition

Example 2-17. Sample Program

```
Private Sub CmdTrackProfile_Click()
Const HeadOffsetStart As Double = -0.000003 ' Start of Head Offset [Meter]
    Const HeadOffsetEnd As Double = 0.000003 ' End of Head Offset [Meter]
    Const NumOfHeadOffsetData As Integer = 21 ' Number of Measurement Points
    '
    Dim trackWidth As Double, ReadWriteOffset As Double
    Dim HeadOffsetData(NumOfHeadOffsetData - 1) As Double
    Dim TrackProfileResult(NumOfHeadOffsetData - 1) As Double
    Dim i As Integer
    Dim Disp As String
    '
    ' Create Array of Head Offset Data
    '
    For i = 0 To NumOfHeadOffsetData - 1
       HeadOffsetData(i) = HeadOffsetStart + _
       i * (HeadOffsetEnd - HeadOffsetStart) / (NumOfHeadOffsetData - 1)
    Next i
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    '
    ' Measurement (Measurement Sequence: Erase -> Write -> Measurement, Track Movement Direction: Both
    '              Measurement Parameter: TAA )
    '
    ErrorCheck hpe5022_measureTrackProfile(HpE5022, hpe5022_SEQ_ER_WR_M, hpe5022_TP_BOTH, hpe5022_MEAS_TAA, _
                                           NumOfHeadOffsetData, HeadOffsetData(0))
    '
    ' Query Measurement Result (Measurement Parameter: TAA, Averaged Data of Forward and Backward)
    '
    ErrorCheck hpe5022_trackProfileData_Q(HpE5022, hpe5022_TP_DATA_AVE, hpe5022_DATA_TAA, _
                                          TrackProfileResult(0))
    ErrorCheck hpe5022_calculateTrackWidth_Q(HpE5022, hpe5022_DATA_TAA, trackWidth)
    ErrorCheck hpe5022_calculateReadWriteOffset_Q(HpE5022, hpe5022_DATA_TAA, ReadWriteOffset)
    '
    ' Display Mesurement Result
    '
    Disp$ = "Track Profile: TAA" + vbCrLf _
        + "Write Track Width: " + Format(trackWidth, "0.00E-##") + vbCrLf _
        + "Read/Write Offset: " + Format(ReadWriteOffset, "0.00E-##") + vbCrLf
    For i = 0 To NumOfHeadOffsetData - 1
       Disp$ = Disp$ + Format(HeadOffsetData(i), "0.00E-##") + " : " + _
                       Format(TrackProfileResult(i), "0.00E-##") + vbCrLf
    Next i
    LblResult.Caption = Disp$
End Sub
```

Description

The “hpe5022_trackProfileData_Q” function returns the array data of the TAA results for the specified segment. As the Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “TrackProfileResult(0)” in line of “hpe5022_trackProfileData_Q” passes the first element of the array by reference.

The “hpe5022_measureTrackProfile” can also measure the Narrow Band TAA, program sequence is similar. The Narrow Band TAA program example is in the furnished sample program.

Functions used in this measurement

“hpe5022_selectPattern” on page 119

“hpe5022_measureTrackProfile” on page 615

“hpe5022_calculateTrackWidth_Q” on page 626

“hpe5022_calculateReadWriteOffset_Q” on page 628

“hpe5022_trackProfileData_Q” on page 623

Micro Track Profile Measurement

Micro Track Profile is a measurement to calculate the read head width. This measurement contains two portions. The first is to create a micro track and the second is to measure the Narrow Band Track Profile of the created micro track.

Example 2-18. Sample Program

```
Private Sub CmdMicroTrackProfile_Click()
'
    ' Parameters to Create a Micro Track
    '
    Const StartErasePointOffset As Double = 0.000003 ' Head Offset at Start Erase Point [Meter]
    Const EraseStep As Double = 0.0000006 ' Step for Erase [Meter]
    Const TAARatio As Double = 0.1 ' TAA Ratio to finsh to create a Micro Track
    '
    ' Parameters to Measure Narroband TAA Track Profile for the Micro Track
    '
    Const HeadOffsetStart As Double = -0.000003 ' Start Offset to Measure a track profile [Meter]
    Const HeadOffsetEnd As Double = 0.000003 ' End Offset to Measure a track profile [Meter]
    Const NumOfHeadOffsetData As Integer = 21 ' Number of Measurement Points
    '
    Dim ReadHeadWidth As Double
    Dim HeadOffsetData(NumOfHeadOffsetData - 1) As Double
    Dim TrackProfileResult(NumOfHeadOffsetData - 1) As Double
    Dim i As Integer
    Dim Disp As String
    '
    ' Create Array of Head Offset Data
    '
    For i = 0 To NumOfHeadOffsetData - 1
       HeadOffsetData(i) = HeadOffsetStart + _
       i * (HeadOffsetEnd - HeadOffsetStart) / (NumOfHeadOffsetData - 1)
    Next i
    '
    ' Create a Micro Track
    '
    ErrorCheck hpe5022_microTrackConfig(HpE5022, StartErasePointOffset, EraseStep, TAARatio)
    ErrorCheck hpe5022_createMicroTrack(HpE5022)
    '
    ' Measure a Narrow Band TAA Track Profile for a Micro Track.
    '
    ErrorCheck hpe5022_measureTrackProfile(HpE5022, hpe5022_SEQ_M, hpe5022_TP_BOTH, hpe5022_MEAS_NB_TAA, _
                                           NumOfHeadOffsetData, HeadOffsetData(0))
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_trackProfileData_Q(HpE5022, hpe5022_TP_DATA_AVE, hpe5022_DATA_NB_TAA, _
                                          TrackProfileResult(0))
    ErrorCheck hpe5022_calculateTrackWidth_Q(HpE5022, hpe5022_DATA_NB_TAA, ReadHeadWidth)
    '
    ' Display Mesurement Result
    '
    Disp$ = "Track Profile: " + vbCrLf + "Read Head Width: " + Format(ReadHeadWidth, "00.0E-##") + vbCrLf
```

```
    Disp$ = Disp$ + vbCrLf + "Head Offset : Narrow Band TAA" + vbCrLf
    For i = 0 To NumOfHeadOffsetData - 1
        Disp$ = Disp$ + Format(HeadOffsetData(i), "0.00E-##") + " : " _
                      + Format(TrackProfileResult(i), "0.00E-##") + vbCrLf
    Next i
    LblResult.Caption = Disp$
    '
End Sub
```

Description

As the output signal of the micro track is low, narrow band track profile is usually used. The recommended head offset of start erase point is approximately 70% of the head width and the step for erase is approximately 10% of it. For example, when the head width is 7μ, the head offset start point for erase is approximately 5μ, the erase step is approximately 0.7μ.

The “hpe5022_trackProfileData_Q” function returns the array data of TAA results for the specified segment. As the Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “TrackProfileResult(0)” in line of “hpe5022_trackProfileData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_microTrackConfig” on page 637

“hpe5022_createMicroTrack” on page 644

“hpe5022_measureTrackProfile” on page 615

“hpe5022_trackProfileData_Q” on page 623

“hpe5022_calculateTrackWidth_Q” on page 626

Write Current Sweep Measurement

The write current sweep measurement is used to observe the write current characteristic of the parameters of the TAA, Narrow Band TAA, PW, Overwrite and NLTS by 5th harmonic method.

Example 2-19. Sample Program

```
Private Sub CmdWriteCurrentSweep_Click()
Const WriteCurrentStart As Double = 0.01 ' Start of Write Current [Ampere]
    Const WriteCurrentEnd As Double = 0.035 ' End of Write Current [Ampere]
    Const NumOfWriteCurrentData As Integer = 20 ' Number of Measurement Points
    '
    Dim WriteCurrentData(NumOfWriteCurrentData - 1) As Double
    Dim WriteCurrentSweepResult(NumOfWriteCurrentData - 1) As Double
    Dim i As Integer
    Dim Disp As String
    '
    ' Create Array of Write Current Data
    '
    For i = 0 To NumOfWriteCurrentData - 1
       WriteCurrentData(i) = WriteCurrentStart + _
       i * (WriteCurrentEnd - WriteCurrentStart) / (NumOfWriteCurrentData - 1)
    Next i
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    '
    ' Measurement (Measurement Sequence: Erase -> Write -> Measurement, Measurement Parameter: TAA)
    '
    ErrorCheck hpe5022_measureWriteCurrentSweep(HpE5022, hpe5022_MEAS_TAA, _
                                            NumOfWriteCurrentData, WriteCurrentData(0))       '
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_writeCurrentSweepData_Q(HpE5022, hpe5022_DATA_TAA, WriteCurrentSweepResult(0))
    '
    ' Display Mesurement Result
    '
    Disp$ = "Write Current Sweep: TAA" + vbCrLf
    For i = 0 To NumOfWriteCurrentData - 1
       Disp$ = Disp$ + Format(WriteCurrentData(i), "0.00E-##") + " A : " _
                     + Format(WriteCurrentSweepResult(i), "0.00E-##") + " V" + vbCrLf
    Next i
    LblResult.Caption = Disp$
End Sub
```

Description

The measurement parameter in the “hpe5022_writeCurrentSweepData_Q” must be in the same category of the measurement done by the “hpe5022_measureWriteCurrentSweep”. For example, when TAA measurement is selected in the “hpe5022_measureWriteCurrentSweep” function, only the parameters of TAA measurement (TAA, TAA positive and TAA negative) can be queried in the “hpe5022_writeCurrentSweepData_Q”.

The “hpe5022_writeCurrentSweepData_Q” function returns the array data of TAA results for the specified segment. As Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “WriteCurrrentResult(0)” in line of “hpe5022_writeCurrentSweepData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_measureWriteCurrentSweep” on page 658

“hpe5022_writeCurrentSweepData_Q” on page 663

“hpe5022_selectPattern” on page 119

Sense Current Sweep Measurement

The sense current sweep measurement is used to observe the sense current characteristics of the measured parameters of the TAA, Narrow Band TAA, PW, Overwrite and NLTS by 5th harmonic method. The program sequence is similar to the Write Sweep Measurement. The sample program is provided in the furnished sample program.

Channel Bit Rate Sweep Measurement

The channel bit rate sweep measurement is used to observe the channel bit rate characteristic of the measured parameters of the TAA, Narrow Band TAA, PW, Baseline, Overwrite and NLTS by 5th harmonic method.

Example 2-20. Sample Program

```
Private Sub CmdChannelBitRateSweep_Click()
Const ChannelBitRateStart As Double = 100000000# ' Start of Channel Bit Rate (bps)
    Const ChannelBitRateEnd As Double = 300000000# ' End of Channel Bit Rate (bps)
    Const NumOfChannelBitRateData As Integer = 21 ' Number of Measurement Points
    '
    Dim ChannelBitRateData(NumOfChannelBitRateData - 1) As Double
    Dim ChannelBitRateResult(NumOfChannelBitRateData - 1) As Double
    Dim i As Integer
    Dim Disp As String
    '
    ' Create the array of Channel Bit Rate data
    '
    For i = 0 To NumOfChannelBitRateData - 1
       ChannelBitRateData(i) = ChannelBitRateStart + _
       i * (ChannelBitRateEnd - ChannelBitRateStart) / (NumOfChannelBitRateData - 1)
    Next i
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    '
    ' Measurement (Measurement : TAA )
    '
    ErrorCheck hpe5022_measureChannelBitRateSweep(HpE5022, hpe5022_MEAS_TAA, _
                                                  NumOfChannelBitRateData, ChannelBitRateData(0))
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_channelBitRateSweepData_Q(HpE5022, hpe5022_DATA_TAA_POS, ChannelBitRateResult(0))
    '
    ' Display Mesurement Result
    '
    Disp$ = "Channel Bit Rate Sweep: TAA Positive" + vbCrLf
    For i = 0 To NumOfChannelBitRateData - 1
       Disp$ = Disp$ + Format(ChannelBitRateData(i), "0.00E-##") + " bps  : " _
                     + Format(ChannelBitRateResult(i), "0.00E-##") + " V" + vbCrLf
    Next i
    LblResult.Caption = Disp$
End Sub
```

Description

The measurement parameter in the “hpe5022_channelBitRateSweepData_Q” must be in the same category of the measurement done by the “hpe5022_measureChannelBitRateSweep”. For example, when TAA measurement is selected in the “hpe5022_measureChannelBitRateSweep” function, only the parameters of TAA measurement (TAA, TAA positive and TAA negative) can be queried in the “hpe5022_channelBitRateSweepData_Q”.

The “hpe5022_channelBitRateSweepData_Q” function returns the array data of TAA results for the specified segment. As Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “ChannelBitRateResult(0)” in line of “hpe5022_channelBitRateSweepData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_selectPattern” on page 119

“hpe5022_measureChannelBitRateSweep” on page 679

“hpe5022_channelBitRateSweepData_Q” on page 684

Roll off Measurement

The roll off measurement is used to observe the flux frequency characteristics of the measured parameters of the TAA and Narrow Band TAA. The measurement sequence is very similar to the channel bit sweep measurement. The sample program is provided in the furnished sample program.

Precompensation Delay Sweep Measurement

The precompensation delay measurement is used to observe the precompensation delay characteristics of the NLTS by the 5th harmonic method.

Example 2-21. Sample Program

```
Private Sub CmdPrecompenSweep_Click()
Const PreCompSweepStart As Double = -0.000000001 ' Start of Precompensation [Second]
    Const PreCompSweepEnd As Double = 0.000000001 ' End of Precompensation [Second]
    Const NumOfPreCompSweep As Integer = 41 ' Number of Measurement Points
    Dim PreCompSweepData(NumOfPreCompSweep - 1) As Double
    Dim NLTSResult(NumOfPreCompSweep - 1) As Double
    Dim i As Integer
    Dim Disp As String
    '
    ' Clear Precompenstion
    '
    ErrorCheck hpe5022_clearPrecomp(HpE5022)
    '
    ' Asign "001x000" into the Precompensation Number 1
    '
    ErrorCheck hpe5022_precompPattern(HpE5022, hpe5022_PREC_PAT_001, hpe5022_PREC_PAT_000, hpe5022_PREC_1)
    '
    ' Precompensation ON for a Test Data Pattern
    ' (The NLTS 5th harmonic Test Data Pattern has a precompensation in this Example)
    '
    ErrorCheck hpe5022_precompState(HpE5022, hpe5022_PAT_NLTS_5TH, VI_TRUE)
    '
    ' Create the array of Precompensation data
    '
    For i = 0 To NumOfPreCompSweep - 1
       PreCompSweepData(i) = PreCompSweepStart + _
       i * (PreCompSweepEnd - PreCompSweepStart) / (NumOfPreCompSweep - 1)
    Next i
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measurePrecompSweep(HpE5022, hpe5022_MEAS_NLTS_5TH, _
                                           NumOfPreCompSweep, PreCompSweepData(0))
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_precompSweepData_Q(HpE5022, hpe5022_DATA_NLTS_5TH, NLTSResult(0))
    '
    ' Display Mesurement Result
    '
    Disp$ = "Precompensation Sweep:" + vbCrLf
    For i = 0 To NumOfPreCompSweep - 1
       Disp$ = Disp$ + Format(PreCompSweepData(i), "0.00E-##") + " : " _
                     + Format(NLTSResult(i), "0.0%") + vbCrLf
    Next i
    LblResult.Caption = Disp$
End Sub
```

Description

The precompensation pattern should be set at “001x 000” (x is a target bit) in order to precompensate the transition for the dibit. As the “hpe5022_measurePrecompSweep” function changes a delay time of the precompensation number 1, the “hpe5022_PREC_1” should be set.

The “hpe5022_precompSweepData_Q” function returns the array data of TAA results for the specified segment. As Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “NLTSResult(0)” in line of “hpe5022_precompSweepData_Q” passes the first element of the array by reference.

Functions used in this measurement

“hpe5022_clearPrecomp” on page 386

“hpe5022_precompPattern” on page 377

“hpe5022_precompState” on page 383

“hpe5022_measurePrecompSweep” on page 696

“hpe5022_precompSweepData_Q” on page 701

Popcorn Noise Measurement (using Counter Module E5041A)

Popcorn noise test is used to detect the head noise referred to as popcorn noise, which occurs shortly after the write operation. The Agilent E5022/E5023 allows you to measure popcorn noise with a programmable time delay after write.

Example 2-22. Sample Program

```
'Sample Program for Popcorn Noise using E5041A
Private Sub CmdPopcorn_Click()

    Const thrPos As Double = 0.0001   '[V] positive Threshold
    Const thrNeg As Double = 0.0001   '[V] negative Threshold
    Const wrTime As Double = 0.0005   '[sec] write time
    Const delay As Double = 0.0001    '[sec] delay time
    Const rdTime As Double = 0.0005   '[sec] read time
    Const counts As Long = 10         ' number of test counts

    Dim writeMode As Integer
    Dim hystLevel As Integer
    Dim failCounPos As Long     'counts popcorn noise from thrPos
    Dim failCounNeg As Long     'counts popcorn noise from thrNeg
    Dim Disp As String

    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    ErrorCheck hpe5022_popcornNoiseCtConfig(HpE5022, thrPos, thrNeg, wrTime, delay, rdTime)
    ErrorCheck hpe5022_popcornNoiseCtWriteModeConfig(HpE5022, hpe5022_WRIT_DATA_POL_FIXED)
    ErrorCheck hpe5022_popcornNoiseCtHysteresisConfig(HpE5022, hpe5022_COMP_HYST_LEVEL_0)

    ' Measurement
    '
    ErrorCheck hpe5022_measurePopcornNoiseCt(HpE5022, counts)

    ' Query Measurement Result
    '
    ErrorCheck hpe5022_popcornNoiseCtWriteModeConfig_Q(HpE5022, writeMode)
    ErrorCheck hpe5022_popcornNoiseCtHysteresisConfig_Q(HpE5022, hystLevel)
    ErrorCheck hpe5022_popcornNoiseCt_Q(HpE5022, failCounPos, failCounNeg)


    ' Display Mesurement Result
    '
    Disp$ = "Popcorn Noise :" + vbCrLf _
                    + "failCoun(+): " + Format(failCounPos, "0.00E-##") + vbCrLf _
                    + "failcount(-)" + Format(failCounNeg, "0.00E-##") + vbCrLf _
                    + "Total failcount" + Format(failCounPos + failCounNeg, "0.00E-##") + vbCrLf

    LblResult.Caption = Disp$

End Sub
```

Description

The Agilent E5041A is a dual counter module designed to detect and count popcorn noise. E5041A has two threshold levels (e.g, hpe5022_COMP_HYST_LEVEL_0) that detect popcorn noise for both positive and negative amplitudes of the input signal. Each threshold has its own hysteresis window that is used to trigger the popcorn noise count. If the number of counted popcorn noise exceeds the maximum number of allowable failcounts, the excess failcounts will not be counted. Refer to “hpe5022_popcornNoiseCt_Q” function of the programming manual for details.

Functions used in this measurement

“hpe5022_popcornNoiseCtConfig” on page 886

“hpe5022_popcornNoiseCtConfig_Q” on page 891

“hpe5022_popcornNoiseCtWriteModeConfig” on page 893

“hpe5022_popcornNoiseCtHysteresisConfig” on page 899

“hpe5022_popcornNoiseCtHysteresisConfig_Q” on page 902

“hpe5022_setupPopcornNoiseCt” on page 906

“hpe5022_popcornNoiseCt_Q” on page 908

Stability Measurement

The stability measurement allows you to observe the amplitude variations during many write-read iterations. See Chapter 4 in the Operation Manual for its measurement definition.

Example 2-23. Sample Program

```
Private Sub CmdStability_Click()
Const NumOfSegment As Integer = 5 ' Number of Segments in a track
    Const NumOfRevolution As Integer = 3 ' Number of Revolutions
    Const writeRatio As Double = 0.4 ' Write Ratio in a Segment
    Const readRatio As Double = 0.4 ' Read Ration in a Segment
    '
    Dim Stability(NumOfRevolution - 1, NumOfSegment - 1) As Double
    Dim mean As Double, min As Double, max As Double, stdDev As Double
    Dim SegmentNo As Integer, RevolutionNo As Integer
    Dim Disp As String
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    '
    ' Setup (Polarity: Same)
    '
    ErrorCheck hpe5022_stabilityConfig(HpE5022, hpe5022_STABILITY_IS_POL_NORM, writeRatio, readRatio)
    '
    ' Measurement (Measurement Parameter : TAA)
    '
    ErrorCheck hpe5022_measureStability(HpE5022, hpe5022_MEAS_TAA, NumOfSegment, NumOfRevolution)
    '
    Disp$ = "Stability: TTA" + vbCrLf
    For SegmentNo = 0 To NumOfSegment - 1
        '
        ' Query Measurement Result
        '
        ErrorCheck hpe5022_stabilityData_Q(HpE5022, SegmentNo, hpe5022_DATA_TAA, Stability(0, SegmentNo))
        ErrorCheck hpe5022_stability_Q(HpE5022, SegmentNo, hpe5022_DATA_TAA, mean, min, max, stdDev)
        '
        ' Display Mesurement Result
        '
        Disp$ = Disp$ + "Segment No." + Str(SegmentNo) + ";" + vbCrLf
        Disp$ = Disp$ + "  TAA:   " + vbCrLf
        For RevolutionNo = 0 To NumOfRevolution - 1
            Disp$ = Disp$ + "    Revolution No." + Str(RevolutionNo) + ":  " _
            + Format(Stability(RevolutionNo, SegmentNo), "0.00E-##") + " V" + vbCrLf
        Next RevolutionNo
        Disp$ = Disp$ + "  Mean:  " + Format(mean, "0.00E-##") + " V" + vbCrLf
        Disp$ = Disp$ + "  Min:   " + Format(min, "0.00E-##") + " V" + vbCrLf
        Disp$ = Disp$ + "  Max:   " + Format(max, "0.00E-##") + " V" + vbCrLf
        Disp$ = Disp$ + "  StdDev:" + Format(stdDev, "0.00E-##") + vbCrLf
```

```
    Next SegmentNo
    LblResult.Caption = Disp$
End Sub
```

Description

The “hpe5022_stabilityData_Q” function returns the array data of the TAA results for the specified segment. As Agilent E5022/E5023 DLL doesn’t accept Automation SAFEARRAYs directly, you have to pass the first element of the array by reference to pass an entire numeric array. See the Visual Basic On-line Manual for the Automation SAFEARRAY. The “Stability(0, segment No)” in line of “hpe5022_stabilityData_Q” passes the first element of the array by reference.

The stability measurement requires to select data pattern. If the required data pattern has already been set before sequence, you need not set in this section.

Functions used in this measurement

“hpe5022_stabilityConfig” on page 511

“hpe5022_measureStability” on page 524

“hpe5022_stabilityData_Q” on page 531

“hpe5022_stability_Q” on page 533

“hpe5022_selectPattern” on page 119

Bit Error Test Measurement

The bit error test measurement allows you to observe the bit error analysis. See Chapter 4 in the Operation Manual for its measurement definition.

Example 2-24. Sample Program

```
Private Sub CmdBer_Click()
    Const BerUserDataBitRate As Double = 120000000# ' User Data Bit Rate [bps]
    Const ReferenceClock As Double = 20000000# ' Channel IC Reference Clock

    Const BerTestPattern1 As String = "11001000" ' Test Pattern in Binary Format

    Const TrackPreambleLength As Long = 1000 ' Track Preamble Length [Byte]
    Const GapLength As Long = 55 ' Dummy if the gap length is Auto [Byte]
    Const SectPreambleLength As Long = 100 ' Sector Preamble Length [Byte]
    Const DataLength As Long = 1024 ' Data Length [Byte]
    Const NoOfSector As Integer = 10 ' Number of Sector per Track
    Const noofave As Integer = 1
    Const MeasByteCount As Double = 1000 ' Measurement Byte Count [Byte]
    Const ErrBitPerByte As Double = 1 ' Error Bit Per Byte
    '
    Dim FilterCutOffFreq As Double, FilterBoost As Double
    Dim bitLength As Long, symbol As Long, totalSector As Long, lostSector As Long
    Dim NumOfSize As Long, ErrorCount() As Long, LostCount() As Long
    Dim ber As Double, delay(2) As Double, Property1() As Double
    Dim Disp As String
    Dim IcName As String * 10
    Dim i As Integer
    ' FilterCutOffFreq = 20000000#
    ' FilterBoost = 5.5
    ' Property1(0) = xxxx
    ' Properry1(1) = xxxx
    '
    ErrorCheck hpe5022_BER_channelIcIdn_Q(HpE5022, IcName)
    ErrorCheck hpe5022_BER_userDataBitRate(HpE5022, BerUserDataBitRate)
    ErrorCheck hpe5022_BER_channelIcReferenceClock(HpE5022, ReferenceClock)
    '
    ' Create BER Data Pattern
    '
    bitLength = Len(BerTestPattern1)
    ErrorCheck hpe5022_BER_userPattern(HpE5022, hpe5022_BER_PAT_1, hpe5022_USER_DATA_BIN, bitLength, _
                                       BerTestPattern1)
    ErrorCheck hpe5022_BER_selectPattern(HpE5022, hpe5022_BER_PAT_1)
    '
    ' Define Track and Sector Format
    '
    ErrorCheck hpe5022_BER_trackFormat(HpE5022, TrackPreambleLength, NoOfSector, VI_FALSE, GapLength)
    ErrorCheck hpe5022_BER_sectorFormat(HpE5022, SectPreambleLength, DataLength)
    ErrorCheck hpe5022_BER_berMeasByteCount(HpE5022, MeasByteCount)
```

```
ErrorCheck hpe5022_BER_errorBitPerByte(HpE5022, ErrBitPerByte)
ErrorCheck hpe5022_BER_channelIcFirCoefReset(HpE5022)
'
' Specify Low Pass Filter
'
' ErrorCheck hpe5022_BER_channelIcLowPassFilter(HpE5022, FilterCutOffFreq, FilterBoost)
'
' Define Precompensation
'
' ErrorCheck hpe5022_BER_channelIcPrecompDelay(HpE5022, Delay(0))
ErrorCheck hpe5022_BER_channelIcPrecompState(HpE5022, VI_TRUE)
'
' Define Encorder
'
ErrorCheck hpe5022_BER_channelIcEndec(HpE5022, VI_TRUE, hpe5022_BER_ENDEC_8_9, VI_TRUE)
'
'  Define Channel IC Property
'
' ErrorCheck hpe5022_BER_channelIcProperty(HpE5022, hpe5022_BER_CIC_xxxx_PROP_xxx, Property1(0))
'
'
' Define Logging Mode
'
ErrorCheck hpe5022_BER_berDataLoggingMode(HpE5022, VI_TRUE, hpe5022_BER_RAW_DATA_ALL)
'
' Define Optimization
'
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_LPF, VI_TRUE)
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_FIR, VI_TRUE)
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_PREC_DEL_COAR, VI_TRUE)
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_PREC_DEL_FINE, VI_TRUE)
ErrorCheck hpe5022_BER_optimize(HpE5022)
'
ErrorCheck hpe5022_BER_channelIcLowPassFilter_Q(HpE5022, FilterCutOffFreq, FilterBoost)
'
' Perform Measurement
'
ErrorCheck hpe5022_BER_measureBer(HpE5022, hpe5022_SEQ_ER_WR_M, noofave)
'
' Query Test Result
'
ErrorCheck hpe5022_BER_ber_Q(HpE5022, ber, symbol, totalSector, lostSector)
ErrorCheck hpe5022_BER_sectorErrorCountSize_Q(HpE5022, NumOfSize)
ReDim ErrorCount(NumOfSize - 1)
ReDim LostCount(NumOfSize - 1)
ErrorCheck hpe5022_BER_sectorErrorCount_Q(HpE5022, ErrorCount(0), LostCount(0))
'
Disp$ = "Bit Error Rate: " + vbCrLf _
      + "BER: " + Format(ber, "0.00E-##") + vbCrLf _
```

```
          + "Symbol: " + Format(symbol, "###0") + vbCrLf _
          + "Total Sector: " + Format(totalSector, "###0") + vbCrLf _
          + "Lost Sector: " + Format(lostSector, "###0") + vbCrLf + vbCrLf _
          + "Filter Cut Off: " + Format(FilterCutOffFreq, "0.00E-##") + " Hz" + vbCrLf _
          + "Filter Boost: " + Format(FilterBoost, "0.00") + " dB" + vbCrLf
    For i = 0 To NumOfSize - 1
       Disp$ = Disp$ + Format(ErrorCount(i), "0.0") + " : " + Format(LostCount(i), "0.0") + vbCrLf
    Next i
    LblChannelICName.Caption = IcName$
    LblResult.Caption = Disp$
    '
End Sub
```

Description

The test data pattern of the bit error measurement is different from the other measurement. The “hpe5022_BER_userPattern” and “hpe5022_BER_userPattern” functions defines the pattern. The “hpe5022_BER_selectPattern” function selects the pattern.

Functions used in this measurement

“hpe5022_BER_userDataBitRate”

“hpe5022_BER_trackFormat”

“hpe5022_BER_sectorFormat”

“hpe5022_BER_selectPattern”

“hpe5022_BER_userPattern”

“hpe5022_BER_channelIc_Idn_Q”

“hpe5022_BER_channelIcReferenceClock”

“hpe5022_BER_channelIcLowPassFilter_Q”

“hpe5022_BER_channelIcFirCoefReset”

“hpe5022_BER_channelIcPrecompState”

“hpe5022_BER_channelIcEndec”

“hpe5022_BER_optimizeState”

“hpe5022_BER_optimize”

“hpe5022_BER_berMeasByteCount”

“hpe5022_BER_errorBitPerByte”

“hpe5022_BER_berDataLoggingMode”

“hpe5022_BER_measureBer”

“hpe5022_BER_ber_Q”

“hpe5022_BER_sectorErrorCountSize_Q”

“hpe5022_BER_sectorErrorCount_Q”

Bit Error Track Profile Measurement

The bit error track profile measurement allows you to observe read offset characteristics of the bit error rate.

Example 2-25. Sample Program

```
Private Sub CmdBerTrackProfile_Click()
    Const BerUserDataBitRate As Double = 120000000#
    Const ReferenceClock As Double = 20000000#
    '
    Const BerTestPattern1 As String = "11001000"
    '
    Const TrackPreambleLength As Long = 1000
    Const GapLength As Long = 55 ' Dummy if the gap length is Auto
    Const SectPreambleLength As Long = 100
    Const DataLength As Long = 1024
    Const NoOfSector As Integer = 32
    Const noofave As Integer = 1
    Const MeasByteCount As Double = 1000
    Const ErrBitPerByte As Double = 1
    Const otcThreshold As Double = 0.001
    '
    Const NumOfPoint As Integer = 21
    Const OffsetRange As Double = 0.000002
    '
    Dim bitLength As Long, symbol As Long, totalSector As Long, lostSector As Long
    Dim ber As Double, otc As Double, otcPos As Double, otcNeg As Double
    Dim offset(NumOfPoint - 1) As Double
    Dim result(NumOfPoint - 1) As Double
    Dim Disp As String
    Dim i As Integer
    '
    ErrorCheck hpe5022_BER_userDataBitRate(HpE5022, BerUserDataBitRate)
    ErrorCheck hpe5022_BER_channelIcReferenceClock(HpE5022, ReferenceClock)
    '
    ' Create BER Data Pattern
    '
    bitLength = Len(BerTestPattern1)
    ErrorCheck hpe5022_BER_userPattern(HpE5022, hpe5022_BER_PAT_1, hpe5022_USER_DATA_BIN, bitLength, _
                                       BerTestPattern1)
    ErrorCheck hpe5022_BER_selectPattern(HpE5022, hpe5022_BER_PAT_1)
    '
    ' Define Track and Sector Format
    '
```

```
    ErrorCheck hpe5022_BER_trackFormat(HpE5022, TrackPreambleLength, NoOfSector, VI_FALSE, GapLength)
    ErrorCheck hpe5022_BER_sectorFormat(HpE5022, SectPreambleLength, DataLength)
    ErrorCheck hpe5022_BER_berMeasByteCount(HpE5022, MeasByteCount)
    ErrorCheck hpe5022_BER_errorBitPerByte(HpE5022, ErrBitPerByte)
    ErrorCheck hpe5022_BER_channelIcFirCoefReset(HpE5022)
    '
    ' Precompensation State
    '
    ErrorCheck hpe5022_BER_channelIcPrecompState(HpE5022, VI_TRUE)
    '
    ' Define Encorder
    '
    ErrorCheck hpe5022_BER_channelIcEndec(HpE5022, VI_TRUE, hpe5022_BER_ENDEC_8_9, VI_TRUE)
    '
    ' Defiene OTC Threshold Level
    '
    ErrorCheck hpe5022_BER_otcThreshold(HpE5022, otcThreshold)
    '
    ' Define Optimization
    '
    ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_LPF, VI_TRUE)
    ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_FIR, VI_TRUE)
    ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_PREC_DEL_COAR, VI_TRUE)
    ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_PREC_DEL_FINE, VI_TRUE)
    ErrorCheck hpe5022_BER_optimize(HpE5022)
    '
    ErrorCheck hpe5022_BER_measureTrackProfile(HpE5022, hpe5022_SEQ_ER_WR_M, hpe5022_MEAS_BER _
                                              , NumOfPoint, OffsetRange, noofave)
    ErrorCheck hpe5022_executionMode(HpE5022, hpe5022_EXEC_WAIT_START)

    ErrorCheck hpe5022_BER_trackProfileData_Q(HpE5022, hpe5022_DATA_BER, offset(0), result(0))
    ErrorCheck hpe5022_BER_calculateOtc_Q(HpE5022, otc, otcPos, otcNeg)
    Disp$ = "Bit Error Rate: " + vbCrLf _
          + "OTC: " + Format(otc, "0.00E-##") + vbCrLf _
          + "OTC Positive: " + Format(otcPos, "0.00E-##") + vbCrLf _
          + "OTC Negative: " + Format(otcNeg, "0.00E-##") + vbCrLf + vbCrLf _
          + "Offset  :  BER" + vbCrLf
    For i = 0 To NumOfPoint - 1
       Disp$ = Disp$ + Format(offset(i), "0.00E-##") _
                     + " : " + Format(result(i), "0.00E-##") + vbCrLf
    Next i
    LblResult.Caption = Disp$
End Sub
```

Functions used in this measurement

“hpe5022_wai” on page 101

“hpe5022_abort” on page 103

“hpe5022_executionMode” on page 97

“hpe5022_BER_userDataBitRate”

“hpe5022_BER_trackFormat”

“hpe5022_BER_sectorFormat”

“hpe5022_BER_selectPattern”

“hpe5022_BER_userPattern”

“hpe5022_BER_channelIc_Idn_Q”

“hpe5022_BER_channelIcReferenceClock”

“hpe5022_BER_channelIcLowPassFilter_Q”

“hpe5022_BER_channelIcFirCoefReset”

“hpe5022_BER_channelIcPrecompState”

“hpe5022_BER_channelIcEndec”

“hpe5022_BER_optimizeState”

“hpe5022_BER_optimize”

“hpe5022_BER_berMeasByteCount”

“hpe5022_BER_errorBitPerByte”

“hpe5022_BER_measureTrackProfile”

“hpe5022_BER_trackProfileData_Q”

Adjacent Track Squeeze Measurement

The objective of this test is to specify the adjacent track spacings or track pitch in order to maximize track density with the condition that the specified error rate level is met or exceeded. Refer to Chapter 5 of the programming manual for details.

Example 2-26. Sample Program

```
'Sample Program for On-track BER with Squeezing Adjacent Track

Private Sub CmdSqueeze_Click()

    Const BerUserDataBitRate As Double = 120000000# ' User Data Bit Rate [bps]
    Const ReferenceClock As Double = 20000000# ' Channel IC Reference Clock

    Const BerTestPattern1 As String = "11001000" ' Test Pattern in Binary Format

    Const TrackPreambleLength As Long = 1000 ' Track Preamble Length [Byte]
    Const GapLength As Long = 55 ' Dummy if the gap length is Auto [Byte]
    Const SectPreambleLength As Long = 100 ' Sector Preamble Length [Byte]
    Const DataLength As Long = 1024 ' Data Length [Byte]
    Const NoOfSector As Integer = 10 ' Number of Sector per Track
    Const NoOfave As Integer = 1
    Const MeasByteCount As Double = 1000 ' Measurement Byte Count [Byte]
    Const ErrBitPerByte As Double = 1 ' Error Bit Per Byte
    Const points As Integer = 50 'Number of measurement points for On-track BER


    Const HeadOffsetStart As Double = -0.000003 ' Start of Head Offset [Meter]
    Const HeadOffsetEnd As Double = 0.000003 ' End of Head Offset [Meter]
    Const NumOfPoints As Integer = 21 ' Number of Measurement Points
    Const back_AdjacentPos As Integer = 0.000003 '[m] Background Adjacent Track Position
    Const fore_AdjacentPos As Integer = 0.000004 '[m] Foreground Adjacent Track Position

    Dim HeadOffsetData(NumOfPoints - 1) As Double
    Dim On_trackProfile(NumOfPoints - 1) As Double


    Dim FilterCutOffFreq As Double, FilterBoost As Double
    Dim ber(NumOfPoints - 1) As Double
    Dim symbol(NumOfPoints - 1) As Long
    Dim totalSector(NumOfPoints - 1) As Long
    Dim lostSector(NumOfPoints - 1) As Long
    Dim bitLength As Long
    Dim NumOfSize As Long
    Dim delay(2) As Double, Property1() As Double
    Dim Disp As String
    Dim IcName As String * 10

    Dim i As Integer
```

```
' FilterCutOffFreq = 20000000#
' FilterBoost = 5.5
' Property1(0) = xxxx
' Properry1(1) = xxxx
'
ErrorCheck hpe5022_BER_channelIcIdn_Q(HpE5022, IcName)
ErrorCheck hpe5022_BER_userDataBitRate(HpE5022, BerUserDataBitRate)
ErrorCheck hpe5022_BER_channelIcReferenceClock(HpE5022, ReferenceClock)
'
' Create BER Data Pattern
'
bitLength = Len(BerTestPattern1)
ErrorCheck hpe5022_BER_userPattern(HpE5022, hpe5022_BER_PAT_1, hpe5022_USER_DATA_BIN, bitLength, _
                                   BerTestPattern1)
ErrorCheck hpe5022_BER_selectPattern(HpE5022, hpe5022_BER_PAT_1)
'
' Define Track and Sector Format
'
ErrorCheck hpe5022_BER_trackFormat(HpE5022, TrackPreambleLength, NoOfSector, VI_FALSE, GapLength)
ErrorCheck hpe5022_BER_sectorFormat(HpE5022, SectPreambleLength, DataLength)
ErrorCheck hpe5022_BER_berMeasByteCount(HpE5022, MeasByteCount)
ErrorCheck hpe5022_BER_errorBitPerByte(HpE5022, ErrBitPerByte)
ErrorCheck hpe5022_BER_channelIcFirCoefReset(HpE5022)
'
' Specify Low Pass Filter
'
' ErrorCheck hpe5022_BER_channelIcLowPassFilter(HpE5022, FilterCutOffFreq, FilterBoost)
'
' Define Precompensation
'
' ErrorCheck hpe5022_BER_channelIcPrecompDelay(HpE5022, Delay(0))
ErrorCheck hpe5022_BER_channelIcPrecompState(HpE5022, VI_TRUE)
'
' Define Encorder
'
ErrorCheck hpe5022_BER_channelIcEndec(HpE5022, VI_TRUE, hpe5022_BER_ENDEC_8_9, VI_TRUE)
'
'  Define Channel IC Property
'
' ErrorCheck hpe5022_BER_channelIcProperty(HpE5022, hpe5022_BER_CIC_xxxx_PROP_xxx, Property1(0))
'
' Define Optimization
'
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_LPF, VI_TRUE)
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_FIR, VI_TRUE)
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_PREC_DEL_COAR, VI_TRUE)
ErrorCheck hpe5022_BER_optimizeState(HpE5022, hpe5022_BER_OPT_PREC_DEL_FINE, VI_TRUE)
ErrorCheck hpe5022_BER_optimize(HpE5022)
'
```

```
    ErrorCheck hpe5022_BER_channelIcLowPassFilter_Q(HpE5022, FilterCutOffFreq, FilterBoost)

    '

    ' Create Array of Head Offset Data
    '
    For i = 0 To NumOfPoints - 1
       HeadOffsetData(i) = HeadOffsetStart + _
       i * (HeadOffsetEnd - HeadOffsetStart) / (NumOfPoints - 1)
    Next i


    ' Execute Measurement

    ErrorCheck hpe5022_BER_measureAdjacentTrackSqueeze(HpE5022, hpe5022_BER_ADJACENT_TRACK_BOTH, hpe5022_BER_PAT_2, _
                                                       back_AdjacentPos, hpe5022_BER_ADJACENT_TRACK_BOTH, _
                                                       hpe5022_BER_PAT_5, NumOfPoints, HeadOffsetData(0), NoOfave)
    ' Query Test Results
    '(Returns the data for every measurement point)
    ErrorCheck hpe5022_BER_adjacentTrackSqueezeData_Q(HpE5022, On_trackProfile(0))

    '(Returns the computed ber, symbol, total sector and lost sector for every measurement point)
    ErrorCheck hpe5022_BER_adjacentTrackSqueezeDataEx_Q(HpE5022, ber(0), symbol(0), totalSector(0), lostSector(0))


    'Display Test Results

     For i = 0 To NumOfPoints - 1
         Disp$ = Disp$ + Format(HeadOffsetData(i), "0.00E-##") + " : " + _
                         Format(On_trackProfile(i), "0.00E-##") + " : " + _
                         Format(ber(i), "0.00E-##") + vbCrLf
     Next i


    'LblChannelICName.Caption = IcName$
    LblResult.Caption = Disp$


End Sub
```

Functions used in this measurement

“hpe5022_BER_userDataBitRate”

“hpe5022_BER_trackFormat”

“hpe5022_BER_sectorFormat”

“hpe5022_BER_selectPattern”

“hpe5022_BER_userPattern”

“hpe5022_BER_channelIc_Idn_Q”

“hpe5022_BER_channelIcReferenceClock”

“hpe5022_BER_channelIcLowPassFilter_Q”

“hpe5022_BER_channelIcFirCoefReset”

“hpe5022_BER_channelIcPrecompState”

“hpe5022_BER_channelIcEndec”

“hpe5022_BER_optimizeState”

“hpe5022_BER_optimize”

“hpe5022_BER_berMeasByteCount”

“hpe5022_BER_errorBitPerByte”

“hpe5022_BER_measureAdjacentTrackSqueeze”

“hpe5022_BER_adjacentTrackSqueezeData_Q”

“hpe5022_BER_adjacentTrackSqueezeDataEx_Q”

Track Offset Compensation

The objective of this program is to compensate the head positon during read sequence in order to make the read head positon be at the center of write track. See the Track Offset Compensation section in the chapter 5 of the operation manual for sequence flow.

Example 2-27. Sample Program

```
'Private Sub CmdTrkOffsetComp_Click()
    Const CalibrationSize As Integer = 21 ' Track Profile Points for Calibration [points]
    Const CalibrationRange As Double = 0.000003 ' Track Profile Range for Calbration [m]
    Const NumOfAverage As Integer = 100
    Const RepeatInterval As Integer = 10
    Const AveForRev As Integer = 2
    '
    Dim HeadOffset As Double, taa As Double, taaPos As Double, taaNeg As Double, taaAsym As Double
    Dim AcTaa As Double, AcTaaPos As Double, AcTaaNeg As Double, AcTaaAsym As Double
    Dim i As Integer
    Dim Disp As String
    '
    Disp$ = "Offset Value: " + vbCrLf
    AcTaa = 0
    AcTaaPos = 0
    AcTaaNeg = 0
    AcTaaAsym = 0
    '
    ' Define Track Profile Measurements (points and range) for Calibration.
    '
    ErrorCheck hpe5022_trackOffsetCompProfile(AgilentE5022, CalibrationSize, CalibrationRange)
    '
    ' Define the Burst Pattern Automatically.
    ' When you define manually, use hpe5022_trackOffsetCompPattern
    '
    ErrorCheck hpe5022_trackOffsetCompPatternAuto(AgilentE5022)
    '
    ' Write Burst Pattern
    '
    ErrorCheck hpe5022_writeTrackOffsetCompPattern(AgilentE5022)
    '
    ' Set the average number for hpe5022_executeTrackOffsetComp
    '
    ErrorCheck hpe5022_trackOffsetCompRevolution(AgilentE5022, AveForRev)
    '
    ' Track Offset Compenstion State ON
    ' Write & Read Data Pattern besides the Burst Pattern area.
    '
    ErrorCheck hpe5022_trackOffsetCompPatternState(AgilentE5022, VI_TRUE)
    '
    ' Erase on Data Area and Write Data Pattern
    '
```

```
    ErrorCheck hpe5022_measureTaa(AgilentE5022, hpe5022_SEQ_ER_WR_M, 1)
    ErrorCheck hpe5022_taa_Q(AgilentE5022, taa, taaPos, taaNeg, taaAsym)
    '
    AcTaa = AcTaa + taa
    AcTaaPos = AcTaaPos + taaPos
    AcTaaNeg = AcTaaNeg + taaNeg
    AcTaaAsym = AcTaaAsym + taaAsym
    '
    For i = 1 To NumOfAverage
        '
        ' Each Repeat Interval time, Drift is compensated.
        ' If you want to know how long drifted, use hpe5022_trackOffsetCompValue_Q
        '
        If i Mod RepeatInterval = 0 Then
            ErrorCheck hpe5022_executeTrackOffsetComp(AgilentE5022)
            ErrorCheck hpe5022_trackOffsetCompValue_Q(AgilentE5022, HeadOffset) ' (optional)
            Disp$ = Disp$ + Format(i, "0") + ": " + Format(HeadOffset, "0.00E-##") + vbCrLf ' (optional)
        End If
        '
        '  Mesure TAA as an example
        '
        ErrorCheck hpe5022_measureTaa(AgilentE5022, hpe5022_SEQ_M, 10)
        ErrorCheck hpe5022_taa_Q(AgilentE5022, taa, taaPos, taaNeg, taaAsym)
        AcTaa = AcTaa + taa
        AcTaaPos = AcTaaPos + taaPos
        AcTaaNeg = AcTaaNeg + taaNeg
        AcTaaAsym = AcTaaAsym + taaAsym
    Next i
    '
    taa = AcTaa / NumOfAverage
    taaPos = AcTaaPos / NumOfAverage
    taaNeg = AcTaaNeg / NumOfAverage
    taaAsym = AcTaaAsym / NumOfAverage
    '
    LblResult.Caption = Disp$ + vbCrLf _
                    + "TAA :" + vbCrLf _
                    + "TAA: " + Format(taa, "0.00E-##V") + vbCrLf _
                    + "TAA+: " + Format(taaPos, "0.00E-##V") + vbCrLf _
                    + "TAA-: " + Format(taaNeg, "0.00E-##V") + vbCrLf _
                    + "TAA Asym: " + Format(taaAsym, "0.0%")

End Sub
```

Functions used in this measurement

“hpe5022_trackOffsetCompProfile” on page 964

“hpe5022_trackOffsetCompPatternAuto” on page 974

“hpe5022_writeTrackOffsetCompPattern” on page 976

“hpe5022_trackOffsetCompPatternState” on page 978

“hpe5022_trackOffsetCompRevolution” on page 981
“hpe5022_executeTrackOffsetComp” on page 983
“hpe5022_trackOffsetCompValue_Q” on page 986

Burst Pattern Write

Write Bursts are position information written along the data track in order to detect the position of the head and to see to it that the head stays on the center of its designated track.

Example 2-28. Sample Program

```
'Sample Program for Burst Pattern Write

Private Sub CmdWriteBurst_Click()

Const segment As Integer = 2
Const offset As Integer = 0.000005     'Write track offset [meter]
Const burstStart As Integer = 0.00005  'Start of burst pattern [sec]
Const burstLength As Integer = 0.0001  'Length of burst pattern [sec]
Const gateDelay As Integer = 0.002     'Time delay of write gate [sec]
Const gateAperture As Integer = 0.0001 'Aperture time of write gate [sec]
Const frequency As Double = 5000000    'Frequency of burst pattern [bps]


'Configure Write Burst Pattern
ErrorCheck hpe5022_burstPatternConfig(HpE5022, burstStart, burstLength)
ErrorCheck hpe5022_burstGateConfig(HpE5022, gateDelay, gateAperture)
ErrorCheck hpe5022_burstFrequencyConfig(HpE5022, frequency)


'To write multi-bursts pattern change the parameters above and re-execute "hpe5022_writeBurst"
'Execute Write Burst Pattern
ErrorCheck hpe5022_writeBurst(HpE5022, segment, offset)

'Query Burst Pattern
ErrorCheck hpe5022_burstPatternConfig_Q(HpE5022, burstStart, burstLength)
ErrorCheck hpe5022_burstGateConfig_Q(HpE5022, gateDelay, gateAperture)

End Sub
```

Description

The E5022/E5023 system allows the user to write burst pattern along the media in any form as required by measurement applications. E5022/E5023 will only allow one burst pattern to be written for each track rotation. Thus in order to write multi-burst pattern the user must set the gate parameters for each track rotation.

Due to internal constraints of E5037A (data generator module), data writing is actually delayed by a value of **30**× **16T** (T=1/ f, where f is the frequency of the burst pattern). Thus, the user must compensate this value when setting the burst length. Refer to “hpe5022_burstPatternConfig” function of the programming manual for details.

Functions used in this measurement

“hpe5022_writeBurst” on page 993

“hpe5022_burstPatternConfig” on page 997

“hpe5022_burstPatternConfig_Q” on page 1000

“hpe5022_burstFrequencyConfig” on page 1001

“hpe5022_burstFrequencyConfig_Q” on page 1003

“hpe5022_burstGateConfig” on page 1004

“hpe5022_burstGateConfig_Q” on page 1007

Setup-then-Measure Program

The setup-then-measure program is provided to give a high through put measurement. The setup for each measurement is done in advance. The program listed below shows some portion of the sample program. The whole sample program is provided by file name “sample2”.

Example 2-29. Setup then Measure Sample Program

```
Private Sub CmdMeasure_Click()

    Dim i As Integer
    Dim Disp As String
    '
    Track(1) = TrackNoForOverWrite1
    Track(2) = TrackNoForOverWrite2
    '
    ErrorCheck hpe5022_measure(hpe5022, IdParametric)
    ErrorCheck hpe5022_parametric_Q(hpe5022, HFTAA, LFTAA, IsolatedTAA, pw, baseline, resolution)

    For i = 1 To 2
        ErrorCheck hpe5022_track(hpe5022, Track(i))
        ErrorCheck hpe5022_measure(hpe5022, IdOverwrite)
        ErrorCheck hpe5022_overwrite_Q(hpe5022, overwrite(i))
    Next i

    ErrorCheck hpe5022_selectPattern(hpe5022, hpe5022_PAT_HF)
    ErrorCheck hpe5022_measure(hpe5022, IdTrackProfile)
    ErrorCheck hpe5022_trackProfileData_Q(hpe5022, hpe5022_TP_DATA_AVE, hpe5022_DATA_TAA, TrackProfileResult(0))
    ErrorCheck hpe5022_calculateTrackWidth_Q(hpe5022, hpe5022_DATA_TAA, trackWidth)
    ErrorCheck hpe5022_calculateReadWriteOffset_Q(hpe5022, hpe5022_DATA_TAA, ReadWriteOffset)

    ErrorCheck hpe5022_selectPattern(hpe5022, hpe5022_PAT_LF)
    ErrorCheck hpe5022_measure(hpe5022, IdWriteCurrentSweep)
    ErrorCheck hpe5022_writeCurrentSweepData_Q(hpe5022, hpe5022_DATA_TAA, WriteCurrentResult(0))


    LblResult(0).Caption = "Parametric Measuremet:" + vbCrLf _
                        + " HF TAA: " + Format(HFTAA, "0.00E-##V") + vbCrLf _
                        + " LF TAA: " + Format(LFTAA, "0.00E-##V") + vbCrLf _
                        + " Isolated TAA: " + Format(IsolatedTAA, "0.00E-##V") + vbCrLf _
                        + " PW: " + Format(pw, "0.00E-##s") + vbCrLf _
                        + " Baseline: " + Format(baseline, "0.00E-##V") + vbCrLf _
                        + " Resolution: " + Format(resolution, "0.0%") + vbCrLf _
                        + "Overwrite Measurement " + vbCrLf _
                        + " Overwrite 1: " + Format(overwrite(1), "0.0dB") + vbCrLf _
                        + " Overwrite 2: " + Format(overwrite(2), "0.0dB")
    '
    '
    Disp$ = "Track Profile: TAA HF" + vbCrLf _
        + "Track Width: " + Format(trackWidth, "0.00E-##m") + vbCrLf _
```

```
        + "ReadWrite Offset: " + Format(ReadWriteOffset, "0.00E-##m") + vbCrLf
    For i = 0 To NumOfHeadOffsetData - 1
      Disp$ = Disp$ + Format(HeadOffsetData(i), "0.00E-##m") + " : " + Format(TrackProfileResult(i), "0.00E-##V") +
vbCrLf
    Next i
    LblResult(1).Caption = Disp$

    Disp$ = "WriteCurrent: TAA LF" + vbCrLf
    For i = 0 To NumOfWriteCurrentData - 1
      Disp$ = Disp$ + Format(WriteCurrentData(i), "0.00E-##A") + " : " + Format(WriteCurrentResult(i), "0.00E-##V") +
vbCrLf
    Next i
    LblResult(2).Caption = Disp$
End Sub
Private Sub CmdSetup_Click()
    '
    ' Instrument Address Set (This exapmple for Option 102/103)
    '
    Const NumberOfInstrument As Integer = 7
    '
    Dim status, StatusArray(NumberOfInstrument - 1) As Long
    Dim rsrc As rsrcArrayType
    Dim ModuleNo, i As Integer
    Dim ModuleDescription As Variant
    Dim Disp As String
    ModuleDescription = Array("E5035A ", "E5036A ", "E5037A ", "E5038A ", _
                              "E5040A ", "Spinstand ", "Oscilloscope ")
    '
    ' Instruments Address Set
    '
    rsrc.instr(0) = "VXI0::5::INSTR" ' E5035A Spinstand I/F Module
    rsrc.instr(1) = "VXI0::6::INSTR" ' E5036A FilterMatrix Module
    rsrc.instr(2) = "VXI0::7::INSTR" ' E5037A Data Generator Module
    rsrc.instr(3) = "VXI0::8::INSTR" ' E5038A Parametric Module
    rsrc.instr(4) = "VXI0::10::INSTR" ' E5040A Spectrum Measurement Module
    rsrc.instr(5) = "ASRL1::INSTR"    ' Spinstand
    rsrc.instr(6) = "GPIB0::7::INSTR" ' Ocilloscope
    '
    ' Initialize and Get Handle ID
    '
    status = hpe5022_init(NumberOfInstrument, rsrc, StatusArray(0), hpe5022)
    '
    ' Error Handle
    '
    If status = 0 Then
        Call MsgBox("Initialize Succeed", vbOKOnly, "Message")
    Else
        Disp$ = ""
        For ModuleNo = 0 To 6
            If StatusArray(ModuleNo) <> 0 Then
              Disp$ = Disp$ + ModuleDescription(ModuleNo)
            End If
        Next ModuleNo
```

```
    Call MsgBox(Disp$ + "Initialize Failed ", vbOKOnly, "Error")
    End
End If
'
' Create Head Track Offset Data Array for Track Profile Measurement
'
For i = 0 To NumOfHeadOffsetData - 1
   HeadOffsetData(i) = HeadOffsetStart + _
   i * (HeadOffsetEnd - HeadOffsetStart) / (NumOfHeadOffsetData - 1)
Next i
'
' Create Write Current Data Array for Write Current Sweep Measurement
'
For i = 0 To NumOfWriteCurrentData - 1
   WriteCurrentData(i) = WriteCurrentStart + _
   i * (WriteCurrentEnd - WriteCurrentStart) / (NumOfWriteCurrentData - 1)
Next i
'
' Spindle RPM
'
ErrorCheck hpe5022_spindleSpeed(hpe5022, SpindleRPM)
'
' Set Drive Configuration
'(You can use the function of either "hpe5022_driveConfigRadiusSkew" or "hpe5022_driveConfigPivot".
' In this example, hpe5022_driveConfigPivot is used.)
'
ErrorCheck hpe5022_driveConfigPivot(hpe5022, PivotToCenter, PivotToGap)
'
' Set the HP Part Number of Cassette
'
ErrorCheck hpe5022_hgaCassette(hpe5022, CassetteID)
'
' Set HGA DImension
'
ErrorCheck hpe5022_hgaDimension(hpe5022, RefBoss, SliderGap, TipBoss, BottomBoss, Bossid)
'
' Set Inhibit Area Margin
'
ErrorCheck hpe5022_inhibitMargin(hpe5022, InhibitMargin)
'
' Track Format
'
ErrorCheck hpe5022_dataArea(hpe5022, IDRadius, NumOfTrack, TrackPitch)
'
' Head Road Radius and Initial Track Number
'
ErrorCheck hpe5022_headLoadConfig(hpe5022, hpe5022_LOAD_DYNAMIC, HeadLoadRadius, InitialTrackNo)
'
' Write, Sense, Erase Current, Erase Type
'
ErrorCheck hpe5022_writeCurrent(hpe5022, WriteCurrent)
ErrorCheck hpe5022_senseCurrent(hpe5022, SenseCurrent)
ErrorCheck hpe5022_eraseCurrent(hpe5022, EraseCurrent)
```

```
ErrorCheck hpe5022_eraseType(hpe5022, hpe5022_ERASE_DC_NEG)
'
' Filter Matrix Filter Select
'
ErrorCheck hpe5022_selectFilter(hpe5022, hpe5022_FILTER3)
'
' Channel Bit Rate
'
ErrorCheck hpe5022_channelBitRate(hpe5022, ChannelBitRate)
'
' Read Track Offset
'
ErrorCheck hpe5022_readTrackOffset(hpe5022, ReadTrackOffset)
'
' Write Track Offset
'
ErrorCheck hpe5022_writeTrackOffset(hpe5022, WriteTrackOffset)
'
' Test Data Pattern
'
ErrorCheck hpe5022_hfPattern(hpe5022, PeriodOfHFPattern)
ErrorCheck hpe5022_lfPattern(hpe5022, PeriodOfLFPattern)
ErrorCheck hpe5022_isolatedPulsePattern(hpe5022, PeriodOfIsolatedPattern)
ErrorCheck hpe5022_overwriteHfPattern(hpe5022, PeriodOfOWHFDataPattern)
ErrorCheck hpe5022_overwriteLfPattern(hpe5022, PeriodOfOWLFDataPattern)
'
' Measurement Mode is set at AUTO
'
ErrorCheck hpe5022_measurementMode(hpe5022, hpe5022_MEAS_AUTO)
'
' Parametric Measurement Setup
'
ErrorCheck hpe5022_pwThreshold(hpe5022, PwRatio, hpe5022_PW_TAA_TRACKED)
ErrorCheck hpe5022_setupParametric(hpe5022, ParametricNumOfAverage, IdParametric)
'
' Overwrite Measurement Setup
'
ErrorCheck hpe5022_setupOverwrite(hpe5022, OWNumOfAverage, IdOverwrite)
'
' Track Profile Measurement Setup
'
ErrorCheck hpe5022_setupTrackProfile(hpe5022, hpe5022_SEQ_ER_WR_M, hpe5022_TP_BOTH, hpe5022_MEAS_TAA, _
                                     NumOfHeadOffsetData, HeadOffsetData(0), IdTrackProfile)
'
' Write Current Sweep Setup
'
ErrorCheck hpe5022_setupWriteCurrentSweep(hpe5022, hpe5022_MEAS_TAA, NumOfWriteCurrentData, _
                                          WriteCurrentData(0), IdWriteCurrentSweep)
'
' Drive On (Moter On, Head Load)
'
ErrorCheck hpe5022_driveState(hpe5022, VI_TRUE)
'
```

```
    ' Track Number
    '
    ErrorCheck hpe5022_track(hpe5022, TrackNo)
    '
    ' AutoConfigration
    '
    ErrorCheck hpe5022_autoConfig(hpe5022, taa, pwLength)
    LblResult(0).Caption = "Auto Config Done" + vbCrLf + "TAA: " + Format(taa, "0.00E-##V") _
                        + " PW: " + Format(pwLength, "0.00E-##s")

End Sub
```

Description

After the “hpe5022_measure” function is executed, you should query for the corresponding measurement at once. Do not execute the “hpe5022_measure” functions continuously as shown below.

```
ErrorCheck hpe5022_measure(hpe5022, 1)
ErrorCheck hpe5022_measure(hpe5022, 2)
ErrorCheck hpe5022_parametric_Q(hpe5022, HFTAA, LFTAA, IsolatedTAA, pw, baseline, resolution)
ErrorCheck hpe5022_overwrite_Q(hpe5022, overwrite(i))
```

Step 7. Abort

The abort step is used to abort long measurements such as BER, Spectrum SNR, Measurement Spectrum and Popcorn Noise.

Example 2-30. Sample Program

```
Public HpE5022Abort As Boolean
Private Sub CmdAbort_Click()
    HpE5022Abort = True
End Sub


Private Sub CmdSepctralSnr_Click()
    Const startFreq As Double = 200000000# ' Start Frequency [Hz]
    Const stopFreq As Double = 400000000# ' Stop Frequency [Hz]
    Const ResBW As Double = 10000# ' Resolution Band Width Frequency [Hz]
    Const VideoBW As Double = 10000# ' Video Band Width Frequency [Hz]
    Dim Sig As Double, noise As Double, snr As Double, EndState As Long
    EndState = 1
    '
    ' Select Test Data Pattern
    '
    ErrorCheck hpe5022_selectPattern(HpE5022, hpe5022_PAT_HF)
    ErrorCheck hpe5022_spectralSnrFrequency(HpE5022, startFreq, stopFreq)
    ErrorCheck hpe5022_spectralSnrBandWidth(HpE5022, ResBW, VideoBW)
    '
    ' Set Execution Mode to accept wait
    '
    ErrorCheck hpe5022_executionMode(HpE5022, hpe5022_EXEC_WAIT_START)
    '
    ' Measurement
    '
    ErrorCheck hpe5022_measureSpectralSnr(HpE5022, hpe5022_SEQ_ER_WR_M)
    Do Until EndState = VI_SUCCESS
        '
        ' Progress the measurement sequence for 0.5 second on the E5022/E5023 measurement thread
        ' Then,if the function finished, "VI_SUCESS" is returned.
        '
        EndState = hpe5022_wai(HpE5022, 0.5)
        '
        ' Check if abort occur
        '
        HpE5022Abort = False
        DoEvents
        If HpE5022Abort = True Then
            ErrorCheck hpe5022_abort(HpE5022)
            LblResult.Caption = "Measurement aborted"
            '
            ' Set Execution Mode to default Setting
```

```
            '
            ErrorCheck hpe5022_executionMode(HpE5022, hpe5022_EXEC_WAIT_COMPLETE)
            Exit Sub
        End If
    Loop
    '
    ' Set Execution Mode to default Setting
    '
    ErrorCheck hpe5022_executionMode(HpE5022, hpe5022_EXEC_WAIT_COMPLETE)
    '
    ' Check if measurement finished properly
    '
    ErrorCheck hpe5022_lastMeasureState_Q(HpE5022, EndState)
    '
    If EndState <> VI_SUCCESS Then
        LblResult.Caption = "Measurement Error"
    Else
    '
    ' Query Measurement Result
    '
    ErrorCheck hpe5022_spectralSnr_Q(HpE5022, Sig, noise, snr)
    '
    ' Display Mesurement Result
    '
    LblResult.Caption = "Spectral SNR: HF " + vbCrLf _
                    + "Signal:  " + Format(Sig, "0.00E-##") + " V" + vbCrLf _
                    + "Noise: " + Format(noise, "0.00E-##") + "rmsV" + vbCrLf _
                    + "SNR:   " + Format(snr, "0.00E-##") + " V" + vbCrLf
    End If
End Sub
```

Description

A sample program of an abort function for “Spectral SNR” measurement is provided above. The execution mode is set at “hpe5022_EXEC_WAIT_ START” in the “hpe5022_executionMode” function in order to execute measurement of E5022/E5023 in another thread. As soon as the “hpe5022_measureSpectralSnr” function is executed. The thread is changed to Visual Basic. In other words, the Visual basic can execute its command. While the “hpe5022_wai” function is executed, thread is changed to measurement of E5022/E5023 and measurement is proceeded for the specified period. Then the “hpe5022_wai” function returns the status if the measurement is completed. After that the thread is returned to Visual Basic again.

DoEvent Function of Visual Basic pass the control to the operating system to detect if the other event occurs.

Functions used in this measurement

“hpe5022_executionMode” on page 97

“hpe5022_wai” on page 101

“hpe5022_abort” on page 103

“hpe5022_lastMeasureState_Q” on page 100

Step 6. Closing

The closing step turns off the drive and closes the system. This section briefly explains how to close the system using the sample program listed below.

Example 2-31. Sample Program

```
Private Sub CmdClose_Click()
    '
    ' Instrument Address Set
    '
    Const NumberOfInstrument As Integer = 6
    '
    Dim CloseStatusArray(NumberOfInstrument - 1) As Long
    '
    ErrorCheck hpe5022_close(HpE5022, CloseStatusArray(0))

End Sub
```

Description

The number of instruments to be closed should not be less than the number of initialized instrument specified in the Initializing Program.

3 Function Reference

About this chapter

This chapter provides information on functions of the “hpe5022_32.dll”. The syntax code of each function are provided in C language and Visual Basic. The type definition of C language is Agilent VISA Type Definition. See the Agilent VISA User’s Manual for definition.

In “Parameters” section of each function, the “Preset Value” is the default value of a parameter when initialization or reset function is executed. The “Values” is a code which contains a numeric number. The codes express the selection items if the parameter is selection type and it also express the minimum and the maximum values if the parameter is numeric number type.

In “Return Values” section, the completion code and the error code is provided. The completion code is a return value when the function is done successfully. The error code is a return value when the function fails to operate for some reason. The sample usage of return values is shown in a sample sub function named “ErrorCheck” in “Step 1. Initialization” on page 18.

System Open/Close/Reset System Function

This section describes the functions that are used to initialize, close, reset and abort the E5022A/B system.

hpe5022_init

C Syntax

ViStatus hpe5022_init(ViInt16 instrNum, ViRsrc rsrcArray[], ViStatus statArray[], ViPSession id);

Visual Basic Syntax

hpe5022_init(ByVal instrNum, ByRef rsrcArray As rsrcArrayType, ByRef statArry As long, ByRef id As Long) As long

Description

The function initializes the software connection to all instruments (module, box instrument and spinstand) and verifies that they are in the system. In addition, it places the instrument in the reset state. The setting of all functions is set to the value shown in the “Preset Value” of the parameters section.

Parameters

- instrNum

  Description — Specifies the number of the instruments (module, oscilloscope, spectrum analyzer, spinstand) incorporated in E5022A/B system.

  Direction — IN

- rsrcArray

  Description — Specify the resources of E5022A/B or E5023A system components. The system includes the E5035A/B, E5036A, E5037A/B/C/D, E5038A/B, E5040A (4395A) and spinstand are all mandatory components, while the oscilloscope, E5040A (4396B), E5039A/B/C bit error test module and E5041A dual counter module are optional components of the system.

  This is an data array and its size is the same as the number of instrNum. Order of components in this array is arbitrary.

  “rsrcArrayType” data type for Visual Basic is defined in the “hpe5022.bas” file as:

```
Public Type rsrcArrayType
instr(hpe5022_INSTR_NUM_MAX) As String
End Type
```

  The component description string should be the same as the resource symbolic name parameter of VISA viOpen() function of the VISA library. The Type data Type for Visual Basic is defined as follows: See the Agilent VISA User’s

Manual.

Direction IN

Values

Components	**Symbolic name of "resource" at factory default setting**
Agilent E5035A/B (Spinstand Interface Module)[a]	"VXI0::5::INSTR"
Agilent E5036A (Filter Matrix Module)	"VXI0::6::INSTR"
Agilent E5037A/B/C/D (Data Generator Module) [b]	"VXI0::7::INSTR"
Agilent E5038A/B (Parametric Measurement Module)[c]	"VXI0::8::INSTR"
Agilent E5039A/B/C (Bit Error Test Module)[d]	"VXI0::9::INSTR"
Agilent E5040A (Spectrum Measurement Module) [e]	"VXI0::10::INSTR"
Agilent E5041A (Popcorn Noise Tester)	"VXI0::11::INSTR"
Agilent 54845A (Oscilloscope)	"GPIB0::7::INSTR"
Agilent 4395A (Spectrum Analyzer) *[e]	"GPIB0::17::INSTR"
3GHz Spectrum Analyzer [g]	"GPIB0::18::INSTR"
Agilent 4396B (High Frequency Spectrum Analyzer) [f,g]	"GPIB0::18::INSTR"
Agilent E5010B/E5011A/E5010C/E5013A (Spindtand)	"ASRL1::INSTR"

a. E5035A and E5035B are exclusive.
b. E5037A, E5037B, E5037C, and E5037D are exclusive.
c. E5038A and E5038B are exclusive.
d. E5039A, E5039B, and E5039C are exclusive.
e. E5040A and 4395A are exclusive.
f. Where the GPIB address of 4396B is 18.
g. 3GHz Spectrum Analyzer and 4396B are exclusive.

NOTE The GPIB address besides 18 can be selectable for 4396B. In that case, change the xx in "GPIB0::xx::INSTR" at desired number. For example, if the 4396B GPIB address is 19, it will be "GPIB0::19::INSTR".

- statArray

  Description Returns the result of initialization for each component (instruments and spinstand). The returned results are actually error codes of each module/instrument.

  The results are returned in array type whose size is the same as the number of 'instNum". The order of contents in array is the same as 'rsrcArray'.

Direction OUT

- id

  Description Returns the system identifier. This value is used to identify the system session when using other E5022A/B functions.

  "VI_NULL" is returned when an error occurs during initialization.

  Direction OUT

Return Values

Completion Code	**Description**
VI_SUCCESS	No Error

Error Code	**Description**
hpe5022_ERROR_MEM_ALLOC	Can't allocate enough memory. Reboot your PC.
hpe5022_ERROR_ERROR_OPEN	Unable to open some instruments. Check the cable connections, VXI ULA and GPIB Address.
hpe5022_ERROR_FUNCTION_NOT_LOADED	Unable to load the DLL file. Perform the E5022A/B installation software or check if file is installed in the PC. The location of the DLL file is described in chapter 2.
hpe5022_ERROR_HARD_MISS	Hardware is missing. Check the system components.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.
hpe5022_ERROR_NO_OPTION	An option for oscilloscope is not installed. This function can only be used when the system has an oscilloscope.
hpe5022_ERROR_NSUP_HAMP	Installed head amplifier is not supported. Confirm the cable connection and if the head amplifier DLL is installed.
hpe5022_ERROR_NSUP_CHAN_IC	Installed channel IC board in the E5039A/B/C is not supported. Confirm the connection of the board and if the channel IC DLL is installed.

Error Code	Description
hpe5022_ERROR_HARD	Hardware Error is detected. Contact Agilent Technologies.
hpe5022_ERROR_REVISION	Some DLLs and/or module firmware revisions are in conflict. Reinstall the system software.

See Also “hpe5022_close” on page 95

hpe5022_close

C Syntax

ViStatus hpe5022_close(ViSession id, ViStatus statArray[]);

Visual Basic Syntax

hpe5022_close(ByVal id As Long, ByRef statArray As Long) As Long

Description

This function terminates the software connection to all components (module, box instrument and spinstand) and de-allocates system resources.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- statArray

  Description: Returns error code array whose size is defined by “instrNum” of hpe5022_init function.

  The order of components in this array corresponds to the ‘rsrcArray’ of the “hpe5022_init” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_CLOSE	Unable to close some components (module, box instrument and spinstand).

See Also

“hpe5022_init” on page 91

hpe5022_reset

C Syntax

ViStatus hpe5022_reset(ViSession id, ViStatus statArray[]);

Visual Basic Syntax

hpe5022_reset(ByVal id As Long, ByRef statArray As Long)As Long

Description

This function places E5022A/B at the default state. The setting for all functions are set to the values shown in the “Preset Value” in parameters section.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- statArray

  Description — Returns the error codes of each module whose size is defined by the (instr_num). Instr_num is a value which is passed to hpe5022_init() function as an instrNum parameter.

  The order of array codes in this array is the same as the order of components in the ‘rsrcArray’ parameter of the “hpe5022_init” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_RESET	Unable to reset on some components (modules, box instrument and spinstand).
hpe5022_ERROR_HARD_MISS	Hardware is missing. Check the system components.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.

See Also

“hpe5022_init” on page 91

Abort Function

This section describes the functions to abort a measurement.

hpe5022_executionMode

C Syntax

ViStatus hpe5022_executionMode(ViSession id, ViInt16 mode);

Visual Basic Syntax

hpe5022_executionMode(ByVal id As Long, ByVal mode As Integer) As Long

Description

This function specifies the type of test execution to be used when the sequence is setup. There are two ways of setting up this mode, one is by “hpe5022_EXEC_WAIT_START” and the other is by “hpe5022_WAIT_ COMPLETE” mode.

When the execution mode is set to “hpe5022_EXEC_WAIT_COMPLETE”, then the control of Visual Basic and VEE will not return from the time the measurement is executed until the measurement is complete. Control is a graphical object, such as a text box, a rectangle, or a command button, that you place on the development environment to display data, or perform an action.

In other hand, when the execution mode is set to “hpe5022_EXEC_WAIT_ START”, then measurement of E5022A/B will be executed in another thread. The control of Visual Basic and VEE will return at once after the time the measurement is executed, so that Visual Basic and VEE can execute its program step. When “hpe5022_wai” is executed, thread is changed then the measurement is proceeded for the specified period. After that, the “hpe5022_wai” function returns the status if the measurement is completed, then the thread is changed to Visual Basic and VEE. Therefore, When you want to abort a measurement, set it to “hpe5022_EXEC_WAIT_START”, then execute “hpe5022_wai” to monitor if the measurement is completed.

This function also affects both the “hpe5022_measure” and “hpe5022_executeSeq” functions.

See “Step 7. Abort” on page 85 for sample program.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Specifies the test execution mode.

  Preset Value: hpe5022_EXEC_WAIT_COMPLETE

Values

Name	Description
hpe5022_EXEC_WAIT_START	This mode executes the E5022 measurement in another thread, i.e, other functions can be executed along the VEE and VB threads.
hpe5022_EXEC_WAIT_COMPLETE	This mode waits for the measurement sequence to complete after its execution until its finished.

Direction IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'mode' is out of range.

See Also

"hpe5022_executionMode_Q" on page 99

"hpe5022_wai" on page 101

"hpe5022_measure" on page 387

"hpe5022_executeSeq"

hpe5022_executionMode_Q

C Syntax

ViStatus hpe5022_executionMode_Q(ViSession id, ViPInt16 mode);

Visual Basic Syntax

hpe5022_executionMode_Q(ByVal id As Long, ByRef mode As Integer) As Long

Description

This function returns the test execution mode of the measurement sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Returns the test execution mode specified by “hpe5022_executionMode”.

  Values: Same as ‘mode’ in the “hpe5022_executionMode” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_executionMode” on page 97

hpe5022_lastMeasureState_Q

C Syntax

ViStatus hpe5022_lastMeasureState_Q(ViSession id, ViPStatus stat);

Visual Basic Syntax

hpe5022_lastMeasureState_Q(ByVal id As Long, ByRef stat As Long) As Long

Description

This function queries the run-time error status of the measurement sequence. This function is used to check if the measurement is finished properly when the “hpe5022_executionMode” is set to “hpe5022_EXEC_WAIT_START”.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- stat

  Description — Returns the error status. If the last measurement sequence is successful then VI_SUCCESS is returned.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘stat’ is invalid.

See Also

“hpe5022_executionMode” on page 97

hpe5022_wai

C Syntax

ViStatus hpe5022_wai(ViSession id, ViReal64 timeout);

Visual Basic Syntax

hpe5022_wai(ByVal id As Long, ByVal timeout As Double) As Long

Description

This function for all modules to complete a sequence. If the execution mode in the “hpe5022_executionMode” function is set to “hpe5022_EXEC_WAIT_ START”, then measurement of E5022A/B will be executed in another thread. When this function is executed, the measurement is proceeded for the specified time.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- timeout

  Description: Specifies the timeout value.

  Unit: Second

  Direction: IN

  Values

Name	Value
hpe5022_TIMEOUT_MAX	4.0×10^{6}
hpe5022_TIMEOUT_INFINITE	-1
hpe5022_TIMEOUT_IMMEDIATE	0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘timeout’ is out of range.
hpe5022_VI_ERROR_TMO	The measurement sequence failed to complete within the specified time period.

See Also

“hpe5022_executionMode” on page 97

“hpe5022_abort” on page 103

hpe5022_abort

C Syntax

ViStatus hpe5022_abort(ViSession id);

Visual Basic Syntax

hpe5022_abort(ByVal id As Long) As Long

Description

This function causes all modules to abort the measurement sequence. The measurement sequence is aborted when the time specified by “hpe5022_wai” function has expired.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_executionMode” on page 97

“hpe5022_wai” on page 101

Self-test Function

This section describes the function of the self-test. The self-test allows you to check the internal test for some components.

hpe5022_self_test

C Syntax

ViStatus hpe5022_self_test(ViSession id, ViInt16 result[], ViPString msg[], ViStatus statArray[]);

Visual Basic Syntax

hpe5022_self_test(ByVal id As Long, ByRef result As Integer, ByRef msg As msgType, ByRef statArray As Long) As Long

Description

This function performs an internal self-test for all instruments (oscilloscope, spectrum analyzer, spinstand) except for the module of the E5022A/B. And returns the result of the self-test. The E5036A doesn't have an internal self-test function in its module, so it does not return any errors.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- result

  Description: Returns pass or fail results of self-test for each component (instruments and spinstand).

  The data is in array and its size is the same as the number of 'instNum' in the "hpe5022_init" function. The order of components in this array corresponds to the 'rsrcArray' of the "hpe5022_init" function.

  Direction: OUT

  Values

Value	Description
0	Test Passed
not 0	Test Failed

- msg

  Description: Returns self-test status message of each component (instrument and spinstand).

  "rmsgType" data type for Visual Basic is defined in the " hpe5022.bas" file as:

```
Public Type msgType
instr(hpe5022_INSTR_NUM_MAX) As String
End Type
```

The returned data is in array and its size and order is the same as ‘result’.

The message of each element of this array, is limited to (hpe5022_MESSASAGE_LENGTH_MAX) 512 characters.

Direction OUT

- statArray

  Description Returns the error number of self-test of each component (instruments and spinstand).

  The returned data is in array and its size and order is the same as ‘result’.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_SELFTEST	Unable to perform self-test on some components (modules, box instrument and spinstand)

See Also

“hpe5022_init” on page 91

Error Handle Function

This section describes the functions related with an error.

hpe5022_error_query

C Syntax

ViStatus hpe5022_error_query(ViSession id, ViPInt32 err_no, ViPString msg);

Visual Basic Syntax

hpe5022_error_query(ByVal id As Long, ByRef err_no As Long, ByVal msg As String) As Long

Description

This function returns the error number and corresponding error message.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- err_no

  Description — Returns the error code. When all errors are read from the queue, “VI_SUCCESS” is returned.

  Direction — OUT

- msg

  Description — Returns the error message. The length of the returned string is limited to (hpe5022_MESSASAGE_LENGTH_MAX) 512 characters. When all errors are read from the queue, null string,“ ”, is returned.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

hpe5022_error_message

C Syntax

ViStatus hpe5022_error_message(ViSession id, ViStatus err_no, ViPString msg);

Visual Basic Syntax

hpe5022_error_message(ByVal id As Long, ByVal err_no As Long, ByVal msg As String) As Long

Description

This function translates the error number to a user-readable string. The error number is returned as the return value from an E5022A/B driver function. For example:

stat = hpe5022_selectPattern(HpE5022, Pattern)

The function returns the error number and it's set to the 'stat' parameter.

hpe5022_error_message(Hpe5022, stat, ErrorMessage)

The user can get the error message from the error number.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- err_no

  Description: Specifies the error number.

  Direction: IN

- msg

  Description: Returns the error message corresponding to the specified error number. The length of the returned string is limited to 512 characters.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

See Also

hpe5022_set_error_callback

C Syntax

ViStatus hpe5022_set_error_callback(ViSession id, void (_VI_FUNCH *err_callback) (ViSession vi, ViStatus err_code, ViAddr user_data), ViAddr user_data);

Visual Basic Syntax

hpe5022_set_error_callback(ByVal id As Long, ByVal err_callback As Long, ByRef user_data As Any) As Long

Description

This function specifies the call back function. The call back function allows you to call a user define function when an error occurs. After the “hpe5022_set_error_callback” function is performed, the user define function installed in the function is performed when an error occurs.

For example: When performing the sub-function (user-defined function) named “errorHandler”, refer to the sample program shown below

The sub-function should be defined in advance. This example shows you the sub-function display, the error message and the error number.

```
Public Sub _
errorHandler(ByVal vi As Long, ByVal errCode As Long, ByRef userData As Long)
    Dim errorNumber As Long
    Dim errorMessage As String
    Dim stat As Long
    Dim strNum As Variant
    ’
    ’
    Call hpe5022_error_query(vi, errorNumber, errorMessage)
    ’
    ’  Trim C String
    ’    Search the end of error message string
    strNum = InStr(1, errorMessage, Chr(0))
    ’
    If strNum = Null Then errorMessage = ""
    ElseIf strNum < 2 Then errorMessage = ""
    ’    Cut only the errormessage
    Else errorMessage = Left(errorMessage, strNum - 1)
    ’    Display the error Message
    MsgBox (Str(errorNumber) & (errorMessage))
End Sub
```

The sub-function “errorHandler” is specified in the “hpe5022_set_error_callback” function.

```
Dim vi As Long
Dim user_data As Long
’ do some operation
Call hpe5022_set_error_callback(vi, AddressOf errorHandler, user_data)
```

After this function is performed, the “errorHandler” subroutine is performed whenever any error occurs.

If you use this example, the sample program of “error check” subroutine in chapter 2 is not necessary.

Only one call back function can be specified. If more than one call back function is to be specified, the previous call back function will be overwritten by the new one.

For Visual Basic, the syntax of user-defined function must be:

Public Sub func(ByVal id As Long, ByVal errCode As Long, ByRef userdata As Any)

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- err_callback

  Description — Specifies an address (a pointer for C language) of the sub function which you want to perform when an error occurs. If this value is set to “VI_NULL”, the specified sub function is removed.

  Direction — IN

  Preset Value — VI_NULL

- user_data

  Description — Specifies the variable (address of a variable for C language) that you want to pass to the sub function. This parameter is passed to the sub function as its third parameter.

  Direction — IN

Return Values

Completion Code	**Description**
VI_SUCCESS	No Error

Error Code	**Description**
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

System/Module Revision Query Function

This section describes the functions to query the system driver revision

hpe5022_revision_query

C Syntax

ViStatus hpe5022_revision_query(ViSession id, ViPString rev);

Visual Basic Syntax

hpe5022_revision_query(ByVal id As Long, ByVal rev As String) As Long

Description

This function returns the revision number of the function library of E5022A/B system.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rev

  Description: Returns the revision number of the function library. The length of the returned string is limited to 512 characters. The string example would be "A.01.00". This is the same as ‘rev’ of the “hpe5022_idn_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_idn_Q” on page 111

hpe5022_idn_Q

C Syntax

ViStatus hpe5022_idn_Q (ViSession id, ViPString rev, ViPString serial);

Visual Basic Syntax

hpe5022_idn_Q(ByVal id As Long, ByVal rev As String, ByVal serial As String) As Long

Description

This function returns the driver revision and the system serial number of E5022A/B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rev

  Description: Returns the driver revision of E5022A/B. The length of the returned string is limited to 512 characters. An example of the returned string would be "A.01.00". This is the same as ‘rev’ of the “hpe5022_revision_query” function.

  Direction: OUT

- serial

  Description: Returns the system serial number. The length of the returned string is limited to 512 characters.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_revision_query” on page 110

hpe5022_subIdn_Q

C Syntax

ViStatus hpe5022_subIdn_Q (ViSession id, ViPString instrName[], ViPString rev[], ViPString serial[]);

Visual Basic Syntax

hpe5022_subIdn_Q(ByVal id As Long, ByRef instrName As instrNameType, ByRef rev As revType, ByRef serial As serialType) As Long

Description

This function returns the name, firmware revision and the serial number of the Agilent E5022A/B components (module and box instrument). The null string (““) is returned for the spinstand. See “Step 1. Initialization” on page 18.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- instrName

  Description: Returns the component model name in array. The size of the data array is the same as the number of “instrNum” of the “hpe5022_init” function. The order of components in the array is the same as the order of components of “rsrcArray” parameter of the “hpe5022_init” function.

  “instrNameType” data type for Visual Basic is defined in the “hpe5022.bas” file as:

```
Public Type instrNameType
    instr(hpe5022_INSTR_NUM_MAX) As String
End Type
```

  The “hpe5022_INSTR_NUM_MAX” is also defined in that file.

  Direction: OUT

- rev

  Description: Returns the firmware revision of components. The size of data array is the same as the ‘instName’ parameter. The length of the returned string is limited to 512 characters. Some components also return the date of revision.

  Since E5036A and E5041A have no firmware in its module, A message “Not Available” is returned.

  “revType” data type for Visual Basic is defined in the “hpe5022.bas” file as:

```
Public Type revType
    instr(hpe5022_INSTR_NUM_MAX) As String
```

```
End Type
```

Direction OUT

- serial

  Description Returns the serial number of components.The size of this data array is the same as the ‘instName’ parameter. The length of the returned string is limited to 512 characters.

  “serialType” data type for Visual Basic is defined in the “hpe5022.bas” file as:

```
Public Type serialType
    instr(hpe5022_INSTR_NUM_MAX) As String
End Type
```

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_init” on page 91

hpe5022_subModuleIdn_Q

C Syntax

ViStatus hpe5022_subModuleIdn_Q(ViSession id, ViInt16 subModuleId, ViPString moduleName, ViPString serialNumber, ViPString description);

Visual Basic Syntax

hpe5022_subModuleIdn_Q(ByVal id As Long, ByVal subModuleId As Integer, ByVal moduleName As String, ByVal serialNumber As String, ByVal description As String) As Long

Description

This function returns the name, serial number and description of the sub modules.

When the hpe5022_SUBMOD_SSIF_xxxx is selected with the E5035A, this function returns “hpe5022_ERROR_NSUP_FUNC” error.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- subModuleId

  Description: Selects the sub module name to query.

  Direction: IN

  Values

Name	Val.	Description
hpe5022_SUBMOD_SSIF_DAUGHTER_BOARD	0	E5035B Fine Position Control (Option 001 or 002)
hpe5022_SUBMOD_SSIF_HEADAMP_CTRL_UNIT	1	Head Amplifier Control Unit (E5043A, E5043B, or E5043C)
hpe5022_SUBMOD_SSIF_ON_STAGE	2	On Stage Buffer Board (E5029K Option)
hpe5022_SUBMOD_SSIF_HEADAMP_BOARD	3	Head Amplifier Option Number (E5029K Option)
hpe5022_SUBMOD_FMAT_FILTER_0	4	E5036A Filter No. 0
hpe5022_SUBMOD_FMAT_FILTER_1	5	E5036A Filter No. 1
hpe5022_SUBMOD_FMAT_FILTER_2	6	E5036A Filter No. 2
hpe5022_SUBMOD_FMAT_FILTER_3	7	E5036A Filter No. 3

Name	Val.	Description
hpe5022_SUBMOD_BER_CHANNEL_IC	8	E5039A/B/C Channel IC Name

- moduleName

  Description Returns the name of the selected sub module.

  Direction OUT

- serialNumber

  Description Returns the serial number of the selected sub module

  Direction OUT

- description

  Description Returns the description of the selected sub module.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'subModuleId' is out of range.
hpe5022_ERROR_NSUP_FUNC	This installed module has no selected sub module.

See Also

Data Definition Function

This section describes the functions for data setup.

hpe5022_channelBitRate

C Syntax

ViStatus hpe5022_channelBitRate(ViSession id, ViReal64 bitRate);

Visual Basic Syntax

hpe5022_channelBitRate(ByVal id As Long, ByVal bitRate As Double) As Long

Description

This function specifies the channel bit rate value.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bitRate

  Description: Specifies the channel bit rate. This value is returned by “hpe5022_channelBitRateRange_Q” function.

  Direction: IN

  Preset Value: 100×10^6

  Unit: bps (bits per second)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘bitRate’ is out of range.

See Also

“hpe5022_channelBitRate_Q” on page 117

“hpe5022_channelBitRateRange_Q” on page 118

hpe5022_channelBitRate_Q

C Syntax

ViStatus hpe5022_channelBitRate_Q(ViSession id, ViPReal64 bitRate);

Visual Basic Syntax

hpe5022_channelBitRate_Q(ByVal id As Long, ByRef bitRate As Double) As Long

Description

This function returns the specified channel bit rate.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bitRate

  Description: Returns the specified channel bit rate.

  Direction: OUT

  Unit: bps (bit per second)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_channelBitRate” on page 116

hpe5022_channelBitRateRange_Q

C Syntax

ViStatus hpe5022_channelBitRateRange_Q(ViSession id, ViPReal64 bitRateMin, ViPReal64 bitRateMax);

Visual Basic Syntax

hpe5022_channelBitRateRange_Q(ByVal id As Long, ByRef bitRateMin As Double, ByRef bitRateMax As Double) As Long

Description

This function returns the range of the channel bit rate. The range is dependent on the system option. E5022A/B or E5023A detects the system option and sets the range automatically. Although the E5037A option 002 is 750 Mbps Data Generator, the maximum channel bit rate is 1 Gbps. Howevor, the bit rate over 750 Mbps is not guaranteed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bitRateMin

  Description: Returns the minimum value that the channel bit rate setting can be set into.

  Direction: OUT

  Unit: bps (bit per second)

- bitRateMax

  Description: Returns the maximum value that the channel bit rate setting can be set into.

  Direction: OUT

  Unit: bps (bit per second)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_channelBitRate” on page 116

hpe5022_selectPattern

C Syntax

ViStatus hpe5022_selectPattern(ViSession id, ViInt16 pat);

Visual Basic Syntax

hpe5022_selectPattern(ByVal id As Long, ByVal pat As Integer) As Long

Description

This function selects the data pattern to be written for write operation.Some measurements such as resolution measurement, selects the data pattern automatically. See each measurement function for details.

Each selected pattern has its own property. The property of each pattern is defined by functions, such as “hpe5022_hfPattern”, “hpe5022_lfPattern” (listed in “See also”).

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- pat

  Description Specifies the data pattern to be used for measurement.

  Direction IN

  Preset Value 2 (hpe5022_PAT_ISO)

  Values

Name	Value	Description
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_REP_2	8	Repetition Pattern2
hpe5022_PAT_REP_3	9	Repetition Pattern3
hpe5022_PAT_REP_4	10	Repetition Pattern4
hpe5022_PAT_USER	20	User-defined Pattern

Name	Value	Description
hpe5022_PAT_USER_2	21	User-defined Pattern2
hpe5022_PAT_USER_3	22	User-defined Pattern3
hpe5022_PAT_USER_4	23	User-defined Pattern4

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘pat’ is out of range.

See Also

“hpe5022_hfPattern” on page 122

“hpe5022_lfPattern” on page 124

“hpe5022_isolatedPulsePattern” on page 126

“hpe5022_prbsPattern” on page 128

“hpe5022_repetitivePattern” on page 131

“hpe5022_overwriteHfPattern” on page 136

“hpe5022_overwriteLfPattern” on page 138

“hpe5022_userPattern” on page 140

hpe5022_selectPattern_Q

C Syntax

ViStatus hpe5022_selectPattern_Q(ViSession id, ViInt16 pat);

Visual Basic Syntax

hpe5022_selectPattern_Q(ByVal id As Long, ByRef pat As Integer) As Long

Description

This function returns the selected data pattern.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- pat

  Description — Returns the selected data pattern.

  Direction — OUT

  Values — Same as the ‘pat’ in the “hpe5022_selectPattern” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_selectPattern” on page 119

hpe5022_hfPattern

C Syntax

ViStatus hpe5022_hfPattern(ViSession id, ViInt16 n);

Visual Basic Syntax

hpe5022_hfPattern(ByVal id As Long, ByVal n As Integer) As Long

Description

This function specifies the repetitive period of the HF pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Specifies the number of repetitive periods.

  Direction: IN

  Preset value: 2

  Values

Name	Value
hpe5022_HF_MIN	1
hpe5022_HF_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The number of repetitive period is out of range.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_hfPattern_Q” on page 123

hpe5022_hfPattern_Q

C Syntax

ViStatus hpe5022_hfPattern_Q(ViSession id, ViPInt16 n);

Visual Basic Syntax

hpe5022_hfPattern_Q(ByVal id As Long, ByRef n As Integer) As Long

Description

This function returns the specified number of repetitive periods of the HF pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Returns the number of repetitive periods.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

"hpe5022_hfPattern_Q" on page 123

hpe5022_lfPattern

C Syntax

ViStatus hpe5022_lfPattern(ViSession id, ViInt16 n);

Visual Basic Syntax

hpe5022_lfPattern(ByVal id As Long, ByVal n As Integer) As Long

Description

This function specifies the repetitive period of the LF pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Specifies the number of repetitive periods.

  Direction: IN

  Preset value: 6

  Values

Name	Value
hpe5022_LF_MIN	1
hpe5022_LF_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The number of repetitive period is out of range.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_lfPattern_Q” on page 125

hpe5022_lfPattern_Q

C Syntax

ViStatus hpe5022_lfPattern_Q(ViSession id, ViPInt16 n);

Visual Basic Syntax

hpe5022_lfPattern_Q(ByVal id As Long, ByRef n As Integer) As Long

Description

This function returns the specified number of repetitive periods of the LF pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Returns the specified number of repetitive periods.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_lfPattern” on page 124

hpe5022_isolatedPulsePattern

C Syntax

ViStatus hpe5022_isolatedPulsePattern(ViSession id, ViInt16 n);

Visual Basic Syntax

hpe5022_isolatedPulsePattern(ByVal id As Long, ByVal n As Integer) As Long

Description

This function specifies the number of repetitive periods of an isolated pulse pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Specifies the number of repetitive periods.

  Direction: IN

  Preset value: 20

  Values

Name	Value
hpe5022_ISO_MIN	1
hpe5022_ISO_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The data transitions period is out of range.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_isolatedPulsePattern_Q” on page 127

hpe5022_isolatedPulsePattern_Q

C Syntax

ViStatus hpe5022_isolatedPulsePattern_Q(ViSession id, ViPInt16 n);

Visual Basic Syntax

hpe5022_isolatedPulsePattern_Q(ByVal id As Long, ByRef n As Integer) As Long

Description

This function returns the specified number of repetitive period of an isolated pulse pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Returns the specified number of repetitive periods.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_isolatedPulsePattern” on page 126

hpe5022_prbsPattern

C Syntax

ViStatus hpe5022_prbsPattern(ViSession id, ViInt32 prbs);

Visual Basic Syntax

hpe5022_prbsPattern(ByVal id As Long, ByVal prbs As Long) As Long

Description

This function sets the pseudo random bit sequence pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- prbs

  Description: Specifies the pseudo random sequence generating polynomial. The LSB (bit 0) of this value represents 0th order coefficient of the generating polynomial, bit 1 represents 1st order, and so on. This function can generate up to 10th order pseudo random data. The initial bit is 100000.....

  For example: if you want to generate the random sequence of x^7+x^3+1, the value 137 ($=2^7+2^3+1$) should be set to this value.

  Direction: IN

  Preset value: 137 (x^7+x^3+1)

  Values

Name	Value
hpe5022_PRBS_MIN	3
hpe5022_PRBS_MAX	1023

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘prbs’ is out of range.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_prbsPattern_Q” on page 130

hpe5022_prbsPattern_Q

C Syntax

ViStatus hpe5022_prbsPattern_Q(ViSession id, ViPInt32 prbs);

Visual Basic Syntax

hpe5022_prbsPattern_Q(ByVal id As Long, ByRef prbs As Long) As Long

Description

This function returns the specified pseudo random generating polynomial.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- prbs

  Description: Returns the pseudo random sequence generating polynomial.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_prbsPattern” on page 128

hpe5022_repetitivePattern

C Syntax

ViStatus hpe5022_repetitivePattern(ViSession id,ViInt16 n);

Visual Basic Syntax

hpe5022_repetitivePattern(ByVal id As Long, ByVal n As Integer) As Long

Description

This function specifies the number of repetitive periods of the repetitive data pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Specifies the number of repetitive periods.

  Direction: IN

  Preset value: 10

  Values

Name	Value
hpe5022_REP_MIN	1
hpe5022_REP_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The number of repetitive period is out of range.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_repetitivePattern_Q” on page 132

hpe5022_repetitivePattern_Q

C Syntax

ViStatus hpe5022_repetitivePattern_Q(ViSession id,ViPInt16 n);

Visual Basic Syntax

hpe5022_repetitivePattern_Q(ByVal id As Long, ByRef n As Integer) As Long

Description

This function returns the specified number of repetitive periods of the repetitive data pattern.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- n

  Description — Returns the number of repetitive periods.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_repetitivePattern” on page 131

hpe5022_repetitivePatternEx

C Syntax

ViStatus hpe5022_repetitivePatternEx(ViSession id,ViInt16 dataPat, VInt16 n);

Visual Basic Syntax

hpe5022_repetitivePatternEx(ByVal id As Long, ByVal dataPat As Integer, ByVal n As Integer) As Long

Description

This function sets the number of repetitive period of the repetitive data pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataPat

  Description: Specifies the type of repetitive data pattern.

  Direction: IN

  Values

Name	Value
hpe5022_PAT_REP	4
hpe5022_PAT_REP_2	8
hpe5022_PAT_REP_3	9
hpe5022_PAT_REP_4	10

- n

  Description: Returns the number of repetitive periods of the repetitive pattern.

  Direction: IN

  Preset Value: 10

  Values

Name	Value
hpe5022_REP_MIN	1
hpe5022_REP_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_repetitivePatternEx_Q” on page 135

“hpe5022_repetitivePattern” on page 131

“hpe5022_selectPattern” on page 119

hpe5022_repetitivePatternEx_Q

C Syntax

ViStatus hpe5022_repetitivePatternEx_Q(ViSession id,ViInt16 dataPat, VPInt16 n);

Visual Basic Syntax

hpe5022_repetitivePatternEx_Q(ByVal id As Long, ByVal dataPat As Integer, ByRef n As Integer) As Long

Description

This function returns the specified number of repetitive periods of the repetitive data pattern.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataPat

  Description — Specifies the type of repetitive data pattern.

  Direction — IN

  Values

Name	Value
hpe5022_PAT_REP	4
hpe5022_PAT_REP_2	8
hpe5022_PAT_REP_3	9
hpe5022_PAT_REP_4	10

- n

  Description — Returns the number of repetitive periods of the repetitive pattern.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_repetitivePatternEx” on page 133

hpe5022_overwriteHfPattern

C Syntax

ViStatus hpe5022_overwriteHfPattern(ViSession id, ViInt16 n);

Visual Basic Syntax

hpe5022_overwriteHfPattern(ByVal id As Long, ByVal n As Integer) As Long

Description

This function specifies the overwrite HF Pattern. This pattern is used in the overwrite measurement even if this pattern is not selected by the “hpe5022_selectPattern” function.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- n

  Description Specifies the number of repetitive periods.

  Direction IN

  Preset value 3

  Values

Name	Value
hpe5022_HF_MIN	1
hpe5022_HF_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The number of repetitive period is out of range.

See Also

“hpe5022_overwriteHfPattern_Q” on page 137

“hpe5022_measureOverwrite” on page 910

“hpe5022_setupOverwrite” on page 913

“hpe5022_selectPattern” on page 119

hpe5022_overwriteHfPattern_Q

C Syntax

ViStatus hpe5022_overwriteHfPattern_Q(ViSession id, ViPInt16 n);

Visual Basic Syntax

hpe5022_overwriteHfPattern_Q(ByVal id As Long, ByRef n As Integer) As Long

Description

This function returns the number of repetitive period of the overwrite HF Pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Returns the specified number of repetitive periods.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

"hpe5022_overwriteHfPattern" on page 136

hpe5022_overwriteLfPattern

C Syntax

ViStatus hpe5022_overwriteLfPattern(ViSession id, ViInt16 n);

Visual Basic Syntax

hpe5022_overwriteLfPattern(ByVal id As Long, ByVal n As Integer) As Long

Description

This function sets the number of repetitive period of the overwrite LF Pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Specifies the number of repetitive periods.

  Direction: IN

  Preset value: 8

  Values

Name	Value
hpe5022_LF_MIN	1
hpe5022_LF_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The data transition’s repetitive period is out of range.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_overwriteLfPattern_Q” on page 139

“hpe5022_measureOverwrite” on page 910

“hpe5022_setupOverwrite” on page 913

hpe5022_overwriteLfPattern_Q

C Syntax

ViStatus hpe5022_overwriteLfPattern_Q(ViSession id, ViPInt16 n);

Visual Basic Syntax

hpe5022_overwriteLfPattern_Q(ByVal id As Long, ByRef n As Integer) As Long

Description

This function returns the number of repetitive periods of the overwrite LF Pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- n

  Description: Returns the number of repetitive periods.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

"hpe5022_overwriteLfPattern" on page 138

hpe5022_userPattern

C Syntax

ViStatus hpe5022_userPattern(ViSession id, ViInt16 dataForm, ViInt32 bitLength, const ViChar data[]);

Visual Basic Syntax

hpe5022_userPattern(ByVal id As Long, ByVal dataForm As Integer, ByVal bitLength As Long, ByVal data As String) As Long

Description

This function specifies the user-defined data pattern. The E5022A/B or E5023A allows you to generate your own pattern. The defined data pattern is repeated when it is used for the output data.

When you generate the data pattern of “A31C”, (“1010001100011101” in binary)

hpe5022_userPattern(hpE5022, hpe5022_USER_DATA_HEX, 16, “A31C”) or

hpe5022_userPattern(hpE5022, hpe5022_USER_DATA_BIN, 16, “1010001100011101”) can be used.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataForm

  Description Specifies the data format.

  Direction IN

  Values

Name	Value	Description
hpe5022_USER_DATA_HEX	0	Hexadecimal format (0-9,A-F or a-f)
hpe5022_USER_DATA_BIN	1	Binary format (0,1)

- bitLength

  Description Specifies the bit length of the user data.

  Direction IN

  Preset value 2

  Values

Name	Value
hpe5022_PAT_LENGTH_MIN	1
hpe5022_PAT_LENGTH_MAX	32,768

- data

  Description Specifies the user data in binary or hexadecimal format.

  Direction IN

  Preset value 11 (binary format)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataform’ and/or ‘bitLength’ is out of range.
hpe5022_ERROR_INV_DATA_SIZE	The length of specified data pattern is shorter than the specified user data bit length.
hpe5022_ERROR_INV_DATA_TYPE	The specified data pattern includes an invalid character. (either binary nor hexadecimal)

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_userPattern_Q” on page 142

hpe5022_userPattern_Q

C Syntax

ViStatus hpe5022_userPattern_Q(ViSession id,ViInt16 dataForm,ViPInt32 bitLength, ViChar data[]);

Visual Basic Syntax

hpe5022_userPattern_Q(ByVal id As Long, ByVal dataForm As Integer, ByRef bitLength As Long, ByVal data As String) As Long

Description

This function returns the specified user-defined data pattern.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataForm

  Description — Specifies the format of data to be returned.

  Direction — IN

  Values — Same as ‘dataForm’ in the “hpe5022_userPattern” function.

- bitLength

  Description — Returns the specified bit length of the user data.

  Direction — OUT

- data

  Description — Returns the specified user data pattern.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataform’ is out of range.

See Also

“hpe5022_userPattern” on page 140

hpe5022_userPatternEx

C Syntax

ViStatus hpe5022_userPatternEx(ViSession id, ViInt16 dataPat, ViInt16 dataForm, ViInt32 bitLength, const ViChar data[]);

Visual Basic Syntax

hpe5022_userPatternEx(ByVal id As Long, ByVal dataPat As Integer, ByVal dataForm As Integer, ByVal bitLength As Long, ByVal data As String) As Long

Description

This function specifies the user-defined data pattern. The E5022A/B or E5023A allows you to generate your own data pattern. The defined data pattern is repeated when it is used as the output data.

When you generate the data pattern of “A31C”, (“1010001100011101” in binary)

hpe5022_userPattern(hpE5022, hpe5022_USER_DATA_HEX, 16, “A31C”) or

hpe5022_userPattern(hpE5022, hpe5022_USER_DATA_BIN, 16, “1010001100011101”) can be used.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataPat

  Description Specifies the user data pattern.

  Direction IN

  Values

Name	Value
hpe5022_PAT_USER	20
hpe5022_PAT_USER_2	21
hpe5022_PAT_USER_3	22
hpe5022_PAT_USER_4	23

- dataForm

  Description Specifies the data format.

  Direction IN

  Values

Name	Value	Description
hpe5022_USER_DATA_HEX	0	Hexadecimal format (0-9,A-F or a-f)
hpe5022_USER_DATA_BIN	1	Binary format (0,1)

- bitLength

  Description Specifies the bit length of the user data.

  Direction IN

  Preset value 2

  Values

Name	Value
hpe5022_PAT_LENGTH_MIN	1
hpe5022_PAT_LENGTH_MAX	32,768

- data

  Description Specifies the user data in binary or hexadecimal format.

  Direction IN

  Preset value 11 (binary format)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataform’ and/or ‘bitLength’ is out of range.
hpe5022_ERROR_INV_DATA_SIZE	The length of specified data pattern is shorter than the specified user data bit length.
hpe5022_ERROR_INV_DATA_TYPE	The specified data pattern includes an invalid character. (either binary nor hexadecimal)

See Also

“hpe5022_userPatternEx_Q” on page 145

“hpe5022_userPattern” on page 140

“hpe5022_selectPattern” on page 119

hpe5022_userPatternEx_Q

C Syntax

ViStatus hpe5022_userPatternEx_Q(ViSession id, ViInt16 dataPat, ViInt16 dataForm, ViPInt32 bitLength, ViChar data[]);

Visual Basic Syntax

hpe5022_userPatternEx_Q(ByVal id As Long, ByVal dataPat As Integer, ByVal dataForm As Integer, ByRef bitLength As Long, ByVal data As String) As Long

Description

This function returns the user-defined data pattern. The E5022A/B or E5023A allows you to generate your own data pattern. The defined data pattern is repeated when it is used as the output data.

When you generate the data pattern of “A31C” (“1010001100011101” in binary), hpe5022_userPatternEx_Q (hpE5022, hpe5022_USER_DATA_HEX, bitLength, data) returns bitLength = 16 and data = “A31C.”

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataPat

  Description Specifies the user data pattern.

  Direction IN

  Values

Name	Value
hpe5022_PAT_USER	20
hpe5022_PAT_USER_2	21
hpe5022_PAT_USER_3	22
hpe5022_PAT_USER_4	23

- dataForm

  Description Specifies the data format.

  Direction IN

  Values

Name	Value	Description
hpe5022_USER_DATA_HEX	0	Hexadecimal format (0-9,A-F or a-f)
hpe5022_USER_DATA_BIN	1	Binary format (0,1)

- bitLength

Description Returns the bit length of the user data.

Direction OUT

Preset value 2

Values

Name	Value
hpe5022_PAT_LENGTH_MIN	1
hpe5022_PAT_LENGTH_MAX	32,768

- data

  Description Returns the user data in binary or hexadecimal format.

  Direction OUT

  Preset value 11 (binary format)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataform’ and/or ‘bitLength’ is out of range.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_userPatternEx” on page 143

Filter Setup Function

This section describes the functions for the filter matrix module setup. The filter matrix module has up to four filters and they are replaceable.

hpe5022_filterState_Q

C Syntax

ViStatus hpe5022_filterState_Q(ViSession id, ViInt16 filter, ViPBoolean state);

Visual Basic Syntax

hpe5022_filterState_Q(ByVal id As Long, ByVal filter As Integer, ByRef state As Integer) As Long

Description

This function returns the status of the selected filter whether it is installed or not.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- filter

  Description: Specifies the filter to be checked.

  Direction: IN

  Values

Name	Value	Description
hpe5022_FILTER0	0	Filter No.0
hpe5022_FILTER1	1	Filter No.1
hpe5022_FILTER2	2	Filter No.2
hpe5022_FILTER3	3	Filter No.3

- state

  Description: Returns the status of the specified filter.

  Direction: OUT

  Values

Name	Value	Description
VI_TRUE	1	Filter is installed
VI_FALSE	0	Filter is not installed

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘filter’ is out of range.

See Also

hpe5022_selectFilter

C Syntax

ViStatus hpe5022_selectFilter(ViSession id,ViInt16 filter);

Visual Basic Syntax

hpe5022_selectFilter(ByVal id As Long, ByVal filter As Integer) As Long

Description

This function selects the filter to be used in the filter matrix module.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- filter

  Description: Selects the filter to be used.

  Direction: IN

  Values

Name	Value	Description
hpe5022_FILTER0	0	Filter No.0
hpe5022_FILTER1	1	Filter No.1
hpe5022_FILTER2	2	Filter No.2
hpe5022_FILTER3	3	Filter No.3

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’filter’ is out of range.
hpe5022_ERROR_HARD_MISS	The selected filter is not installed. Check if the filter is installed.

See Also

“hpe5022_selectFilter_Q” on page 150

hpe5022_selectFilter_Q

C Syntax

ViStatus hpe5022_selectFilter_Q(ViSession id,ViPInt16 filter);

Visual Basic Syntax

hpe5022_selectFilter_Q(ByVal id As Long, ByRef filter As Integer) As Long

Description

This function returns the selected filter.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- filter

  Description: Returns the selected filter.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_FILTER0	0	Filter No.0
hpe5022_FILTER1	1	Filter No.1
hpe5022_FILTER2	2	Filter No.2
hpe5022_FILTER3	3	Filter No.3

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_selectFilter” on page 149

hpe5022_filterInformation_Q

C Syntax

ViStatus hpe5022_filterInformation_Q(ViSession id, ViInt16 filter, ViPInt16 filterType, ViPInt16 rollOff, ViPInt16 order, ViPReal64 fc, ViPReal64 dcGain, ViPString part_num, ViPString desc);

Visual Basic Syntax

hpe5022_filterInformation_Q(ByVal id As Long, ByVal filter As Integer, ByRef filterType As Integer, ByRef rollOff As Integer, ByRef order As Integer, ByRef fc As Double, ByRef dcGain As Double, ByVal part_num As String, ByVal desc As String) As Long

Description

This function returns the information of the selected filter from the filter matrix module.

NOTE

The parameters ‘filterType’, ‘rollOff’, ‘order’, ‘fc’ and ‘dcGain’ returns 0 currently. These parameters are for future use.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- filter

  Description: Specifies the filter from which to get the information.

  Direction: IN

  Values

Name	Value	Description
hpe5022_FILTER0	0	Filter No.0
hpe5022_FILTER1	1	Filter No.1
hpe5022_FILTER2	2	Filter No.2
hpe5022_FILTER3	3	Filter No.3

- filterType

  Description: Returns the type of the selected filter.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_FILT_TYPE_LPAS	1	Low pass filter
hpe5022_FILT_TYPE_HPAS	2	High pass filter

Name	Value	Description
hpe5022_FILT_TYPE_BPAS	3	Band pass filter

- rollOff

  Description Returns the roll-off characteristics of the filter (i.e, Bessel, Butterworth, Chebyshev)

  Direction OUT

  Values

Name	Value	Description
hpe5022_FILT_ROFF_NA	0	Not Applicable
hpe5022_FILT_ROFF_BESS	1	Bessel
hpe5022_FILT_ROFF_BUTT	2	Butterworth
hpe5022_FILT_ROFF_CHEB	3	Chebyshev

- order

  Description Returns the filter order.

  Direction OUT

- fc

  Description Returns the cut-off frequency of the selected filter.

  Direction OUT

  Unit Hz

- dcGain

  Description Returns the DC gain of the selected filter.

  Direction OUT

  Unit dB

- part_num

  Description Returns the Agilent part number of the selected filter. The length of this string is defined as “hpe5022_PART_NUM_LENGTH”.

  Direction OUT

- desc

  Description Returns the filter’s description. The length of this string is limited to “hpe5022_MESSAGE_LENGTH_MAX”.

Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘filter’ is out of range.
hpe5022_ERROR_HARD_MISS	The filter is not installed or not properly installed. Check the hardware.

See Also

Drive Setup Function

This section describes the functions for drive setup. The drive setup functions specifies the head dimension, head loading/unloading position, the data area on the media and the spindle speed. These functions should be specified before the drive is turned on.

hpe5022_driveConfigPivot

C Syntax

ViStatus hpe5022_driveConfigPivot(ViSession id, ViReal64 pc, ViReal64 pg);

Visual Basic Syntax

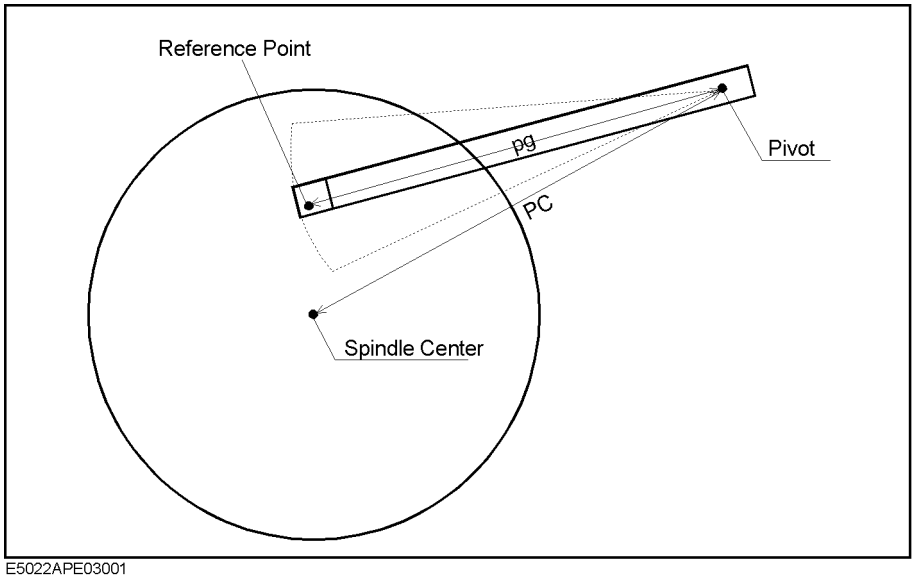
hpe5022_driveConfigPivot(ByVal id As Long, ByVal pc As Double, ByVal pg As Double) As Long

Description

This function specifies the parameters to calculate the head position and head skew angle. Although the spinstand adopts an X-Y stage, the E5022A/B simulates the real drive by changing the head position and the head skew angle depending on the specified track number (See the “hpe5022_track” function). Each parameter is shown in Figure 3-1.

E5022A/B provides two ways to configure the drive. One is by using this function and another is by using the “hpe5022_driveConfigRadiusSkew” function which specifies the head radius and its head skew angle for two locations.

Figure 3-1 Drive Configuration Parameters (Pivot, Spindle Center, Reference Point)



Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

Direction IN

- pc

  Description Specifies the length between the pivot and the center of the spindle.

  Direction IN

  Preset Value 100×10^{-3}

  Unit Meter

- pg

  Description Specifies the length between the pivot and the reference point on the HGA. See “hpe5022_hgaDimension” functions for the reference point.

  Direction IN

  Preset Value 100×10^{-3}

  Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has already been turned on. Turn it off before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_driveConfigPivot_Q” on page 156

“hpe5022_driveConfigRadiusSkew” on page 157

“hpe5022_hgaDimension” on page 161

“hpe5022_driveState” on page 209

“hpe5022_track” on page 229

hpe5022_driveConfigPivot_Q

C Syntax

ViStatus hpe5022_driveConfigPivot_Q(ViSession id, ViPReal64 pc, ViPReal64 pg);

Visual Basic Syntax

hpe5022_driveConfigPivot_Q(ByVal id As Long, ByRef pc As Double, ByRef pg As Double) As Long

Description

This function returns the specified parameters shown in Figure 3-1 on page 154. Even if the drive configuration is specified by the “hpe5022_driveConfigRadiusSkew” function, this function returns the calculated parameters.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pc

  Description: Returns the length between the pivot and the center of spindle.

  Direction: OUT

  Unit: Meter

- pg

  Description: Returns the length between the pivot and the reference point on the HGA.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_driveConfigPivot” on page 154

“hpe5022_driveConfigRadiusSkew” on page 157

hpe5022_driveConfigRadiusSkew

C Syntax

hpe5022_driveConfigRadiusSkew(ViSession id,ViReal64 radius1,ViReal64 radius2,ViReal64 skew1,ViReal64 skew2);

Visual Basic Syntax

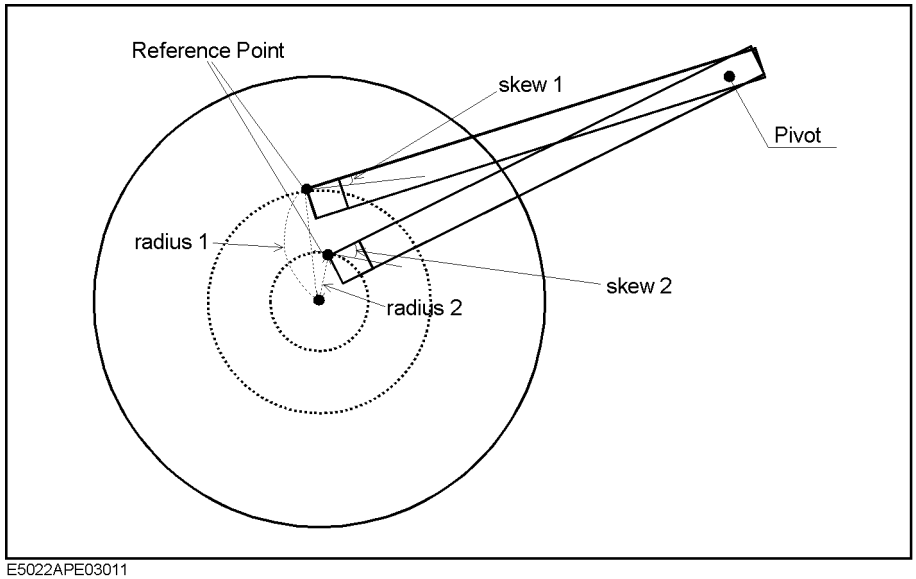
hpe5022_driveConfigRadiusSkew(ByVal id As Long, ByVal radius1 As Double, ByVal radius2 As Double, ByVal skew1 As Double, ByVal skew2 As Double) As Long

Description

This function specifies the parameters to calculate the head position and head skew angle. Although the spinstand adopts an X-Y stage, the E5022A/B simulates the real drive by changing the head position and the head skew angle depending on the specified track number (See the “hpe5022_track” function). Each parameter is shown in Figure 3-2. The position 1 and 2 are arbitrary location on the disk.

E5022A/B provides two ways to configure the drive. One is by using this function and another is by using the “hpe5022_driveConfigPivot” function which specifies the dimensions of spindle center and pivot.

Figure 3-2 **Drive Configuration Parameters (Radius, skew)**



E5022APE03011

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- radius1

  Description: Specifies the radius for position 1

  Direction: IN

  Unit: Meter

Preset Values 21×10^{-3}

- radius2

  Description Specifies the radius for position 2

  Direction IN

  Unit Meter

  Preset Values 23×10^{-3}

- skew1

  Description Specifies the skew angle for position 1

  Direction IN

  Unit Degree

- skew2

  Description Specifies the skew angle for position 2

  Direction IN

  Unit Degree

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has already been turned on. Turn it off before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_driveConfigPivot” on page 154

“hpe5022_driveConfigRadiusSkew_Q” on page 159

“hpe5022_driveState” on page 209

“hpe5022_track” on page 229

hpe5022_driveConfigRadiusSkew_Q

C Syntax

hpe5022_driveConfigRadiusSkew_Q(ViSession id,ViPReal64 radius1,ViPReal64 radius2,ViPReal64 skew1,ViPReal64 skew2);

Visual Basic Syntax

hpe5022_driveConfigRadiusSkew_Q(ByVal id As Long, ByVal radius1 As Double, ByVal radius2 As Double, ByRef skew1 As Double, ByRef skew2 As Double) As Long

Description

This function returns the specified parameters shown in Figure 3-2 on page 157. Even if the drive configuration is specified by the “hpe5022_driveConfigPivot” function, this function returns the calculated parameters.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- radius1

  Description: Specifies the radius for position 1.

  Direction: OUT

  Unit: Meter

- radius2

  Description: Specifies the radius for position 2.

  Direction: OUT

  Unit: Meter

- skew1

  Description: Returns the skew angle for position 1.

  Direction: OUT

  Unit: Degree

- skew2

  Description: Returns the skew angle for position 2.

  Direction: OUT

  Unit: Degree

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also “hpe5022_driveConfigRadiusSkew” on page 157

hpe5022_hgaDimension

C Syntax

ViStatus hpe5022_hgaDimension(ViSession id, ViReal64 td, ViReal64 sg, ViReal64 tip, ViReal64 bot, ViReal64 boss_id);

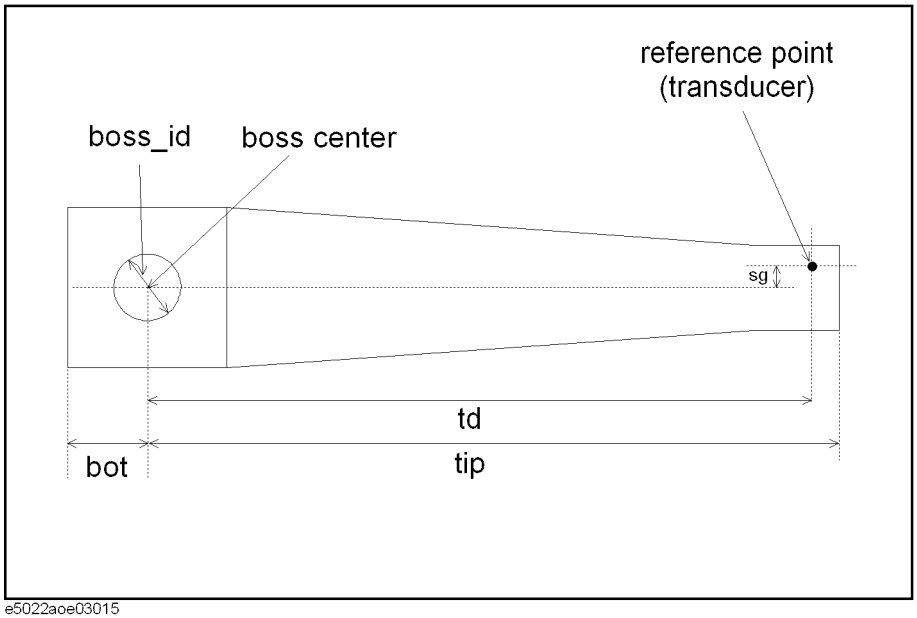
Visual Basic Syntax

hpe5022_hgaDimension(ByVal id As Long, ByVal td As Double, ByVal sg As Double, ByVal tip As Double, ByVal bot As Double, ByVal boss_id As Double) As Long

Description

This function specifies the dimension of the HGA (Head Gimbal Assembly). The parameters are shown in Figure 3-3. See Chapter 4 of the Operation Manual.

Figure 3-3 **HGA Dimension**



e5022aoe03015

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- td

  Description: Specifies the perpendicular distance between the reference point and the boss center. The reference point is defined as a point at the center of the track when the write track offset and read track offset are zero. Generally, the location of write head transducer is used as the reference point.

  Direction: IN

  Unit: Meter

  Preset Value: hpe5022_HGA_TRAN_DIST_MIN (0.1)

Values

Name	Value
hpe5022_HGA_TRAN_DIST_MIN	0
hpe5022_HGA_TRAN_DIST_MAX	0.1

- sg

  Description: Specifies the perpendicular distance between the boss center and the reference point.

  Direction: IN

  Unit: Meter

  Preset Value: 1×10^{-3}

  Values

Name	Value
hpe5022_HGA_SLID_GAP_MIN	0
hpe5022_HGA_SLID_GAP_MAX	10×10^{-3}

- tip

  Description: Specifies the distance between the tip of the HGA and the boss center.

  Direction: IN

  Unit: Meter

  Preset Value: hpe5022_HGA_TIP_LENG_MAX (0.1)

  Values

Name	Value
hpe5022_HGA_TIP_LENG_MIN	0
hpe5022_HGA_TIP_LENG_MAX	0.1

- bot

  Description: Specifies the distance between the base of the HGA and the boss center.

  Direction: IN

  Unit: Meter

  Preset Value: hpe5022_HGA_BOT_LENG_MAX (0.1)

Values

Name	Value
hpe5022_HGA_BOT_LENG_MIN	0
hpe5022_HGA_BOT_LENG_MAX	0.1

- boss_id

  Description Specifies the inner diameter of the boss.

  Direction IN

  Unit Meter

  Preset Value hpe5022_HGA_BOSS_ID_MAX (10×10^{-3})

  Values

Name	Value
hpe5022_HGA_BOSS_ID_MIN	1×10^{-3}
hpe5022_HGA_BOSS_ID_MAX	10×10^{-3}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘td’, ‘sg’, ‘tip’, ‘bot’, or/and ‘boss_id’ is out of range

See Also

“hpe5022_hgaDimension_Q” on page 164

hpe5022_hgaDimension_Q

C Syntax

ViStatus hpe5022_hgaDimension_Q(ViSession id, ViPReal64 td, ViPReal64 sg, ViPReal64 tip, ViPReal64 bot, ViPReal64 boss_id);

Visual Basic Syntax

hpe5022_hgaDimension_Q(ByVal id As Long, ByRef td As Double, ByRef sg As Double, ByRef tip As Double, ByRef bot As Double, ByRef boss_id As Double) As Long

Description

This function returns the dimension of the HGA (Head Gimbal Assembly).

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- td

  Description: Returns the perpendicular distance between the reference point and boss center.

  Direction: IN

  Unit: Meter

  Values: Same as the ‘td’ parameter in the “hpe5022_hgaDimension” function.

- sg

  Description: Returns the perpendicular distance between boss center and the reference point.

  Direction: OUT

  Unit: Meter

  Values: Same as the ‘sg’ parameter in the “hpe5022_hgaDimension” function.

- tip

  Description: Returns the distance between the tip of the HGA and boss center.

  Direction: OUT

  Unit: Meter

  Values: Same as the ‘tip’ parameter in the “hpe5022_hgaDimension” function.

- bot

  Description — Returns the distance between the base of the HGA and boss center.

  Direction — OUT

  Unit — Meter

  Values — Same as the ‘bot’ parameter in the “hpe5022_hgaDimension” function.

- boss_id

  Description — Returns the boss inner diameter.

  Direction — OUT

  Unit — Meter

  Values — Same as the ‘boss_id’ parameter in the “hpe5022_hgaDimension” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

See Also

“hpe5022_hgaDimension” on page 161

hpe5022_inhibitMargin

C Syntax

ViStatus hpe5022_inhibitMargin(ViSession id, ViReal64 margin);

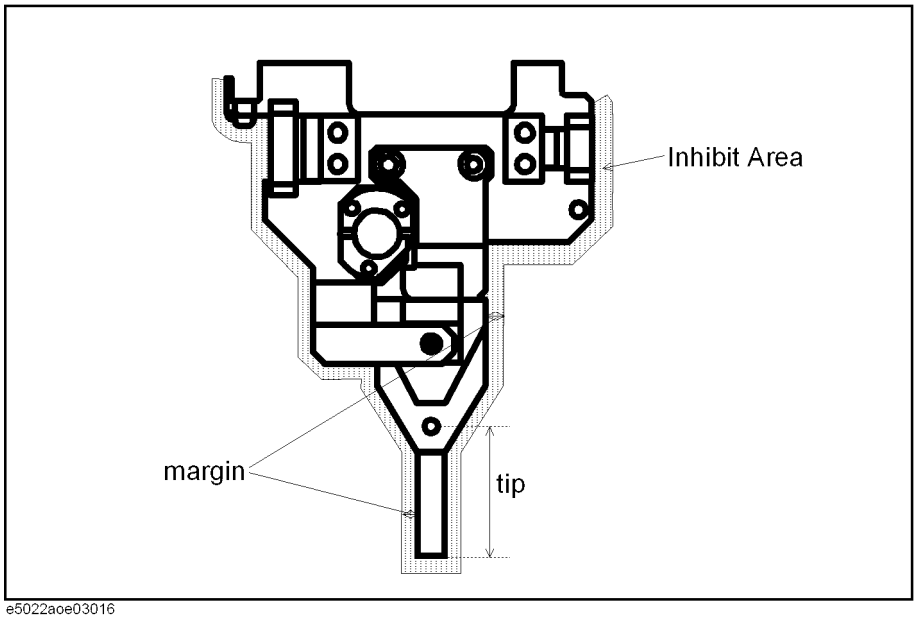
Visual Basic Syntax

hpe5022_inhibitMargin(ByVal id As Long, ByVal margin As Double) As Long

Description

This function specifies the margin of inhibit area. An inhibit area or marginal distance has to be established to keep the head and cassette from spindle contact. See Chapter 4 of the Operation Manual.

Figure 3-4 **Inhibit Area.**



e5022aoe03016

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- margin

  Description: Specifies the inhibit margin.

  Direction: IN

  Unit: Meter

  Preset Value: 1×10^{-3}

Values

Name	Value
hpe5022_INH_MARG_MIN	0
hpe5022_INH_MARG_MAX	10×10^{-3}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘margin’ is out of range

See Also “hpe5022_hgaDimension_Q” on page 164

hpe5022_inhibitMargin_Q

C Syntax

ViStatus hpe5022_inhibitMargin_Q(ViSession id, ViPReal64 margin);

Visual Basic Syntax

hpe5022_inhibitMargin_Q(ByVal id As Long, ByRef margin As Double) As Long

Description

This function returns the inhibit margin.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- margin

  Description: Returns the inhibit margin.

  Direction: OUT

  Unit: Meter

  Values: Same as the ‘margin’ parameter in the “hpe5022_inhibitMargin” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

See Also

“hpe5022_hgaDimension_Q” on page 164

hpe5022_hgaCassette

C Syntax

ViStatus hpe5022_hgaCassette(ViSession id, ViString part_num);

Visual Basic Syntax

hpe5022_hgaCassette(ByVal id As Long, ByVal part_num As String) As Long

Description

This function selects the HGA cassette type. E5022A/B allows you to use several types of HGA cassette. Each type of HGA cassette is designed for FIC, CAPS, TSA and paddle suspension.

This function is used to let the E5022A/B know the dimension of the HGA cassette, in order to avoid contact with the spindle.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- part_num

  Description: Specifies Agilent part number of the HGA cassette to be used. The Agilent part number is labelled on top of the HGA cassette. A sample string would be “E5010-61121”. When the customer-designed cassette is used then “GENERIC1” must be specified. The character length of this parameter is defined by hpe5022_PART_NUM_LENGTH.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_NSUP_HGA_CASSETTE	Specified HGA cassette type is not supported. Check the specified Agilent part number. If it is correct, the DLL for the cassette is not installed. Contact the Agilent Technologies office.

See Also

“hpe5022_hgaCassette_Q” on page 170

hpe5022_hgaCassette_Q

C Syntax

ViStatus hpe5022_hgaCassette_Q(ViSession id, ViPString part_num);

Visual Basic Syntax

hpe5022_hgaCassette_Q(ByVal id As Long, ByVal part_num As String) As Long

Description

This function returns the type of HGA cassette.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- part_num

  Description: Returns the Agilent part number of the HGA cassette to be used. If the part number has not been defined, a null string “” will be reported. The character length of this parameter is defined by hpe5022_PART_NUM_LENGTH.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_NDEF_HGA	The HGA cassette type is undefined.

See Also

“hpe5022_hgaCassette” on page 169

hpe5022_hgaCassetteDimension

C Syntax

ViStatus hpe5022_hgaCassetteDimension(ViSession id, ViReal64 boss_cent_x, ViReal64 boss_cent_y, ViInt16 points, const ViReal64 outline_x[], const ViReal64 outline_y[]);

Visual Basic Syntax

hpe5022_hgaCassetteDimension(ByVal id As Long, ByVal boss_cent_x As Double, ByVal boss_cent_y As Double, ByVal points As Integer, ByRef outline_x As Double, ByRef outline_y As Double) As Long

Description

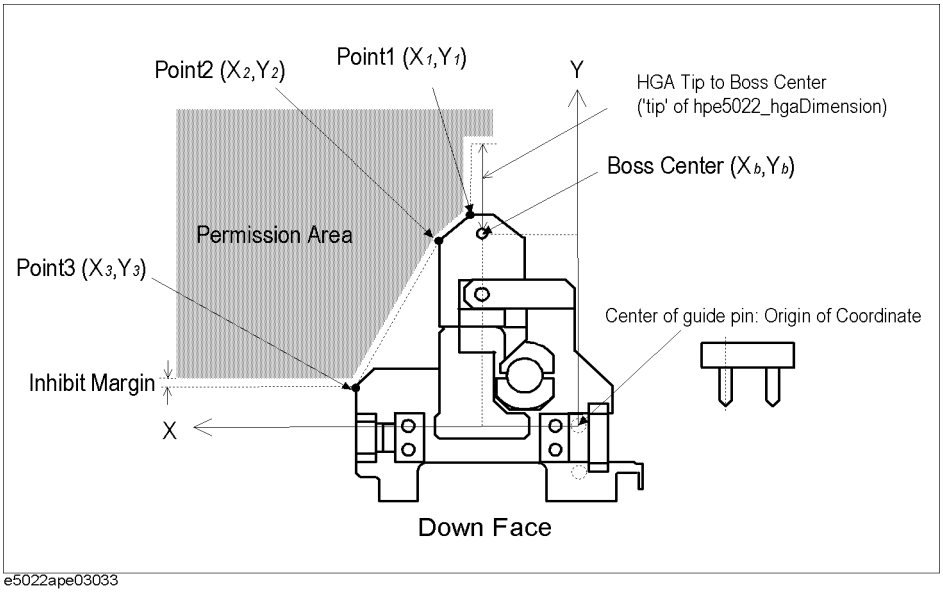
This function allows you to use a customer designed HGA cassette. When using the customer designed HGA cassette, you must let the system know the dimensions of the cassette you are using in order to avoid contacts with the spindle and shroud. The location of the HGA boss center and several points outside of the down face cassette must be specified. Since up face is defined as the line asymmetry of the down face it is impossible to define the down face and up face separately.

The location of the center of right side guide pin is defined as the origin of the coordinates. When a cassette has two guide pins in one side, the head pin becomes the origin.

The x and y coordinates of HGA boss center is defined as the position. In the x and y coordinates, two or three location points must be defined to let the system know the shape of the cassette. The user can specify some points outside of the cassette. At least two points must be specified.

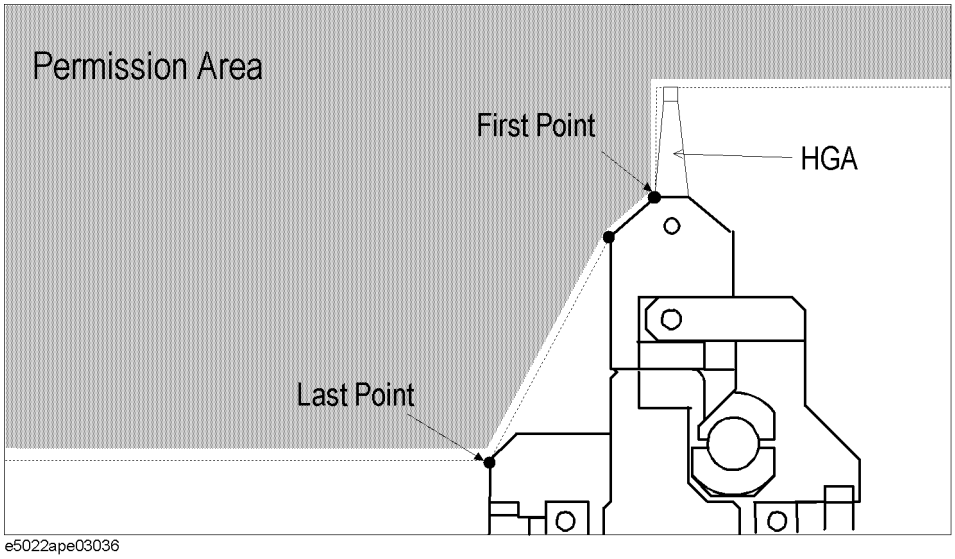
The outer area covered by the dotted line from the specified point is defined as the permission area where the spindle and shroud are permitted to exist. On the other hand, the inner area is defined as the inhibit area where the spindle is not permitted to exists.

Figure 3-5 Coordinate of Cassette Dimension



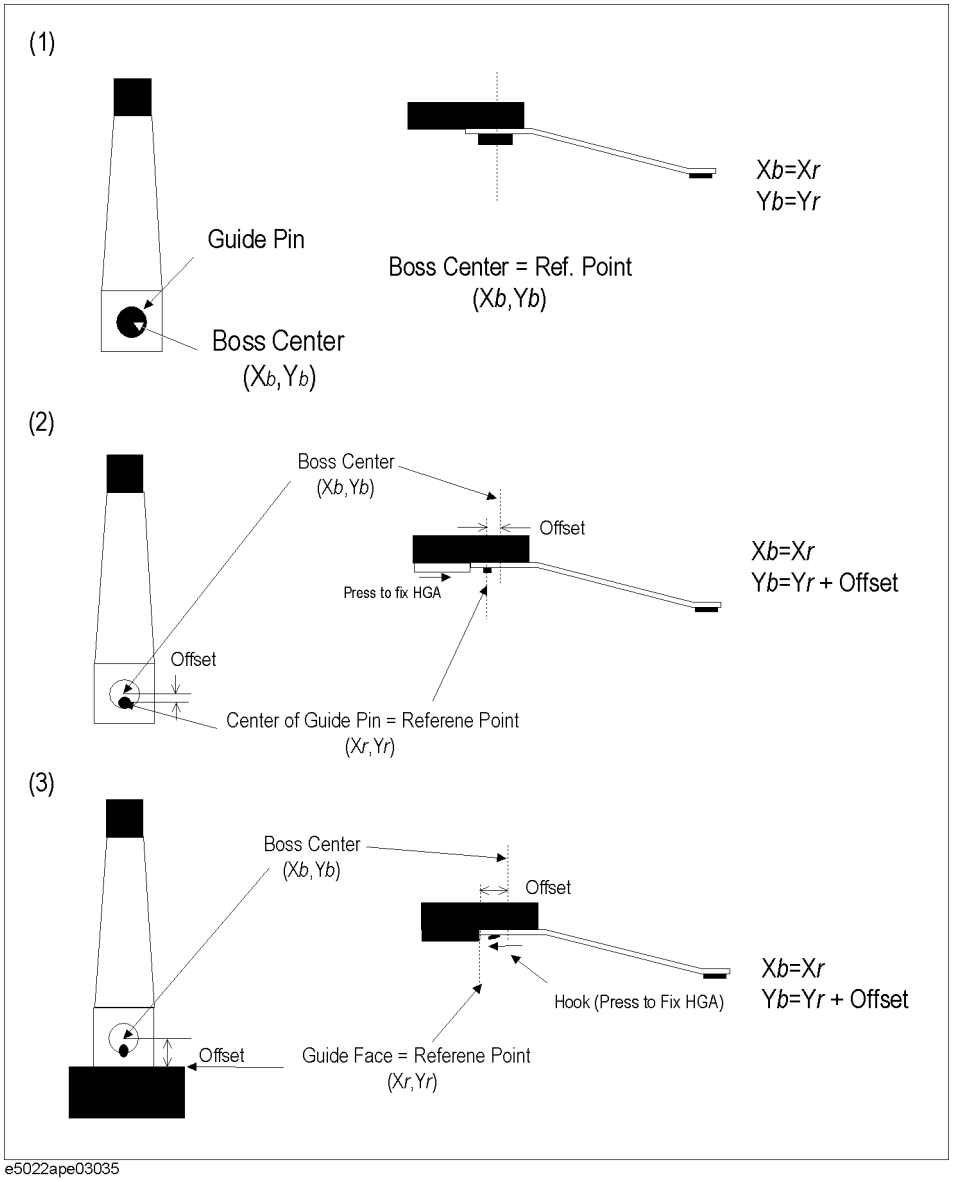
The first point should be at the base of the HGA as shown in Figure 3-6. Permission area is defined as shown in Figure 3-6, a parallel line with the X axis through the last point represents the boundary line of the permission area.

Figure 3-6 First Point and Last Point



There are various types of attaching the HGA on cassette. When the boss center is not the same as reference position (which positions the HGA), you have to calculate the boss center. The compensated value of the boss center position should be entered.

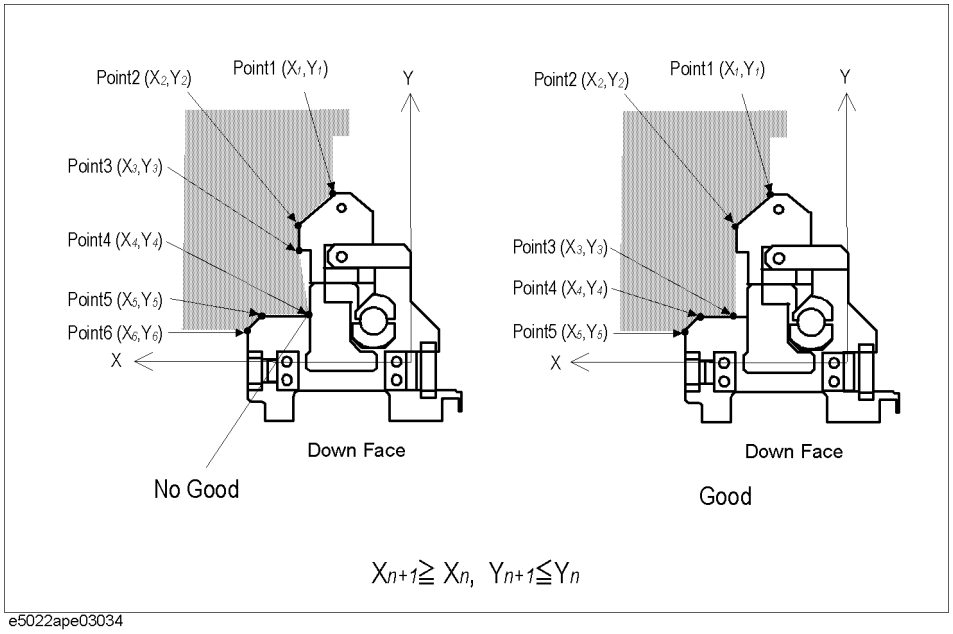
Figure 3-7 **Boss Center Compensation for various types of HGA attachment**



e5022ape03035

The value of x-coordinate at a certain point must be greater than or equal to the value of the previous point. On the other hand, the value of y-coordinate at a certain point must be less than or equal to the value of the previous point. See Figure 3-8.

Figure 3-8 **Rules for X,Y coordinate values**



e5022ape03034

The inhibit margin specified by “hpe5022_inhibitMargin” is valid for this function.

To use the customer designed HGA cassette, the “GENERIC1” must be selected in the “hpe5022_hgaCassette” function.

NOTE An auto disk alignment swings the skew angle from -20 to +20 degree. The customer designed HGA should be designed for the range of skew angle.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- boss_cent_x

  Description: Specifies the x-coordinate value of the boss center of the HGA (X_b in Figure 3-5 and Figure 3-7).

  Direction: IN

  Unit: Meter

  Preset Value: hpe5022_HGA_CASS_BOSS_CENT_MAX (1)

  Values

Name	Value
hpe5022_HGA_CASS_BOSS_CENT_MIN	-1

Name	Value
hpe5022_HGA_CASS_BOSS_CENT_MAX	1

- boss_cent_y

  Description: Specifies the y-coordinate value of the boss center of the HGA (Y_b in Figure 3-5 and Figure 3-7).

  Direction: IN

  Unit: Meter

  Preset Value: hpe5022_HGA_CASS_BOSS_CENT_MAX (1)

  Values

Name	Value
hpe5022_HGA_CASS_BOSS_CENT_MIN	-1
hpe5022_HGA_CASS_BOSS_CENT_MAX	1

- points

  Description: Specifies the number of points of outline vertex.

  Direction: IN

  Preset Value: hpe5022_HGA_CASS_OUTL_COUN_MIN (2)

  Values

Name	Value
hpe5022_HGA_CASS_OUTL_COUN_MIN	2
hpe5022_HGA_CASS_OUTL_COUN_MAX	256

- outline_x

  Description: Specifies the x-coordinate value of outline vertex as the data array ($X_{1\ to\ n}$ in Figure 3-5). The size of array must be the same as ‘points’

  Direction: IN

  Unit: Meter

  Preset Value: {1, 1}

  Values

Name	Value
hpe5022_HGA_CASS_OUTL_MIN	-1

Name	Value
hpe5022_HGA_CASS_OUTL_MAX	1

- outline_y

  Description — Specifies the y-coordinate value of the outline vertex as the data array ($Y_{1\ to\ n}$ in Figure 3-5). The size of array must be the same as ‘points’

  Direction — IN

  Unit — Meter

  Preset Value — {1, -1}

  Values

Name	Value
hpe5022_HGA_CASS_OUTL_MIN	-1
hpe5022_HGA_CASS_OUTL_MAX	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘boss_cent_x’, ‘boss_cent_y’, ‘points’, ‘outline_x’ or/and ‘outline_y’ is out of range

See Also

“hpe5022_hgaCassette” on page 169

“hpe5022_inhibitMargin” on page 166

hpe5022_hgaCassetteDimension_Q

C Syntax

ViStatus hpe5022_hgaCassetteDimension_Q(ViSession id, ViPReal64 boss_cent_x, ViPReal64 boss_cent_y, ViPInt16 points, const ViPReal64 outline_x[], const ViPReal64 outline_y[]);

Visual Basic Syntax

hpe5022_hgaCassetteDimension_Q(ByVal id As Long, ByRef boss_cent_x As Double, ByRef boss_cent_y As Double, ByRef points As Integer, ByRef outline_x As Double, ByRef outline_y As Double) As Long

Description

This function returns the dimensions of the customer designed HGA cassette specified by the “hpe5022_hgaCassetteDimension” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- boss_cent_x

  Description — Returns the x-coordinate value of the boss center of the HGA.

  Direction — OUT

  Unit — Meter

- boss_cent_y

  Description — Returns the y-coordinate value of the boss center of the HGA.

  Direction — OUT

  Unit — Meter

- points

  Description — Returns the number of points of outline vertex.

  Direction — OUT

- outline_x

  Description — Returns the x-coordinate value of outline vertex as the data array. The size of array is the same as ‘points’

  Direction — OUT

  Unit — Meter

- outline_y

  Description — Returns the y-coordinate value of outline vertex as the data array. The size of array is the same as ‘points’

Direction	OUT
Unit	Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

See Also

“hpe5022_hgaCassetteDimension” on page 171

hpe5022_hgaFace_Q

C Syntax

ViStatus hpe5022_hgaFace_Q (ViSession id, ViPInt16 face);

Visual Basic Syntax

hpe5022_hgaFace_Q(ByVal id As Long, ByRef face As Integer) As Long

Description

This function returns HGA cassette face type. The face type is detected by the photo coupler diode on the HLM.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- face

  Description: Returns the face type of HGA cassette.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_HGA_FACE_UP	0	Up Face
hpe5022_HGA_FACE_DOWN	1	Down Face

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_MISS	The HGA cassette is not installed. Check if the cassette is installed properly.

See Also

“hpe5022_hgaCassette” on page 169

hpe5022_dataArea

C Syntax

ViStatus hpe5022_dataArea(ViSession id, ViReal64 id_rad, ViInt32 max_trk, ViReal64 pitch);

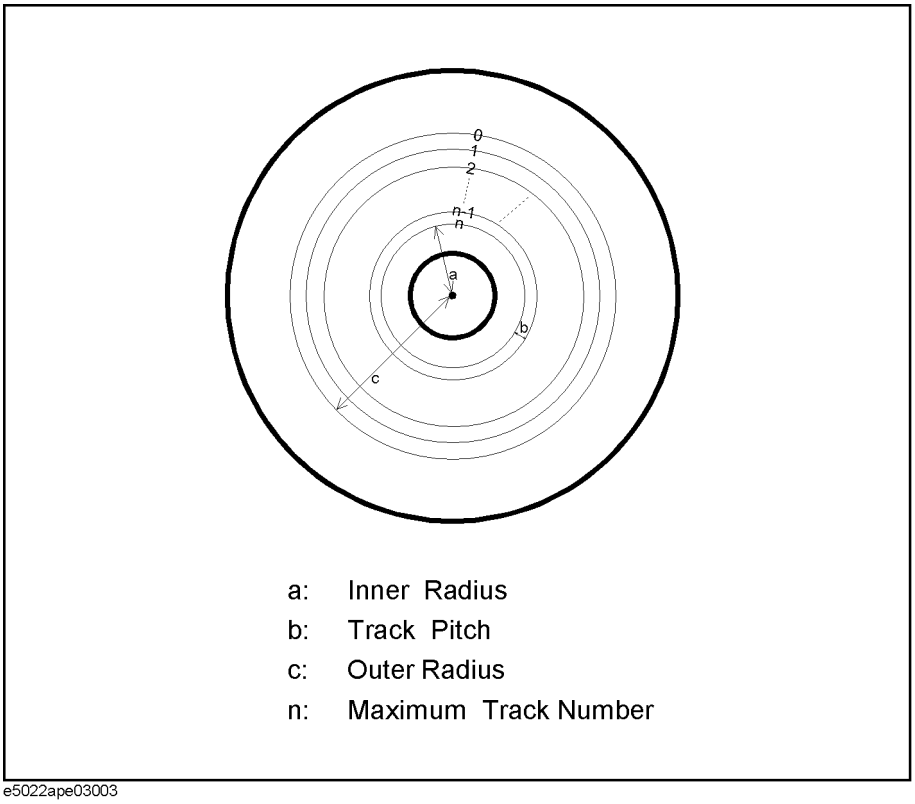
Visual Basic Syntax

hpe5022_dataArea(ByVal id As Long, ByVal id_rad As Double, ByVal max_trk As Long, ByVal pitch As Double) As Long

Description

This function specifies the data area. The data area is defined by the inmost track radius, the track pitch and the maximum track number. The track number begins at 0, so that the number of tracks equals the maximum track number plus 1.

Figure 3-9 Data Area Definition



e5022ape03003

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- id_rad

  Description — Specifies the inmost track radius of the data area on the disk. The distance between the reference position of HGA and the spindle center is defined as the radius.

Direction IN

Unit Meter

Preset values

Disk size	Preset Value
48mm (1.8")	14.5×10^{-3}
65mm (2.5")	18.5×10^{-3}
95mm (3.5")	21.5×10^{-3}
130mm (5.25")	29×10^{-3}

- max_trk

  Description Specifies the maximum track number. Tracks are numbered from 0 to ‘max_trk’ from outer side.

  Direction IN

  Preset values

Disk size	Preset Value
48mm(1.8”)	2500
65mm (2.5")	4000
95mm (3.5")	8000
130mm (5.25")	11000

  Value

hpe5022_TRACK_MIN	0
hpe5022_TRACK_MAX	1000000

- pitch

  Description Specifies the track pitch.

  Direction IN

  Preset value 3×10^{-6}

  Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘max_trk’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive is running. The spinstand drive should be turned off before this function is executed. See the “hpe5022_driveState” function.

See Also

“hpe5022_dataArea_Q” on page 183

“hpe5022_driveState” on page 209

hpe5022_dataArea_Q

C Syntax

ViStatus hpe5022_dataArea(ViSession id, ViPReal64 id_rad, ViPInt32 max_trk, ViPReal64 pitch);

Visual Basic Syntax

hpe5022_dataArea_Q(ByVal id As Long, ByRef id_rad As Double, ByRef max_trk As Long, ByRef pitch As Double) As Long

Description

This function returns the parameters which defines the data area. They are the inner radius, the track pitch and the maximum track number.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- id_rad

  Description — Returns the innermost track radius of the data area on the disk.

  Direction — OUT

  Unit — Meter

- max_trk

  Description — Returns the maximum track number.

  Direction — OUT

- pitch

  Description — Returns the track pitch.

  Direction — OUT

  Unit — Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_dataArea” on page 180

hpe5022_dataAreaOuterRadius_Q

C Syntax

ViStatus hpe5022_dataAreaOuterRadius_Q(ViSession id, ViPReal64 od_rad);

Visual Basic Syntax

hpe5022_dataAreaOuterRadius_Q(ByVal id As Long, ByRef od_rad As Double) As Long

Description

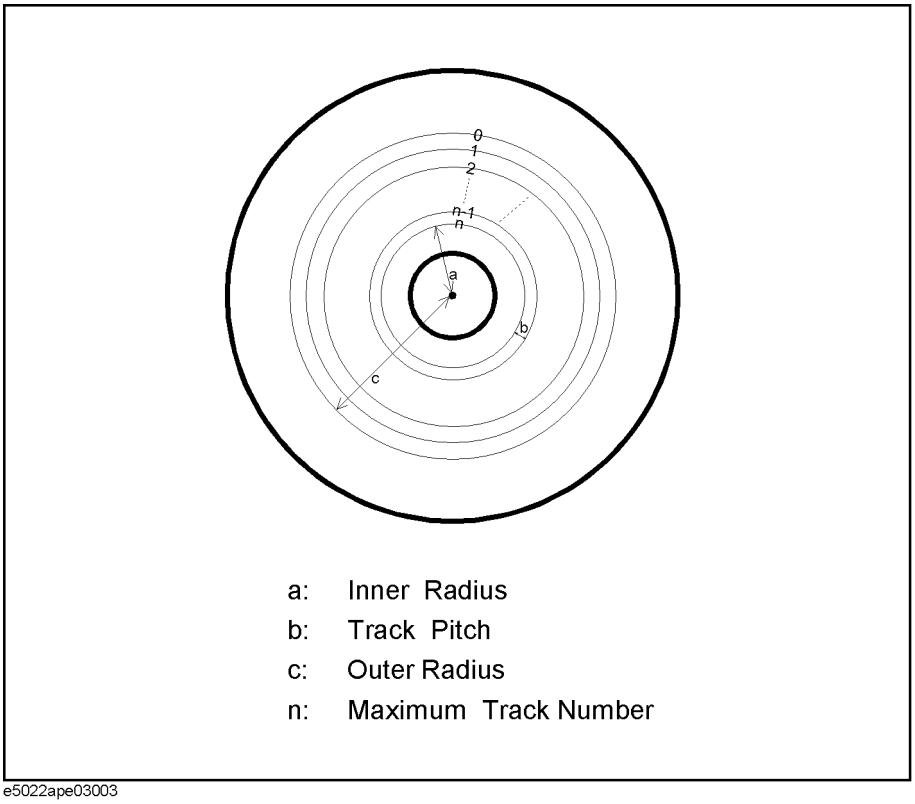
This function returns the outmost radius of the data area. The outmost radius is the distance between the center of the disk and the center of track No 0.

The outmost radius is calculated from the following formula.

$$OuterRadius = InnnerRadius + (MaximumTrackNo \times TrackPitch)$$

The inmost radius, the maximum track number and the track pitch are determined by the “hpe5022_dataArea” function.

Figure 3-10 Data Area Definition



e5022ape03003

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- od_rad

  Description Returns the outer radius of the data area on the disk.

  Direction OUT

  Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_dataArea” on page 180

hpe5022_headLoadType

C Syntax

ViStatus hpe5022_headLoadType(ViSession id, ViInt16 loadType);

Visual Basic Syntax

hpe5022_headLoadType(ByVal id As Long, ByVal loadType As Integer) As Long

Description

This function specifies the method to load the head. In case that hpe5022_LOAD_TYPE_NORMAL or hpe5022_LOAD_TYPE_TILT are specified as the parameter loadType, the head is loaded with the configuration set up by “hpe5022_headLoadConfig” or “hpe5022_headLoadConfigEx”. In case that hpe5022_LOAD_TYPE_RAMP is specified as loadType, the head is loaded using a ramp loading mechanism with the configuration set up by “hpe5022_headLoadRampConfig”.

When ramp load is used, the ramp load position and media load position must be configured to the spinstand using a ramp load positioning utility program.

When tilt load is used, the tilt head loader is used instead of the standard HLM.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- loadType

  Description Specifies the type of the head loading.

  Direction IN

  Values

Name	Value	Description
hpe5022_LOAD_TYPE_NORMAL	0	Normal Loading
hpe5022_LOAD_TYPE_RAMP	1	Ramp Loading
hpe5022_LOAD_TYPE_TILT	4112	Tilt Loading

  Preset Values hpe5022_LOAD_TYPE_NORMAL (0)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'loadType' is out of range.
hpe5022_ERROR_NSUP_CONF	• Tilt loading mechanism or Ramp loading mechanism is not installed. • Load/Initial position is not specified with ramp load.

See Also

"hpe5022_headLoadType_Q" on page 188

"hpe5022_headLoadConfig" on page 192

"hpe5022_headLoadConfigEx" on page 195

"hpe5022_headLoadRampConfig" on page 201

hpe5022_headLoadType_Q

C Syntax

ViStatus hpe5022_headLoadType_Q(ViSession id, ViInt16 loadType);

Visual Basic Syntax

hpe5022_headLoadType_Q(ByVal id As Long, ByRef loadType As Integer) As Long

Description

This function returns the type of the head loading method.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- loadType

  Description: Returns the type of the head loading.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_LOAD_TYPE_NORMAL	0	Normal Loading
hpe5022_LOAD_TYPE_RAMP	1	Ramp Loading
hpe5022_LOAD_TYPE_TILT	4112	Tilt Loading

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_headLoadType” on page 186

hpe5022_headLoadTypeList_Q

C Syntax

ViStatus hpe5022_headLoadTypeList_Q(ViSession id, ViPInt16 size, ViInt16 loadTypes[]);

Visual Basic Syntax

hpe5022_headLoadTypeList_Q(ByVal id As Long, ByRef size As Integer, ByRef loadTypes As Integer) As Long

Description

This function returns the list of head loading method types supported by the currently used spinstand.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the number of head load types supported by the spinstand.

  Direction — OUT

- loadTypes

  Description — Returns the array of the head load types. The maximum array size is (hpe5022_HEAD_LOAD_TYPE_SIZE_MAX).

  Direction — OUT

  Values

Name	Value
hpe5022_HEAD_LOAD_TYPE_SIZE_MAX	256

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_headLoadType” on page 186

“hpe5022_headLoadType_Q” on page 188

“hpe5022_headLoadTypeName_Q” on page 190

hpe5022_headLoadTypeName_Q

C Syntax

ViStatus hpe5022_headLoadTypeName_Q(ViSession id, ViInt16 loadType, ViPString typeName);

Visual Basic Syntax

hpe5022_headLoadTypeName_Q(ByVal id As Long, ByVal loadType As Integer, ByVal typeName As String) As Long

Description

This function returns the name of specified load method type.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- loadType

  Description — Specifies the head load type. This value should be included in the array returned by the function “hpe5022_headLoadTypeList_Q” on page 189.

  Direction — IN

- typeName

  Description — Returns the head load type name. The maximum number of characters is (hpe5022_HEAD_LOAD_NAME_SIZE_MAX).

  Direction — OUT

  Values

Name	Value
hpe5022_HEAD_LOAD_NAME_SIZE_MAX	256

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The specified load method type is not supported by the currently used spinstand. Supported load method types are returned by the function “hpe5022_headLoadTypeList_Q” on page 189.

See Also

“hpe5022_headLoadTypeList_Q” on page 189

“hpe5022_headLoadType” on page 186

“hpe5022_headLoadType_Q” on page 188

hpe5022_headLoadConfig

C Syntax

ViStatus hpe5022_headLoadConfig(ViSession id, ViInt16 method, ViReal64 loadPos, ViInt32 initTrack);

Visual Basic Syntax

hpe5022_headLoadConfig(ByVal id As Long, ByVal method As Integer, ByVal loadPos As Double, ByVal initTrack As Long) As Long

Description

This function specifies the load position and initial track when the hpe5022_LOAD_TYPE_NORMAL or hpe5022_LOAD_TYPE_TILT is specified as the loadType by using “hpe5022_headLoadType” function. The head moves from the home position to the load position, as it is loaded. Then positions at the initial track when the drive is turned on by the “hpe5022_driveState” function.

Either the “hpe5022_headLoadConfigEx” or this function should be defined. If both functions are defined, then the last executed function becomes valid.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- method

  Description: Specifies the head loading method. Dynamic loading can load and unload the head while the spindle is rotating. While static loading stops the spindle rotation before it loads and unloads the head.

  Direction: IN

  Values

Name	Value	Description
hpe5022_LOAD_DYNAMIC	0	Dynamic Loading
hpe5022_LOAD_STATIC	1	Static Loading

  Preset Values: hpe5022_LOAD_DYNAMIC (0)

- loadPos

  Description: Specifies the distance between the spindle rotation axis and the head position, where the head is loaded and unloaded.

  Direction: IN

  Unit: Meter

Preset Values

Disk size	Preset Value
4mm (1.8")	13.5×10^{-3}
65mm (2.5")	17.5×10^{-3}
95mm (3.5")	20×10^{-3}
130mm (5.25")	27.5×10^{-3}

initTrack

Description Specifies the initial track number.

Direction IN

Preset value 0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘loadPos’ and/or ‘initTrack’ are out of range.

See Also

“hpe5022_headLoadConfig_Q” on page 194

“hpe5022_driveState” on page 209

“hpe5022_headLoadConfigEx” on page 195

“hpe5022_headLoadType” on page 186

hpe5022_headLoadConfig_Q

C Syntax

ViStatus hpe5022_headLoadConfig_Q(ViSession id, ViPInt16 method, ViPReal64 loadPos, ViPInt32 initTrack);

Visual Basic Syntax

hpe5022_headLoadConfig_Q(ByVal id As Long, ByRef method As Integer, ByRef loadPos As Double, ByRef initTrack As Long) As Long

Description

This function returns the load position and initial track number when the hpe5022_LOAD_TYPE_NORMAL or hpe5022_LOAD_TYPE_TILT is specified as the loadType by using “hpe5022_headLoadType” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- method

  Description — Returns the head loading method.

  Direction — OUT

  Values — Same as the ‘method’ in the “hpe5022_headLoadConfig” function.

- loadPos

  Description — Returns the distance between the spindle rotation axis and the head position, where the head is loaded or unloaded.

  Direction — OUT

  Unit — Meter

- initTrack

  Description — Returns the initial track number.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_headLoadConfig” on page 192

hpe5022_headLoadConfigEx

C Syntax

ViStatus hpe5022_headLoadConfigEx(ViSession id, ViInt32 loadRpm, ViReal64 loadRadius, ViReal64 loadSkew, ViReal64 initRadius, ViReal64 initSkew, ViInt32 unloadRpm, ViReal64 unloadRadius, ViReal64 unloadSkew);

Visual Basic Syntax

hpe5022_headLoadConfigEx(ByVal id As Long, ByVal loadRpm As Long, ByVal loadRadius As Double, ByVal loadSkew As Double, ByVal initRadius As Double, ByVal initSkew As Double, ByVal unloadRpm As Double, ByVal unloadRadius As Double, ByVal unloadSkew As Double) As Long

Description

This function configures the head loading conditions when the hpe5022_LOAD_TYPE_NORMAL or hpe5022_LOAD_TYPE_TILT is specified as the loadType by using “hpe5022_headLoadType” function. This function allows you to set the spindle rotation speed and head load position where the head is loaded/unloaded. Parameters such as load rpm, load radius, load skew, initial radius, initial skew, unload rpm, unload radius and unload skew are specified in this function.

Either the “hpe5022_headLoadConfig” or this function should be defined. If both functions are defined, then the last executed function becomes valid. In the “hpe5022_headLoadConfig” you are only allowed to specify the type of loading and head position as the head loads. In this function you are allowed to set these parameters independent from unload operation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- loadRpm

  Description: Specifies the type of spindle rotation speed when the head is loaded. The range of rpm value is returned by the “hpe5022_spindleSpeed_Q” function.

  The “hpe5022_LOAD_RPM_COUPLED” means that the type of loading is Dynamic. A dynamic loading is when the spindle rotation is continuous as the head loads. Static on the other hand, means that the spindle rotation stops as the head loads, this value is represented by “hpe5022_LOAD_RPM_STATIC”.

  Direction: IN

  Unit: rpm

  Preset Value: hpe5022_LOAD_RPM_COUPLED (-1)

Values

Name	Value
hpe5022_LOAD_RPM_COUPLED	-1
hpe5022_LOAD_RPM_STATIC	0

- loadRadius

  Description Specifies the radius when the head is loaded.

  Direction IN

  Unit Meter

  Preset Value At reset, this value is set according to the following disk sizes.

Disk Size	Value
48mm (1.8")	13.5×10^{-3}
65mm (2.5")	17.5×10^{-3}
95mm (3.5")	20×10^{-3}
130mm (5.25")	27.5×10^{-3}

- loadSkew

  Description Specifies the skew angle when the head is loaded.

  Direction IN

  Unit Degree

  Preset Value 0

Name	Value
hpe5022_SKEW_ANGLE_MIN	-60
hpe5022_SKEW_ANGLE_MAX	30

- initRadius

  Description Specifies the radius at initial position after the head is loaded.

  Direction IN

  Unit Meter

  Preset Value Radius of track 0.

- initSkew

Description Specifies the skew angle at the initial position after the head is loaded.

Direction IN

Unit Degree

Preset Value 0

Name	Value
hpe5022_SKEW_ANGLE_MIN	-60
hpe5022_SKEW_ANGLE_MAX	30

- unloadRpm

  Description Specifies the spindle rotation speed when the head is unloaded. At reset, this value is set to the same value of the loadRpm reset value.

  Direction IN

  Unit hpe5022_LOAD_RPM_COUPLED.(-1)

- unloadRadius

  Description Specifies the radius when the head is unloaded.

  Direction IN

  Unit Meter

  Preset Value Load radius reset value

- unloadSkew

  Description Specifies the skew angle when the head is unloaded.

  Direction IN

  Unit Degree

  Preset Value 0

Name	Value
hpe5022_SKEW_ANGLE_MIN	-60
hpe5022_SKEW_ANGLE_MAX	30

NOTE When parameters “loadSkew”, “initSkew” and “unloadSkew” are set below -30 degrees the E5022A/B system can not guarantee a track offset of ± 6µm, this could result to an error “hpe5022_ERROR_INV_PARAMETER”. When skew angle is set below -30 degrees the track offset must be set to a value lower than ± 6µm to avoid this error. It is recommended that the skew angle be set between 30 and -30 degrees for ± 6µm track offset.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	Either one of the following parameters ‘loadRpm’, ‘loadRadius’, ‘loadSkew’, ‘initRadius’, ‘initSkew’, ‘unloadRpm’, ‘unloadRadius’ and/or ‘unloadSkew’ is out of range.

See Also

“hpe5022_headLoadConfigEx_Q” on page 199

“hpe5022_headLoadConfig” on page 192

“hpe5022_driveState” on page 209

“hpe5022_headLoad” on page 237

“hpe5022_spindleSpeed_Q” on page 214

“hpe5022_headLoadType” on page 186

hpe5022_headLoadConfigEx_Q

C Syntax

ViStatus hpe5022_headLoadConfigEx_Q(ViSession id, ViPInt32 loadRpm, ViPReal64 loadRadius, ViPReal64 loadSkew, ViPReal64 initRadius, ViPReal64 initSkew, ViPInt32 unloadRpm, ViPReal64 unloadRadius, ViPReal64 unloadSkew);

Visual Basic Syntax

hpe5022_headLoadConfigEx_Q(ByVal id As Long, ByRef loadRpm As Long, ByRef loadRadius As Double, ByRef loadSkew As Double, ByRef initRadius As Double, ByRef initSkew As Double, ByRef unloadRpm As Double, ByRef unloadRadius As Double, ByRef unloadRpm As Double) As Long

Description

This function returns the specified head loading configurations when the hpe5022_LOAD_TYPE_NORMAL or hpe5022_LOAD_TYPE_TILT is specified as the loadType by using “hpe5022_headLoadType” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- loadRpm

  Description — Returns the spindle rotation speed in rpm when the head is loaded.

  Direction — OUT

  Unit — rpm

- loadRadius

  Description — Returns the radius when the head is loaded.

  Direction — OUT

  Unit — Meter

- loadSkew

  Description — Returns the skew angle when the head is loaded.

  Direction — OUT

  Unit — Degree

- initRadius

  Description — Returns the radius at initial position of the head after loading.

  Direction — OUT

  Unit — Meter

- initSkew

  Description Returns the skew angle at the initial position of the head after loading.

  Direction OUT

  Unit Degree

- unloadRpm

  Description Returns the spindle rotation speed in rpm when the head is unloaded.

  Direction OUT

  Unit rpm

- unloadRadius

  Description Returns the radius when the head is unloaded.

  Direction OUT

  Unit Meter

- unloadSkew

  Description Returns the skew angle when the head is unloaded.

  Direction OUT

  Unit Degree

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_headLoadConfigEx” on page 195

“hpe5022_headLoadType” on page 186

hpe5022_headLoadRampConfig

C Syntax

ViStatus hpe5022_headLoadRampConfig(ViSession id, ViReal64 headSpeedOnRamp, ViReal64 initRadius, ViReal64 initSkew);

Visual Basic Syntax

hpe5022_headLoadRampConfig(ByVal id As Long, ByVal headSpeedOnRamp As Double, ByVal initRadius As Double, ByVal initSkew As Double) As Long

Description

This function specifies the head loading conditions when the hpe5022_LOAD_TYPE_RAMP is specified as the loadType by “hpe5022_headLoadType” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- headSpeedOnRamp

  Description: Specifies the head speed between ramp load position and media load position. The range of the head speed is returned by the “hpe5022_headPositionSpeedRange_Q”function.

  Direction: IN

  Unit: Meter per Second

  Preset Values: maximum speed of the head speed

- initRadius

  Description: Specifies the radius that the head is moved to after the head is loaded.

  Direction: IN

  Unit: Meter

  Preset Values: (inner diameter radius + outer diameter radius) / 2

- initSkew

  Description: Specifies the skew angle that the head is moved to after the head is loaded.

  Direction: IN

  Unit: Degree

  Values

Name	Value
hpe5022_SKEW_ANGLE_MIN	-60.0

Name	Value
hpe5022_SKEW_ANGLE_MAX	30.0

Preset Values 0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The handle specified by ‘headSpeedOnRamp’, ‘initRadius’ and/or ‘initSkew’ is invalid.

See Also

“hpe5022_headLoadType” on page 186

“hpe5022_headLoadRampConfig_Q” on page 203

hpe5022_headLoadRampConfig_Q

C Syntax

ViStatus hpe5022_headLoadRampConfig_Q(ViSession id, ViPReal64 headSpeedOnRamp, ViPReal64 initRadius, ViPReal64 initSkew);

Visual Basic Syntax

hpe5022_headLoadRampConfig_Q(ByVal id As Long, ByRef headSpeedOnRamp As Double, ByRef initRadius As Double, ByRef initSkew As Double) As Long

Description

This function returns the head loading conditions when the hpe5022_LOAD_TYPE_RAMP is specified as the loadType by “hpe5022_headLoadType” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- headSpeedOnRamp

  Description — Returns the head speed on ramp.

  Direction — OUT

  Unit — Meter per Second

- initRadius

  Description — Returns the radius, that the head is moved to after the head is loaded.

  Direction — OUT

  Unit — Meter

- initSkew

  Description — Returns the skew angle, that the head is moved to after the head is loaded.

  Direction — OUT

  Unit — Degree

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_headLoadType” on page 186

“hpe5022_headLoadRampConfig” on page 201

hpe5022_diskSizeRestriction

C Syntax

ViStatus hpe5022_diskSizeRestriction(ViSession id, ViBoolean restrict);

Visual Basic Syntax

hpe5022_diskSizeRestriction(ByVal id As Long, ByVal restrict As Integer) As Long

Description

This function removes a restriction of the head position. Normally, the head loading/unloading position and a position itself must be within the disk size. This function allows you to set the loading/unloading position and the moved position at the outside of the disk media.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- restrict

  Description: Select a restriction of the head position.

  When this parameter is set to “VI_TRUE”, the head position and loading/unloading position must be inside of the disk size.

  When this parameter is set to “VI_FALSE”, the head position can be set outside of the disk size.

  This function is effective at the following functions:

  — “hpe5022_headLoadConfigEx”

  — “hpe5022_headLoadConfig”

  — “hpe5022_headLoadRampConfig”

  — “hpe5022_headPosition”

  — “hpe5022_headLoad”

  — “hpe5022_headLoadImmediate”

  Direction: IN

  Preset Value: VI_TRUE

  Values

Name	Value
VI_FALSE	0
VI_TRUE	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘restrict’ is out of range.

See Also

“hpe5022_headLoadConfigEx” on page 195

“hpe5022_headLoadConfig” on page 192

“hpe5022_headLoadRampConfig” on page 201

“hpe5022_headPosition” on page 241

“hpe5022_headLoad” on page 237

“hpe5022_headLoadImmediate” on page 247

hpe5022_diskSize_Q

C Syntax

ViStatus hpe5022_diskSize_Q(ViSession id, ViPReal64 disk_size);

Visual Basic Syntax

hpe5022_diskSize_Q(ByVal id As Long, ByRef disk_size As Double) As Long

Description

This function returns the disk substrate outside diameter.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- disk_size

  Description — Returns the disk diameter in meter.When either of the 3 inch or the 3.5 inch disk is attached, this parameter will be set to the out diameter value of the 3.5 inch disk regardless of the actual disk size.

  Direction

Disk Size	Return Value
1.8 inch	0.048 m
2.5 inch	0.065 m
3 / 3.5 inch	0.095 m

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

hpe5022_diskSizeRestriction_Q

C Syntax

ViStatus hpe5022_diskSizeRestriction_Q(ViSession id, ViPBoolean restrict);

Visual Basic Syntax

hpe5022_diskSizeRestriction_Q(ByVal id As Long, ByRef restrict As Integer) As Long

Description

This function returns a status of the head position restriction.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- restrict

  Description: Select a restriction of the head position.

  When “VI_TRUE” is selected, the head position and loading/unloading position must be inside of the disk size.

  When “VI_FALSE” is selected, the head position can be set outside of the disk size.

  Direction: OUT

  Values

Name	Value
VI_FALSE	0
VI_TRUE	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_diskSizeRestriction” on page 205

Drive Control Function

This section describes the functions to control the spinstand drive. This function allows you to rotate the spindle and load the head.

hpe5022_driveState

C Syntax

ViStatus hpe5022_driveState(ViSession id, ViBoolean drvOn);

Visual Basic Syntax

hpe5022_driveState(ByVal id As Long, ByVal drvOn As Integer) As Long

Description

This function controls the spinstand drive. This function performs the following sequences.

- Drive On sequence:
  1. Clamp the disk.
  2. Rotate the spindle.
  3. Move the head to the load position where it is specified by the “hpe5022_headLoadConfig” function.
  4. Load it.
  5. Move it at the initial track where it is specified by the “hpe5022_headLoadConfig”function.
- Drive Off sequence:
  1. Move the head to the load position.
  2. Unload it.
  3. Move it at the home position.
  4. Stop the rotation of the spindle.
  5. Unclamp the disk.

Parameters

- id

Description	Specifies the system identifier. This is given by the "hpe5022_init" function.
Direction	IN

- drvOn

Description	Turn-on or turn-off the drive.
Preset Value	VI_FALSE (0)
Direction	IN

Values

Name	Value	Description
VI_TRUE	1	Drive On
VI_FALSE	0	Drive Off

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD	Hardware error is detected.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_HARD_MISS	The HGA cassette is not installed or the media is not placed. Check if the cassette is installed properly and if the media is set properly.
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_POSITION	The specified head position is in the inhibit area. Check the Agilent part number of the cassette, the inhibit area, the HGA dimension, the data area and the specified track number. See the "hpe5022_hgaCassette", "hpe5022_hgaDimension" and "hpe5022_inhibitMargin" functions.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.
hpe5022_ERROR_NDEF_HGA	The HGA dimension is not defined. See the "hpe5022_hgaCassette" function.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.
hpe5022_ERROR_SHROUD_OPEN	The shroud cover is open. Close this cover when the drive is on.

See Also

"hpe5022_hgaDimension" on page 161

"hpe5022_inhibitMargin" on page 166

"hpe5022_hgaCassette" on page 169

“hpe5022_driveState_Q” on page 212

hpe5022_driveState_Q

C Syntax

ViStatus hpe5022_driveState_Q(ViSession id,ViPBoolean drvOn);

Visual Basic Syntax

hpe5022_driveState_Q(ByVal id As Long, ByRef drvOn As Integer) As Long

Description

This function returns the status of the spinstand drive.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- drvOn

  Description — Returns the status of the spinstand drive

  Direction — OUT

  Values — Same as the ‘drvOn’ parameter in the“hpe5022_driveState” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_driveState” on page 209

hpe5022_spindleSpeed

C Syntax

ViStatus hpe5022_spindleSpeed(ViSession id, ViInt32 rpm);

Visual Basic Syntax

hpe5022_spindleSpeed(ByVal id As Long, ByVal rpm As Long) As Long

Description

This function specifies the speed of the spindle rotation. The spindle speed at loading and unloading can be changed by the “hpe5022_headLoadConfigEx” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rpm

  Description: Specifies the speed of the spindle rotation. The range of the rpm is returned by the “hpe5022_spindleSpeedRange_Q” function.

  Direction: IN

  Preset value: 5400

  Unit: rpm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The specified rotation speed is out of range.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.

See Also

“hpe5022_spindleSpeed_Q” on page 214

“hpe5022_spindleAcceleration” on page 216

“hpe5022_headLoadConfigEx” on page 195

hpe5022_spindleSpeed_Q

C Syntax

ViStatus hpe5022_spindleSpeed_Q(ViSession id, ViPInt32 rpm);

Visual Basic Syntax

hpe5022_spindleSpeed_Q(ByVal id As Long, ByRef rpm As Long) As Long

Description

This function returns the speed of the spindle rotation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rpm

  Description: Returns the speed of the spindle rotation.

  Direction: OUT

  Unit: rpm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spindleSpeed” on page 213

hpe5022_spindleSpeedRange_Q

C Syntax

ViStatus hpe5022_spindleSpeedRange_Q(ViSession id, ViPInt32 rpmMin, ViPInt32 rpmMax);

Visual Basic Syntax

hpe5022_spindleSpeedRange_Q(ByVal id As Long, ByRef rpmMin As Long, ByRef rpmMax As Long) As Long

Description

This function returns the range of the spindle speed which is used as a reference to specify “rpm” in the “hpe5022_spindleSpeed” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- rpmMin

  Description — Returns the minimum speed of spindle rotation.

  Direction — OUT

  Unit — rpm

- rpmMax

  Description — Returns the maximum speed of spindle rotation.

  Direction — OUT

  Unit — rpm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spindleSpeed” on page 213

hpe5022_spindleAcceleration

C Syntax

ViStatus hpe5022_spindleAcceleration(ViSession id, ViInt32 accel);

Visual Basic Syntax

hpe5022_spindleAcceleration(ByVal id As Long, ByVal accel As Long) As Long

Description

This function is used to set up the acceleration of the spindle.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- accel

  Description: Specifies the acceleration of the spindle rotation. The acceleration range (i.e, max. acceleration and min. acceleration) to which this parameter can be set is returned by the “hpe5022_spindleAccelerationRange_Q” function.

  Direction: IN

  Unit: RPM/sec

NOTE

The VEE program supports only two acceleration settings (i.e, 2500 for slow and 10000 for fast acceleration). With this function the user can specify the acceleration at any values within the allowable acceleration range as returned by “hpe5022_spindleAccelerationRange_Q” function. However, when the user sets the acceleration value anywhere within this range (i.e, 3000 or 3500) the system can not guarantee that the specified acceleration. Due to electronics and mechanical constraints there are some regions where the user can not set the acceleration value, which could result to an error. This error message will appear in the LCD display of the spinstand. The user will have to do trial and error to determine the acceptable setting.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter “accel” is out of range.

See Also

“hpe5022_spindleAcceleration_Q” on page 217

hpe5022_spindleAcceleration_Q

C Syntax

ViStatus hpe5022_spindleAcceleration_Q(ViSession id, ViPInt32 accel);

Visual Basic Syntax

hpe5022_spindleAcceleration_Q(ByVal id As Long, ByRef accel As Long) As Long

Description

This function returns the specified acceleration of the spindle rotation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- accel

  Description: Returns the acceleration of the spindle rotation.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spindleAcceleration” on page 216

hpe5022_spindleAccelerationRange_Q

C Syntax

ViStatus hpe5022_spindleAccelerationRange_Q(ViSession id, ViPInt32 accelMin, ViPInt32 accelMax);

Visual Basic Syntax

hpe5022_spindleAccelerationRange_Q(ByVal id As Long, ByRef accelMin As Long, ByRef accelMax as Long) As Long

Description

This function returns the maximum and minimum allowable accelerations of the spindle which is used as a reference to specify the acceleration “accel” in the “hpe5022_spindleAcceleration” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- accelMin

  Description — Returns the minimum acceleration range of the spindle rotation.

  Direction — IN

  Unit — RPM/sec

- accelMax

  Description — Returns the maximum acceleration range of the spindle rotation.

  Direction — IN

  Unit — RPM/sec

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spindleAcceleration” on page 216

hpe5022_readTrackOffset

C Syntax

ViStatus hpe5022_readTrackOffset(ViSession id, ViReal64 offset);

Visual Basic Syntax

hpe5022_readTrackOffset(ByVal id As Long, ByVal offset As Double) As Long

Description

This function specifies the read track offset from the center of the track. The inward direction is positive. The head moves to track offset position during read sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the read track offset. If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is narrowed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -6.0×10^{-6}, respectively. See the track offset compensation in the operation manual.

  Direction: IN

  Preset value: 0

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q” function.

See Also

“hpe5022_readTrackOffset_Q” on page 221

“hpe5022_trackOffsetCompValue” on page 984

hpe5022_readTrackOffset_Q

C Syntax

ViStatus hpe5022_readTrackOffset_Q(ViSession id, ViPReal64 offset);

Visual Basic Syntax

hpe5022_readTrackOffset_Q(ByVal id As Long, ByRef offset As Double) As Long

Description

This function returns the read track offset value. The inward direction is positive.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Returns the read track offset.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_readTrackOffset” on page 219

hpe5022_writeTrackOffset

C Syntax

ViStatus hpe5022_writeTrackOffset(ViSession id, ViReal64 offset);

Visual Basic Syntax

hpe5022_writeTrackOffset(ByVal id As Long, ByVal offset As Double) As Long

Description

This function specifies the write track offset from the center of track. The inward direction is positive. The head moves to the write track offset position during an erase and write sequence in all measurement functions.

This write offset value does not apply to the “hpe5022_write” and the “hpe5022_erase” functions.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the write track offset. If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction: IN

  Preset value: 0

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'offset' is out of range. Check if the head offset compensation value by using "hpe5022_trackOffsetCompValue_Q" function.

See Also

"hpe5022_writeTrackOffset_Q" on page 224

"hpe5022_write" on page 326

"hpe5022_erase" on page 362

"hpe5022_trackOffsetCompValue" on page 984

hpe5022_writeTrackOffset_Q

C Syntax

ViStatus hpe5022_writeTrackOffset_Q(ViSession id, ViPReal64 offset);

Visual Basic Syntax

hpe5022_writeTrackOffset_Q(ByVal id As Long, ByRef offset As Double) As Long

Description

This function returns the write track offset value. The inward direction is positive.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Returns the write track offset value

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_writeTrackOffset” on page 222

hpe5022_trackOffsetImmediate

C Syntax

ViStatus hpe5022_trackOffsetImmediate(ViSession id, ViReal64 offset);

Visual Basic Syntax

hpe5022_trackOffsetImmediate(ByVal id As Long, ByVal offset As Double) As Long

Description

This function allows you to move the fine stage position (piezo) without a trigger by the index pulse. This function works even if the head is unloaded.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the absolute track offset position to move. If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction: IN

  Preset value: 0

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’ is out of range.

See Also “hpe5022_trackOffsetCompValue” on page 984

hpe5022_trackOffsetUserSettlingTime

C Syntax

ViStatus hpe5022_trackOffsetUserSettlingTime(ViSession id, ViReal64 time);

Visual Basic Syntax

hpe5022_trackOffsetUserSettlingTime(ByVal id As Long, ByVal time As Double) As Long

Description

This function allows you to have a wait time after the fine positioning (piezo movement) before the next step start.

Normally, the piezo is moved during one or two spindle revolutions, then the write/read sequences is done at the next revolution. If you wait one or two more revolution before the write/read sequences start, the piezo position might be more stable. So that, this might improve the measurement stability.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- time

  Description: Specifies the additional setting time. The time of (one revolution period)*n [n=1, 2, 3..] is recommended.

  Direction: IN

  Preset value: 0

  Unit: second

  Values

Name	Value
hpe5022_TRACK_OFFSET_USER_SETT_MIN	0
hpe5022_TRACK_OFFSET_USER_SETT_MAX	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘time’ is out of range.

See Also

“hpe5022_trackOffsetUserSettlingTime_Q” on page 228

hpe5022_trackOffsetUserSettlingTime_Q

C Syntax

ViStatus hpe5022_trackOffsetUserSettlingTime_Q(ViSession id, ViPReal64 time);

Visual Basic Syntax

hpe5022_trackOffsetUserSettlingTime_Q(ByVal id As Long, ByRef time As Double) As Long

Description

This function returns the additional settling time of the fine positioning.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- time

  Description: Returns the additional setting time.

  Direction: OUT

  Unit: second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

"hpe5022_trackOffsetUserSettlingTime" on page 227

hpe5022_track

C Syntax

ViStatus hpe5022_track(ViSession id, ViInt32 track_no);

Visual Basic Syntax

hpe5022_track(ByVal id As Long, ByVal track_no As Long) As Long

Description

This function moves the head at the specified track from the current track.

The head position and the head skew angle are calculated using the parameters specified by the “hpe5022_driveConfigPivot” or “hpe5022_driveConfigRadiusSkew”, and “hpe5022_hgaDimension” functions.

When you want to set the head position and the skew angle manually, use the “hpe5022_headPosition” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- track_no

  Description: Specifies the target track number. The outermost track is track number 0. The maximum track number is specified by the “hpe5022_dataArea” function.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.

Error Code	Description
hpe5022_ERROR_INV_POSITION	The specified head position is in the inhibit area. Check the Agilent part number of the cassette, the inhibit area, the HGA dimension, the data area and the specified track number. See the “hpe5022_hgaCassette” “hpe5022_inhibitMargin”, “hpe5022_hgaDimension” and “hpe5022_dataArea” functions.
hpe5022_ERROR_NDEF_HGA	The HGA dimenstion is not defined. See the “hpe5022_hgaCassette” and “hpe5022_hgaDimension” functions.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.

See Also

“hpe5022_driveConfigPivot” on page 154

“hpe5022_driveConfigRadiusSkew” on page 157

“hpe5022_hgaDimension” on page 161

“hpe5022_inhibitMargin” on page 166

“hpe5022_hgaCassette” on page 169

“hpe5022_dataArea” on page 180

“hpe5022_driveState” on page 209

“hpe5022_track_Q” on page 231

“hpe5022_headPosition” on page 241

hpe5022_track_Q

C Syntax

ViStatus hpe5022_track_Q(ViSession id, ViPInt32 track_no);

Visual Basic Syntax

hpe5022_track_Q(ByVal id As Long, ByRef track_no As Long) As Long

Description

This function returns the track number where the head is located.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- track_no

  Description: Returns the current track number of the head position. During operation of the “hpe5022_track” function, when a failure has occurred just before calling this function the ‘track_no’ is set to -1. It is also set to -1 when this function is called just after calling the “hpe5022_headPosition” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_dataArea” on page 180

“hpe5022_track” on page 229

“hpe5022_headPosition” on page 241

hpe5022_motorState

C Syntax

ViStatus hpe5022_motorState(ViSession id, ViBoolean motorOn);

Visual Basic Syntax

hpe5022_motorState(ByVal id As Long, ByVal motorOn As Integer) As Long

Description

This function rotates and stops the spindle. The sequence is described below.

- Motor on sequence:
  1. Clamp the disk.
  2. Rotate the spindle.
- Motor off sequence:
  1. Stop the spindle rotation.
  2. Unclamp the disk.

As the “hpe5022_driveState” functions includes this function, you will use this function when you want to just rotate and stop the spindle while the drive is turned on (“hpe5022_driveState” is set to “VI_TRUE”).

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- motorOn

  Description: Specifies to rotate or stop the spindle.

  Direction: IN

  Preset value: 0 (VI_FALSE)

  Values

Name	Value	Description
VI_TRUE	1	Motor on
VI_FALSE	0	Motor off

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_MISS	The media is not set. Check if the media is set properly.
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the "hpe5022_driveState" function.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.
hpe5022_ERROR_NDEF_HGA	The HGA dimenstion is not defined.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.
hpe5022_ERROR_SHROUD_OPEN	The shroud cover is open. Close this cover when the drive is on.

See Also

"hpe5022_driveState" on page 209

"hpe5022_motorState_Q" on page 234

hpe5022_motorState_Q

C Syntax

ViStatus hpe5022_motorState_Q(ViSession id, ViPBoolean motorOn);

Visual Basic Syntax

hpe5022_motorState_Q(ByVal id As Long, ByRef motorOn As Integer) As Long

Description

This function returns when the spindle rotates.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- motorOn

  Description: Returns if the spindle rotates.

  Direction: OUT

  Values: Same as ‘motorOn’ parameter in the “hpe5022_motorState” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_motorState” on page 232

hpe5022_motorStateEx

C Syntax

ViStatus hpe5022_motorStateEx(ViSession id, ViInt16 motorState);

Visual Basic Syntax

hpe5022_motorStateEx(ByVal id As Long, ByVal motorState As Integer) As Long

Description

This function allows you to stop the spindle rotation with clamping the disk. The sequence is described below.

- Motor on sequence: (same as the “Motor on” sequence of the “hpe5022_motorState” function)
  1. Clamp the disk.
  2. Rotate the spindle.
- Motor off sequence: (same as the “Motor off” sequence of the “hpe5022_motorState” function)
  1. Stop the spindle rotation.
  2. Unclamp the disk.
- Motor stop sequence:
  1. Stop the spindle rotation with clamping the disk.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- motorOn

  Description: Specifies the motor condition.

  Direction: IN

  Values

Name	Value	Description
pe5022_MOTOR_STATE_ON	0	Motor on
hpe5022_MOTOR_STATE_OFF	1	Motor off
hpe5022_MOTOR_STATE_STOP	2	Motor stop

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_MISS	The media is not set. Check if the media is set properly.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.
hpe5022_ERROR_NDEF_HGA	The HGA dimenstion is not defined.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.
hpe5022_ERROR_SHROUD_OPEN	The shroud cover is open. Close this cover when the drive is on.

See Also

“hpe5022_driveState” on page 209

“hpe5022_motorState_Q” on page 234

hpe5022_headLoad

C Syntax

ViStatus hpe5022_headLoad(ViSession id, ViBoolean load);

Visual Basic Syntax

hpe5022_headLoad(ByVal id As Long, ByVal load As Integer) As Long

Description

This function loads and unloads the head. The sequences are as follows:

- When hpe5022_LOAD_TYPE_NORMAL or hpe5022_LOAD_TYPE_TILT is specified as the loadType by the “hpe5022_headLoadType” function.

  — Head Load sequence:

  1. Move the head to the load position where it is specified by the “hpe5022_headLoadConfig” or “hpe5022_headLoadConfigEx” function.
  2. Load the head.
  3. Move the head to the initial position where it is specified by the “hpe5022_headLoadConfig” or “hpe5022_headLoadConfigEx” function.

  — Head Unload sequence:

  1. Move the head to the load position where it is specified by the “hpe5022_headLoadConfig” or “hpe5022_headLoadConfigEx” function.
  2. Unload the head.
  3. Move the head to the home position.

- When hpe5022_LOAD_TYPE_RAMP is specified as the loadType by the “hpe5022_headLoadType” function.

  — Head Load sequence:

  1. Move the ramp to Ramp In position.
  2. Move the head to the ramp load position where it is configured by the Ramp Positioning Utility.
  3. Load the head.
  4. Move the head to the media load position where it is configured by the Ramp Positioning Utility, with the head speed cconfigured by the “hpe5022_headLoadRampConfig” function.
  5. Move the ramp to Ramp Out position.
  6. Move the head to the initial position where it is specified by the “hpe5022_headLoadRampConfig”.

  — Head Unload sequence:

1. Move the ramp to Ramp In position
2. Move the head to the media load position.
3. Move the head to the ramp load position with the head speed configured by the “hpe5022_headLoadRampConfig” function.
4. Unload the head.
5. Move the ramp to Ramp Out position.

   Move the head to the home position.

As the “hpe5022_driveState” function includes this function, use this function when you want to just unload and load while the spindle is rotating.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- load

  Description: Specifies loading or unloading.

  Direction: IN

  Preset value: 0 (VI_FALSE)

  Values

Name	Value	Description
VI_TRUE	1	Load the Head
VI_FALSE	0	Unload the head

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_HARD	Hardware error is detected.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.

Error Code	Description
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.
hpe5022_ERROR_INV_POSITION	The specified head position is in the inhibit area. Check the Agilent part number of the cassette, the inhibit area, the HGA dimension, the data area and the specified track number. See the “hpe5022_hgaCassette”, “hpe5022_hgaDimension” and “hpe5022_inhibitMargin” functions.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.

See Also

“hpe5022_headLoadConfig” on page 192

“hpe5022_driveState” on page 209

“hpe5022_headLoad_Q” on page 240

“hpe5022_headLoadType” on page 186

hpe5022_headLoad_Q

C Syntax

ViStatus hpe5022_headLoad_Q(ViSession id, ViPBoolean load);

Visual Basic Syntax

hpe5022_headLoad_Q(ByVal id As Long, ByRef load As Integer) As Long

Description

This function returns if the head is loaded.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- load

  Description: Returns if the head is loaded.

  Direction: OUT

  Values: Same as ‘load’ parameter in the “hpe5022_headLoad” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_HARD	Hardware error is detected.

See Also

“hpe5022_headLoad” on page 237

hpe5022_headPosition

C Syntax

ViStatus hpe5022_headPosition(ViSession id, ViReal64 radius, ViReal64 skew);

Visual Basic Syntax

hpe5022_headPosition(ByVal id As Long, ByVal radius As Double, ByVal skew As Double) As Long

Description

This function moves the head at the specified position and controls the skew angle at the specified value.

This function moves the head to specified position from its current position.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- radius

  Description: Specifies the target radius of the head position.

  Direction: IN

  Unit: Meter

- skew

  Description: Specifies the target skew angle.

  Direction: IN

  Unit: Degree

  Values

Name	Value
hpe5022_SKEW_ANGLE_MIN	-60
hpe5022_SKEW_ANGLE_MAX	30

NOTE

When skew is set below -30 degrees the E5022A/B system can not guarantee a track offset of ± 6µm, this could result to an error “hpe5022_ERROR_INV_PARAMETER”. When skew angle is set below -30 degrees the track offset must be set to a value lower than ± 6µm to avoid this error. It is recommended that the skew angle be set between 30 and -30 degrees.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the "hpe5022_driveState" function.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.
hpe5022_ERROR_INV_POSITION	The specified head position is in the inhibit area. Check the Agilent part number of the cassette, the inhibit area, the HGA dimension, the data area and the specified track number. See the "hpe5022_hgaCassette", "hpe5022_hgaDimension", "hpe5022_inhibitMargin" and "hpe5022_dataArea" functions.
hpe5022_ERROR_NDEF_HGA	The HGA dimension is not defined. See the "hpe5022_hgaCassette" and "hpe5022_hgaDimension" functions.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.

See Also

"hpe5022_hgaDimension" on page 161

"hpe5022_inhibitMargin" on page 166

"hpe5022_hgaCassette" on page 169

"hpe5022_dataArea" on page 180

"hpe5022_driveState" on page 209

"hpe5022_track" on page 229

hpe5022_headPosition_Q

C Syntax

ViStatus hpe5022_headPosition_Q(ViSession id, ViPReal64 radius, ViPReal64 skew);

Visual Basic Syntax

hpe5022_headPosition_Q(ByVal id As Long, ByRef radius As Double, ByRef skew As Double) As Long

Description

This function returns the current position of the head from its previous position.

Even if the “hpe5022_track” function is used to set the head position and the skew angle, this function returns the current values that is set automatically.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- radius

  Description — Returns the current radius of the head position.

  Direction — OUT

  Unit — Meter

- skew

  Description — Returns the current skew angle.

  Direction — OUT

  Unit — Degree

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_hgaDimension” on page 161

“hpe5022_inhibitMargin” on page 166

“hpe5022_hgaCassette” on page 169

“hpe5022_dataArea” on page 180

“hpe5022_driveState” on page 209

“hpe5022_track” on page 229

hpe5022_rampPosition

C Syntax

ViStatus hpe5022_rampPosition(ViSession id, ViInt16 position);

Visual Basic Syntax

hpe5022_rampPosition(ByVal id As Long, ByVal position As Integer) As Long

Description

This function controls the position of the ramp load mechanism of the spinstand. In case that the spinstand in use does not support ramp mechanism, “hpe5022_ERROR_NSUP_FUNC” will take place.

Parameters

- id

  Description: Specifies the system identifier. This is given by the “hpe5022_init” function.

  Direction: IN

- position

  Description: Specifies the position of the ramp load mechanism.

  Direction: IN

  Values: The following constants can be used.

Name	Value
hpe5022_RAMP_POSITION_EVACUATED	0
hpe5022_RAMP_POSITION_OPERATING	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The handle specified by ‘position’ is invalid.
hpe5022_ERROR_NSUP_FUNC	The ramp load mechanism is not installed.

See Also

“hpe5022_rampPosition_Q” on page 246

“hpe5022_headLoadRampConfig” on page 201

“hpe5022_headLoadRampConfig_Q” on page 203

hpe5022_rampPosition_Q

C Syntax

ViStatus hpe5022_rampPosition_Q(ViSession id, ViPInt16 position);

Visual Basic Syntax

hpe5022_rampPosition_Q(ByVal id As Long, ByVal position As Integer) As Long

Description

This function returns the position of the ramp load mechanism of the spinstand. In case that the spinstand in use does not support ramp mechanism, “hpe5022_ERROR_NSUP_FUNC” will take place.

Parameters

- id

  Description — Specifies the system identifier. This is given by the “hpe5022_init” function.

  Direction — IN

- position

  Description — Returns the position of the ramp load mechanism.

  Direction — OUT

  Values — The following constants can be used.

Name	Value
hpe5022_RAMP_POSITION_EVACUATED	0
hpe5022_RAMP_POSITION_OPERATING	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NSUP_FUNC	The ramp load mechanism is not installed.

See Also

“hpe5022_rampPosition” on page 245

“hpe5022_headLoadRampConfig” on page 201

“hpe5022_headLoadRampConfig_Q” on page 203

hpe5022_headLoadImmediate

C Syntax

ViStatus hpe5022_headLoadImmediate(ViSession id, ViBoolean load);

Visual Basic Syntax

hpe5022_headLoadImmediate(ByVal id As Long, ByVal load As Integer) As Long

Description

This function loads or unloads the head at the current head position. Head seek action is not done by this function. This function will succeed when the head is located at the home position or upon the disk. Motor on/off state does not affect the operation of this function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the “hpe5022_init” function.

  Direction: IN

- load

  Description: Loads and unloads the head.

  Direction: IN

  Values

Name	Value	Description
VI_TRUE	1	Load head
VI_FALSE	0	Unload head

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD	Hardware error is detected.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INTERLOCK	The spinstand is out of air. Check if air is supplied to the spinstand.

Error Code	Description
hpe5022_ERROR_INV_POSITION	The specified head position is in the inhibit area. Check the Agilent part number of the cassette, the inhibit area, the HGA dimension, the data area and the specified track number. See the “hpe5022_hgaCassette”, “hpe5022_hgaDimension” and “hpe5022_inhibitMargin” functions.

See Also

“hpe5022_hgaCassette” on page 169

“hpe5022_hgaDimension” on page 161

“hpe5022_inhibitMargin” on page 166

hpe5022_headPositionSpeed

C Syntax

ViStatus hpe5022_headPositionSpeed(ViSession id, ViReal64 vel);

Visual Basic Syntax

hpe5022_headPositionSpeed(ByVal id As Long, ByVal vel As Double) As Long

Description

This function controls a velocity of the head movement by the coarse stage. The head movement speeds at not only on-media but also off-media (the loading/unloading and a movement from/to the home position) are changed. The speed of the piezo movement is not changed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- vel

  Description: Specifies a velocity of the head movement. The range of velocity is returned by the “hpe5022_headPositionSpeedRange_Q” function.

  Direction: IN

  Unit: Meter/Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘vel’ is out of range.

See Also

“hpe5022_headPositionSpeed_Q” on page 250

“hpe5022_headPositionSpeedRange_Q” on page 251

hpe5022_headPositionSpeed_Q

C Syntax

ViStatus hpe5022_headPositionSpeed_Q(ViSession id, ViPReal64 vel);

Visual Basic Syntax

hpe5022_headPositionSpeed(ByVal id As Long, ByRef vel As Double) As Long

Description

This function returns the velocity of the head movement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- vel

  Description: Returns a velocity of the head movement.

  Direction: OUT

  Unit: Meter/Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_headPositionSpeed” on page 249

hpe5022_headPositionSpeedRange_Q

C Syntax

ViStatus hpe5022_headPositionSpeedRange_Q(ViSession id, ViPReal64 velMin, ViPReal64 velMax);

Visual Basic Syntax

hpe5022_headPositionSpeedRange_Q(ByVal id As Long, ByRef velMin As Double, ByRef velMax As Double) As Long

Description

This function returns the minimum and maximum values of the head movement velocity. This values are depending on a spinstand type.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- velMin

  Description: Returns the minimum velocity of the head movement.

  Direction: OUT

  Unit: Meter/Second

- velMax

  Description: Returns the maximum velocity of the head movement.

  Direction: OUT

  Unit: Meter/Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_headPositionSpeed” on page 249

hpe5022_trackToRadiusSkew

C Syntax

ViStatus hpe5022_trackToRadiusSkew(ViSession id, ViInt32 track_no, ViPReal64 radius, ViPReal64 skew);

Visual Basic Syntax

hpe5022_trackToRadiusSkew(ByVal id As Long, ByVal track_no As Long, ByRef radius As Double, ByRef skew As Double) As Long

Description

This function computes and returns the radius and skew angle of the head as specified by the “hpe5022_track” function which specifies the track number position of the head.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- track_no

  Description — Specifies the track number from where to calculate the radius and skew angle.

  Direction — IN

  Unit — Meter

- radius

  Description — Returns the radius corresponding to its specified track number.

  Direction — OUT

  Unit — Meter

- skew

  Description — Returns the skew angle corresponding to its specified track number.

  Direction — OUT

  Unit — Degree

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘track_no’ is out of range.

See Also “hpe5022_track” on page 229

Input Range Control Function

This section describes the control function of the input range.

hpe5022_measurementMode

C Syntax

ViStatus hpe5022_measurementMode(ViSession id,ViInt16 mode);

Visual Basic Syntax

hpe5022_measurementMode(ByVal id As Long, ByVal mode As Integer) As Long

Description

This function specifies the measurement mode.

When the measurement mode is set to auto, the values specified by the “hpe5022_inputRange” and “hpe5022_pwReference” functions are calculated from the value specified by the “hpe5022_isolatedPulseReference” function.

When this function is set to manual, the user must specify two other functions (the “hpe5022_inputRange” and “hpe5022_pwReference” functions).

See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Specifies the type of measurement mode. If set to “hpe5022_MEAS_AUTO”, then the input range and PW reference will automatically be set to its optimum value.

  Direction: IN

  Preset Value: hpe5022_MEAS_AUTO

  Values

Name	Value	Description
hpe5022_MEAS_AUTO	0	The auto level control is ON
hpe5022_MEAS_MAN	1	The auto level control is OFF

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘mode’ is out of range.

See Also

“hpe5022_measurementMode_Q” on page 256

“hpe5022_autoConfig” on page 257

“hpe5022_isolatedPulseReference” on page 265

“hpe5022_inputRange” on page 271

“hpe5022_pwReference” on page 403

hpe5022_measurementMode_Q

C Syntax

ViStatus hpe5022_measurementMode_Q(ViSession id,ViPInt16 mode);

Visual Basic Syntax

hpe5022_measurementMode_Q(ByVal id As Long, ByRef mode As Integer) As Long

Description

This function returns the specified measurement mode.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Returns the measurement mode.

  Direction: OUT

  Values: Same as ‘mode’ in the “hpe5022_measurementMode” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_measurementMode” on page 254

hpe5022_autoConfig

C Syntax

ViStatus hpe5022_autoConfig(ViSession id,ViPReal64 taa, ViPReal64 pwLength);

Visual Basic Syntax

hpe5022_autoConfig(ByVal id As Long, ByRef taa As Double, ByRef pwLength As Double) As Long

Description

This function measures the TAA and the PW length of an isolated pulse pattern and sets the result into the “hpe5022_isolatedPulseReference” function.

When this function is executed and “hpe5022_measurementMode” function is set to auto mode, the input range is automatically controlled even if the pulse is affected by the Inter-Symbol Interference (ISI).

When magnetic transitions are written close enough, the isolated pulses from each transition start to overlap. This overlapping effect is called ISI.

The auto configuration measures the TAA and the PW length for an isolated pulse. E5022A/B controls the input level using these values. Since these values are set automatically in the “hpe5022_isolatedPulseReference” function, there is no need to execute this function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- taa

  Description: Returns the TAA value of the reference isolated pulse.

  Direction: OUT

  Unit: Volt

- pwLength

  Description: Returns the Pulse Width (PW) length of the reference isolated pulse.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_AUTO_CONFIG_FAILED	The auto configuration has failed.
hpe5022_ERROR_MOTOR_FAULT	An error is detected in the motor or servo of the spinstand. Check the spinstand and restart the system.

See Also

“hpe5022_measurementMode” on page 254

“hpe5022_isolatedPulseReference” on page 265

“hpe5022_inputRange” on page 271

hpe5022_autoConfigEraseBand

C Syntax

ViStatus hpe5022_autoConfigEraseBand(ViSession id, ViReal64 range, ViReal64 pitch);

Visual Basic Syntax

hpe5022_autoConfigEraseBand(ByVal id As Long, ByVal range As Double, ByVal pitch As Double) As Long

Description

This function specifies parameters for a band erase in the auto configuration sequence. The band erase performs erases between -(specified range) and +(specified range) with the specified pitch.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- range

  Description: Specifies the erase range. the minimum value is 0.

  Direction: IN

  Unit: Meter

  Preset Value: 3×10^{-6}

  Values

Name	Value
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- pitch

  Description: Specifies the erase pitch.

  Direction: IN

  Unit: Meter

  Preset Value: 0.3×10^{-6}

  Values

Name	Value
hpe5022_ERASE_BAND_PITCH_MIN	50×10^{-9}
hpe5022_ERASE_BAND_PITCH_MAX	6×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘range’ and/or ‘pitch’ is out of range.

See Also

“hpe5022_autoConfig” on page 257

“hpe5022_autoConfigEraseBand_Q” on page 261

hpe5022_autoConfigEraseBand_Q

C Syntax

ViStatus hpe5022_autoConfigEraseBand_Q(ViSession id, ViPReal64 range, ViPReal64 pitch);

Visual Basic Syntax

hpe5022_autoConfigEraseBand(ByVal id As Long, ByRef range As Double, ByRef pitch As Double) As Long

Description

This function returns parameters for a band erase in the auto configuration sequence.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- range

  Description — Returns the width to erase.

  Direction — OUT

  Unit — Meter

- pitch

  Description — Returns the erase pitch.

  Direction — OUT

  Unit — Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_autoConfigEraseBand” on page 259

“hpe5022_autoConfig” on page 257

hpe5022_autoConfigTrackProfile

C Syntax

ViStatus hpe5022_autoConfigTrackProfile(ViSession id, ViReal64 range, ViInt16 points);

Visual Basic Syntax

hpe5022_autoConfigTrackProfile(ByVal id As Long, ByVal range As Double, ByVal points As Integer) As Long

Description

This function specifies parameters for a track profile measurement in the auto configuration sequence. The track profile measurement performs TAA measurements between -(specified range) and +(specified range) with the specified number of points in order to find the optimum input range.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- range

  Description: Specifies the width to make a track profile measurement. The minimum value is 0.

  Direction: IN

  Unit: Meter

  Preset Value: 2×10^{-6}

  Values

Name	Value
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- points

  Description: Specifies number of points for the track profile measurement.

  Direction: IN

  Preset Value: 101

  Values

Name	Value
hpe5022_TRACK_PROFILE_SIZE_MIN	1
hpe5022_TRACK_PROFILE_SIZE_MAX	201

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘range’ and/or ‘points’is out of range.

See Also

“hpe5022_autoConfig” on page 257

“hpe5022_autoConfigTrackProfile_Q” on page 264

hpe5022_autoConfigTrackProfile_Q

C Syntax

ViStatus hpe5022_autoConfigTrackProfile_Q(ViSession id, ViPReal64 range, ViPInt16 points);

Visual Basic Syntax

hpe5022_autoConfigTrackProfile(ByVal id As Long, ByRef range As Double, ByRef points As Integer) As Long

Description

This function returns parameters for a track profile measurement in the auto configuration sequence.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- range

  Description — Returns the width to make a track profile measurement.

  Direction — OUT

  Unit — Meter

- points

  Description — Returns number of points for the track profile measurement.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_autoConfig” on page 257

“hpe5022_autoConfigTrackProfile” on page 262

hpe5022_isolatedPulseReference

C Syntax

ViStatus hpe5022_isolatedPulseReference(ViSession id,ViReal64 taa, ViReal64 pwLength);

Visual Basic Syntax

hpe5022_isolatedPulseReference(ByVal id As Long, ByVal taa As Double, ByVal pwLength As Double) As Long

Description

This function specifies the TAA and the PW length of the reference isolated pulse. Typical TAA and PW length for an isolated pulse of measurement must be specified.

When magnetic transitions are written close enough, the isolated pulses from each transition start to overlap according to the principle of superposition. This overlapping effect is called ISI. E5022A/B calculates the level and the pulse width of an input signal even if the pulse is affected by ISI.

Since “hpe5022_autoConfig” function measures an isolated pulse and determines the parameters automatically, there is no need to set the parameters of this function when auto configuration is executed.

See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- taa

  Description: Specifies the TAA of the reference isolated pulse.

  Direction: IN

  Preset Value: 2.3333×10^{-3} (= $350\times10^{-3}/150$)

  Unit: Volt

- pwLength

  Description: Specifies the PW length of the reference isolated pulse.

  Direction: IN

  Preset Value: 0.5×10^{-3}

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

"hpe5022_measurementMode" on page 254

"hpe5022_autoConfig" on page 257

"hpe5022_isolatedPulseReference_Q" on page 267

"hpe5022_inputRange" on page 271

"hpe5022_pwReference" on page 403

hpe5022_isolatedPulseReference_Q

C Syntax

ViStatus hpe5022_isolatedPulseReference_Q(ViSession id,ViPReal64 taa, ViPReal64 pwLength);

Visual Basic Syntax

hpe5022_isolatedPulseReference_Q(ByVal id As Long, ByRef taa As Double, ByRef pwLength As Double) As Long

Description

This function returns the specified TAA and PW length of the reference isolated pulse.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- taa

  Description — Returns the TAA of the reference isolated pulse.

  Direction — OUT

  Unit — Volt

- pwLength

  Description — Returns the PW of the reference isolated pulse.

  Direction — IN

  Unit — Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_isolatedPulseReference” on page 265

hpe5022_isolatedPulseReferenceTaaAuto

C Syntax

ViStatus hpe5022_isolatedPulseReferenceTaaAuto(ViSession id, ViBoolean state);

Visual Basic Syntax

hpe5022_isolatedPulseReferenceTaaAuto(ByVal id As Long, ByVal state As Integer) As Long

Description

This function automatically controls the reference isolated pulse of TAA to an appropriate value in order to avoid data overflow. This function will only work when track profile is measured.

When an overflow occurs during track profile measurement and “state” parameter of this function is set to “VI_TURE”, this function will search the appropriate range for TAA by tracing the TAA’s peak value from the center of the pulse width. The input range will automatically adjusts itself until the overflow disappears.

This new range will be the basis from which all other measurements will be performed.

When track profile is measured and “state” is set to VI_FALSE, the input range setting will not change even if an overflow is detected.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- state

  Description: Specifies the auto setting state of the isolated pulse TAA.

  Direction: IN

  Preset Value: VI_FALSE

  Values

Name	Value	Description
VI_FALSE	0	The auto range control is OFF
VI_TRUE	1	The auto range control is ON

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

"hpe5022_inputRange" on page 271

"hpe5022_measurementMode" on page 254

"hpe5022_measureTrackProfile" on page 615

"hpe5022_measureTripleTrack" on page 714

hpe5022_isolatedPulseReferenceTaaAuto_Q

C Syntax

ViStatus hpe5022_isolatedPulseReferenceTaaAuto_Q(ViSession id, ViPBoolean state);

Visual Basic Syntax

hpe5022_isolatedPulseReferenceTaaAuto_Q(ByVal id As Long, ByRef state As Integer) As Long

Description

This function returns the specified automatic setting state of the reference isolated pulse of the TAA.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- state

  Description: Returns the auto setting state of the isolated pulse TAA.

  Direction: OUT

  Values

Name	Value	Description
VI_FALSE	0	The auto setting state is OFF
VI_TRUE	1	The auto setting state is ON

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_isolatedPulseReferenceTaaAuto” on page 268

hpe5022_inputRange

C Syntax

ViStatus hpe5022_inputRange(ViSession id,ViReal64 level);

Visual Basic Syntax

hpe5022_inputRange(ByVal id As Long, ByVal level As Double) As Long

Description

This function specifies the input range. Although E5022A/B adjusts its attenuators and amplifiers in order to execute proper measurement, an approximate voltage level must be set at the head.

Input range is affected by the parameters, channel bit rate, data pattern, spindle speed and head position radius when you measure the same head, E5022A/B provides a function that allows the user to calculate this parameters automatically. See the “hpe5022_isolatedPulseReference” or “hpe5022_autoConfig” functions.

The measurement mode defines which data is to be used. When this mode is set to auto, the values specified by the “hpe5022_isolatedPulseReference” function will be used to set the input range. When manual mode is selected, the value specified in this function will be used to set the input range.

See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- level

  Description: Specifies the input range.

  Direction: IN

  Unit: Volt

  Preset Values

Disk size	Preset Value
48mm (1.8")	2.3169×10^{-3}
65mm (2.5")	2.322×10^{-3}
95mm (3.5")	2.3282×10^{-3}
130mm (5.25")	2.3306×10^{-3}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

"hpe5022_measurementMode" on page 254

"hpe5022_autoConfig" on page 257

"hpe5022_isolatedPulseReference" on page 265

"hpe5022_inputRange_Q" on page 273

hpe5022_inputRange_Q

C Syntax

ViStatus hpe5022_inputRange_Q(ViSession id,ViPReal64 level);

Visual Basic Syntax

hpe5022_inputRange_Q(ByVal id As Long, ByRef level As Double) As Long

Description

This function returns the specified input range. The returned value depends on the specified measurement mode. When set to auto mode, the value specified by “hpe5022_isolatedPulseReference” function will be used to set the input range. When set to manual mode, the value specified by the “hpe5022_inputRange” function will be used to set the input range.

See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- level

  Description: Returns the specified input range.

  Direction: OUT

  Unit: Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_isolatedPulseReference” on page 265

“hpe5022_measurementMode” on page 254

“hpe5022_inputRange” on page 271

hpe5022_ampGain_Q

C Syntax

ViStatus hpe5022_ampGain_Q(ViSession id,ViPReal64 gain);

Visual Basic Syntax

hpe5022_ampGain_Q(ByVal id As Long, ByRef gain As Double) As Long

Description

This function returns the amplitude gain. This function allows you to calculate the signal level at the head from the output level of the filter matrix module.

$$SignalLevel_{atFilterMatrixOutput} = ampGain \times SignalLevel_{atHead}$$

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- gain

  Description: Returns the amplitude gain.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

Head Amplifier Function

This section describes the function that controls the head amplifier.

hpe5022_headAmpName_Q

C Syntax

ViStatushpe5022_headAmpName_Q(ViSession id, ViPString name);

Visual Basic Syntax

hpe5022_headAmpName_Q(ByVal id As Long, ByVal name As String) As Long

Description

This function returns Agilent part number of the head amplifier which is currently installed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- name

  Description: Returns the Agilent part number of the installed head amplifier. The character length of this parameter is defined by hpe5022_PART_NUM_LENGTH.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

hpe5022_headAmpInputImpedance_Q

C Syntax

ViStatus hpe5022_headAmpInputImpedance_Q(ViSession id, ViPReal64 impd);

Visual Basic Syntax

hpe5022_headAmpInputImpedance_Q(ByVal id As Long, ByRef impd As Double) As Long

Description

This function returns the input impedance of the head amplifier which is currently installed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- impd

  Description: Returns the input impedance of the installed head amplifier.

  Direction: OUT

  Unit: Ohms

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

hpe5022_headAmpProperty

C Syntax

ViStatus hpe5022_headAmpProperty(ViSession id, ViInt32 property, ViReal64 value);

Visual Basic Syntax

hpe5022_headAmpProperty(ByVal id As Long, ByVal property As Long, ByVal value As Double) As Long

Description

This function sets the property of the installed head amplifier. Property characteristics is dependent on each head amplifier. The property and its value is described in a corresponding header file for head amplifier. This file can be found in the directory named “c:\Program Files\Agilent \E5022\include\” with file name e5022_66xxx.h (for C Language) or e5022_66xxx.bas (for Visual Basic). (e5022_66xxx is part number of head amplifier). This function is available when the head amplifier is the E5029H and E5029J Option 701 to 704 and 706. When it is the E5029K, E5029J Option 705 and 707 and above, use the “hpe5022_headAmpRegister” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- property

  Description — Specifies property of the head amplifier. The property is named as “hpe5022_HAP_PROP_XXXXX_YYYYY”. (where XXXXX is a lower 5 digit part number of the head amplifier.)

  Direction — IN

- value

  Description — Specifies the data value for property.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘value’ is out of range.

Error Code	Description
hpe5022_ERROR_NSUP_FUNC	The E5035B does not support this function.

See Also

“hpe5022_headAmpProperty_Q” on page 279

“hpe5022_headAmpRegister” on page 280

hpe5022_headAmpProperty_Q

C Syntax

ViStatus hpe5022_headAmpProperty_Q(ViSession id, ViInt32 property, ViPReal64 value);

Visual Basic Syntax

hpe5022_headAmpProperty_Q(ByVal id As Long, ByVal property As Long, ByRef value As Double) As Long

Description

This function returns the property of the installed head amplifier. This function is available when the system has the E5035A. This function is available when the head amplifier is the E5029H and E5029J Option 701 to 704 and 706. When it is E5029K, E5029J Option 705 and 707 and above, use the “hpe5022_headAmpRegister” function.

Parameters

- Description — Specifies the system identifier. This is given by the "hpe5022_init" function.
  Direction — IN
- property
  Description — Specifies property of the head amplifier.
  Direction — IN
- value
  Description — Returns the data value for the specified property.
  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_NSUP_FUNC	The E5035B does not support this function.

See Also

“hpe5022_headAmpProperty” on page 277

“hpe5022_headAmpRegister_Q” on page 282

hpe5022_headAmpRegister

C Syntax

ViStatus hpe5022_headAmpRegister(ViSession id, ViInt16 regNo, ViInt16 upperBitPos, ViInt16 lowerBitPos, ViInt32 data);

Visual Basic Syntax

hpe5022_headAmpRegister(ByVal id As Long, ByVal regNo As Integer, ByVal upperBitPos As Integer, ByVal lowerBitPos As Integer, ByVal data As Long) As Long

Description

This function sets a specified data at the bit field of the head amplifier IC register. This function sets the data at the specified bits between upper and lower bit positions in the specified register number. The bit position of LSB of the register is zero.

This function is available when the head amplifier is the E5029J Option 705 and 707 and above, E5029K. When it is the E5029H and E5029J Option 701 to 704 and 706, use the “hpe5022_headAmpProperty” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- regNo

  Description: Specifies the register number of the installed head amplifier.

  Direction: IN

- upperBitPos

  Description: Specifies the upper bit position of the register bit field.

  Direction: IN

- lowerBitPos

  Description: Specifies the lower bit position of the register bit field.

  Direction: IN

- data

  Description: Specifies the value to be set at the specified bits.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter is out of range.
hpe5022_ERROR_NSUP_FUNC	The E5035A does not support this function.

See Also "hpe5022_headAmpProperty" on page 277

hpe5022_headAmpRegister_Q

C Syntax

ViStatus hpe5022_headAmpRegister_Q(ViSession id, ViInt16 regNo, ViInt16 upperBitPos, ViInt16 lowerBitPos, ViPInt32 data);

Visual Basic Syntax

hpe5022_headAmpRegister_Q(ByVal id As Long, ByVal regNo As Integer, ByVal upperBitPos As Integer, ByVal lowerBitPos As Integer, ByRef data As Long) As Long

Description

This function returns an established data at the specified bit field of the head amplifier IC register. This function is available when the head amplifier is the E5029J Option 705 and 707 and above, E5029K. When it is the E5029H and E5029J Option 701 to 704 and 706, use the “hpe5022_headAmpProperty_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- regNo

  Description: Specifies the register number of the installed head amplifier.

  Direction: IN

- upperBitPos

  Description: Specifies the upper (left) bit position of the register bit field.

  Direction: IN

- lowerBitPos

  Description: Specifies the lower (right) bit position of the register bit field.

  Direction: IN

- data

  Description: Returns the established value at the specified bits.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter is out of range.
hpe5022_ERROR_NSUP_FUNC	The E5035A does not support this function.

See Also “hpe5022_headAmpRegister” on page 280

hpe5022_headAmpActiveChannel

C Syntax

ViStatus hpe5022_headAmpActiveChannel(ViSession id, ViInt16 chan);

Visual Basic Syntax

hpe5022_headAmpActiveChannel(ByVal id As Long, ByVal chan As Integer) As Long

Description

This function specifies the active channel number of head amplifier which is currently installed. This function is not supported on the E5035A spinstand interface module.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- chan

  Description — Specifies the active channel number of the head amplifier which is currently installed.

  The upper and lower limit values are queryable by the hpe5022_headAmActiveChannelRange_Q function. Preset value is dependent on the head amplifier's configuration.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter is out of range.
hpe5022_ERROR_NSUP_FUNC	The E5035A does not support this function.

See Also

hpe5022_headAmpActiveChannel_Q

C Syntax

ViStatus hpe5022_headAmpActiveChannel_Q(ViSession id, ViPInt16 chan);

Visual Basic Syntax

hpe5022_headAmpActiveChannel_Q(ByVal id As Long, ByRef chan As Integer) As Long

Description

This function returns the active channel number of the head amplifier which is currently installed. This function is not supported on the E5035A spinstand interface module.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- chan

  Description — Returns the active channel number of the head amplifier which is currently installed.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter is out of range.
hpe5022_ERROR_NSUP_FUNC	The E5035A does not support this function.

See Also

hpe5022_headAmpActiveChannelRange_Q

C Syntax

ViStatus hpe5022_headAmpActiveChannelRange_Q(ViSession id, ViPInt16 chanMin, ViPInt16 chanMax);

Visual Basic Syntax

hpe5022_headAmpActiveChannelRange_Q(ByVal id As Long, ByRef chanMin As Integer, ByRef chanMax As Integer) As Long

Description

This function returns the active channel number range of the head amplifier which is currently installed. This function is not supported on the E5035A spinstand interface module.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- chanMin

  Description: Returns the smallest value of the selectable channel number of head amplifier which is currently installed.

  Direction: OUT

- chanMax

  Description: Returns the largest value of the selectable channel number of head amplifier which is currently installed.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter is out of range.
hpe5022_ERROR_NSUP_FUNC	The E5035A does not support this function.

See Also

hpe5022_updateHeadAmp

C Syntax

ViStatus hpe5022_updateHeadAmp(ViSession id);

Visual Basic Syntax

hpe5022_updateHeadAmp(ByVal id As Long) As Long

Description

This function is used when the head amplifier on cassette is installed. Query functions related with the head amplifier do not refresh the detected head amplifier's information.

When the user decides to change the type of head amplifier on cassette, the user must execute "hpe5022_updateHeadAmp" function, so that information query (i.e, "hpe5022_headAmpName_Q", "hpe5022_headAmpBufferName_Q", etc.) will report the correct information of the installed head amplifier on cassette.

When the head is loaded and the user executes this function, the head amplifier's information is automatically refreshed by the system. Thus, even if this function is not executed during measurement, correct setting and measurement is guaranteed by the system.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

hpe5022_headAmpSerialNumber_Q

C Syntax

ViStatus hpe5022_headAmpSerialNumber_Q(ViSession id, ViPString serial_number);

Visual Basic Syntax

hpe5022_headAmpSerialNumber_Q(ByVal id As Long, ByVal serial_number As String) As Long

Description

This function returns the serial number of the head amplifier currently installed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- serial_number

  Description: Returns the serial number of the head amplifier board (i.e. Agilent serial number). The character length of this parameter is defined by hpe5022_SERIAL_NUM_ LENGTH. If UHA (universal head amp) is used, null string is returned.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

hpe5022_headAmpBufferName_Q

C Syntax

ViStatus hpe5022_headAmpBufferName_Q(ViSession id, ViPString name);

Visual Basic Syntax

hpe5022_headAmpBufferName_Q(ByVal id As Long, ByVal name As String) As Long

Description

This function returns the name of the installed head amplifier buffer board (E5029J).

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- name

  Description: Returns the name of the head amplifier's buffer board (i.e. Agilent part number). The character length of this parameter is defined by hpe5022_PART_NUM_LENGTH. If a head amplifier on cassette is not used (i.e, head amplifier buffer board is not installed), the head amplifier's name (part number) is returned.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

hpe5022_headAmpBufferSerialNumber_Q

C Syntax

ViStatus hpe5022_headAmpBufferSerialNumber_Q(ViSession id, ViPString serial_number);

Visual Basic Syntax

hpe5022_headAmpBufferSerialNumber_Q(ByVal id As Long, ByVal serial_number As String) As Long

Description

This function returns the serial number of the head amplifier buffer board which is currently installed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- serial_number

  Description: Returns the serial number of the head amplifier buffer board (i.e. Agilent serial number). The character length of this parameter is defined by hpe5022_SERIAL_NUM_LENGTH. If a head amplifier on cassette is not used (i.e, head amplifier buffer board is not installed), then the serial number of the head amplifier is returned.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

hpe5022_headAmpInfoFile

C Syntax

ViStatus hpe5022_headAmpInfoFile(ViSession id, ViString bdFileName);

Visual Basic Syntax

hpe5022_headAmpInfoFile(ByVal id As Long, ByVal bdFileName As String) As Long

Description

This function specifies the head amplifier information file.

Generally, the system detects which head amplifier is installed automatically because it refers the information in the EEPROM in the head amplifier and reads the corresponding information file under the system software directory.

This function allows you to use the head amplifiers without the EEPROM in itself. The system reads the specified information file. The assistance from Agilent is required when users develop the information files.

When the head amplifier information file is specified by this function and the head amplifier with the EEPROM is installed, the information of the EEPROM is used and the file name specified by this function is ignored.

This function is available when the system has the E5035B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bdFileName

  Description: Specifies the file name of the head amp information. The file employs the XML format.

  The default directory is "C:\Program Files\Agilent\E5022\config\headamp". When the head amplifier information file is located under the other directory, specify the file name with the directory path.

  The maximum length of the string is defined by the (hpe5022_FILE_NAME_LENGTH_MAX).

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The specified file name is too long.
hpe5022_ERROR_NSUP_HAMP	Error occurred while loading and parsing
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has already been turned on. Turn it off before executing this function. See the "hpe5022_driveState" function.
hpe5022_ERROR_NSUP_FUNC	The E5035A is installed.

See Also "hpe5022_headAmpInfoFile_Q" on page 293

hpe5022_headAmpInfoFile_Q

C Syntax

ViStatus hpe5022_headAmpInfoFile_Q(ViSession id, ViPString bdFileName);

Visual Basic Syntax

hpe5022_headAmpInfoFile_Q(ByVal id As Long, ByRef bdFileName As String) As Long

Description

This function returns the selected head amplifier information file.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bdFileName

  Description: Returns the file name specified by the “hpe5022_headAmpInfoFile” function. Even if the head amplifier with EEPROM is installed, the file name specified by the “hpe5022_headAmpInfoFile” function is returned. The maximum length of the string is defined by the (hpe5022_FILE_NAME_LENGTH_MAX).

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_headAmpInfoFile” on page 291

hpe5022_headAmpIcName_Q

C Syntax

ViStatus hpe5022_headAmpIcName_Q(ViSession id, ViPString icName, ViPString icDesc);

Visual Basic Syntax

hpe5022_headAmpIcName_Q(ByVal id As Long, ByVal icName As String, ByVal icDesc As String) As Long

Description

This function returns the head amp IC name and its description. This function is available when your system has the E5035B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- icName

  Description: Returns the head amplifier IC name. The maximum length of the string is defined by the (hpe5022_MESSAGE_LENGTH_MAX).

  Direction: OUT

- icDesc

  Description: Returns the description for the head amplifier IC. The maximum length of the string is defined by the (hpe5022_MESSAGE_LENGTH_MAX).

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_NSUP_FUNC	The E5035A is installed.

See Also

Port Configuration Function

This section describes the functions to determine trigger ports configuration.

hpe5022_triggerPort

C Syntax

ViStatus hpe5022_triggerPort(ViSession id,ViInt16 inst, ViInt16 trigPort);

Visual Basic Syntax

hpe5022_triggerPort(ByVal id As Long, ByVal inst As Integer, ByVal trigPort As Integer) As Long

Description

This function controls the trigger port between E5035A and other instruments. When cable connection particularly BNC trigger is set to default at the factory, you need not change this configuration.

When 3GHz Spectrum Analyzer (E4402B) is installed, the E5023A system uses two trigger port internally: TTLTRG* for E5040A and EXT1 (BNC) for 3GHz Spectrum Analyzer. This function can configure only TTLTRG* port for E5040A in this case. The trigger port for the 3GHz Spectrum Analyzer is fixed to EXT1 port and cannot be configured.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- inst

  Description Specifies the instrument.

  Direction IN

  Values

Name	Value	Description
hpe5022_INSTR_DGEN	2	Data Generator Module (Agilent E5037A/B/C/D)
hpe5022_INSTR_PARA	3	Parametric Module (Agilent E5038A/B)
hpe5022_INSTR_DSO	4	Oscilloscope (Option)
hpe5022_INSTR_SA	5	Spectrum Analyzer

- trigPort

  Description Specifies the trigger port and TTL trigger lines. the TTLTRG0 through the TTLTRG7 are the lines of the VXI

bus. EXT0 and EXT1 are the designated ports oscilloscope and spectrum analyzer respectively.

When the ‘inst’ parameter is set to either “hpe5022_INSTR_DSO” or “hpe5022_INSTR_SA”, the “hpe5022_TRIG_PORT_TTLTRG0” to “hpe5022_TRIG_PORT_TTLTRG7” can not be selected.

Direction IN

Values

Name	Value	Description
hpe5022_TRIG_PORT_TTLTRG0	0	TTLTRG0
hpe5022_TRIG_PORT_TTLTRG1	1	TTLTRG1
hpe5022_TRIG_PORT_TTLTRG2	2	TTLTRG2
hpe5022_TRIG_PORT_TTLTRG3	3	TTLTRG3
hpe5022_TRIG_PORT_TTLTRG4	4	TTLTRG4
hpe5022_TRIG_PORT_TTLTRG5	5	TTLTRG5
hpe5022_TRIG_PORT_TTLTRG6	6	TTLTRG6
hpe5022_TRIG_PORT_TTLTRG7	7	TTLTRG7
hpe5022_TRIG_PORT_EXT0	8	EXT0
hpe5022_TRIG_PORT_EXT1	9	EXT1

Preset values — When ‘inst’ is “hpe5022_INSTR_DGEN”:

Name	Value
hpe5022_TRIG_PORT_TTLTRG0	0

— When ‘inst’ is “hpe5022_INSTR_PARA”:

Name	Value
hpe5022_TRIG_PORT_TTLTRG1	1

— When ‘inst’ is “hpe5022_INSTR_DSO”:

Name	Value
hpe5022_TRIG_PORT_EXT0	8

— When ‘inst’ is “hpe5022_INSTR_SA”:

Name	Value
hpe5022_TRIG_PORT_EXT1	9

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘inst’ and/or ‘trigPort’ is out of range.
hpe5022_ERROR_INV_TRIG_PORT	Invalid trigger port is selected as the parameter ‘trigPort’.

See Also

“hpe5022_triggerPort_Q” on page 298

hpe5022_triggerPort_Q

C Syntax

ViStatus hpe5022_triggerPort_Q(ViSession id,ViPInt16 inst, ViPInt16 trigPort);

Visual Basic Syntax

hpe5022_triggerPort_Q(ByVal id As Long, ByVal inst As Integer, ByRef trigPort As Integer) As Long

Description

This function returns the configuration of the trigger port or TTL trigger line.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- inst

  Description — Specifies the instrument type.

  Direction — IN

  Values — Same as ‘inst’ parameter in the “hpe5022_triggerPort” function.

- trigPort

  Description — Returns the trigger port or TTL trigger line.

  Direction — OUT

  Values — Same as ‘trigPort’ parameter in the “hpe5022_triggerPort” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘inst’ is out of range.

See Also

“hpe5022_triggerPort” on page 295

hpe5022_triggerImmediate

C Syntax

ViStatus hpe5022_triggerImmediate(ViSession id,ViInt16 trigPort, ViReal64 delay, ViReal64 aperture);

Visual Basic Syntax

hpe5022_triggerImmediate(ByVal id As Long, ByVal trigPort As Integer, ByVal delay As Double, ByVal aperture As Double) As Long

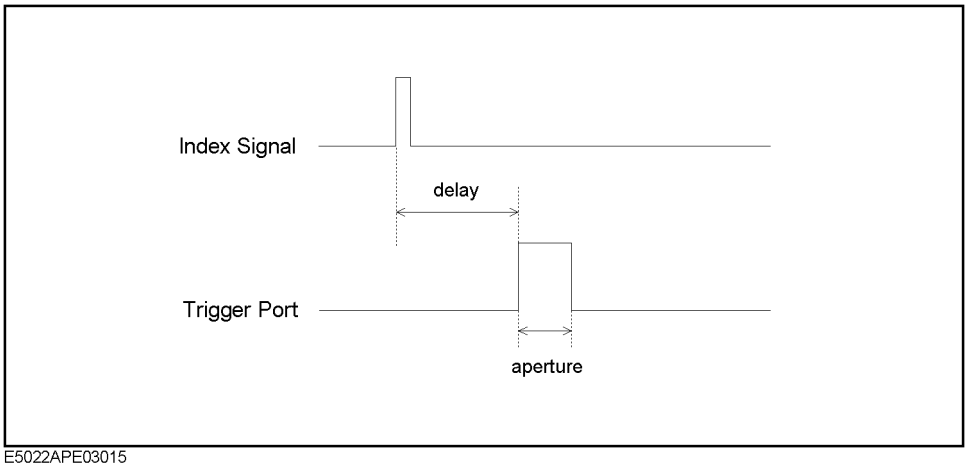
Description

This function produces trigger on the specified trigger port or TTL trigger line. The trigger is synchronized with the spinstand index pulse and occurs just once after the function is executed. The sum of delay time and aperture is limited to a value as defined by the equation below:

Equation 3-1 Delay and Aperture Limit

$$Delay + Aperture \leq \frac{60}{SpindleSpeed} - 400 \times 10^{-6}$$

Figure 3-11 Delay and Aperture



E5022APE03015

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- trigPort

  Description: Specifies the trigger port and TTL trigger line.

  Direction: IN

  Values: Same as ‘trigPort’ parameter in the “hpe5022_triggerPort” function.

- delay

  Description Specifies the delay time from the index pulse.

  Direction IN

  Unit Second

  Values

Name	Value
hpe5022_EXT_TRIG_DEL_MIN	0

- aperture

  Description Specifies the aperture time of the trigger. The polarity is positive. When ‘trigPort’ parameter is set to one of the TTLTRG lines i.e, “hpe5022_TRIG_PORT_TTLTRG0” to “hpe5022_TRIG_PORT_TTLTRG7”, the aperture time will have no effect when trigPort is specified as TTLTRGx, since it has no capability to control the aperture time.

  Direction IN

  Unit Second

  Values

Name	Value
hpe5022_EXT_TRIG_APER_MIN	1.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘trigPort’ is out of range.

See Also

“hpe5022_triggerPort” on page 295

Output State Function

This section describes the function used to enable/disable the output of write current, sense stimulus, write gate and write data.

hpe5022_outputState

C Syntax

ViStatus hpe5022_outputState(ViSession id, ViBoolean state);

Visual Basic Syntax

hpe5022_outputState(ByVal id As Long, ByVal state As Integer) As Long

Description

This function enables/disables the output of write current, sense stimulus, write gate and write data outputs.

They are controlled automatically when the provided measurement function such as “hpe5022_measureParametric” is used. In general, this function is seldom used.

Do not attach or remove the cassette of AOC when the “hpe5022_outputState” function is turned on. This function turns on the head amp power at the same time.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- state

  Description: Sets the output status.

  Direction: IN

  Preset value: 0 (VI_FALSE)

  Values

Name	Value	Description
VI_TRUE	1	Output Enable
VI_FALSE	0	Output Disable

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_HARD_AMP	Hardware error (Head Amp) is detected.

See Also “hpe5022_outputState_Q” on page 303

hpe5022_outputState_Q

C Syntax

ViStatus hpe5022_outputState_Q(ViSession id, ViPBoolean state);

Visual Basic Syntax

hpe5022_outputState_Q(ByVal id As Long, ByRef state As Integer) As Long

Description

This function returns the output state.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- state

  Description: Returns the output state.

  Direction: OUT

  Values: Same as ‘state’ in the “hpe5022_outputState” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_outputState” on page 301

External Write Data Source Function

This section describes the function of the external data source.

hpe5022_writeDataSource

C Syntax

ViStatus hpe5022_writeDataSource(ViSession id, ViInt16 dataSource);

Visual Basic Syntax

hpe5022_writeDataSource(ByVal id As Long, ByVal dataSource As Integer) As Long

Description

This function allows you to use the external write data source.

When hpe5022_WRITE_DATA_SOURCE_EXTERNAL is selected, the system uses external data source and BNC connector on front panel is selected as write gate. In case of the E5037A/B, the data into the Ext Data In connectors are used as a write data. In case of the E5037C/D, the data into the port 2 on the E5043A/B/C are used as a write data. The E5022/E5023 measurement functions implicitly control write data source.

When hpe5022_WRITE_DATA_SOURCE_INTERNAL is selected, the system uses the signal generated by the E5037A/B/C/D as a write data, and VXI backplane is selected as write gate.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataSource

  Description: Selects the write data source.

  Direction: IN

  Preset value: 0 (hpe5022_WRITE_DATA_SOURCE_INTERNAL)

  Values

Name	Value	Description
hpe5022_WRITE_DATA_SOURCE_EXTERNAL	1	Write Date Source External
hpe5022_WRITE_DATA_SOURCE_INTERNAL	0	Write Date Source Internal

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataSource’ is out of range.

See Also

“hpe5022_writeDataSource_Q” on page 306

hpe5022_writeDataSource_Q

C Syntax

ViStatus hpe5022_writeDataSource_Q(ViSession id, ViPInt16 dataSource);

Visual Basic Syntax

hpe5022_writeDataSource_Q(ByVal id As Long, ByRef dataSource As Integer) As Long

Description

This function returns the status of the external write data source.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataSource

  Description: Return the status of the write data source.

  Direction: OUT

  Preset value: 0

  Values: Same as the “hpe5022_writeDataSource” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_writeDataSource” on page 304

Spinstand Controller J2 Port I/O Function

This section describes the function allows you to control the J2 I/O port on the rear panel of the spinstand controller.

hpe5022_spinStand_ttlOutput

C Syntax

ViStatus hpe5022_spinStand_ttlOutput(ViSession id, ViInt32 pinNo, ViInt32 level);

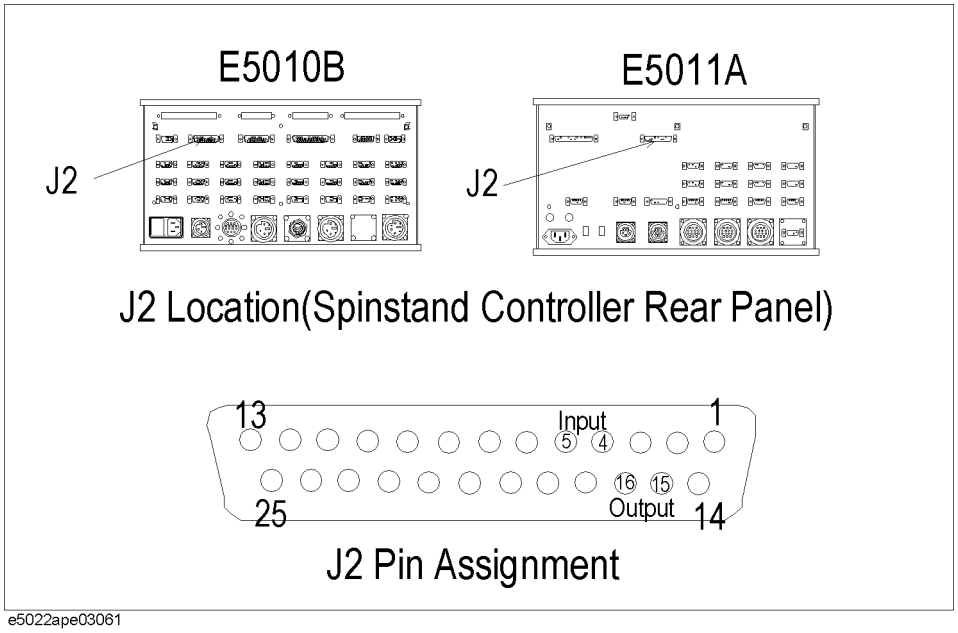
Visual Basic Syntax

hpe5022_spinStand_ttlOutput(ByVal id As Long, ByVal pinNo As Integer, ByVal level As Long) As Long

Description

This function controls the high/low output of the J2 I/O port on the rear panel of the spinstand controller. The pins of No. 15 and 16 are available for users as output. The other output pins are used for the system. The location of the J2 port and its pin assignment are shown in the Figure 3-12.

Figure 3-12 J2 Location and Pin Assignment



e5022ape03061

In case of the E5011A, the cable is already connected on the J2 connector and some pins are used for the system. When you use the output/input pins, you have to modify the cable in order to use the output/input signal and do not disconnect the other system used pins. The pin assignment of the J2 connector for the E5011A is

shown in the Table 3-1.

Table 3-1 E5011A J2 Pin Assignment

Pin No.	Description	Pin No.	Description
1	GND (Used)	14	GND (Used)
2	TTL Input 0 (Used)	15	TTL Output 0
3	TTL Input 1 (Used)	16	TTL Output 1
4	TTL Input 2	17	TTL Output 2 (Used)
5	TTL Input 3	18	TTL Output 3 (Used)
6	TTL Input 4 (Used)	19	TTL Output 4 (Used)
7	TTL Input 4 (Used)	20	TTL Output 4 (Used)
8	TTL Input 6 (Used)	21	TTL Output 6 (Used)
9	TTL Input 7 (Used)	22	TTL Output 7 (Used)
10	N/C	23	GND (Used)
11	N/C	24	GND
12	N/C	25	GND
13	+5 V (Used)		

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- pinNo

  Description Selects the pin number.

  Direction IN

  Values

Name	Value	Description
hpe5022_SS_TTLIO_PIN15	15	Pin 15
hpe5022_SS_TTLIO_PIN16	16	Pin 16

- level

  Description Sets the level for the specified pin.

  Direction IN

  Preset value 0 (hpe5022_TTL_LOW)

Values

Name	Value	Description
hpe5022_TTL_HIGH	1	TTL High (More than 2.0 V)
hpe5022_TTL_LOW	0	TTL Low (less than 0.8 V)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_UNAVAILABLE_PINNO	The selected pin number is valid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘level’ is out of range.

See Also

“hpe5022_spinStand_ttlOutput_Q” on page 310

“hpe5022_spinStand_ttlInput_Q” on page 312

hpe5022_spinStand_ttlOutput_Q

C Syntax

ViStatus hpe5022_spinStand_ttlOutput_Q(ViSession id, ViInt32 pinNo, ViPInt32 level);

Visual Basic Syntax

hpe5022_spinStand_ttlOutput_Q(ByVal id As Long, ByVal pinNo As Integer, ByRef level As Integer) As Long

Description

This function returns the high/low status of the No. 15 or 16 pin on the J2 I/O port.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pinNo

  Description: Selects the pin number.

  Direction: IN

  Values

Name	Value	Description
hpe5022_SS_TTLIO_PIN15	15	Pin 15
hpe5022_SS_TTLIO_PIN16	16	Pin 16

- level

  Description: Returns the current setting for the specified pin.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_TTL_HIGH	1	TTL High (More than 2.0 V)
hpe5022_TTL_LOW	0	TTL Low (less than 0.8 V)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_UNAVAILABLE_PINNO	The selected pin number is valid.

See Also “hpe5022_spinStand_ttlOutput_Q” on page 310

hpe5022_spinStand_ttlInput_Q

C Syntax

ViStatus hpe5022_spinStand_ttlInput_Q(ViSession id, ViInt32 pinNo, ViPInt32 level);

Visual Basic Syntax

hpe5022_spinStand_ttlInput_Q(ByVal id As Long, ByVal pinNo As Integer, ByRef level As Long) As Long

Description

This function queries the high/low status of the J2 I/O port on the rear panel of the spinstand controller. The pins of No. 4 and 5 are available for users as the input. The other input pins are used for the system. The location of the J2 port and its pin assignment are shown in the Figure 3-12.

In case of the E5011A, the cable is already connected on the J2 connector and some pins are used for the system. When you use the output/input pins, you have to modify the cable in order to use the output/input signal, and do not disconnect the other system used pins. The pin assignment of the J2 connector for the E5011A is shown in the Table 3-1.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pinNo

  Description: Selects the pin number.

  Direction: IN

  Values

Name	Value	Description
hpe5022_SS_TTLIO_PIN4	4	Pin 4
hpe5022_SS_TTLIO_PIN5	5	Pin 5

- level

  Description: Returns the level of the specified pin.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_TTL_HIGH	1	TTL High (More than 2.0 V)
hpe5022_TTL_LOW	0	TTL Low (less than 0.8 V)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_UNAVAILABLE_PINNO	The selected pin number is valid.

See Also

“hpe5022_spinStand_ttlOutput” on page 307

Shroud Cover Function

This section describes the function related with a shroud cover.

hpe5022_spindleShroud_Q

C Syntax

ViStatus hpe5022_spindleShroud_Q(ViSession id, ViPBoolean openStat);

Visual Basic Syntax

hpe5022_spindleShroud_Q(ByVal id As Long, ByRef openStat as Integer) As Long

Description

This function returns the status of the spindle shroud.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- openStat

  Description: Returns the status of the spindle shroud.

  Direction: OUT

  Values

Name	Value	Description
VI_TRUE	1	Spindle shroud is open.
VI_FALSE	0	Spindle shroud is close.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

hpe5022_checkShroudForm

C Syntax

ViStatus hpe5022_checkShroudForm(ViSession id, ViBoolean check_on_off);

Visual Basic Syntax

hpe5022_checkShroudForm(ByVal id As Long, ByVal check_on_off As Integer) As Long

Description

This function activates the collusion check between the shroud and cassette. In the case of ON, the hpe5022_ERROR_INV_POSITION error occurs and the head is not moved when the specified head position causes the collusion of the shroud and cassette. In the case of OFF, there is no check and it might cause the collution.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- check_on_off

  Description: Selects the status of the collusion check.

  Direction: IN

  Preset Value: VI_TRUE

  Values

Name	Value	Description
VI_TRUE	1	Collusion check is ON
VI_FALSE	0	Collusion check is OFF

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_checkShroudForm_Q” on page 316

hpe5022_checkShroudForm_Q

C Syntax

ViStatus hpe5022_checkShroudForm_Q(ViSession id, ViPBoolean check_on_off);

Visual Basic Syntax

hpe5022_checkShroudForm_Q(ByVal id As Long, ByRef check_on_off As Integer) As Long

Description

This function returns the status of the collusion check between the shroud and cassette.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- check_on_off

  Description: Returns the status of the collusion check.

  Direction: OUT

  Values

Name	Value	Description
VI_TRUE	1	Collusion check is ON
VI_FALSE	0	Collusion check is OFF

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_checkShroudForm” on page 315

hpe5022_shroudForm

C Syntax

ViStatus hpe5022_shroudForm(ViSession id, ViReal64 r, ViReal64 offset);

Visual Basic Syntax

hpe5022_shroudForm(ByVal id As Long, ByVal r As Double, ByVal offset As Double) As Long

Description

This function allows you to use the customer designed shroud. Define the shroud dimension when the customer designed shroud is used. This dimension is used to avoid a collision between the shroud and cassette.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- r

  Description: Specifies the distance between the spindle center and shroud inner surface.

  Direction: IN

  Preset Value: 0.06833 (This is a distance of the standard shroud)

  Unit: Meter

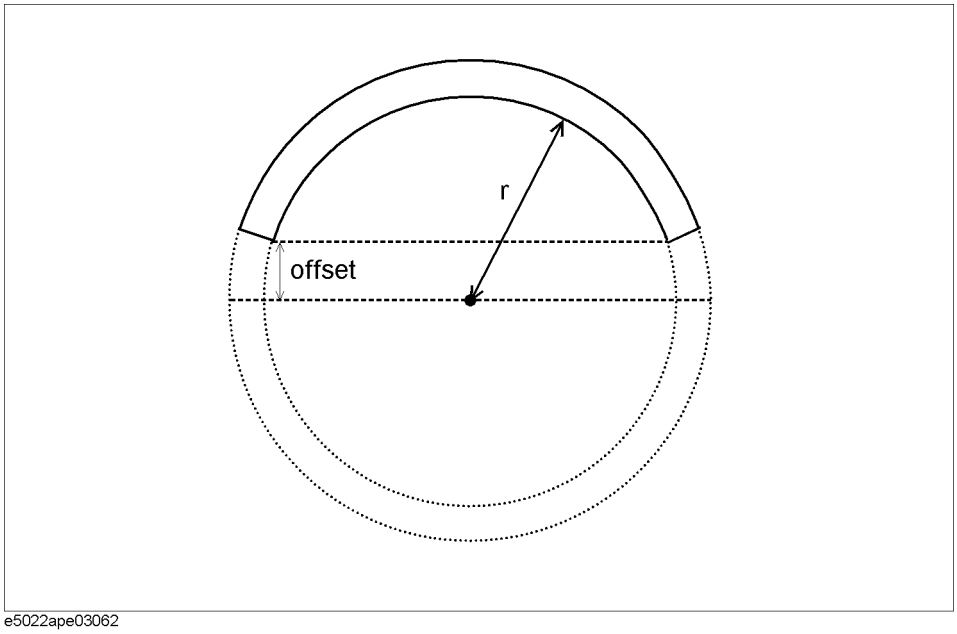
  Values

Name	Value
hpe5022_SHROUD_RAD_MIN	0.01
hpe5022_SHROUD_RAD_MAX	1.00

- offset

  Description: Specifies the distance shown in the figure.

Figure 3-13 Shroud Form Parameter



e5022ape03062

Direction IN

Preset Value 0.00635 (This is a distance of the standard shroud)

Unit Meter

Values

Name	Value
hpe5022_SHROUD_OFFS_MIN	0
hpe5022_SHROUD_OFFS_MAX	1.00

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_checkShroudForm” on page 315

“” on page 318

hpe5022_shroudForm_Q

C Syntax

ViStatus hpe5022_shroudForm_Q(ViSession id, ViPReal64 r, ViPReal64 offset);

Visual Basic Syntax

hpe5022_shroudForm_Q(ByVal id As Long, ByRef r As Double, ByRef offset As Double) As Long

Description

This function returns the shroud dimension.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- r

  Description: Specifies the distance between the spindle center and shroud inner surface.

  Direction: OUT

  Unit: Meter

- offset

  Description: Specifies the distance shown in the figure.

  Direction: IN

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_shroudForm” on page 317

Spectrum Analyzer Display Function

This section describes the function that controls the display status of the spectrum analyzer. Normally, during measurement nothing is displayed on the screen of the spectrum analyzer due to screen save. This function allows the user to view the waveform on the screen of the 4395A spectrum analyzer. This function can not be used for E5040A because E5040A has no display.

hpe5022_spectrumDisplayState

C Syntax

hpe5022_spectrumDisplayState(ViSession id, ViBoolean dispOn);

Visual Basic Syntax

hpe5022_spectrumDisplayState(ByVal id As Long, ByVal dispOn As Integer) As Long

Description

This function controls the display state of the 4396A spectrum analyzer.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dispOn

  Description: Sets the display to viewable state.

  Direction: IN

  Preset Value: VI_FALSE

  Values

Name	Value	Description
VI_TRUE	1	Display is ON
VI_FALSE	0	Display is OFF

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_INSTR_ERROR_NSUP_CONF	This function does not support E5040A.

See Also “hpe5022_spectrumDisplayState_Q” on page 322

hpe5022_spectrumDisplayState_Q

C Syntax

hpe5022_spectrumDisplayState_Q(ViSession id, ViPBoolean dispOn);

Visual Basic Syntax

hpe5022_spectrumDisplayState_Q(ByVal id As Long, ByRef dispOn As Integer) As Long

Description

This function returns display state of the 4395A spectrum analyzer.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dispOn

  Description: Returns the display state of the spectrum analyzer.

  Direction: OUT

  Values: Same as ‘dispOn’ in the “hpe5022_spectrumDisplayState” function

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectrumDisplayState” on page 320

Write Function

This section describes the functions to write data.

hpe5022_writeCurrent

C Syntax

ViStatus hpe5022_writeCurrent(ViSession id,ViReal64 current);

Visual Basic Syntax

hpe5022_writeCurrent(ByVal id As Long, ByVal current As Double) As Long

Description

This function specifies the write current applied to the write head.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Specifies the write current. The limit of write current is dependent on the head amplifier installed in the system. The upper and lower limit of the write current can be queried by the “hpe5022_writeCurrentRange_Q” function.

  Direction: IN

  Preset value: 0

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘current’ is out of range.

See Also

“hpe5022_writeCurrent_Q” on page 324

“hpe5022_writeCurrentRange_Q” on page 325

hpe5022_writeCurrent_Q

C Syntax

ViStatus hpe5022_writeCurrent_Q(ViSession id,ViPReal64 current);

Visual Basic Syntax

hpe5022_writeCurrent_Q(ByVal id As Long, ByRef current As Double) As Long

Description

This function returns the specified write current.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Returns the specified write current.

  Direction: OUT

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_writeCurrent” on page 323

hpe5022_writeCurrentRange_Q

C Syntax

ViStatus hpe5022_writeCurrentRange_Q(ViSession id, ViPReal64 currMin, ViPReal64 currMax);

Visual Basic Syntax

hpe5022_writeCurrentRange_Q(ByVal id As Long, ByRef currMin As Double, ByRef currMax As Double) As Long

Description

This function returns the range of the write current. The write current range depends on the installed head amplifier. E5022A/B senses the head amplifier and sets the range automatically.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- currMin

  Description: Returns the minimum value that the write current can be set.

  Direction: OUT

  Unit: Ampere

- currMax

  Description: Returns the maximum value that the write current can be set.

  Direction: OUT

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_writeCurrent” on page 323

hpe5022_write

C Syntax

ViStatus hpe5022_write(ViSession id, ViReal64 offset);

Visual Basic Syntax

hpe5022_write(ByVal id As Long, ByVal offset As Double) As Long

Description

This function executes write sequence. The write sequence is as follows.

1. Set the write current specified by the “hpe5022_writeCurrent” function.
2. Move the head to the write track offset position.
3. Write the data pattern which is specified by “hpe5022_selectPattern”.

As functions which execute measurement, such as the “hpe5022_measureParametric” function, performs the write sequence automatically, use this function when you want to write only.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the write track offset position from the center of the track.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is narrowed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -6.0×10^{-6}, respectively. See the track offset compensation in the operation manual.

  Direction: IN

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q ” function.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_driveState” on page 209

“hpe5022_writeCurrent” on page 323

hpe5022_initiateFreeRunWrite

C Syntax

ViStatus hpe5022_initiateFreeRunWrite(ViSession id);

Visual Basic Syntax

hpe5022_initiateFreeRunWrite(ByVal id As Long) As Long

Description

This function writes the data pattern continuously. The write current, data pattern are specified by the “hpe5022_writeCurrent” and “hpe5022_selectPattern” function, respectively. To stop write, use the “hpe5022_abortFreeRunWrite” function. However, writing is aborted when any function is called.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_writeCurrent” on page 323

“hpe5022_abortFreeRunWrite” on page 329

hpe5022_abortFreeRunWrite

C Syntax

ViStatus hpe5022_abortFreeRunWrite(ViSession id);

Visual Basic Syntax

hpe5022_abortFreeRunWrite(ByVal id As Long) As Long

Description

This function aborts the write-freerun function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_initiateFreeRunWrite” on page 328

hpe5022_writeMultiple

C Syntax

ViStatus hpe5022_writeMultiple(ViSession id, ViReal64 offset, ViInt16 counts);

Visual Basic Syntax

hpe5022_writeMultiple(ByVal id As Long, ByVal offset As Double, ByVal counts As Integer) As Long

Description

This function writes the data for several revolution. The write sequence is as follows.

1. Set the write current specified by the “hpe5022_writeCurrent” function.
2. Move the head to the write track offset position.
3. Write the data pattern which is specified by “hpe5022_selectPattern” for the entire track.
4. Repeat the step 3 ‘counts’ times if it is more than 1.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the write offset position from track center.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction: IN

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

  Unit: Meter

- counts

  Description: Specifies the number of repeat for write.

  Direction: IN

Values

Name	Value
hpe5022_WRITE_COUN_MIN	1
hpe5022_WRITE_COUN_MAX	1000000

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’, and/or ‘counts’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_write” on page 326

hpe5022_setupWriteMultiple

C Syntax

ViStatus hpe5022_setupWriteMultiple(ViSession id, ViReal64 offset, ViInt16 counts, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupWriteMultiple(ByVal id As Long, ByVal offset As Double, ByVal counts As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the write-multiple sequence to the specified test identifier. See the “hpe5022_writeMultiple” function about the details of the sequence. This function does not execute the write-multiple sequence. The write-multiple is executed by the “hpe5022_measure” function with its test identifier specified in this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- offset

  Description — Specifies the write offset position from track center.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction — IN

  Values — Same as ‘offset’ in the “hpe5022_writeMultiple” function.

  Unit — Meter

- counts

  Description — Specifies the number of repeat for write.

  Direction — IN

  Values — Same as ‘counts’ in the “hpe5022_writeMultiple” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the write-multiple by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’, and/or ‘counts’ is out of range.

See Also

“hpe5022_writeMultiple” on page 330

“hpe5022_measure” on page 387

hpe5022_writeReferenceDcr

C Syntax

ViStatus hpe5022_writeReferenceDcr(ViSession id, ViReal64 refDcr);

Visual Basic Syntax

hpe5022_writeReferenceDcr(ByVal id As Long, ByVal refDcr As Double) As Long

Description

This function specifies the DC resistance of the write head in Ω. This value is used to apply precise write current to the write head.

Parameters

- id

  Description: Specifies the system identifier. This is given by the “hpe5022_init” function.

  Direction: IN

- refDcr

  Description: Specifies the DC resistance of the write head.

  Direction: IN

  Values

Name	Value
hpe5022_WRITE_REFERENCE_DCR_MIN	0.1
hpe5022_WRITE_REFERENCE_DCR_MAX	1×10^{3}

  Preset Value: 10

  Unit: Ω

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘refDcr’ is out of range.

See Also

“hpe5022_writeReferenceDcr_Q” on page 335

hpe5022_writeReferenceDcr_Q

C Syntax

ViStatus hpe5022_writeReferenceDcr_Q(ViSession id, ViPReal64 refDcr);

Visual Basic Syntax

hpe5022_writeReferenceDcr_Q(ByVal id As Long, ByRef refDcr As Double) As Long

Description

This function returns the DC resistance of the write head currently set in Ω.

Parameters

- id

  Description: Specifies the system identifier. This is given by the “hpe5022_init” function.

  Direction: IN

- refDcr

  Description: Returns the DC resistance of the write head.

  Direction: OUT

  Values

Name	Value
hpe5022_WRITE_REFERENCE_DCR_MIN	0.1
hpe5022_WRITE_REFERENCE_DCR_MAX	1×10^3

  Unit: Ω

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_writeReferenceDcr” on page 334

Sense Current Function

This section describes the function to set a sense current.

hpe5022_senseCurrent

C Syntax

ViStatus hpe5022_senseCurrent(ViSession id, ViReal64 current);

Visual Basic Syntax

hpe5022_senseCurrent(ByVal id As Long, ByVal current As Double) As Long

Description

This function specifies the sense current (MR bias current) applied to the read head. When the stimulus mode in the “hpe5022_senseStimulusMode” is set at any value except hpe5022_SENSE_STIMULUS_CURRENT, this function can not be executed and an error is returned.

When the mode in the “hpe5022_senseStimulusMode” function is set at hpe5022_SENSE_STIMULUS_CURRENT, both the “hpe5022_senseCurrent” and “hpe5022_senseStimulus” functions can change the sense current.

Parameters

- id

  Description: Specifies the system identifier. This is given by the “hpe5022_init” function.

  Direction: IN

- current

  Description: Specifies the sense current. The limit value of the sense current is dependent on the installed head amplifier. The upper and lower limit values of the sense current can be queried by the “hpe5022_senseCurrentRange_Q” function.

  Direction: IN

  Preset value: ‘curMin’ of “hpe5022_senseCurrentRange_Q”

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘current’ is out of range.

Error Code	Description
hpe5022_ERROR_INV_SETUP	The sense stimulus mode in the "hpe5022_senseStimulusMode" is not set at hpe5022_SENSE_STIMULUS_CURRENT.

See Also

"hpe5022_senseCurrent_Q" on page 338

"hpe5022_senseCurrentRange_Q" on page 339

"hpe5022_senseStimulusMode" on page 343

hpe5022_senseCurrent_Q

C Syntax

ViStatus hpe5022_senseCurrent_Q(ViSession id, ViPReal64 current);

Visual Basic Syntax

hpe5022_senseCurrent_Q(ByVal id As Long, ByRef current As Double) As Long

Description

This function returns the specified sense current. When the stimulus mode in the “hpe5022_senseStimulusMode” is set at any value except hpe5022_SENSE_STIMULUS_CURRENT, this function can not be executed and an error is returned.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Returns the specified sense current.

  Direction: OUT

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_SETUP	The sense stimulus mode in the “hpe5022_senseStimulusMode” is not set at hpe5022_SENSE_STIMULUS_CURRENT.

See Also

“hpe5022_senseCurrent” on page 336

“hpe5022_senseStimulusMode” on page 343

hpe5022_senseCurrentRange_Q

C Syntax

ViStatus hpe5022_senseCurrentRange_Q(ViSession id, ViPReal64 currMin, ViPReal64 currMax);

Visual Basic Syntax

hpe5022_senseCurrentRange_Q(ByVal id As Long, ByRef currMin As Double ByRef currMax As Double) As Long

Description

This function returns the range of the sense current.The sense current range depends on the installed head amplifier.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- currMin

  Description: Returns the minimum value that the sense current can be set.

  Direction: OUT

  Unit: Ampere

- currMax

  Description: Returns the maximum value that the sense current can be set.

  Direction: OUT

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_senseCurrent” on page 336

hpe5022_senseCurrentPolarity

C Syntax

ViStatus hpe5022_senseCurrentPolarity(ViSession id, ViInt16 pol);

Visual Basic Syntax

hpe5022_senseCurrentPolarity (ByVal id As Long, ByVal pol As Integer) As Long

Description

This function specifies the sense current polarity during a write operation. You can set at reverse and/or off only if the installed head amplifier has the capability.

The setting of this function affects the one of the “hpe5022_senseCurrentPolarity” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- pol

  Description — Specifies the sense current polarity during a write operation. When “hpe5022_SENS_CURR_POL_REV” is selected, the direction of sense current is reversed during a write sequence. When “hpe5022_SENS_CURR_POL_OFF” is selected, the sense current is set to zero during a write sequence.

  This setting is valid for all of measurement includes stability measurements.

  Direction — IN

  Values

Name	Value	Description
hpe5022_SENS_CURR_POL_NORM	0	Same polarity
hpe5022_SENS_CURR_POL_REV	1	Reverse polarity
hpe5022_SENS_CURR_POL_OFF	2	Off Mode

  Preset Value — hpe5022_SENS_CURR_POL_NORM

Return Values

Completion Code	Description
VI_SUCCESS	No error.

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘pol’ is out of range.
hpe5022_ERROR_NSUP_SENS_CURR_POL	Reverse and/or off sense current modes are not supported.

See Also

“hpe5022_stabilityConfig” on page 511

“hpe5022_stabilityAbsSenseModeConfig” on page 537

“hpe5022_senseStimulusMode” on page 343

“hpe5022_senseStimulusPolarity” on page 352

hpe5022_senseCurrentPolarity_Q

C Syntax

ViStatus hpe5022_senseCurrentPolarity_Q(ViSession id, ViPInt16 pol);

Visual Basic Syntax

hpe5022_senseCurrentPolarity_Q Lib "hpe5022_32.dll" (ByVal id As Long, ByRef pol As Integer) As Long

Description

This function returns the sense current polarity during a write sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pol

  Description: Returns the sense current polarity during a write sequence.

  Direction: OUT

  Values: Same as ‘pol’ in the “hpe5022_senseCurrentPolarity” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.

See Also

“hpe5022_senseCurrentPolarity” on page 340

Sense Stimulus Function

This section describes the function to set a sense stimulus.

hpe5022_senseStimulusMode

C Syntax

ViStatus hpe5022_senseStimulusMode(ViSession id, ViInt16 mode);

Visual Basic Syntax

hpe5022_senseStimulusMode(ByVal id As Long, ByVal mode As Integer) As Long

Description

This function allows you to set a MR bias type at either current, voltage, or power. The value of the MR bias is set by the “hpe5022_senseStimulus” function. You can set at voltage and/or power only if the installed head amplifier has the capability.

When the mode is set at hpe5022_SENSE_STIMULUS_CURRENT, both the “hpe5022_senseCurrent” and “hpe5022_senseStimulus” functions can change the sense current.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Selects the sense stimulus mode.

  Direction: IN

  Preset value: When the installed head amplifier has the current mode, hpe5022_SENSE_STIMULUS_CURRENT is set as default.

  When it does not have the current mode, hpe5022_SENSE_STIMULUS_VOLTAGE is set as default.

  Values

Name	Value	Description
hpe5022_SENSE_STIMULUS_CURRENT	0	Current
hpe5022_SENSE_STIMULUS_VOLTAGE	1	Voltage
hpe5022_SENSE_STIMULUS_POWER	2	Power

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘mode’ is out of range.
hpe5022_ERROR_NSUP_CONF	The installed head amplifier does not support this function.
pe5022_ERROR_CAL_FAILED	There is something wrong in the calibration data. Contact to the Agilent office.

See Also

“hpe5022_senseStimulusMode_Q” on page 345

“hpe5022_senseStimulus” on page 346

“hpe5022_senseCurrent” on page 336

hpe5022_senseStimulusMode_Q

C Syntax

ViStatus hpe5022_senseStimulusMode_Q(ViSession id, ViPInt16 mode);

Visual Basic Syntax

hpe5022_senseStimulusMode_Q(ByVal id As Long, ByRef mode As Integer) As Long

Description

This function returns the setting of a MR bias mode.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Returns the sense stimulus mode.

  Direction: OUT

  Preset value: hpe5022_SENSE_STIMULUS_CURRENT

  Values: Same as ‘mode’ in the “hpe5022_senseStimulusMode” function

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_senseStimulusMode” on page 343

hpe5022_senseStimulus

C Syntax

ViStatus hpe5022_senseStimulus(ViSession id, ViReal64 stimulus);

Visual Basic Syntax

hpe5022_senseStimulus(ByVal id As Long, ByVal stimulus As Double) As Long

Description

This function specifies a value of the MR bias. The “hpe5022_senseStimulusMode” function can select a MR bias type from current, voltage and power. The “hpe5022_senseStimulusRange_Q” function returns the range of the setting.

The value for each mode is memorized. If the mode is changed from the current to the voltage, then changed again from the voltage to the current, the current value is still remained.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- stimulus

  Description: Specifies the sense stimulus value.

  Direction: IN

  Preset value: ‘stimulusMin’ returned by the “hpe5022_senseStimulusRange_Q” function.

  Unit: Depends on the type of “hpe5022_senseStimulusMode”.

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘stimulus’ is out of range.

See Also

“hpe5022_senseStimulusMode” on page 343

“hpe5022_senseStimulusRange_Q” on page 348

hpe5022_senseStimulus_Q

C Syntax

ViStatus hpe5022_senseStimulus_Q(ViSession id, ViPReal64 stimulus);

Visual Basic Syntax

hpe5022_senseStimulus_Q(ByVal id As Long, ByRef stimulus As Double) As Long

Description

This function returns a value of the MR bias.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- stimulus

  Description: Returns the sense stimulus value.

  Direction: OUT

  Unit: Depends on the type of “hpe5022_senseStimulusMode”.

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_senseStimulus” on page 346

hpe5022_senseStimulusRange_Q

C Syntax

ViStatus hpe5022_senseStimulusRange_Q(ViSession id, ViInt16 mode, ViPReal64 stimulusMin, ViPReal64 stimulusMax);

Visual Basic Syntax

hpe5022_senseStimulusRange_Q(ByVal id As Long, ByVal mode As Integer, ByRef stimulusMin As Double, ByRef stimulusMax As Double) As Long

Description

This function returns a range of the MR bias setting. The range depends on the installed head amplifier.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- mode

  Description — Selects the sense stimulus mode

  Direction — IN

  Values

Name	Value	Description
hpe5022_SENSE_STIMULUS_CURRENT	0	Current
hpe5022_SENSE_STIMULUS_VOLTAGE	1	Voltage
hpe5022_SENSE_STIMULUS_POWER	2	Power

- stimulusMin

  Description — Returns the minimum value of the range for the selected mode.

  Direction — OUT

  Unit — Depend on the mode of “hpe5022_senseStimulusMode”.

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

- stimulusMax

  Description — Returns the maximum value of the range for the selected mode.

  Direction — OUT

  Unit — Same as the ‘stimulusMin’

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘mode’ is out of range.

See Also

“hpe5022_senseStimulus” on page 346

hpe5022_senseReferenceDcr

C Syntax

ViStatus hpe5022_senseReferenceDcr(ViSession id, ViReal64 refDcr);

Visual Basic Syntax

hpe5022_senseReferenceDcr(ByVal id As Long, ByVal refDcr As Double) As Long

Description

This function specifies the DC resistance of the read head. The resistance requires in order to apply a precise sense stimulus voltage to the read head. You need to set this when the voltage is selected in the “hpe5022_senseStimulusMode” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- refDcr

  Description — Specifies the DC resistance of the MR head.

  Direction — IN

  Values

Name	Value
hpe5022_SENSE_REFERENCE_DCR_MIN	0.1
hpe5022_SENSE_REFERENCE_DCR_MAX	1×10^3

  Unit — Ohm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘refDcr’ is out of range.

See Also

“hpe5022_senseStimulusMode” on page 343

“hpe5022_senseStimulus” on page 346

hpe5022_senseReferenceDcr_Q

C Syntax

ViStatus hpe5022_senseReferenceDcr_Q(ViSession id, ViPReal64 refDcr);

Visual Basic Syntax

hpe5022_senseReferenceDcr_Q(ByVal id As Long, ByRef refDcr As Double) As Long

Description

This function returns the setting value of the DC resistance of the read head.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- refDcr

  Description: Returns the setting of DC resistance.

  Direction: OUT

  Unit: Ohm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_senseReferenceDcr” on page 350

hpe5022_senseStimulusPolarity

C Syntax

ViStatus hpe5022_senseStimulusPolarity(ViSession id, ViInt16 pol);

Visual Basic Syntax

hpe5022_senseStimulusPolarity(ByVal id As Long, ByVal pol As Integer) As Long

Description

This function specifies the polarity of a MR bias during a write operation. You can set at reverse and/or off only if the installed head amplifier has the capability. The setting of this function affects the one of the “hpe5022_senseCurrentPolarity” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- pol

  Description — Specifies the MR bias polarity during a write operation. When “hpe5022_SENS_STIM_POL_REV” is selected, the direction of MR bias is reversed during a write operation. When “hpe5022_SENS_STIM_POL_OFF” is selected, the sense current is set to zero during a write operation.

  Direction — IN

  Preset value — hpe5022_SENS_STIM_POL_NORM

  Values

Name	Value	Description
hpe5022_SENS_STIM_POL_NORM	0	Same polarity
hpe5022_SENS_STIM_POL_REV	1	Reverse polarity
hpe5022_SENS_STIM_POL_OFF	2	Off Mode

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘pol’ is out of range.
hpe5022_ERROR_NSUP_SENS_CURR_POL	The installed head amplifier does not support this function.

See Also

“hpe5022_senseStimulusMode_Q” on page 345

“hpe5022_senseStimulus” on page 346

“hpe5022_senseCurrentPolarity” on page 340

hpe5022_senseStimulusPolarity_Q

C Syntax

ViStatus hpe5022_senseStimulusPolarity_Q(ViSession id, ViPInt16 pol);

Visual Basic Syntax

hpe5022_senseStimulusPolarity_Q(ByVal id As Long, ByRef pol As Integer) As Long

Description

This function returns the setting of the polarity of a MR bias during write operation.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- pol

  Description — Returns the MR bias polarity during a write sequence.

  Direction — OUT

  Values — Same as ‘pol’ in the “hpe5022_senseStimulusPolarity” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_senseStimulusPolarity” on page 352

Erase Function

This section describes the functions to erase a track with a DC signal.

hpe5022_eraseType

C Syntax

ViStatus hpe5022_eraseType(ViSession id, ViInt16 eraseType);

Visual Basic Syntax

hpe5022_eraseType(ByVal id As Long, ByVal eraseType As Integer) As Long

Description

This function selects the erase type; DC erase positive, DC erase negative and AC erase.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- type

  Description: Specifies the erase type. DC erase positive means that current flows the ports from WD+ to WD- on the head amplifier. The DC erase negative means that current flows the ports from WD- to WD+ on the head amplifier. When AC erase is selected, the AC frequency must be specified by the “hpe5022_eraseAcFrequency” function.

  Direction: IN

  Preset value: 0 (hpe5022_ERASE_DC_NEG)

  Values

Name	Value	Description
hpe5022_ERASE_DC_NEG	0	DC negative erase
hpe5022_ERASE_DC_POS	1	DC positive erase
hpe5022_ERASE_AC	2	AC Erase

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘type’ is out of range.

See Also

“hpe5022_eraseAcFrequency” on page 360

“hpe5022_eraseType_Q” on page 357

hpe5022_eraseType_Q

C Syntax

ViStatus hpe5022_eraseType_Q(ViSession id, ViPInt16 eraseType);

Visual Basic Syntax

hpe5022_eraseType_Q(ByVal id As Long, ByRef eraseType As Integer) As Long

Description

This function returns the specified erase type.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- type

  Description: Returns the specified erase type.

  Direction: OUT

  Values: Same as ‘type’ in the “hpe5022_eraseType” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_eraseType” on page 355

hpe5022_eraseCurrent

C Syntax

ViStatus hpe5022_eraseCurrent(ViSession id, ViReal64 current);

Visual Basic Syntax

hpe5022_eraseCurrent(ByVal id As Long, ByVal current As Double) As Long

Description

This function specifies the erase current applied to the write head.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Specifies the erase current. The limit value of erase current is dependent on the installed head amplifier.

  Direction: IN

  Preset value: 0

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘current’ is out of range.

See Also

“hpe5022_eraseCurrent_Q” on page 359

“hpe5022_writeCurrentRange_Q” on page 325

hpe5022_eraseCurrent_Q

C Syntax

ViStatus hpe5022_eraseCurrent_Q(ViSession id,ViPReal64 current);

Visual Basic Syntax

hpe5022_eraseCurrent_Q(ByVal id As Long, ByRef current As Double) As Long

Description

This function returns the specified erase current.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Returns the specified erase current.

  Direction: OUT

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_eraseCurrent” on page 358

hpe5022_eraseAcFrequency

C Syntax

ViStatus hpe5022_eraseAcFrequency(ViSession id,ViReal64 freq);

Visual Basic Syntax

hpe5022_eraseAcFrequency(ByVal id As Long, ByVal freq As Double) As Long

Description

This function specifies the frequency of an AC erase.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Specifies the AC erase frequency. The maximum limit of this value is returned by the “hpe5022_channelBitRateRange_Q” function.

  Direction: IN

  Preset value: 200×10^6

  Unit: Flux per second

  Values

Name	Value
hpe5022_ERASE_FREQ_MIN	4×10^6

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freq’ is out of range.

See Also

“hpe5022_channelBitRateRange_Q” on page 118

“hpe5022_eraseType” on page 355

“hpe5022_eraseAcFrequency_Q” on page 361

“hpe5022_erase” on page 362

hpe5022_eraseAcFrequency_Q

C Syntax

ViStatus hpe5022_eraseAcFrequency_Q(ViSession id,ViPReal64 freq);

Visual Basic Syntax

hpe5022_eraseAcFrequency_Q(ByVal id As Long, ByRef freq As Double) As Long

Description

This function returns the specified frequency of AC erase.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Returns the specified frequency of AC erase.

  Direction: OUT

  Unit: Flux per second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_eraseAcFrequency” on page 360

hpe5022_erase

C Syntax

ViStatus hpe5022_erase(ViSession id, ViReal64 offset);

Visual Basic Syntax

hpe5022_erase(ByVal id As Long ByVal offset As Double) As Long

Description

This function executes erase. The sequence is as follows:

1. Set the erase current at a specified value by “hpe5022_eraseCurrent”.
2. Move the head to the track offset position from the center of the track.
3. Erase with the current polarity specified by “hpe5022_eraseType”.

As functions which execute measurements such as the “hpe5022_measureParametric” function performs the erase sequence automatically, you may use this function when you want to execute an erase track only.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the write track offset position from the center of the track.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction: IN

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q ” function.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_driveState” on page 209

“hpe5022_eraseType” on page 355

“hpe5022_eraseCurrent” on page 358

hpe5022_eraseMultiple

C Syntax

ViStatus hpe5022_eraseMultiple(ViSession id, ViReal64 offset, ViInt16 counts);

Visual Basic Syntax

hpe5022_eraseMultiple(ByVal id As Long, ByVal offset As Double, ByVal counts As Integer) As Long

Description

This function executes multiple erase on a same track. The sequence is as follows:

1. Set the erase current at a specified value by “hpe5022_eraseCurrent”.
2. Move the head to the track offset position from the center of the track.
3. Set the current polarity specified by “hpe5022_eraseType”.
4. Erase the whole track.
5. Repeat the step 4 ‘counts’ times if it is more than 1.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the erase offset position from track center.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction: IN

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

  Unit: Meter

- counts

  Description: Specifies the number of repeat for erase.

  Direction: IN

Values

Name	Value
hpe5022_WRITE_COUN_MIN	1
hpe5022_WRITE_COUN_MAX	1000000

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’, and/or ‘counts’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_erase” on page 362

hpe5022_setupEraseMultiple

C Syntax

ViStatus hpe5022_setupEraseMultiple(ViSession id, ViReal64 offset, ViInt16 counts, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupEraseMultiple(ByVal id As Long, ByVal offset As Double, ByVal counts As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the erase-multiple sequence to the specified test identifier. See the “hpe5022_eraseMultiple” function about the details of the sequence. This function does not execute the erase-multiple sequence. The erase-multiple is executed by the “hpe5022_measure” function with its test identifier specified in this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- offset

  Description — Specifies the write (erase) offset position from track center.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction — IN

  Values — Same as ‘offset’ in the “hpe5022_eraseMultiple” function.

  Unit — Meter

- counts

  Description — Specifies the number of repeat for write.

  Direction — IN

  Values — Same as ‘counts’ in the “hpe5022_eraseMultiple” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the write-multiple by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’, and/or ‘counts’ is out of range.

See Also

“hpe5022_eraseMultiple” on page 364

“hpe5022_measure” on page 387

hpe5022_eraseBand

C Syntax

ViStatus hpe5022_eraseBand(ViSession id, ViReal64 range, ViReal64 pitch);

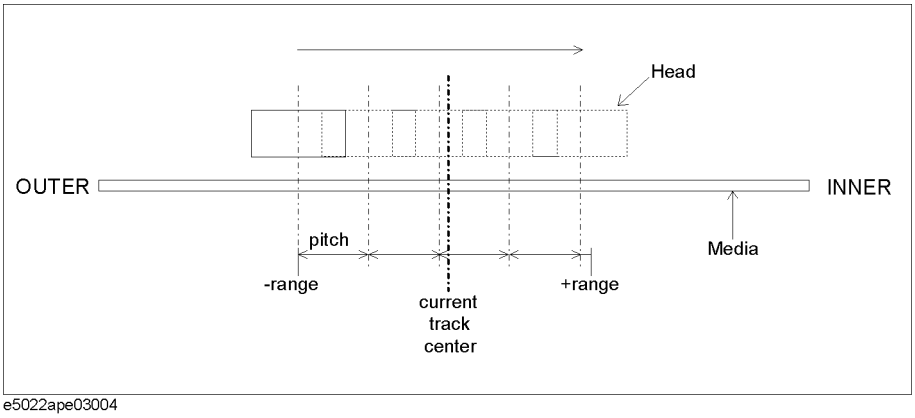
Visual Basic Syntax

hpe5022_eraseBand(ByVal id As Long, ByVal range As Double, ByVal pitch As Double) As Long

Description

This function performs a erase with a specified track pitch and erase range from the current track.The erase range is specified by the parameter “range”. The erase type and erase current is specified by the “hpe5022_eraseType” and “hpe5022_eraseCurrent”functions respectively.

Figure 3-14 Band Erase Sequence



e5022ape03004

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- range

  Description: Specifies the erase range.

  Direction: IN

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- pitch

  Description: Specifies the erase pitch.

  Direction: IN

Unit Meter

Values

Name	Value
hpe5022_ERASE_BAND_PITCH_MIN	50×10^{-9}
hpe5022_ERASE_BAND_PITCH_MAX	6×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘range’ and/or ‘pitch’ is out of range.

See Also

“hpe5022_dataArea” on page 180

“hpe5022_driveState” on page 209

“hpe5022_eraseType” on page 355

“hpe5022_eraseCurrent” on page 358

hpe5022_setupEraseBand

C Syntax

ViStatus hpe5022_setupEraseBand(ViSession id, ViReal64 range, ViReal64 pitch, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupEraseBand(ByVal id As Long, ByVal range As Double, ByVal pitch As Double, ByRef testHndl As Long) As Long

Description

This function sets the band erase sequence from a specified range and specified pitch.This function does not execute the band erase operation. Band erase is executed by the “hpe5022_measure” function with the test identifier given by this function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- range

  Description: Specifies the erase range.

  Direction: IN

  Unit: Meter

  Value

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- pitch

  Description: Specifies the erase pitch.

  Direction: IN

  Unit: Meter

  Values

Name	Value
hpe5022_ERASE_BAND_PITCH_MIN	50×10^{-9}
hpe5022_ERASE_BAND_PITCH_MAX	6×10^{-6}

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the band erase by the “hpe5022_eraseBand” function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘range’ and/or ‘pitch’ is out of range.

See Also

“hpe5022_eraseType” on page 355

“hpe5022_eraseCurrent” on page 358

“hpe5022_eraseBand” on page 368

“hpe5022_measure” on page 387

hpe5022_eraseWideBand

C Syntax

ViStatus hpe5022_eraseWideBand(ViSession id, ViReal64 range, ViReal64 pitch);

Visual Basic Syntax

hpe5022_eraseWideBand(ByVal id As Long, ByVal range As Double, ByVal pitch As Double) As Long

Description

This function erases a whole specified area. The “hpe5022_eraseBand” function erases by the piezo movement. That’s why it can erase only the area between $+6\times10^{-6}$ [m] and -6×10^{-6} [m] from the center of a current track. In other hand, this function erases by the piezo and the X-Y stage movements.

1. Move a track center position at the -(range)+4×10^{-6}[m] by actuating X-Y stages.
2. Perform a band erase.
   (same as hpe5022_eraseBand:
   range: $+6\times10^{-6}$
   pitch: same as ‘pitch’ of this function)
3. Move a track center position by $+8\times10^{-6}$[m] by actuating X-Y stages.
4. Repeat the steps 2 and 3 until the track center position reaches the (+range).

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- range

  Description: Specifies a range for erase. The values (radius±range) should be between ID and OD.

  Direction: IN

  Values

Name	Value
hpe5022_DISK_RANGE_MAX	47.5×10^{-3}

  Unit: Meter

- Pitch

  Description: Specifies a pitch for erase within the range.

  Direction: IN

Values

Name	Value
hpe5022_ERASE_BAND_PITCH_MIN	50×10^{-9}
hpe5022_ERASE_BAND_PITCH_MAX	6×10^{-6}

Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘range’ or/and ‘pitch” is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has already been turned on. Turn it off before executing this function. See the “hpe5022_driveState” function.

See Also

Precompensation Function

This section describes the functions related with the precompensation. E5022A/B can delay a write timing for the specified patterns. The pattern is defined as three bits previous and behind the target bit. Three kinds of delay time can be specified. In addition, you can select which test data pattern will have a precompensation such as the NLTS 5th pattern.

hpe5022_precompDelay

C Syntax

ViStatus hpe5022_precompDelay(ViSession id,ViInt16 precompNum, ViReal64 delay);

Visual Basic Syntax

hpe5022_precompDelay(ByVal id As Long, ByVal precompNum As Integer, ByVal delay As Double) As Long

Description

This function specifies the precompensation delay time. Up to three delay's can be set. The pattern which has a delay time (precompensation) is defined by the "hpe5022_precompPattern" function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- precompNum

  Description: Specifies the precompensation number to set the delay time.

  Direction: IN

  Values

Name	Value	Description
hpe5022_PREC_1	1	Precompensation No.1
hpe5022_PREC_2	2	Precompensation No.2
hpe5022_PREC_3	3	Precompensation No.3

- delay

  Description: Specifies the delay time for the selected precompNum.

  Direction: IN

  Preset value: 0

  Unit: Second

Values

Name	Value
hpe5022_PREC_VAL_MIN	-1900×10^{-12}
hpe5022_PREC_VAL_MAX	1900×10^{-12}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘precompNum’ and/or ‘delay’ is out of range.

See Also

“hpe5022_precompDelay_Q” on page 376

“hpe5022_precompPattern” on page 377

“hpe5022_precompState” on page 383

“hpe5022_clearPrecomp” on page 386

hpe5022_precompDelay_Q

C Syntax

ViStatus hpe5022_precompDelay_Q(ViSession id,ViInt16 precompNum, ViPReal64 delay);

Visual Basic Syntax

hpe5022_precompDelay_Q(ByVal id As Long, ByVal precompNum As Integer, ByRef delay As Double) As Long

Description

This function returns the delay time which is assigned to the precompensation number.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- precompNum

  Description — Specifies the precompensation number to query the delay time.

  Direction — IN

  Values — Same as the ‘precompNum’ in the “hpe5022_precompDelay” function.

- delay

  Description — Returns the delay time for the selected precompensation number.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘precompNum’ is out of range.

See Also

“hpe5022_precompDelay” on page 374

hpe5022_precompPattern

C Syntax

ViStatus hpe5022_precompPattern(ViSession id,ViInt16 preBit, ViInt16 postBit, ViInt16 precompNum);

Visual Basic Syntax

hpe5022_precompPattern(ByVal id As Long, ByVal preBit As Integer, ByVal postBit As Integer, ByVal precompNum As Integer) As Long

Description

This function specifies the precompensation pattern which has a delay time. E5022A/B allows you to specify three bits before and after the target bit.

1. If the user wants to have a precompensation delay specified by “hpe5022_PREC_1” for the pattern “001x000” (x is a target bit), the function should be:

   ```
   hpe5022_precompPattern(hpe5022,
   hpe5022_PREC_PAT_001,hpe5022_PREC_PAT_000, hpe5022_PREC_1)
   ```

   The delay time of “hpe5022_PREC_1” is specified by the “hpe5022_precompDelay” function.

   The setting of precompensation is hold until the “hpe5022_clearPrecomp” or “hpe5022_reset” function is performed.

2. The setting of precompensation pattern is added to the current setting. For example:

   If the user wants to have the precompensation delay for patterns “001x000” and “010x000”, the following functions are to be performed.

   ```
   hpe5022_precompPattern(hpe5022,
   hpe5022_PREC_PAT_010,hpe5022_PREC_PAT_000, hpe5022_PREC_1)
   ```

3. When the setting for a pattern overlaps, then the last setting for the pattern becomes valid. For Example: when the following function is performed,

   hpe5022_precompPattern(hpe5022,
   hpe5022_PREC_PAT_0,hpe5022_PREC_PAT_000, hpe5022_PREC_1)

   The precompensation delay specified by “hpe5022_PREC_1” is set for the patterns of “000x000”, “010x000”, “100x000” and “110x000”.

   Then the following function is performed,

   ```
   hpe5022_precompPattern(hpe5022,
   hpe5022_PREC_PAT_110,hpe5022_PREC_PAT_000, hpe5022_PREC_2)
   ```

   The precompensation delay specified by “hpe5022_PREC_2” is set for the pattern “110x000”, and the precompensation delay specified by “hpe5022_PREC1” is set for the patterns of “000x000”, “010x000” and “100x000”. (The pattern of “110x000” is removed).

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

Direction IN

- preBit

  Description Specifies the precompensation of pre-bit pattern.

  Direction IN

  Values

Name	Value	Description
hpe5022_PREC_PAT_X	0	Any
hpe5022_PREC_PAT_0	1	000, 010, 100, or 110 (**0)[a]
hpe5022_PREC_PAT_1	2	001, 011, 101, or 111 (**1)[*a]
hpe5022_PREC_PAT_00	3	000 or 100 (*00) [*a]
hpe5022_PREC_PAT_01	4	001 or 101 (*01) [*a]
hpe5022_PREC_PAT_10	5	010 or 110 (*10) [*a]
hpe5022_PREC_PAT_11	6	011 or 111 (*11) [*a]
hpe5022_PREC_PAT_000	7	000
hpe5022_PREC_PAT_001	8	001
hpe5022_PREC_PAT_010	9	010
hpe5022_PREC_PAT_011	10	011
hpe5022_PREC_PAT_100	11	100
hpe5022_PREC_PAT_101	12	101
hpe5022_PREC_PAT_110	13	110
hpe5022_PREC_PAT_111	14	111

a. * means 0 or 1

- postBit

  Description Specifies the precompensation of post-bit pattern.

  Direction IN

  Values

Name	Value	Description
hpe5022_PREC_PAT_X	0	Any
hpe5022_PREC_PAT_0	1	000, 001, 010, or 011 (0**) [a]

Name	Value	Description
hpe5022_PREC_PAT_1	2	100, 101, 110, or 111 (1**) [*a]
hpe5022_PREC_PAT_00	3	001 or 000 (00*) [*a]
hpe5022_PREC_PAT_01	4	010 or 011 (01*) [*a]
hpe5022_PREC_PAT_10	5	100 or 101 (10*) [*a]
hpe5022_PREC_PAT_11	6	110 or 111 (11*) [*a]
hpe5022_PREC_PAT_000	7	000
hpe5022_PREC_PAT_001	8	001
hpe5022_PREC_PAT_010	9	010
hpe5022_PREC_PAT_011	10	011
hpe5022_PREC_PAT_100	11	100
hpe5022_PREC_PAT_101	12	101
hpe5022_PREC_PAT_110	13	110
hpe5022_PREC_PAT_111	14	111

a. * means 0 or 1

- precompNum

  Description — Specifies the precompensation delay number. The delay time is set by the “hpe5022_precompDelay” function.

  Direction — IN

  Preset value — 0 (hpe5022_PREC_OFF)

  Value

Name	Value	Description
hpe5022_PREC_OFF	0	Set delay time at 0.
hpe5022_PREC_1	1	Set precompensation No.1
hpe5022_PREC_2	2	Set precompensation No.2
hpe5022_PREC_3	3	Set precompensation No.3

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘prebit’, ‘postbit’ and/or ‘precompNum’ is out of range.

See Also

“hpe5022_reset” on page 96

“hpe5022_precompDelay” on page 374

“hpe5022_precompPattern_Q” on page 381

“hpe5022_precompState” on page 383

“hpe5022_clearPrecomp” on page 386

hpe5022_precompPattern_Q

C Syntax

ViStatus hpe5022_precompPattern_Q(ViSession id, ViInt16 preBit, ViInt16 postBit, ViPInt16 precompNum);

Visual Basic Syntax

hpe5022_precompPattern_Q(ByVal id As Long, ByVal preBit As Integer, ByVal postBit As Integer, ByRef precompNum As Integer) As Long

Description

This function returns the precompensation delay number for the specified pattern. A sample program which shows the current setting of precompensation is shown in chapter 2.

When the user selects a pattern that includes a wildcard (*) in the pre-bit or post-bit patterns, such as “hpe5022_PREC_PAT_0”, an error may occur if the precompensation number is duplicated. For example:

When the following functions are performed,

```
hpe5022_precompPattern(hpe5022,
hpe5022_PREC_PAT_01,hpe5022_PREC_PAT_000, hpe5022_PREC_1)
```

```
hpe5022_precompPattern(hpe5022,
hpe5022_PREC_PAT_11,hpe5022_PREC_PAT_000, hpe5022_PREC_2)
```

The patterns of “001x000” and “101x000” (x is a target bit) has a precompensation delay specified by the “hpe5022_PREC_1” and the pattern of “011x000” and “111x000” has one specified by the “hpe5022_PREC_2”.

If the following functions are performed. You will have an error.

```
hpe5022_precompPattern_Q(hpe5022,
hpe5022_PREC_PAT_1,hpe5022_PREC_PAT_000, hpe5022_PREC_1)
```

Because, the “**1x000” is true for both “hpe5022_PREC_1” and “hpe5022_PREC_2”.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- preBit

  Description — Specifies the precompensation of pre-bit pattern.

  Direction — IN

  Values — Same as ‘preBit’ in the “hpe5022_precompPattern” function.

- postBit

  Description — Specifies the precompensation of post-bit pattern.

  Direction — IN

Values — Same as ‘postBit’ in the “hpe5022_precompPattern” function.

- precompNum

  Description — Returns the precompensation of the delay number.

  Direction — OUT

  Value — Same as ‘precompNum’ in the “hpe5022_precompPattern” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘preBit’ and/or ‘postBit’ is out of range.
hpe5022_ERROR_CONFLICT_PREC_PATTERN	The precompensation pattern is conflict.

See Also

“hpe5022_precompPattern” on page 377

“hpe5022_precompDelay” on page 374

hpe5022_precompState

C Syntax

ViStatus hpe5022_precompState(ViSession id,ViInt16 dataPat, ViBoolean state);

Visual Basic Syntax

hpe5022_precompState(ByVal id As Long, ByVal dataPat As Integer, ByVal state As Integer) As Long

Description

This function controls the precompensation state of the specified data pattern. When pattern is set to ON in the parameter “state”, then the selected data pattern will have a precompensation. Delay time and pattern for precompensation is defined by the “hpe5022_precompDelay” and “hpe5022_precompPattern” functions.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataPat

  Description: Specifies the data pattern to be selected for precompensation.

  Direction: IN

  Values

Name	Value	Description
hpe5022_PAT_HF	0	HF Pattern
hpe5022_PAT_LF	1	LF Pattern
hpe5022_PAT_ISO	2	Isolated Pulse Pattern
hpe5022_PAT_PRBS	3	PRBS Pattern
hpe5022_PAT_REP	4	Repetition Pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th Pattern
hpe5022_PAT_OWHF	6	Overwrite HF Pattern
hpe5022_PAT_OWLF	7	Overwrite LF Pattern
hpe5022_PAT_USER	20	User-defined Pattern

- state

  Description: Sets the precompensation state.

  Direction: IN

  Preset value: OFF (VI_FALSE)

Values

Name	Value	Description
VI_TRUE	1	Precompensation ON
VI_FALSE	0	Precompensation OFF

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_NSUP_PREC	The installed Data Generator Module does not support procompensation function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataPat’ is out of range.

See Also

“hpe5022_precompDelay” on page 374

“hpe5022_precompPattern” on page 377

“hpe5022_precompState_Q” on page 385

“hpe5022_clearPrecomp” on page 386

hpe5022_precompState_Q

C Syntax

ViStatus hpe5022_precompState_Q(ViSession id,ViInt16 dataPat, ViPBoolean state);

Visual Basic Syntax

hpe5022_precompState_Q(ByVal id As Long, ByVal dataPat As Integer, ByRef state As Integer) As Long

Description

This function returns the precompensation state of the specified data pattern.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataPat

  Description — Specifies the data pattern for precompensation.

  Direction — IN

  Values — Same as ‘dataPat’ in the “hpe5022_precompState” function.

- state

  Description — Returns the precompensation state.

  Direction — OUT

  Values — Same as the ‘state’ in the “hpe5022_precompState” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataPat’ is out of range.

See Also

“hpe5022_precompState” on page 383

hpe5022_clearPrecomp

C Syntax

ViStatus hpe5022_clearPrecomp(ViSession id);

Visual Basic Syntax

hpe5022_clearPrecomp(ByVal id As Long) As Long

Description

This function clears all precompensation configurations; all precompensation pre/post-bit patterns are set to “hpe5022_PREC_OFF”, the precompensation delay is set to zero for all precompensation delay number, and precompensation state is set to OFF for all data patterns.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_precompDelay” on page 374

“hpe5022_precompPattern” on page 377

“hpe5022_precompState” on page 383

Test Execution Function

This function executes the test sequence of the setup-then-measure measurement. The test sequence is defined by the “hpe5022_setupxxxxx” function. See “Setup-then-measure Measurement” on page 17.

hpe5022_measure

C Syntax

ViStatus hpe5022_measure(ViSession id, ViObject testHndl);

Visual Basic Syntax

hpe5022_measure(ByVal id As Long, ByVal testHndl As Long) As Long

Description

This function executes the test sequence of the specified test identifier. Before this function is executed, the measurement setup must be performed. For example:

```
hpe5022_setupTaa(id, hpe5022_SEQ_ER_WR_M, 5, IdTaa)
hpe5022_measure(id, IdTaa)
```

The measurement setup function is shown in the “See Also”. In case of lack of PC resources, the “hpe5022_releaseSetup” function can be used to release the setup from the PC resource. There are two ways of executing the test sequence as defined by “hpe5022_executionMode” function before measurement. One is by “hpe5022_EXEC_WAIT_COMPLETE” and the other is by “hpe5022_EXEC_WAIT_START” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- testHndl

  Description: Specifies the test identifier to execute. This test identifier defined by the hpe5022_setupxxxxx function is used for this function.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_AUTO_RANGE_FAIL	The autoranging in the track profile measurement is failed. Check if the paramter setting and head is correct.
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_EXCESS_TRACK_DATA	The data overflow a track at one revolution at the bit error measurement. Change the setting of the user data rate, rpm, track number, track format or sector format.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘testHndl’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_NO_OPTION	An option is not installed. This function can only be used if the system has option.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_PARAMETER_SET_FAILED	The measurement data bytes exceeds the limit. See “hpe5022_BER_berMeasByte Count” function.
hpe5022_ERROR_PLL_UNLOCK	The PLL is unlocked. Check if parameter setting is correct.
hpe5022_ERROR_RESAMP_FAILED	The waveform re-sampling has failed.

Error Code	Description
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

"hpe5022_releaseSetup" on page 390

"hpe5022_setupBaseline" on page 450

"hpe5022_setupBaselinePopping" on page 806

"hpe5022_setupChannelBitRateSweep" on page 682

"hpe5022_setupNarrowBandTaa" on page 574

"hpe5022_setupNlts5th" on page 813

"hpe5022_setupOverwrite" on page 913

"hpe5022_setupParametric" on page 469

"hpe5022_setupPartialErasure" on page 937

"hpe5022_setupPw" on page 433

"hpe5022_setupResolution" on page 504

"hpe5022_setupRollOff" on page 689

"hpe5022_setupSenseCurrentSweep" on page 668

"hpe5022_setupSnr" on page 825

"hpe5022_setupStability" on page 529

"hpe5022_setupTaa" on page 412

"hpe5022_setupTrackProfile" on page 620

"hpe5022_setupTripleTrack" on page 719

"hpe5022_setupPrecompSweep" on page 699

"hpe5022_setupWave" on page 1031

"hpe5022_setupWriteCurrentSweep" on page 661

Release setup test function

This function releases the test sequence from the PC resource.

hpe5022_releaseSetup

C Syntax

ViStatus hpe5022_releaseSetup(ViSession id, ViObject testHndl);

Visual Basic Syntax

hpe5022_releaseSetup(ByVal id As Long, ByVal testHndl As Long) As Long

Description

This function releases the test sequence from the memory of the PC and modules. The setup function shown in “See Also” uses some portion of the PC/Modules resource. When many setup function is to be performed once, an error might occur “hpe5022_ERROR_MEM_ALLOC”. In that case, you can keep PC/Modules memory by using this function for the finished setup function. For example:

```
hpe5022_setupTaa(id, hpe5022_SEQ_ER_WR_M, 5, IdTaa)
hpe5022_setupPw(id, hpe5022_SEQ_ER_WR_M, 5, IdPw)
hpe5022_measure(id, IdTaa)
hpe5022_taa_Q(id, taa, TAAPos, TAANeg, TAAAsym)
hpe5022_measure(id, IdPw)
hpe5022_pw_Q(id, pw, PWPos, PWNeg, PWAsym)

hpe5022_releaseSetup(id, IdTaa)
hpe5022_releaseSetup(id, IdPw)

hpe5022_setupBaseline(id, hpe5022_SEQ_ER_WR_M, 5, IdBaseline)
hpe5022_measure(id, IdBaseline)
hpe5022_baseline_Q(id, Bl, BlPos, BlNeg, Separation)
```

When you have enough the resource, you do not need to use this function. The “hpe5022_releaseSetupAll” releases all of the test sequences.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- testHndl

  Description — Specifies test identifier to release. This test identifier defined by the hpe5022_setupxxxxx function is used for this.

Direction IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'testHndl' is out of range.

See Also

"hpe5022_releaseSetupAll" on page 392

"hpe5022_setupTaa" on page 412

"hpe5022_setupNarrowBandTaa" on page 574

"hpe5022_setupPw" on page 433

"hpe5022_setupBaseline" on page 450

"hpe5022_setupResolution" on page 504

"hpe5022_setupParametric" on page 469

"hpe5022_setupNlts5th" on page 813

"hpe5022_setupSnr" on page 825

"hpe5022_setupOverwrite" on page 913

"hpe5022_setupPartialErasure" on page 937

"hpe5022_setupTrackProfile" on page 620

"hpe5022_setupWriteCurrentSweep" on page 661

"hpe5022_setupSenseCurrentSweep" on page 668

"hpe5022_setupChannelBitRateSweep" on page 682

"hpe5022_setupRollOff" on page 689

"hpe5022_setupPrecompSweep" on page 699

"hpe5022_setupBaselinePopping" on page 806

"hpe5022_setupTripleTrack" on page 719

"hpe5022_setupStability" on page 529

"hpe5022_setupWave" on page 1031

hpe5022_releaseSetupAll

C Syntax

ViStatus hpe5022_releaseSetup(ViSession id);

Visual Basic Syntax

hpe5022_releaseSetup(ByVal id As Long) As Long

Description

This function releases all of the test sequences from the memory of the PC and modules.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_releaseSetup” on page 390

DC Resistance Measurement Function

This section describes the functions related with DC resistance measurement. DC is measured by the spinstand interface module.

hpe5022_measureReadHeadDcr

C Syntax

ViStatus hpe5022_measureReadHeadDcr(ViSession id, ViReal64 current);

Visual Basic Syntax

hpe5022_measureReadHeadDcr(ByVal id As Long, ByVal current As Double) As Long

Description

This function measures the read head's DC resistance. DC resistance measurement should be made under the condition that the drive is off. When you measure the read head DC resistance only, the only required function before this function is executed is the "hpe5022_init" function. The "hpe5022_readHeadDcr_Q" function returns the result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Specifies the measurement current to measure a DC resistance. The range of current value is returned by the "hpe5022_senseCurrentRange_Q" function.

  Direction: IN

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.

See Also

“hpe5022_init” on page 91

“hpe5022_readHeadDcr_Q” on page 397

“hpe5022_senseCurrentRange_Q” on page 339

hpe5022_measureReadHeadDcrEx

C Syntax

ViStatus hpe5022_measureReadHeadDcrEx(ViSession id, ViInt16 senseStimulusMode, ViReal64 stimulus);

Visual Basic Syntax

hpe5022_measureReadHeadDcrEx(ByVal id As Long, ByVal senseStimulusMode As Integer, ByVal stimulus As Double) As Long

Description

This function measures the DC resistance of the read head. Measurement of DC resistance should be made in condition that the drive is off. When you measure read head DC resistance only, the only required function before this function is executed is the “hpe5022_init” function. The “hpe5022_readHeadDcr_Q” function returns the result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- mode

  Description — Selects the sense stimulus mode to measure a read head DCR.

  Direction — IN

  Preset value — When the installed head amplifier has the current mode, hpe5022_SENSE_STIMULUS_CURRENT is set as default.

  When it does not have the current mode, hpe5022_SENSE_STIMULUS_VOLTAGE is set as default.

  Values

Name	Value	Description
hpe5022_SENSE_STIMULUS_CURRENT	0	Current
hpe5022_SENSE_STIMULUS_VOLTAGE	1	Voltage
hpe5022_SENSE_STIMULUS_POWER	2	Power

- stimulus

  Description — Specifies the sense stimulus value.

  Direction — IN

  Preset value — ‘stimulusMin’ returned by the “hpe5022_senseStimulusRange_Q” function.

  Unit — Depends on the type of “hpe5022_senseStimulusMode”.

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

pe5022_SENSE_STIMULUS_POWER : Watt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘stimulus’ is out of range.
hpe5022_ERROR_NSUP_CONF	The installed head amplifier does not support this function.

See Also

“hpe5022_init” on page 91

“hpe5022_readHeadDcr_Q” on page 397

“hpe5022_senseStimulusRange_Q” on page 348

hpe5022_readHeadDcr_Q

C Syntax

ViStatus hpe5022_readHeadDcr_Q(ViSession id,ViPReal64 dcr);

Visual Basic Syntax

hpe5022_readHeadDcr_Q(ByVal id As Long, ByRef dcr As Double) As Long

Description

This function returns the read head DC resistance.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dcr

  Description: Returns the DC resistance of the read head.

  Direction: OUT

  Unit: ohm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The DC resistance data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureReadHeadDcr” on page 393

“hpe5022_measureReadHeadDcrEx” on page 395

hpe5022_measureWriteHeadDcr

C Syntax

ViStatus hpe5022_measureWriteHeadDcr(ViSession id, ViReal64 current);

Visual Basic Syntax

hpe5022_measureWriteHeadDcr(ByVal id As Long, ByVal current As Double) As Long

Description

This function measures the DC resistance of the write head. Measurement of DC resistance should be made in condition that the drive is off. When you measure write head DC resistance only, the only required function before this function is executed is the “hpe5022_init” function. The “hpe5022_writeHeadDcr_Q” function returns the result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- current

  Description — Specifies the measurement current. The limit value of the write current is dependent on the installed head amplifier. The limit value of the current can be queried using “hpe5022_writeCurrentRange_Q” function.

  Direction — IN

  Unit — Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_init” on page 91

“hpe5022_writeHeadDcr_Q” on page 399

“hpe5022_writeCurrentRange_Q” on page 325

hpe5022_writeHeadDcr_Q

C Syntax

ViStatus hpe5022_writeHeadDcr_Q(ViSession id,ViPReal64 dcr);

Visual Basic Syntax

hpe5022_writeHeadDcr_Q(ByVal id As Long, ByRef dcr As Double) As Long

Description

This function returns the DC resistance of the write head.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dcr

  Description: Returns the DC resistance of the write head.

  Direction: OUT

  Unit: ohm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The DC resistance data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureWriteHeadDcr” on page 398

Reference value for Parametric Test Function

This section describes the functions that define the reference values for the parametric test by a parametric module.

hpe5022_taaReference

C Syntax

ViStatus hpe5022_taaReference(ViSession id,ViReal64 taaRef);

Visual basic Syntax

hpe5022_taaReference(ByVal id As Long, ByVal taaRef As Double) As Long

Description

This function specifies the reference value of the TAA to be used for measurements executed by the parametric module. E5022A/B allows you to define the TAA value to calculate PW and baseline parameters. In general, when “hpe5022_PW_TAA_ TRACKED” is selected in the hpe5022_pwThreshold function, there is no need to specify this function.

When “hpe5022_PW_TAA_ FIXED” is selected in the “hpe5022_pwThreshold” function, the “taaRef” value of this function will be used to calculate PW. When track mode is selected, the real-time-measured TAA value is used as the reference.

In addition, when fixed mode is selected in the “hpe5022_baselinePllThreshold” function, this value will be used for baseline measurement. When track mode is selected, the real-time-measured TAA value will be used.

See chapter 5 in the Operaion Manual for details.

This value is not used for baseline and PW measurements using an oscilloscope.(“hpe5022_waveBaseline_Q”, “hpe5022_wavePw_Q”)

Parameters

- id

Description	Specifies the system identifier. This is given by the "hpe5022_init" function.
Direction	IN

- taaRef

Description	Specifies the reference value of TAA.
Direction	IN
Preset Value	2.33×10^{-3} (350×10^{-3} /150)
Unit	Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_baselinePllThreshold” on page 463

“hpe5022_pwThreshold” on page 428

hpe5022_taaReference_Q

C Syntax

ViStatus hpe5022_taaReference_Q(ViSession id,ViPReal64 taaRef);

Visual Basic Syntax

hpe5022_taaReference_Q(ByVal id As Long, ByRef taaRef As Double) As Long

Description

This function returns the reference value of TAA.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- taaRef

  Description: Returns the reference value of TAA.

  Direction: OUT

  Unit: Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_taaReference” on page 400

hpe5022_pwReference

C Syntax

ViStatus hpe5022_pwReference(ViSession id,ViReal64 pwRef);

Visual basic Syntax

hpe5022_pwReference(ByVal id As Long, ByVal pwRef As Double) As Long

Description

This function specifies the reference value of PW. Pulse Width reference value is used for TAA measurement because the parametric module requires rough value of PW to measure TAA.

The “hpe5022_pwReferenceRange_Q” function returns the range of PW reference.

Since this value is affected by parameters, spindle speed and the head position radius when you measure the same head, E5022A/B provides a function which allows you to calculate this reference PW automatically. See the “hpe5022_isolatedPulseReference” or “hpe5022_autoConfig” functions.

The measurement mode defines which data is to be used. When this mode is set to auto, the values specified by the “hpe5022_isolatedPulseReference” function will be used to set the PW reference value. When set to manual, the values specified by this function will be used to set the PW reference value.

See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pwRef

  Description: Specifies the reference value of PW.

  Direction: IN

  Unit: Second

  Preset Values

Disk size	Preset Value
48mm (1.8")	35.0×10^{-9}
65mm (2.5")	29.0×10^{-9}
95mm (3.5")	19.4×10^{-9}
130mm (5.25")	14.2×10^{-9}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘pwRef’ is out of range.

See Also

“hpe5022_pwReferenceRange_Q” on page 406

“hpe5022_pwReference_Q” on page 405

“hpe5022_isolatedPulseReference” on page 265

“hpe5022_autoConfig” on page 257

“hpe5022_measurementMode” on page 254

hpe5022_pwReference_Q

C Syntax

ViStatus hpe5022_pwReference_Q(ViSession id,ViPReal64 pwRef);

Visual Basic Syntax

hpe5022_pwReference_Q(ByVal id As Long, ByRef pwRef As Double) As Long

Description

This function returns the current setting value of PW reference.

This value is determined depending on the measurement mode. When it is set to auto, the values specified by the “hpe5022_isolatedPulseReference” function is used to set the PW reference. When it is set to manual, the values specified by this “hpe5022_pwReference” function is used to set the PW reference.

See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pwRef

  Description: Returns the current setting value of PW reference.

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_pwReference” on page 403

“hpe5022_measurementMode” on page 254

“hpe5022_isolatedPulseReference” on page 265

hpe5022_pwReferenceRange_Q

C Syntax

ViStatus hpe5022_pwReferenceRange_Q(ViSession id, ViPReal64 pwRefMin, ViPReal64 pwRefMax);

Visual Basic Syntax

hpe5022_pwReferenceRange_Q(ByVal id As Long, ByRef pwRefMin As Double, ByRef pwRefMax As Double) As Long

Description

This function returns the PW reference range. The PW reference range depends on the installed parametric module (E5038A or E5038B).

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pwRefMin

  Description: Returns the minimum value that the PW reference can be set.

  Direction: OUT

  Unit: Second

- pwRefMax

  Description: Returns the maximum value that the PW reference can be set.

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_pwReference” on page 403

hpe5022_baselineReference

C Syntax

ViStatus hpe5022_baselineReference(ViSession id,ViReal64 baseRef);

Visual Basic Syntax

hpe5022_baselineReference(ByVal id As Long, ByVal baseRef As Double) As Long

Description

This function specifies the baseline reference value. The baseline reference is a reference amplitude level for the TAA and the PW measurement. See chapter 5 in the Operaion Manual for details.

This value is not used in PW measurement using an oscilloscope. (“hpe5022_wavePw_Q”)

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- baseRef

  Description — Specifies the Baseline reference value.

  Direction — IN

  Preset Value — 0

  Unit — Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_baselineReference_Q” on page 408

hpe5022_baselineReference_Q

C Syntax

ViStatus hpe5022_baselineReference_Q(ViSession id,ViPReal64 baseRef);

Visual Basic Syntax

hpe5022_baselineReference_Q(ByVal id As Long, ByRef baseRef As Double) As Long

Description

This function returns the baseline reference value.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- baseRef

  Description: Returns the Baseline reference value.

  Direction: OUT

  Unit: Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

Track Average Amplitude (TAA) Measurement Function

This section describes the functions related with track average amplitude (TAA) measurement. This is measured by the parametric module.

hpe5022_measureTaa

C Syntax

ViStatus hpe5022_measureTaa(ViSession id, ViInt16 seqType, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureTaa(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer) As Long

Description

This function measures the TAA parameters according to the specified measurement sequence. There are three types of sequence : 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The detailed sequences are as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If ‘seqType’ (see parameters) is set to “hpe5022_SEQ_ER_WR_M”, perform erase (same as the “hpe5022_erase” function) of an entire track.
3. If ‘seqType’ is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write the data pattern specified by the “hpe5022_selectPattern” function for an entire track at the write track offset position.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure the TAA.
6. Repeat the procedure steps 3 to 5 ‘ave’ (see parameters) times if it is more than 1.

The “hpe5022_taa_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measure
hpe5022_SEQ_WR_M	1	Write->Measure
hpe5022_SEQ_M	2	Measure

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_TAA_COUN_MIN	1
hpe5022_TAA_COUN_MAX	100

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘seqType’ and/or ‘ave’ are out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit

Error Code	Description
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_taa_Q” on page 414

“hpe5022_writeTrackOffset” on page 222

“hpe5022_readTrackOffset” on page 219

hpe5022_setupTaa

C Syntax

ViStatus hpe5022_setupTaa(ViSession id, ViInt16 seqType, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupTaa(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the TAA measurement sequence to the specified test identifier. See the “hpe5022_measureTaa” function about the details of the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with its test identifier specified in this function.

The “hpe5022_taa_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as ‘seqType’ in the “hpe5022_measureTaa” function.

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as ‘ave’ in the “hpe5022_measureTaa” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the TAA measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup ” function.

See Also

“hpe5022_measureTaa” on page 409

“hpe5022_taa_Q” on page 414

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_taa_Q

C Syntax

ViStatus hpe5022_taa_Q(ViSession id, ViPReal64 taa, ViPReal64 taaPos, ViPReal64 taaNeg, ViPReal64 taaAsym);

Visual Basic Syntax

hpe5022_taa_Q(ByVal id As Long, ByRef taa As Double, ByRef taaPos As Double, ByRef taaNeg As Double, ByRef taaAsym As Double) As Long

Description

This function returns the results of the execution through TAA measurement sequence. The returned parameters are TAA, TAA positive (TAA+), TAA negative (TAA-) and TAA asymmetry.

See chapter 5 in the Operaion Manual for the definition of these parameters.

Each result is the mean of the corresponding parameter of the “hpe5022_taaData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- taa

  Description: Returns the value of TAA.

  Direction: OUT

  Unit: Volt

- taaPos

  Description: Returns the value of TAA positive.

  Direction: OUT

  Unit: Volt

- taaNeg

  Description: Returns the value of TAA negative.

  Direction: OUT

  Unit: Volt

- taaAsym

  Description: Returns the value of TAA asymmetry.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The TAA data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

“hpe5022_taaData_Q” on page 419

hpe5022_taaStatistic_Q

C Syntax

ViStatus hpe5022_taaStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_taaStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the specified TAA parameter. The data returned by the “hpe5022_taaData_Q” function are used for this statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- mean

  Description: Returns the mean value of measurement from the specified data type. The returned value is the same as the corresponding values returned by the “hpe5022_taa_Q” function.

  Direction: OUT

  Unit: Volt (except for TAA asymmetry)

- min

  Description: Returns the minimum value of measurement.

  Direction: OUT

Unit Volt (except for TAA asymmetry)

- max

  Description Returns the maximum value of measurement.

  Direction OUT

  Unit Volt (except for TAA asymmetry)

- stdDev

  Description Returns the standard deviation value which is calculated using the measurement result of the specified data type.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The TAA data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_taaData_Q” on page 419

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

hpe5022_taaDataSize_Q

C Syntax

ViStatus hpe5022_taaDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_taaDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the TAA data as returned by the “hpe5022_taaData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the array size of TAA data.

  The size will be the same as the value of ‘ave’ specified by the “hpe5022_measureTaa” or “hpe5022_setupTaa” function if the measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The TAA data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_taaData_Q” on page 419

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

hpe5022_taaData_Q

C Syntax

ViStatus hpe5022_taaData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_taaData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the TAA data for each revolution. When the ‘ave’ parameter in the “hpe5022_measureTaa” or the “hpe5022_setupTaa” function is set to more than one, this function allows you to get the TAA data for each revolution. If ‘ave’ is set to 1, then the output of this function will be the same as the “hpe5022_taa_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA Positive
hpe5022_DATA_TAA_NEG	3	TAA Negative
hpe5022_DATA_TAA_ASYM	4	TAA Asymmetry

- data

  Description: Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_taaDataSize_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The TAA data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_taaDataSize_Q” on page 418

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

hpe5022_taaRawDataSize_Q

C Syntax

ViStatus hpe5022_taaRawDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_taaRawDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the data returned by “hpe5022_taaRawData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the array size of TAA raw data.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The TAA raw data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_taaRawData_Q” on page 422

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

.

hpe5022_taaRawData_Q

C Syntax

ViStatus hpe5022_taaRawData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_taaRawData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the TAA raw data. The parametric module measures TAA by observing its peak detector voltage. The peak detector voltage is monitored at every 5 micro seconds. The monitored data at each revolution are averaged and the averaged data is returned by the “hpe5022_taaData_Q” function.

The raw data averages the monitored data for every 5 micro seconds on the last revolution. The data at the previous revolution cannot be retrieved because data in the memory is overwritten for every revolution

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA Positive
hpe5022_DATA_TAA_NEG	3	TAA Negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- data

  Description: Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_taaRawDataSize_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The TAA raw data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_taaRawDataSize_Q” on page 421

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

hpe5022_taaRawStatistic_Q

C Syntax

ViStatus hpe5022_taaRawStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_taaRawStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the TAA raw data. See the “hpe5022_taaRawData_Q” function for the raw data. The data returned by the “hpe5022_taaRawData_Q” function is used for statistic analysis.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA Positive
hpe5022_DATA_TAA_NEG	3	TAA Negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- mean

  Description — Returns the mean value for the raw data of the specified data type.

  Direction — OUT

  Unit — Volt

- min

  Description — Returns the minimum value in the raw data of the specified data type.

  Direction — OUT

  Unit — Volt

- max

  Description — Returns the maximum value in the raw data of the specified data type.

  Direction — OUT

  Unit — Volt

- stdDev

  Description — Returns the standard deviation value which is calculated using the raw data of the specified data type.

  Direction — OUT

  Unit — Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The TAA raw data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_taaRawData_Q” on page 422

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

hpe5022_taaModulation_Q

C Syntax

ViStatus hpe5022_taaModulation_Q(ViSession id, ViPReal64 modPP, ViPReal64 modPos, ViPReal64 modNeg);

Visual Basic Syntax

hpe5022_taaModulation_Q(ByVal id As Long, ByRef modPP As Double, ByRef modPos As Double) As Long

Description

This function returns the TAA modulation from the last revolution. The data returned by the “hpe5022_taaRawData_Q” function is used for this calculation. The definition of modulation is described in the operation manual.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- modPP

  Description — Returns the value of peak to peak modulation

  Direction — OUT

- modPos

  Description — Returns the value of the positive modulation.

  Direction — OUT

- modNeg

  Description — Returns the value of the negative modulation.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The TAA raw data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_taaRawData_Q” on page 422

“hpe5022_measureTaa” on page 409

“hpe5022_setupTaa” on page 412

Pulse Width (PW) Measurement Function

This section describes the functions related with pulse width (PW) measurement. The pulse width measurement in this function category is measured by the parametric module.

hpe5022_pwThreshold

C Syntax

ViStatus hpe5022_pwThreshold(ViSession id, ViReal64 threshold, ViInt16 taaMode);

Visual Basic Syntax

hpe5022_pwThreshold(ByVal id As Long, ByVal threshold As Double, ByVal taaMode As Integer) As Long

Description

This function specifies the threshold level and the reference TAA for PW measurement. The real-time-measured TAA or the user-defined reference TAA value can be selected as the reference TAA. See chapter 5 in the Operaion Manual for details.

This function is not used for pulse width measurement using an Oscilloscope. (“hpe5022_wavePw_Q” function)

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- threshold

  Description — Specifies the threshold level which is shown with the ratio to TAA value.

  Direction — IN

  Preset Value — 0.5

  Values

Name	Value
hpe5022_PW_THR_MIN	0.2
hpe5022_PW_THR_MAX	0.8

- taaMode

  Description — Specifies the reference TAA mode for PW threshold.

  When the “hpe5022_PW_TAA_TRACKED” is selected, the real-time-measured TAA positive and TAA negative levels

will be used to measure PW. When "hpe5022_PW_TAA_FIXED" is selected, the reference TAA level as specified by the "hpe5022_taaReference" function will be used to calculate PW.

Direction IN

Preset value 1 (hpe5022_PW_TAA_TRACKED)

Values

Name	Value	Description
hpe5022_PW_TAA_TRACKED	1	Tracked mode
hpe5022_PW_TAA_FIXED	2	Fixed mode

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'threshold' and/or 'taaMode' is out of range.

See Also

"hpe5022_measurePw" on page 431

"hpe5022_setupPw" on page 433

"hpe5022_pwThreshold_Q" on page 430

"hpe5022_taaReference" on page 400

"hpe5022_wavePw_Q" on page 1043

hpe5022_pwThreshold_Q

C Syntax

ViStatus hpe5022_pwThreshold_Q(ViSession id, ViPReal64 threshold, ViPInt16 taaMode);

Visual Basic Syntax

hpe5022_pwThreshold_Q(ByVal id As Long, ByRef threshold As Double, ByRef taaMode As Integer) As Long

Description

This function returns the configuration of the PW threshold.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- threshold

  Description — Returns the threshold level.

  Direction — OUT

- taaMode

  Description — Returns the specified TAA mode setting.

  Direction — OUT

  Values — Same as the ‘taaMode’ in the “hpe5022_pwThreshold”“hpe5022_pwThreshold” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_pwThreshold” on page 428

hpe5022_measurePw

C Syntax

ViStatus hpe5022_measurePw(ViSession id, ViInt16 seqType, ViInt16 ave);

Visual Basic Syntax

hpe5022_measurePw(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer) As Long

Description

This function measures the PW parameters according to the specified sequence type. There are three sequence types: 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The detailed sequences are as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If ‘seqType’ (see parameters) is set to “hpe5022_SEQ_ER_WR_M”, perform a erase (same as the “hpe5022_erase” function) for an entire track.
3. If ‘seqType’ is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write the data pattern specified by the “hpe5022_selectPattern” function for an entire track at the write track offset position.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure the PW parameter.
6. Repeat the procedure steps 3 to 5 ‘ave’ (see parameters) times if it is more than 1.

The “hpe5022_pw_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measure
hpe5022_SEQ_WR_M	1	Write->Measure
hpe5022_SEQ_M	2	Measure

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_PW_COUN_MIN	1
hpe5022_PW_COUN_MAX	100

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing the function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_selectPattern” on page 119

“hpe5022_pw_Q” on page 435

“hpe5022_write” on page 326

hpe5022_setupPw

C Syntax

ViStatus hpe5022_setupPw(ViSession id, ViInt16 seqType, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupPw(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer, ByVal testHndl As Long) As Long

Description

This function assigns the PW measurement sequence to the specified test identifier. See the “hpe5022_measurePw” function about the details of the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_pw_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measurePw” function.

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as the ‘ave’ in the “hpe5022_measurePw” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the PW measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measurePw” on page 431

“hpe5022_pw_Q” on page 435

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_pw_Q

C Syntax

ViStatus hpe5022_pw_Q(ViSession id, ViPReal64 pw, ViPReal64 pwPos, ViPReal64 pwNeg, ViPReal64 pwAsym);

Visual Basic Syntax

hpe5022_pw_Q(ByVal id As Long, ByRef pw As Double, ByRef pwPos As Double, ByRef pwNeg As Double, ByRef pwAsym As Double) As Long

Description

This function returns the measurement results of PW. The returned parameters are PW, PW positive, PW negative and PW asymmetry. See chapter 5 in the Operaion Manual about the definition of these parameters.

Each result is the mean of the corresponding parameter of the “hpe5022_pwData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pw

  Description: Returns the PW.

  Direction: OUT

  Unit: Second

- pwPos

  Description: Returns the positive value of PW.

  Direction: OUT

  Unit: Second

- pwNeg

  Description: Returns the negative value of PW.

  Direction: OUT

  Unit: Second

- pwAsym

  Description: Returns the asymmetry of PW.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The PW data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measurePw” on page 431

“hpe5022_setupPw” on page 433

“hpe5022_pwData_Q” on page 438

hpe5022_pwDataSize_Q

C Syntax

ViStatus hpe5022_pwDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_pwDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of PW data returned by the “hpe5022_pwData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the array size of PW data.

  The size will be the same as the value of ‘ave’ specified by the “hpe5022_measurePw” or “hpe5022_setupPw” function if the measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The PW data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_pwData_Q” on page 438

“hpe5022_measurePw” on page 431

“hpe5022_setupPw” on page 433

hpe5022_pwData_Q

C Syntax

ViStatus hpe5022_pwData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_pwData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the PW data for each revolution.

When the ‘ave’ parameter in the “hpe5022_measurePw” or the “hpe5022_setupPw” function is set to more than one, this function allows you to get the PW data for each revolution. If ‘ave’ is set to 1, the output of this function will be the same as “hpe5022_pw_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry

- data

  Description — Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_pwDataSize_Q” function.

  Direction — OUT

  Unit — Second (except for PW asymmetry)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The PW data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_pwDataSize_Q” on page 437

“hpe5022_measurePw” on page 431

“hpe5022_setupPw” on page 433

hpe5022_pwStatistic_Q

C Syntax

ViStatus hpe5022_pwStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_pwStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function returns the statistics (mean, minimum, maximum and standard deviation) of the specified PW parameter. The data returned by the “hpe5022_pwData_Q” function is used for statistic analysis.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry

- mean

  Description — Returns the mean value of measurement from the specified data type. The returned value is the same as the corresponding values returned by the “hpe5022_pw_Q” function.

  Direction — OUT

  Unit — Second (excepts for PW asymmetry)

- min

  Description Returns the minimum value of measurement.

  Direction OUT

  Unit Second (excepts for PW asymmetry)

- max

  Description Returns the maximum value of measurement.

  Direction OUT

  Unit Second (except for PW asymmetry)

- stdDev

  Description Returns the standard deviation value which is calculated using the measurement results of the specified data type.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The PW data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_pwData_Q” on page 438

“hpe5022_measurePw” on page 431

“hpe5022_setupPw” on page 433

hpe5022_pwRawDataSize_Q

C Syntax

ViStatus hpe5022_pwRawDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_pwRawDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of PW raw data returned by the “hpe5022_pwRawData_Q” function. The returned size is the number of measurement points at the last revolution.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the array size of PW raw data.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The PW raw data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_pwRawData_Q” on page 443

“hpe5022_measurePw” on page 431

“hpe5022_setupPw” on page 433

hpe5022_pwRawData_Q

C Syntax

ViStatus hpe5022_pwRawData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_pwRawData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the PW raw data.

The parametric module measures a PW by observing its pulse width detector voltage. The pulse width detector voltage is monitored at every 5 micro seconds. The monitored data at each revolution is averaged and the averaged data is returned by the “hpe5022_pwData_Q” function.

The raw data means the monitored data at the last revolution. The data at the previous revolution cannot be retrieved because the data in the memory is overwritten for every revolution.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry

- data

  Description: Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_pwRawDataSize_Q” function.

  Direction: OUT

  Unit: Second (except for PW asymmetry)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The PW raw data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_pwRawDataSize_Q” on page 442

“hpe5022_pwData_Q” on page 438

hpe5022_pwRawStatistic_Q

C Syntax

ViStatus hpe5022_pwRawStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_pwRawStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the PW raw data. See the “hpe5022_pwRawData_Q” function for the raw data. The data returned by the “hpe5022_pwRawData_Q” function are used for this statistic analysis.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry

- mean

  Description — Returns the mean value of raw data from the specified data type.

  Direction — OUT

  Unit — Second (except for PW asymmetry)

- min

  Description — Returns the minimum value of the raw data from the specified data type.

  Direction — OUT

  Unit — Second (except for PW asymmetry)

- max

  Description — Returns the maximum value of the raw data from the specified data type.

  Direction — OUT

  Unit — Second (except for PW asymmetry)

- stdDev

  Description — Returns the standard deviation value which is calculated using the raw data of the specified data type.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The PW raw data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_pwRawData_Q” on page 443

Baseline Measurement Function

This section describes the functions related with baseline measurement. The baseline measurement in this function category is measured by the parametric module.

hpe5022_measureBaseline

C Syntax

ViStatus hpe5022_measureBaseline(ViSession id, ViInt16 seqType, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureBaseline(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer) As Long

Description

This function measures the baseline parameters according to the specified sequence type. There are three types of sequence : 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The detailed sequences are as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If ‘seqType’ (see parameters) is set to “hpe5022_SEQ_ER_WR_M”, perform a erase (same as the “hpe5022_erase” function) for an entire track.
3. If ‘seqType’ is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write data pattern specified by the “hpe5022_selectPattern” function for an entire track at the write track offset position.
4. Move the head to the read track offset specified by the “hpe5022_readTrackOffset” function.
5. Measure the baseline parameter.
6. Repeat the steps 3 to 5 ‘ave’ (see parameters) times if it is more than 1.

The “hpe5022_baseline_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measure
hpe5022_SEQ_WR_M	1	Write->Measure
hpe5022_SEQ_M	2	Measure

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_BASE_COUN_MIN	1
hpe5022_BASE_COUN_MAX	100

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module.Check if the parameter setting is correct.

Error Code	Description
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module.Check if the parameter setting is correct.
hpe5022_ERROR_PLL_UNLOCK	The PLL is unlocked. Check if the setting parameter is correct.

See Also

“hpe5022_writeTrackOffset” on page 222

“hpe5022_readTrackOffset” on page 219

“hpe5022_baseline_Q” on page 452

“hpe5022_driveState” on page 209

hpe5022_setupBaseline

C Syntax

ViStatus hpe5022_setupBaseline(ViSession id, ViInt16 seqType, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupBaseline(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer, ByVal testHndl As Long) As Long

Description

This function assigns the baseline measurement sequence to the specified test identifier. Refer to the “hpe5022_measureBaseline” function about the details of the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_baseline_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measureBaseline” function

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as the ‘ave’ in the “hpe5022_measureBaseline” function

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the baseline measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureBaseline” on page 447

“hpe5022_baseline_Q” on page 452

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_baseline_Q

C Syntax

ViStatus hpe5022_baseline_Q(ViSession id, ViPReal64 base, ViPReal64 basePos, ViPReal64 baseNeg, ViPReal64 sep);

Visual Basic Syntax

hpe5022_baseline_Q(ByVal id As Long, ByRef base As Double, ByRef basePos As Double, ByRef baseNeg As Double, ByRef sep As Double) As Long

Description

This function returns the measurement results of baseline measurement. The returned parameters are Baseline, Baseline positive (Baseline+), Baseline negative (Baseline-) and Baseline separation. See chapter 5 in the Operaion Manual for the definition of these parameters.

The each result is the mean of corresponding parameter of the “hpe5022_baselineData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- base

  Description: Returns the value of Baseline.

  Direction: OUT

  Unit: Volt

- basePos

  Description: Returns the value of baseline positive.

  Direction: OUT

  Unit: Volt

- baseNeg

  Description: Returns the value of baseline negative.

  Direction: OUT

  Unit: Volt

- sep

  Description: Returns the value of the baseline separation.

  Direction: OUT

  Unit: Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The baseline data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureBaseline” on page 447

“hpe5022_setupBaseline” on page 450

“hpe5022_baselineData_Q” on page 455

hpe5022_baselineDataSize_Q

C Syntax

ViStatus hpe5022_baselineDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_baselineDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of baseline data returned by the “hpe5022_baselineData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the array size of baseline data.

  The size will be the same as the value of ‘ave’ specified by the “hpe5022_measureBaseline” or “hpe5022_setupBaseline” function if the measurement done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The baseline data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_baselineData_Q” on page 455

“hpe5022_measureBaseline” on page 447

“hpe5022_setupBaseline” on page 450

hpe5022_baselineData_Q

C Syntax

ViStatus hpe5022_baselineData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_baselineData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the baseline data for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureBaseline” or the “hpe5022_setupBaseline” function is set to more than one, the function allows you to get the baseline data for each revolution. If ‘ave’ is set to 1, the output of this function will be the same as the “hpe5022_baseline_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_BL	9	Baseline
hpe5022_DATA_BL_POS	10	Baseline positive
hpe5022_DATA_BL_NEG	11	Baseline negative
hpe5022_DATA_BL_SEP	12	Baseline separation

- data

  Description: Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_baselineDataSize_Q” function.

  Direction: OUT

  Unit: Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The baseline data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_baselineDataSize_Q” on page 454

“hpe5022_measureBaseline” on page 447

“hpe5022_setupBaseline” on page 450

hpe5022_baselineStatistic_Q

C Syntax

ViStatus hpe5022_baselineStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_baselineStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the specified baseline parameter. The data returned by the “hpe5022_baselineData_Q” function are used for this statistic analysis

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_BL	9	Baseline
hpe5022_DATA_BL_POS	10	Baseline positive
hpe5022_DATA_BL_NEG	11	Baseline negative
hpe5022_DATA_BL_SEP	12	Baseline separation

- mean

  Description: Returns the mean value in the measurement result of the specified data type. The returned value is the same as the corresponding values returned by the “hpe5022_baseline_Q” function.

  Direction: OUT

  Unit: Volt

- min

  Description: Returns the minimum value of measurement result from the specified data type.

  Direction: OUT

Unit Volt

- max

  Description Returns the maximum value of measurement from the specified data type.

  Direction OUT

  Unit Volt

- stdDev

  Description Returns the standard deviation value which is calculated using the measurement result of the specified data type.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The baseline data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_baselineData_Q” on page 455

“hpe5022_measureBaseline” on page 447

“hpe5022_setupBaseline” on page 450

hpe5022_baselineHistogram_Q

C Syntax

ViStatus hpe5022_baselineHistogram_Q(ViSession id, ViReal64 volt[], ViInt32 basePos[], ViInt32 baseNeg[]);

Visual Basic Syntax

hpe5022_baselineHistogram_Q(ByVal id As Long, ByRef volt As Double, ByRef basePos As Long, ByRef baseNeg As Long) As Long

Description

This function returns the histogram of Baseline positive and Baseline negative The returned histogram is generated as follows, the measurement range is equally divided into 256 parts, and these parts are used as range of each category in histogram. The baseline raw data (i.e. all measurement values which are used to calculate baseline parameters of each track) at last track is classified into each category to create the histogram of Baseline+ and Baseline-.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- volt

  Description: Returns 256 center voltage of histogram. The returned data is set in array. This parameter is the divisions of a scale for the ‘basePos’ and ‘baseNeg’ parameters. The relation among these parameters is shown in Figure 3-15.

  Direction: OUT

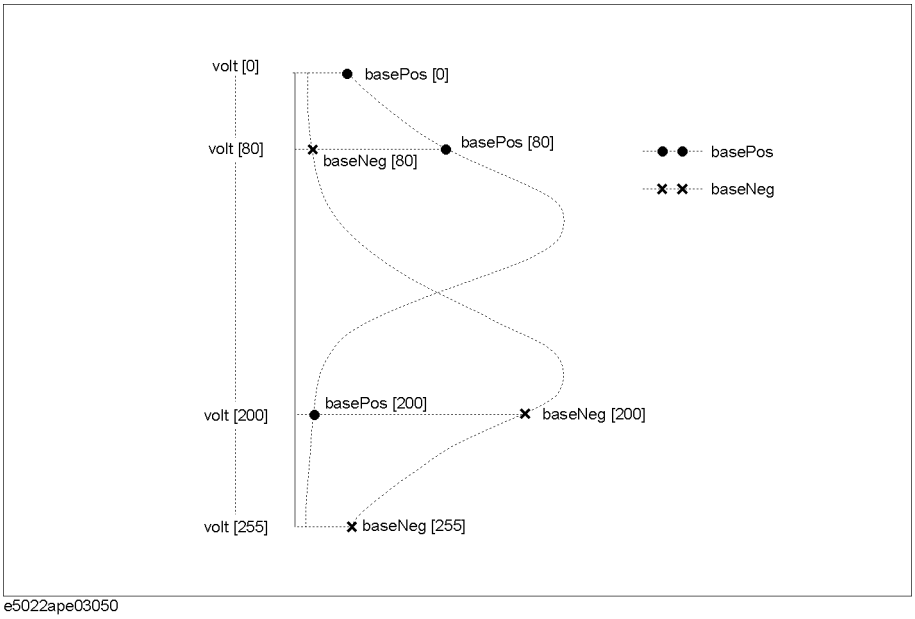
  Unit: Volt

- basePos

  Description: Returns the counts of data in each category of the histogram of Baseline positive. The returned data is set in array, and its size is 256. The basePos [XX] is the count of data in category of which the center value is volt [XX]. (XX is an integer from 0 to 255.).

  Direction: OUT

Figure 3-15 The relation among ‘volt’, ‘basePos’ and ‘baseNeg’



e5022ape03050

- baseNeg

  Description Returns counts of data in each category of the histogram of Baseline-. The returned data is set in array and its size is 256. The baseNeg[YY] is the count of data in category of which the center value is volt[YY]. (YY is an integer from 0 to 255.).

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The baseline data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureBaseline” on page 447

“hpe5022_setupBaseline” on page 450

hpe5022_baselineSamplingPhase

C Syntax

ViStatus hpe5022_baselineSamplingPhase(ViSession id,ViReal64 phase);

Visual Basic Syntax

hpe5022_baselineSamplingPhase(ByVal id As Long, ByVal phase As Double) As Long

Description

This function specifies the sampling phase of baseline measurement. Generally, it is not necessary to specify this function. See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- phase

  Description: Specifies the sampling phase.

  Direction: IN

  Preset Value: 0

  Unit: Degree

  Values

Name	Value
hpe5022_BASE_SAMP_PHAS_MIN	-90
hpe5022_BASE_SAMP_PHAS_MAX	90

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘phase’ is out of range.

See Also

“hpe5022_baselineSamplingPhase_Q” on page 462

hpe5022_baselineSamplingPhase_Q

C Syntax

ViStatus hpe5022_baselineSamplingPhase_Q(ViSession id,ViPReal64 phase);

Visual Basic Syntax

hpe5022_baselineSamplingPhase_Q(ByVal id As Long, ByRef phase As Double) As Long

Description

This function returns the sampling phase for the baseline measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- phase

  Description: Returns the sampling phase.

  Direction: OUT

  Unit: Degree

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_baselineSamplingPhase” on page 461

hpe5022_baselinePllThreshold

C Syntax

ViStatus hpe5022_baselinePllThreshold(ViSession id,ViReal64 level, ViInt16 taaRef);

Visual Basic Syntax

hpe5022_baselinePllThreshold(ByVal id As Long, ByVal level As Double, ByVal taaRef As Integer) As Long

Description

This function specifies the PLL threshold level for the baseline measurement. Generally, it is not necessary to specify this. See chapter 5 in the Operaion Manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- level

  Description: Specifies the threshold level.

  Direction: IN

  Preset Value: 0.5

  Values

Name	Value
hpe5022_PLL_THR_MIN	0.2
hpe5022_PLL_THR_MAX	0.8

- taaRef

  Description: Specifies the TAA reference.

  If (hpe5022_BL_TAA_TRACKED), TAA reference tracks the peak of input signal while baseline measuring.

  If (hpe5022_BL_TAA_FIXED), TAA reference is specified by the “hpe5022_taaReference” function.

  Direction: IN

  Preset Value: 1 (hpe5022_BL_TAA_TRACKED)

  Values

Name	Value	Description
hpe5022_BL_TAA_TRACKED	1	Tracked mode.
hpe5022_BL_TAA_FIXED	2	Fixed mode.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘level’ and/or ‘taaRef’ is out of range.

See Also

“hpe5022_baselinePllThreshold_Q” on page 465

hpe5022_baselinePllThreshold_Q

C Syntax

ViStatus hpe5022_baselinePllThreshold_Q(ViSession id,ViPReal64 level, ViPInt16 taaRef);

Visual Basic Syntax

hpe5022_baselinePllThreshold_Q(ByVal id As Long, ByRef level As Double, ByRef taaRef As Integer) As Long

Description

This function returns the PLL threshold level for baseline measurement.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- level

  Description — Returns the threshold level.

  Direction — OUT

- taaRef

  Description — Returns the TAA reference.

  Direction — OUT

  Values — Same as ‘taaRef’ in the “hpe5022_baselinePllThreshold” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_baselinePllThreshold” on page 463

Parametric Measurement Function

This section describes the functions related with parametric measurement. Parametric measurement measures some typical parameters. Measurement in this function is done by a parametric module.

hpe5022_measureParametric

C Syntax

ViStatus hpe5022_measureParametric(ViSession id, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureParametric(ByVal id As Long, ByVal ave As Integer) As Long

Description

This function measures the following parameters.

- TAA measured using HF pattern (HF TAA)
- TAA measured using LF pattern (LF TAA)
- TAA measured using isolated pulse pattern (Isolated pulse TAA)
- PW measured using isolated pulse pattern (PW)
- Resolution calculated using the HF-TAA and the LF-TAA (Resolution)

The measurement sequence of this function is as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. Perform a erase (same as the “hpe5022_erase” function) for an entire track.
3. Write HF data pattern specified by the “hpe5022_hfPattern” function for an entire track.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure TAA for the HF data pattern.
6. Move the head to the write offset position.
7. Perform a erase (same as the “hpe5022_erase” function) for an entire track.
8. Write an LF data pattern specified by the “hpe5022_lfPattern” function.
9. Move the head to the read offset position.
10. Measure a TAA parameter for the LF data pattern.
11. Move the head to the write offset position.
12. Write the isolated pulse data pattern specified by the “hpe5022_isolatedPulsePattern” function
13. Move the head to the read offset position.

14. Measure a TAA parameter for the isolated data pattern.

15. Measure a PW parameter for the isolated data pattern.

16. Repeat the steps 1 to 15 ‘ave’ (see parameters) times if it is more than 1.

The data patterns of HF, LF and isolated pulse patterns are used automatically. The data pattern specified by the “hpe5022_selectPattern” function is not used for this measurement.

The “hpe5022_parametric_Q”, “hpe5022_parametricHfTaa_Q”, “hpe5022_parametricLfTaa_Q”, “hpe5022_parametricIsolatedPulseTaa_Q” and “hpe5022_parametricPw_Q” functions return the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_PARAMETRIC_COUN_MIN	1
hpe5022_PARAMETRIC_COUN_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

Error Code	Description
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module.Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module.Check if the parameter setting is correct.

See Also

"hpe5022_parametric_Q" on page 471

"hpe5022_parametricHfTaa_Q" on page 473

"hpe5022_parametricLfTaa_Q" on page 480

"hpe5022_parametricIsolatedPulseTaa_Q" on page 487

"hpe5022_parametricPw_Q" on page 494

"hpe5022_erase" on page 362

"hpe5022_readTrackOffset" on page 219

"hpe5022_hfPattern" on page 122

"hpe5022_lfPattern" on page 124

"hpe5022_isolatedPulsePattern" on page 126

hpe5022_setupParametric

C Syntax

ViStatus hpe5022_setupParametric(ViSession id, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupParametric(ByVal id As Long, ByVal ave As Integer, ByVal testHndl As long) As Long

Description

This function assigns the parametric measurement sequence to the specified test identifier. See the “hpe5022_measureParametric” function about the details of this sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_parametric_Q”, “hpe5022_parametricHfTaa_Q”, “hpe5022_parametricLfTaa_Q”, “hpe5022_parametricIsolatedPulseTaa_Q” and “hpe5022_parametricPw_Q” functions returns the measurement results.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- ave

  Description: Specifies the number of measurements for averaging.

  Direction: IN

  Values: Same as the ‘ave’ in the “hpe5022_measureParametric” function.

- testHndl

  Description: Returns the test identifier. This identifier is used to execute the parametric measurement by the “hpe5022_measure” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_parametric_Q” on page 471

“hpe5022_parametricHfTaa_Q” on page 473

“hpe5022_parametricLfTaa_Q” on page 480

“hpe5022_parametricIsolatedPulseTaa_Q” on page 487

“hpe5022_parametricPw_Q” on page 494

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_parametric_Q

C Syntax

ViStatus hpe5022_parametric_Q(ViSession id, ViPReal64 hf_taa, ViPReal64 lf_taa, ViPReal64 iso_taa,ViPReal64 pw, ViPReal64 baseline, ViPReal64 resolution);

Visual Basic Syntax

hpe5022_parametric_Q(ByVal id As Long, ByRef hf_taa As Double, ByRef lf_taa As Double, ByRef iso_taa As Double, ByRef pw As Double, ByRef baseline As Double, ByRef resolution As Double) As Long

Description

This function returns the results of the parametric measurement. The returned parameters are as follows.

- TAA measured using the specified HF pattern (HF TAA).
- TAA measured using the specified LF pattern (LF TAA).
- TAA measured using the specified isolated pulse pattern (Isolated pulse TAA).
- PW measured using the specified isolated pulse pattern (PW).
- Resolution calculated using the HF-TAA and the LF-TAA (Resolution).

Refer to the chapter 5 in the Operaion Manual regarding the definition of this parameters.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- hf_taa

  Description: Returns the value of the HF TAA.

  Direction: OUT

  Unit: Volt

- lf_taa

  Description: Returns the value of the LF TAA.

  Direction: OUT

  Unit: Volt

- iso_taa

  Description: Returns the value of the Isolated pulse TAA.

  Direction: OUT

  Unit: Volt

- pw

  Description Returns the value of the PW.

  Direction OUT

  Unit Second

- baseline

  Description Returns the (hpe5022_NAN) always. This parameter remains for the backward compatibility of a software whose revision is below A.02.10.

  Direction OUT

  Value

Name	Value
hpe5022_NAN	9.91×10^{37}

- resolution

  Description Returns the value of Resolution.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

hpe5022_parametricHfTaa_Q

C Syntax

ViStatus hpe5022_parametricHfTaa_Q(ViSession id,ViPReal64 taa,ViPReal64 taaPos,ViPReal64 taaNeg,ViPReal64 taaAsym);

Visual Basic Syntax

hpe5022_parametricHfTaa_Q(ByVal id As Long, ByRef taa As Double, ByRef taaPos As Double, ByRef taaNeg As Double, ByRef taaAsym As Double,) As Long

Description

This function returns the TAA, the TAA positive, the TAA negative and the TAA asymmetry of the measurement using an HF data pattern.

Each result is the mean of the corresponding parameter of the “hpe5022_parametricHfTaaData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- taa

  Description — Returns the HF TAA.

  Direction — OUT

  Unit — Volt

- taaPos

  Description — Returns the HF TAA positive.

  Direction — OUT

  Unit — Volt

- taaNeg

  Description — Returns the HF TAA negative.

  Direction — OUT

  Unit — Volt

- taaAsym

  Description — Returns the HF TAA asymmetry.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricHfTaaData_Q” on page 476

hpe5022_parametricHfTaaDataSize_Q

C Syntax

ViStatus hpe5022_parametricHfTaaDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_parametricHfTaaDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of HF TAA data returned by the “hpe5022_parametricHfTaaData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- size

  Description: Returns the array size of the HF TAA data.

  The size is the same as the value of ‘ave’ specified by the “hpe5022_measureParametric” or “hpe5022_setupParametric” function if the measurement is done properly.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_parametricHfTaaData_Q” on page 476

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

hpe5022_parametricHfTaaData_Q

C Syntax

ViStatus hpe5022_parametricHfTaaData_Q(ViSession id,ViInt16 dataType,ViReal64 data[]);

Visual Basic Syntax

hpe5022_parametricHfTaaData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the HF TAA data of the measurement for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureParametric” or the “hpe5022_setupParametric” function is set to more than one, the function allows you to get the HF TAA data for each revolution. If ‘ave’ is set to 1, the output of this function is the same as the “hpe5022_parametricHfTaa_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- data

  Description: Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_parametricHfTaaDataSize_Q” function.

  Direction: OUT

  Unit: Volt (except for TAA asymmetry)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricHfTaa_Q” on page 473

“hpe5022_parametricHfTaaDataSize_Q” on page 475

hpe5022_parametricHfTaaStatistic_Q

C Syntax

ViStatus hpe5022_parametricHfTaaStatistic_Q(ViSession id,ViInt16 dataType,ViPReal64 mean,ViPReal64 min,ViPReal64 max,ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_parametricHfTaaStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the TAA parametric measurement. The data returned by the “hpe5022_parametricHfTaaData_Q” function is used for statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- mean

  Description: Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_parametricHfTaa_Q” function.

  Direction: OUT

  Unit: Volt

- min

  Description: Returns the minimum value.

  Direction: OUT

  Unit: Volt

- max

  Description Returns the maximum value.

  Direction OUT

  Unit Volt

- stdDev

  Description Returns the standard deviation value.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_parametricHfTaaData_Q” on page 476

hpe5022_parametricLfTaa_Q

C Syntax

ViStatus hpe5022_parametricLfTaa_Q(ViSession id,ViPReal64 taa,ViPReal64 taaPos,ViPReal64 taaNeg,ViPReal64 taaAsym);

Visual Basic Syntax

hpe5022_parametricLfTaa_Q(ByVal id As Long, ByRef taa As Double, ByRef taaPos As Double, ByRef taaNeg As Double, ByRef taaAsym As Double,) As Long

Description

This function returns the TAA, the TAA positive (TAA+), the TAA negative (TAA-) and the TAA asymmetry of the measurement using LF data pattern.

Each result is the mean of the corresponding parameter of the “hpe5022_parametricLfTaaData_Q” function.

Parameters

- id
  - Description: Handle
  - Direction: IN
- taa
  - Description: Returns the LF TAA.
  - Direction: OUT
  - Unit: Volt
- taaPos
  - Description: Returns the LF TAA positive.
  - Direction: OUT
  - Unit: Volt
- taaNeg
  - Description: Returns the LF TAA negative.
  - Direction: OUT
  - Unit: Volt
- taaAsym
  - Description: Returns the LF TAA asymmetry.
  - Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricLfTaaData_Q” on page 483

hpe5022_parametricLfTaaDataSize_Q

C Syntax

ViStatus hpe5022_parametricLfTaaDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_parametricLfTaaDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the LF TAA data returned by the “hpe5022_parametricLfTaaData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the LF TAA data.

  The size is the same as the value of ‘ave’ specified by the “hpe5022_measureParametric” or “hpe5022_setupParametric” function if the measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_parametricLfTaaData_Q” on page 483

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

hpe5022_parametricLfTaaData_Q

C Syntax

ViStatus hpe5022_parametricLfTaaData_Q(ViSession id,ViInt16 dataType,ViReal64 data[]);

Visual Basic Syntax

hpe5022_parametricLfTaaData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the LF TAA data of the measurement for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureParametric” or the “hpe5022_setupParametric” function is set to more than one, the function allows you to get the LF TAA data for each revolution. If the ‘ave’ is set to 1, the output of this function will be the same as the “hpe5022_parametricLfTaa_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- data

  Description — Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_parametricLfTaaData_Q” function.

  Direction — OUT

  Unit — Volt (except for TAA asymmetry)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricLfTaaDataSize_Q” on page 482

“hpe5022_parametricLfTaa_Q” on page 480

hpe5022_parametricLfTaaStatistic_Q

C Syntax

ViStatus hpe5022_parametricLfTaaStatistic_Q(ViSession id,ViInt16 dataType,ViPReal64 mean,ViPReal64 min,ViPReal64 max,ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_parametricLfTaaStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the LF TAA parametric measurement. The data returned by the “hpe5022_parametricLfTaaData_Q” function is used for statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- mean

  Description: Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_parametricLfTaa_Q” function.

  Direction: OUT

  Unit: Volt

- min

  Description: Returns the minimum value.

  Direction: OUT

  Unit: Volt

- max

  Description Returns the maximum value.

  Direction OUT

  Unit Volt

- stdDev

  Description Returns the standard deviation value.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘datatype’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_parametricLfTaaData_Q” on page 483

hpe5022_parametricIsolatedPulseTaa_Q

C Syntax

ViStatus hpe5022_parametricIsolatedPulseTaa_Q(ViSession id,ViPReal64 taa,ViPReal64 taaPos,ViPReal64 taaNeg,ViPReal64 taaAsym);

Visual Basic Syntax

hpe5022_parametricIsolatedPulseTaa_Q(ByVal id As Long, ByRef taa As Double, ByRef taaPos As Double, ByRef taaNeg As Double, ByRef taaAsym As Double,) As Long

Description

This function returns the TAA, the TAA positive (TAA+), the TAA negative (TAA-) and the TAA asymmetry of the measurement using isolated pulse data pattern. Each result is the mean of the corresponding parameter of the “hpe5022_parametricIsolatedPulseTaaData_Q” function.

Parameters

- id
  - Description: Specifies the system identifier. This is given by the "hpe5022_init" function.
  - Direction: IN
- taa
  - Description: Returns TAA of the isolated pulse data pattern.
  - Direction: OUT
  - Unit: Volt
- taaPos
  - Description: Returns TAA positive of the isolated pulse data pattern.
  - Direction: OUT
  - Unit: Volt
- taaNeg
  - Description: Returns TAA negative of the isolated pulse data pattern.
  - Direction: OUT
  - Unit: Volt
- taaAsym
  - Description: Returns TAA asymmetry of the isolated data pattern.
  - Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricIsolatedPulseTaaData_Q” on page 490

hpe5022_parametricIsolatedPulseTaaDataSize_Q

C Syntax

ViStatus hpe5022_parametricIsolatedPulseTaaDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_parametricIsolatedPulseTaaDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the isolated pulse TAA data returned by the “hpe5022_parametricIsolatedPulseTaaData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the isolated pulse TAA data.

  The size is the same as the value of ‘ave’ as specified by the “hpe5022_measureParametric” or “hpe5022_setupParametric” function if the measurement is done properly.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_parametricIsolatedPulseTaaData_Q” on page 490

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

hpe5022_parametricIsolatedPulseTaaData_Q

C Syntax

ViStatus hpe5022_parametricIsolatedPulseTaaData_Q(ViSession id,ViInt16 dataType,ViReal64 data[]);

Visual Basic Syntax

hpe5022_parametricIsolatedPulseTaaData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the isolated pulse TAA data of the measurement for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureParametric” or the “hpe5022_setupParametric” function is set to more than one, this function allows you to get the isolated pulse TAA data for each revolution. If ‘ave’ is set to 1, the output of this function will be the same as the “hpe5022_parametricIsolatedPulseTaaData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- data

  Description — Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_parametricIsolatedPulseTaaDataSize_Q” function.

  Direction — OUT

  Unit — Volt (except for TAA asymmetry)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricIsolatedPulseTaaDataSize_Q” on page 489

“hpe5022_parametricIsolatedPulseTaaData_Q” on page 490

hpe5022_parametricIsolatedPulseTaaStatistic_Q

C Syntax

ViStatus hpe5022_parametricIsolatedPulseTaaStatistic_Q(ViSession id,ViInt16 dataType,ViPReal64 mean,ViPReal64 min,ViPReal64 max,ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_parametricIsolatedPulseTaaStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the isolated pulse TAA measurement. The data returned by the “hpe5022_parametricIsolatedPulseTaaData_Q” function is used for statistic analysis.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry

- mean

  Description — Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_parametricIsolatedPulseTaa_Q” function.

  Direction — OUT

  Unit — Volt

- min

  Description — Returns the minimum value.

  Direction — OUT

  Unit — Volt

- max

  Description Returns the maximum value.

  Direction OUT

  Unit Volt

- stdDev

  Description Returns the standard deviation value.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_parametricIsolatedPulseTaaData_Q”

hpe5022_parametricPw_Q

C Syntax

ViStatus hpe5022_parametricPw_Q(ViSession id,ViPReal64 pw,ViPReal64 pwPos,ViPReal64 pwNeg,ViPReal64 pwAsym);

Visual Basic Syntax

hpe5022_parametricPw_Q(ByVal id As Long, ByRef pw As Double, ByRef pwPos As Double, ByRef pwNeg As Double, ByRef pwAsym As Double,) As Long

Description

This function returns the PW, the PW positive (PW+), the PW negative (PW-) and the PW asymmetry of the measurement using the isolated pulse data pattern. Each result is the mean of the corresponding parameter of the “hpe5022_parametricPwData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pw

  Description: Returns the PW.

  Direction: OUT

  Unit: Second

- pwPos

  Description: Returns the positive value of PW.

  Direction: OUT

  Unit: Second

- pwNeg

  Description: Returns the negative value of PW.

  Direction: OUT

  Unit: Second

- pwAsym

  Description: Returns the PW asymmetry.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricPwData_Q” on page 497

hpe5022_parametricPwDataSize_Q

C Syntax

ViStatus hpe5022_parametricPwDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_parametricPwDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of PW data returned by the “hpe5022_parametricPwData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the PW data.

  The size will be the same as the value of ‘ave’ specified by the “hpe5022_measureParametric” or “hpe5022_setupParametric” function if the measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricPwData_Q” on page 497

hpe5022_parametricPwData_Q

C Syntax

ViStatus hpe5022_parametricPwData_Q(ViSession id,ViInt16 dataType,ViReal64 data[]);

Visual Basic Syntax

hpe5022_parametricPwData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the PW data of the measurement for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureParametric” or the “hpe5022_setupParametric” function is set to more than one, the function allows you to get the PW data for each revolution. If ‘ave’ is set to 1, the output of this function will be the same as one of the “hpe5022_parametricPwData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry

- data

  Description: Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_parametricPwDataSize_Q” function.

  Direction: OUT

  Unit: Second (except for PW asymmetry)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureParametric” on page 466

“hpe5022_setupParametric” on page 469

“hpe5022_parametricPwData_Q” on page 497

“hpe5022_parametricPwDataSize_Q” on page 496

hpe5022_parametricPwStatistic_Q

C Syntax

ViStatus hpe5022_parametricPwStatistic_Q(ViSession id,ViInt16 dataType,ViPReal64 mean,ViPReal64 min,ViPReal64 max,ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_parametricPwStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the parametric PW measurement. The data returned by the “hpe5022_parametricPwData_Q” function is used for statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry

- mean

  Description: Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_parametricPw_Q” function.

  Direction: OUT

  Unit: Second

- min

  Description: Returns the minimum value.

  Direction: OUT

  Unit: Second

- max

  Description Returns the maximum value.

  Direction OUT

  Unit Second

- stdDev

  Description Returns the standard deviation value.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The parametric data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_parametricPwData_Q” on page 497

Resolution Measurement Function

This section describes the functions related with resolution measurement. The baseline measurement in this function category is measured by the parametric module.

hpe5022_measureResolution

C Syntax

ViStatus hpe5022_measureResolution(ViSession id, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureResolution(ByVal id As Long, ByVal ave As Integer) As Long

Description

This function measures the resolution. The measurement sequences are as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. Perform a erase (same as the hpe5022_erase function) for an entire track.
3. Write a HF data pattern specified by the “hpe5022_hfPattern” function for an entire track at the write track offset position.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure a TAA parameter for the HF data pattern.
6. Move the head to the write track offset position.
7. Perform a erase (same as the hpe5022_erase function) for an entire track.
8. Write LF data pattern specified by the “hpe5022_lfPattern” function for an entire track at the write track offset position.
9. Move the head to the read track offset position.
10. Measure a TAA parameter for the LF data pattern.
11. Repeat the steps 1 to 10 ‘ave’ (see parameters) times if it is more than 1.

The data patterns HF and LF are used automatically. The data pattern specified by the “hpe5022_selectPattern” function is not used for this measurement.

The “hpe5022_resolution_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- ave

Description Specifies the number of measurements for averaging.

Direction IN

Values

Name	Value
hpe5022_RESO_COUN_MIN	1
hpe5022_RESO_COUN_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	A hardware error is detected on the head amplifer.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand had been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module.Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module.Check if the parameter setting is correct.

See Also

See “hpe5022_resolution_Q” on page 506.

See “hpe5022_readTrackOffset” on page 219.

See “hpe5022_writeTrackOffset” on page 222.

See “hpe5022_hfPattern” on page 122.

See “hpe5022_lfPattern” on page 124.

See “hpe5022_selectPattern” on page 119.

hpe5022_setupResolution

C Syntax

ViStatus hpe5022_setupResolution(ViSession id, ViIny16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupResolution(ByVal id As Long, ByVal ave As Integer, ByVal testHndl As Long) As Long

Description

This function assigns the resolution measurement sequence to the specified test identifier. See the “hpe5022_measureResolution” function for details of the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_resolution_Q” function returns the measurement result

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as ‘ave’ in the “hpe5022_measureResolution” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the resolution measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.

Error Code	Description
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureResolution” on page 501

“hpe5022_resolution_Q” on page 506

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_resolution_Q

C Syntax

ViStatus hpe5022_resolution_Q(ViSession id,ViPReal64 res);

Visual Basic Syntax

hpe5022_resolution_Q(ByVal id As Long, ByRef res As Double) As Long

Description

This function returns the result of the resolution measurement sequence. See chapter 5 in the Operaion Manual for the definition of this parameter.

Each result is the0 mean value of the corresponding parameter of the “hpe5022_resolutionData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- res

  Description — Returns the result of the resolution measurement.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The resolution data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureResolution” on page 501

“hpe5022_setupResolution” on page 504

hpe5022_resolutionDataSize_Q

C Syntax

ViStatus hpe5022_resolutionDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_resolutionDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of resolution data returned by the “hpe5022_resolutionData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- size

  Description: Returns the size of the resolution data.

  The size is the same as the value of ‘ave’ specified by the “hpe5022_measureResolution” or “hpe5022_setupResolution” function if the measurement is done properly.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The resolution data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_resolutionData_Q” on page 508

“hpe5022_measureResolution” on page 501

“hpe5022_setupResolution” on page 504

hpe5022_resolutionData_Q

C Syntax

ViStatus hpe5022_resolutionData_Q(ViSession id,ViReal64 data[]);

Visual Basic Syntax

hpe5022_resolutionData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the resolution data of the measurement results.

When the ‘ave’ parameter in the “hpe5022_measureResolution” or the “hpe5022_setupResolution” function is set to more than one, the function allows you to get the resolution data for each revolution. If the ‘ave’ is set to 1, the output of this function will be the same as one of the “hpe5022_resolution_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the resolution data in array. The array size is returned by the “hpe5022_resolutionDataSize_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The resolution data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureResolution” on page 501

“hpe5022_setupResolution” on page 504

“hpe5022_resolution_Q” on page 506

“hpe5022_resolutionDataSize_Q” on page 507

hpe5022_resolutionStatistic_Q

C Syntax

ViStatus hpe5022_resolutionStatistic_Q(ViSession id,ViPReal64 mean, ViPReal64 min,ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_resolutionStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the resolution measurement. The data returned by the “hpe5022_resolutionData_Q” function is used for statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mean

  Description: Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_resolution_Q” function.

  Direction: OUT

- min

  Description: Returns the minimum value.

  Direction: OUT

- max

  Description: Returns the maximum value.

  Direction: OUT

- stdDev

  Description: Returns the standard deviation value.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The resolution data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_resolutionData_Q” on page 508

Stability Measurement Function (Ratio Setting)

This section describes the functions related with the stability measurement. This measurement allows you to observe the TAA, the PW and the narrow band TAA variations during many write-read iterations. Ratio Setting means that the period of write/read is defined by ratio.

hpe5022_stabilityConfig

C Syntax

ViStatus hpe5022_stabilityConfig(ViSession id, ViInt16 isMode, ViReal64 writeRatio, ViReal64 readRatio);

Visual Basic Syntax

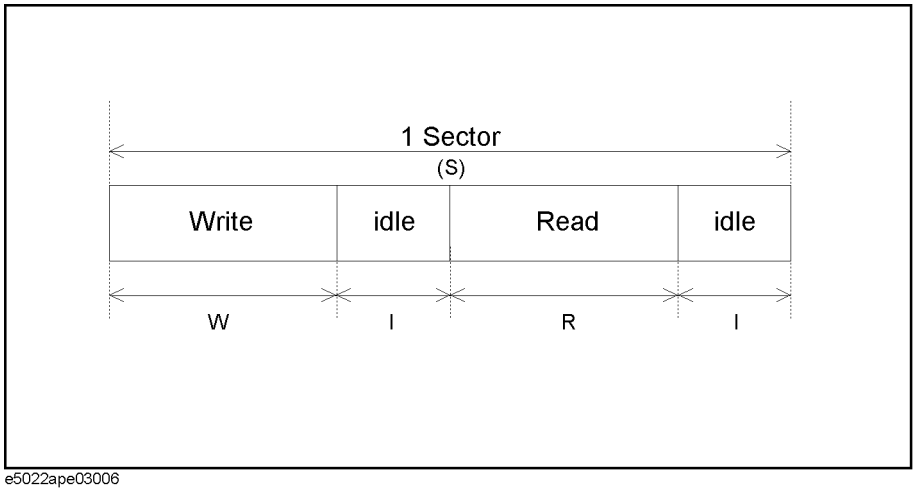
hpe5022_stabilityConfig(ByVal id As Long, ByVal isMode As Integer, ByVal writeRatio As Double, ByVal readRatio As Double) As Long

Description

This function controls the sense current polarity, the write aperture ratio and the read aperture ratio for stability measurement.

The sense current polarity can be selected only when the head amplifier has that capability. The write aperture ratio is defined as the length of write excitation in one segment/sector. Read aperture ratio is defined as the length of the read operation in a segment (sector). Idle period stands in the gap between write excitation and read operation as shown in Figure 3-16. Idle periods have the same length in between write/read iterations.

Figure 3-16 Read and Write Ratios



e5022ape03006

The value of ‘writeRatio’ + ‘readRatio’ must be less than the value of (hpe5022_TAA_STABILITY_RATIO_MAX + hpe5022_TAA_STABILITY_RATIO_MIN)

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- isMode

  Description Specifies the sense current polarity at write operation. When “hpe5022_STABILITY_IS_POL_REV” is selected, the direction of sense current is reversed. When “hpe5022_STABILITY_IS_OFF” is selected, the sense current is set to zero. Some head amplifier does not have the capability to operate at reverse and off sense current modes.

  This setting is used for the stability measurement instead of the setting of “hpe5022_senseCurrentPolarity” function. When this function is not executed, the setting of “hpe5022_senseCurrentPolarity” is used for the stability measurement (Ratio Setting).

  Direction IN

  Preset value hpe5022_STABILITY_IS_POL_NORM

  Values

Name	Value	Description
hpe5022_STABILITY_IS_POL_NORM	0	Same polarity
hpe5022_STABILITY_IS_POL_REV	1	Reverse polarity
hpe5022_STABILITY_IS_OFF	2	Off Mode

- writeRatio

  Description Specifies the write aperture ratio.

  Direction IN

  Preset value 0.4

  Values

Name	Value
hpe5022_STABILITY_RATIO_MIN	0.01
hpe5022_STABILITY_RATIO_MAX	0.98

- readRatio

  Description Specifies the read aperture ratio.

  Direction IN

Preset value 0.4

Values

Name	Value
hpe5022_STABILITY_RATIO_MIN	0.01
hpe5022_STABILITY_RATIO_MAX	0.98

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘isMode’, ‘writeRatio’ and/or ‘readRatio’ are out of range.
hpe5022_ERROR_NSUP_SENS_CURR_POL	The head does not support the reverse polarity and off modes. Select hpe5022_STABILITY_IS_POL_NORM for ‘isMode’ parameter.

See Also

“hpe5022_stabilityConfig_Q” on page 514

“hpe5022_measureStability” on page 524

“hpe5022_setupStability” on page 529

“hpe5022_senseCurrentPolarity” on page 340

hpe5022_stabilityConfig_Q

C Syntax

ViStatus hpe5022_stabilityConfig_Q(ViSession id, ViPInt16 isMode, ViPReal64 writeRatio, ViPReal64 readRatio);

Visual Basic Syntax

hpe5022_stabilityConfig_Q(ByVal id As Long, ByRef isMode As Integer, ByRef writeRatio As Double, ByRef readRatio As Double) As Long

Description

This function returns the specified sense current polarity, the write aperture ratio and the read aperture ratio for stability measurement.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- isMode

  Description — Returns the specified sense current polarity.

  Direction — IN

  Values — Same as the ‘isMode’ in the “hpe5022_stabilityConfig”“hpe5022_stabilityConfig” function.

- writeRatio

  Description — Returns the write aperture ratio.

  Direction — OUT

- readRatio

  Description — Returns the read aperture ratio.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_stabilityConfig” on page 511

hpe5022_stabilitySequenceConfig

C Syntax

ViStatus hpe5022_stabilitySequenceConfig(ViSession id, ViInt16 seqType);

Visual Basic Syntax

hpe5022_stabilitySequenceConfig(ByVal id As Long, ByVal seqType As Integer) As Long

Description

This function specifies the type of sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- seqType

  Description: Specifies the type of sequence. When the “hpe5022_SEQ_M” is selected, the erase and write sequences are not executed.

  Direction: IN

  Preset value: hpe5022_SEQ_ER_WR_M

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Excitation and Measurement
hpe5022_SEQ_M	2	Excitation and Measurement

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘seqType’ is out of range.

See Also

“hpe5022_measureStability” on page 524

“hpe5022_setupStability” on page 529

hpe5022_stabilitySequenceConfig_Q

C Syntax

ViStatus hpe5022_stabilitySequenceConfig_Q(ViSession id, ViPInt16 seqType);

Visual Basic Syntax

hpe5022_stabilitySequenceConfig_Q(ByVal id As Long, ByRef seqType As Integer) As Long

Description

This function returns the type of sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- seqType

  Description: Returns the type of sequence.

  Direction: OUT

  Values: Same as the ‘seqType’ of “hpe5022_stabilitySequenceConfig”

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_stabilitySequenceConfig” on page 515

hpe5022_stabilityWritePatternConfig

C Syntax

ViStatus hpe5022_stabilityWritePatternConfig(ViSession id, ViInt16 writePat);

Visual Basic Syntax

hpe5022_stabilityWritePatternConfig(ByVal id As Long, ByVal writePat As Integer) As Long

Description

This function specifies the stimulus write pattern written along the sector(segment) of the track whose length is specified by the write ratio. Nineteen data patterns are supported by this function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writePat

  Description: Specifies the data pattern used for writing.

  Direction: IN

  Preset value: hpe5022_PAT_DEFAULT

  Values

Name	Value	Description
hpe5022_PAT_DEFAULT	-1	Default pattern
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_REP_2	8	Repetition Pattern2
hpe5022_PAT_REP_3	9	Repetition Pattern3
hpe5022_PAT_REP_4	10	Repetition Pattern4
hpe5022_PAT_USER	20	User-defined Pattern
hpe5022_PAT_USER_2	21	User-defined Pattern2
hpe5022_PAT_USER_3	22	User-defined Pattern3

Name	Value	Description
hpe5022_PAT_USER_4	23	User-defined Pattern4
hpe5022_PAT_ERASE	101	Default Erase Pattern selected by the “hpe5022_eraseType” function
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter writePat is out of range.

See Also

“hpe5022_stabilityWritePatternConfig_Q” on page 519

hpe5022_stabilityWritePatternConfig_Q

C Syntax

ViStatus hpe5022_stabilityWritePatternConfig_Q(ViSession id, ViPInt16 writePat);

Visual Basic Syntax

hpe5022_stabilityWritePatternConfig_Q(ByVal id As Long, ByRef writePat As Integer) As Long

Description

This function returns the write data pattern specified by the “hpe5022_stabilityWritePatternConfig” function for stability measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writePat

  Description: Returns the specified write pattern of the “hpe5022_stabilityWritePatternConfig”.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_stabilityWritePatternConfig” on page 517

hpe5022_stabilityWriteModeConfig

C Syntax

ViStatus hpe5022_stabilityWriteModeConfig(ViSession id, ViInt16 writeMode);

Visual Basic Syntax

hpe5022_stabilityWriteModeConfig(ByVal id As Long, ByVal writeMode As Integer) As Long

Description

This function controls the write current direction at the beginning of write operation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writeMode

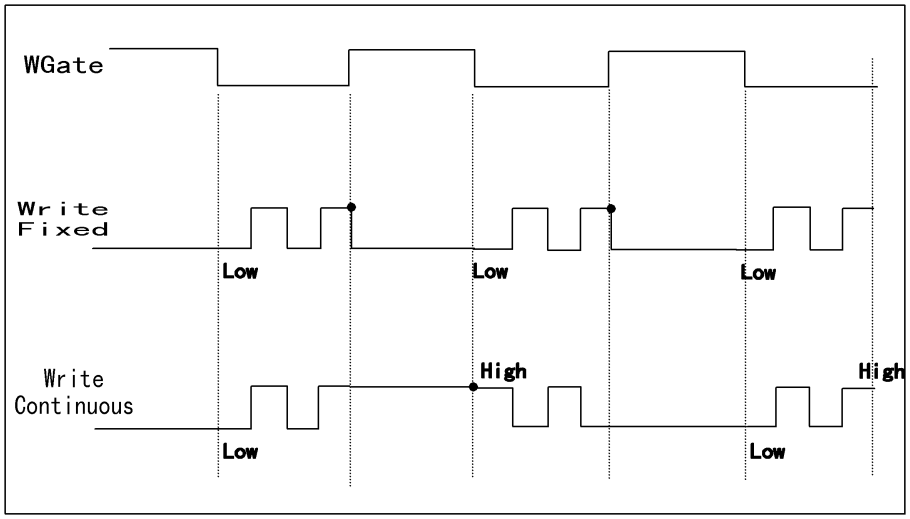
  Description: Specifies the write current direction at the beginning of write operation.

  Direction: IN

  Values

Name	Value	Description
hpe5022_WRIT_DATA_POL_FIXED	0	The write current direction at the beginning of each write operation is always the same.
hpe5022_WRIT_DATA_POL_CONTINUOUS	1	The write current direction at the beginning of write operation is the same as the last current direction of the previous write operation.

Figure 3-17 Write Mode



When [Write Mode] is set to “fixed”, the write current direction at the beginning of each write operation is always the same. On the other hand, when [Write Mode] is set to “continuous” the write current direction is the same as the last current direction of the previous write operation. Each time the write current alternates it allows you to check the polarity of the output pulse at the end of the previous write. You must set the repetitive period of the data pattern T to 1T, 2T, 4T and 8T due to the system’s internal constraints. If you use other repetitive periods of the data pattern, E5022A/B system can not guarantee the change of write current polarity. This function makes it possible for popcorn noise measurement (i.e. failcount) to achieve an acceptable level of TAA under similar usage of the drive. Before this function was introduced, the write current direction always starts at a constant direction, thus you can only observe one side of the output.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writeMode’ is invalid.
hpe5022_ERROR_NSUP_CONF	The installed head amplifier has no capability to control the write current direction.

See Also

“hpe5022_stabilityWriteModeConfig_Q” on page 522

hpe5022_stabilityWriteModeConfig_Q

C Syntax

ViStatus hpe5022_stabilityWritePatternConfig_Q(ViSession id, ViPInt16 writeMode);

Visual Basic Syntax

hpe5022_stabilityWritePatternConfig_Q(ByVal id As Long, ByRef writeMode As Integer) As Long

Description

This function returns the write current direction at the beginning of write operation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writeMode

  Description: Returns the specified write current direction at the beginning of write operation.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_WRIT_DATA_POL_FIXED	0	The write current direction at the beginning of each write operation is always the same.
hpe5022_WRIT_DATA_POL_CONTINUOUS	1	The write current direction at the beginning of write operation is the same as the last current direction of the previous write operation.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writeMode’ is invalid.

See Also “hpe5022_stabilityWriteModeConfig” on page 520

hpe5022_measureStability

C Syntax

ViStatus hpe5022_measureStability(ViSession id, ViInt16 measFunc, ViInt16 segment, ViInt16 rev);

Visual Basic Syntax

hpe5022_measureStability(ByVal id As Long, ByVal measFunc As Integer, ByVal segment As Integer, ByVal rev As Integer) As Long

Description

This function measures the stability. This allows you to divide a track into some segments to get many write-read iterations. The sequence of this function is as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If ‘seqType’ of “hpe5022_stabilitySequenceConfig” fuction is set at “hpe5022_SEQ_ER_WR_M”, erase (same as the “hpe5022_erase” function) for an entire track.
3. If ‘seqType’ of “hpe5022_stabilitySequenceConfig” fuction is set at “hpe5022_SEQ_ER_WR_M”, write the data pattern as specified by the “hpe5022_selectPattern” function.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Set i to zero. (i = 0): a number of revolutions.
6. Set j to zero. (j = 0): a number of segments.
7. Write the data pattern as specified by the “hpe5022_stabilityWritePatternConfig” function during the write ratio period in segment (No. i). The write ratio is specified by the “hpe5022_stabilityConfig” function.
8. Wait for the delay time to elapse.
9. Measure the specified parameter (specified by ‘measFunc’) for the read ratio period in segment (No. i). The read ratio is specified by the “hpe5022_stabilityConfig” function.
10. Increment j by 1. (j = j+1)
11. If j is less than ‘segment’ (see parameters), go to the step 7.
12. Increment i by 1. (i = i +1)
13. If j is less than ‘rev’ (see parameters), go to the step 6.

The “hpe5022_stabilityData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

Direction IN

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_PW	2	Measure PW
hpe5022_MEAS_NB_TAA	3	Measure Narrow Band TAA

- segment

  Description Specifies the number of segments per track,i.e, the number of times a track is divided. If the measurement function is “hpe5022_MEAS_NB_TAA”, then segment must be 1 (“hpe5022_SEG_PER_TRACK_MIN”). Only one measurement per revolution i.e. not sectored.

  Direction IN

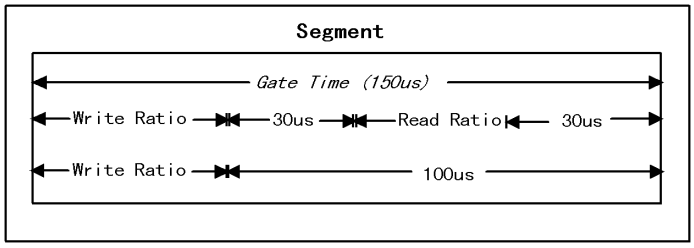
  Values

Name	Value
hpe5022_SEG_PER_TRACK_MIN	1
hpe5022_SEG_PER_TRACK_MAX	80

Gate Time Limitations
track time = (60/rpm) -aper (aper is specified by the hpe5022_trackOffsetComPattern function.) aper = 0 when the stat of the hpe5022_trackOffset CompPattenState is set at VI_FALSE.
gateTime = 60/ RPM/ segment (i.e, gateTime ≥ 150 µsec.)
rpm = spindle speed (where spindle speed ≥ 2400 rpm) = 2400 rpm (where spindle speed < 2400)
writeRatio × gateTime ≥ 1.0 µsec.
(1-writeRatio) × gateTime ≥ 100 µsec. (where ‘segment’ ≥ 1)

Gate Time Limitations
(for TAA and PW Measurement) gateTime× (1- (writeRatio + readRatio)) /2 ≥ 30 µsec. readRatio× gateTime ≥ 20 µsec.
(for Narrow Band TAA Measurement) gateTime × (1- (writeRatio + readRatio)) /2 ≥ 100 µsec. readRatio × gateTime ≥ 0.8 msec.

Example : where rpm = 10000, segment =40, period/rev =6 msec.



From the illustration above the following parameters will be assumed for computation:

1. WriteTime + ReadTime < 90 µsec.

2. Write Ratio ≤ 30% (i.e, Write Time ≤ 45 µsec.

3. Read Ratio ≤ 30% (i.e, Write Time ≤ 45 µsec.

When measuring TAA and PW, a 30µsec. delay time is required this is computed based from **gateTime × (1 - (writeRatio + readRatio)) / 2 ≥ 30 µsec**. and the sum of Write Ratio and Read Ratio. Since the sum of Write Ratio and Read Ratio is 90 µsec. (i.e, [150-(30+30)], the sum of Write Ratio and Read Ratio must be ≤ 60 % (i.e, [90/150=0.6).

Also, **(1-writeRatio) × gateTime** must be ≥ 100 µsec, thus read time should be **40µsec**.

- rev

  Description Specifies the number of revolutions for measurement.

  Direction IN

  Values

Name	**Value**
hpe5022_STABILITY_REV_MIN	1
hpe5022_STABILITY_REV_MAX	100

Number of Counts	Value
hpe5022_STABILITY_COUN_MIN ≤ **segment** × **rev** ≤ hpe5022_STABILITY_COUN_MAX	
hpe5022_STABILITY_COUN_MIN	1
hpe5022_STABILITY_COUN_MAX	1000

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘sector’ and/or ‘rev’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before this function is executed. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.

Error Code	Description
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

"hpe5022_selectPattern" on page 119

"hpe5022_driveState" on page 209

"hpe5022_readTrackOffset" on page 219

"hpe5022_writeTrackOffset" on page 222

"hpe5022_stabilityConfig" on page 511

"hpe5022_stabilityData_Q" on page 531

hpe5022_setupStability

C Syntax

ViStatus hpe5022_setupStability(ViSession id, ViInt16 measFunc, ViInt16 segment, ViInt16 rev, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupStability(ByVal id As Long, ByVal measFunc As integer, ByVal segment As Integer, ByVal rev As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the stability measurement sequence to the specified test identifier. Refer to the “hpe5022_measureStability” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_stabilityData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measureStability”“hpe5022_measureStability” function.

- segment

  Description — Specifies the number of sector/segment per track.

  Direction — IN

  Values — Same as the ‘sector’ in the “hpe5022_measureStability”“hpe5022_measureStability” function.

- rev

  Description — Specifies the number of revolutions of measurement.

  Direction — IN

  Values — Same as the ‘rev’ in the “hpe5022_measureStability”“hpe5022_measureStability” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the stability measurement by the “hpe5022_measure”

function.

Direction	OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘measFunc’, ‘sector’ and/or ‘rev’ are out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureStability” on page 524

“hpe5022_stabilityData_Q” on page 531

“hpe5022_stabilityConfig” on page 511

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_stabilityData_Q

C Syntax

ViStatus hpe5022_stabilityData_Q(ViSession id, ViInt16 segment_no, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_stabilityData_Q(ByVal id As Long, ByVal segment_no As Integer, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the results of stability measurement. The returned data are set in array, and its size is specified by the ‘rev’ parameter in the “hpe5022_measureStability” or “hpe5022_setupStability” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- segment_no

  Description: Specifies the sector number to query the measurement result. The sector number begins at 0.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- data

  Description Returns the data (set in array) at the sector specified by the 'sectorNo'. The size of array is specified by the ‘rev’ parameter in the “hpe5022_measureStability” or “hpe5022_setupStability” function.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘sectorNo’ and/or ‘dataType’ are out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureStability” on page 524

“hpe5022_setupStability” on page 529

hpe5022_stability_Q

C Syntax

ViStatus hpe5022_stability_Q(ViSession id, ViInt16 segment_no, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_stability_Q(ByVal id As Long, ByVal segment_no As Integer, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) for a certain sector of the stability measurement. The data returned by the “hpe5022_stabilityData_Q” function is used for this statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- segment_no

  Description: Specifies the sector number to query the measurement result. The sector number begins at 0.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- mean

  Description Returns the mean value of the stability measurement.

  Direction OUT

- min

  Description Returns the minimum value of the stability measurement.

  Direction OUT

- max

  Description Returns the maximum value of the stability measurement.

  Direction OUT

- stdDev

  Description Returns the standard deviation value of the stability measurement.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘sectorNo’ and/or ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_stabilityData_Q” on page 531

hpe5022_stabilityStatistic_Q

C Syntax

ViStatus hpe5022_stabilityStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_stabilityStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (i.e, mean, minimum, maximum and standard deviation) of the specified data type for stability measurement. The data returned by “hpe5022_stabilityData_Q” function is used for statistic analysis of this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- mean

  Description — Returns the mean value of the specified data type of stability measurement.

  Direction — OUT

- min

  Description Returns the minimum value of the specified data type of stability measurement.

  Direction OUT

- max

  Description Returns the maximum value of the specified data type of stability measurement.

  Direction OUT

- stdDev

  Description Returns the standard deviation value of the specified data type of stability measurement.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_stabilityData_Q” on page 531

Stability Measurement (Absolute Time Setting)

This section describes the functions related with the stability measurement. This measurement allows you to observe the TAA, the PW and the narrow band TAA variations during many write-read iterations. Absolute time setting means that the period of write/read is defined by time.

hpe5022_stabilityAbsSenseModeConfig

C Syntax

ViStatus hpe5022_stabilityAbsSenseModeConfig(ViSession id, ViInt16 isMode);

Visual Basic Syntax

hpe5022_stabilityAbsSenseModeConfig(ByVal id As Long, ByVal isMode As Integer) As Long

Description

This function controls the direction of the sense current polarity for absolute stability measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- isMode

  Description: Specifies the sense current polarity at write operation. When “hpe5022_STABILITY_IS_POL_REV” is selected, the direction of sense current is opposite to the direction of write current. When “hpe5022_STABILITY_IS_OFF” is selected, the sense current is set to zero. Some head amplifier does not have the capability to operate at reverse and/or off sense current modes.

  This setting is used for the stability measurement instead of the setting of “hpe5022_senseCurrentPolarity” function. When this function is not executed, the setting of “hpe5022_senseCurrentPolarity” is used for the stability measurement (Absolute Time Setting).

  Direction: IN

  Preset value: hpe5022_STABILITY_IS_POL_NORM

  Values

Name	Value	Description
hpe5022_STABILITY_IS_POL_NORM	0	Same polarity
hpe5022_STABILITY_IS_POL_REV	1	Reverse polarity

Name	Value	Description
hpe5022_STABILITY_IS_OFF	2	Off Mode

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter specified by ‘isMode’ is invalid.
hpe5022_ERROR_NSUP_SENS_CURR_POL	The head does not support the reverse polarity and off modes. Select hpe5022_STABILITY_IS_POL_NORM for ‘isMode’ parameter.

See Also

“hpe5022_stabilityAbsSenseModeConfig_Q” on page 539

“hpe5022_senseCurrentPolarity” on page 340

hpe5022_stabilityAbsSenseModeConfig_Q

C Syntax

ViStatus hpe5022_stabilityAbsSenseModeConfig_Q(ViSession id, ViPInt16 isMode);

Visual Basic Syntax

hpe5022_stabilityAbsSenseModeConfig_Q(ByVal id As Long, ByRef isMode As Integer) As Long

Description

This function returns the specified sense current polarity of the absolute stability measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- isMode

  Description: Returns the specified sense current polarity during read operation.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_stabilityAbsSenseModeConfig” on page 537

hpe5022_stabilityAbsWritePatternConfig

C Syntax

ViStatus hpe5022_stabilityAbsWritePatternConfig(ViSession id, ViInt16 writePat);

Visual Basic Syntax

hpe5022_stabilityAbsWritePatternConfig(ByVal id As Long, ByVal writePat As Integer) As Long

Description

This function specifies the write data pattern to be written on the test track for stability measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writePat

  Description: Specifies the data pattern used for writing. When this parameter is set to hpe5022_PAT_DEFAULT, the write data pattern is the same as the pattern specified by “hpe5022_selectPattern” function.

  Direction: IN

  Preset value: hpe5022_PAT_DEFAULT

  Values

Name	Value	Description
hpe5022_PAT_DEFAULT	-1	Default pattern
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_REP_2	8	Repetition Pattern2
hpe5022_PAT_REP_3	9	Repetition Pattern3
hpe5022_PAT_REP_4	10	Repetition Pattern4
hpe5022_PAT_USER	20	User-defined Pattern

Name	Value	Description
hpe5022_PAT_USER_2	21	User-defined Pattern2
hpe5022_PAT_USER_3	22	User-defined Pattern3
hpe5022_PAT_USER_4	23	User-defined Pattern4
hpe5022_PAT_ERASE	101	Default Erase Pattern selected by the “hpe5022_eraseType” function
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter writePat is out of range.

See Also

“hpe5022_stabilityAbsWritePatternConfig_Q” on page 542

hpe5022_stabilityAbsWritePatternConfig_Q

C Syntax

ViStatus hpe5022_stabilityAbsWritePatternConfig_Q(ViSession id, ViPInt16 writePat);

Visual Basic Syntax

hpe5022_stabilityAbsWritePatternConfig_Q(ByVal id As Long, ByRef writePat As Integer) As Long

Description

This function returns the specified write data pattern written on the test track.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- writePat

  Description — Returns the specified write data pattern.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_stabilityAbsWritePatternConfig” on page 540

hpe5022_stabilityAbsWriteModeConfig

C Syntax

ViStatus hpe5022_stabilityAbsWriteModeConfig(ViSession id, ViInt16 writeMode);

Visual Basic Syntax

hpe5022_stabilityAbsWriteModeConfig(ByVal id As Long, ByVal writeMode As Integer) As Long

Description

This function controls the write current direction at the beginning of write operation.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- writeMode

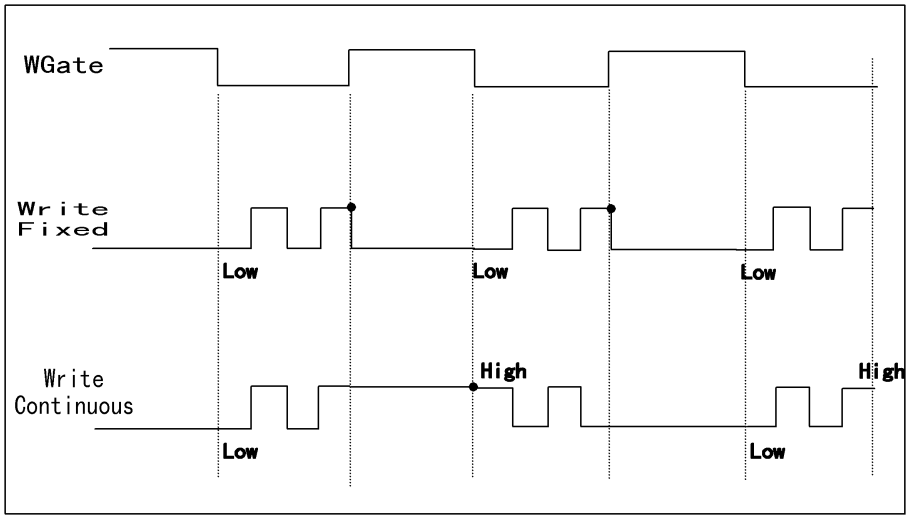
  Description Specifies the write current direction at the beginning of write operation.

  Direction IN

  Values

Name	Value	Description
hpe5022_WRIT_DATA_POL_FIXED	0	The write current direction at the beginning of each write operation is always the same.
hpe5022_WRIT_DATA_POL_CONTINUOUS	1	The write current direction at the beginning of write operation is the same as the last current direction of the previous write operation.

Figure 3-18 Write Mode



When [Write Mode] is set to “fixed”, the write current direction at the beginning of each write operation is always the same. On the other hand, when [Write Mode] is set to “continuous” the write current direction is the same as the last current direction of the previous write operation. Each time the write current alternates it allows you to check the polarity of the output pulse at the end of the previous write. You must set the repetitive period of the data pattern T to 1T, 2T, 4T and 8T due to the system’s internal constraints. If you use other repetitive periods of the data pattern, E5022A/B system can not guarantee the change of write current polarity. This function makes it possible for stability measurement to achieve an acceptable level of TAA under similar usage of the drive. Before this function was introduced, the write current direction always starts at a constant direction, thus you can only observe one side of the output.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writeMode’ is invalid.
hpe5022_ERROR_NSUP_CONF	The installed head amplifier has no capability to control the write current direction.

See Also

“hpe5022_stabilityAbsWriteModeConfig_Q” on page 545

hpe5022_stabilityAbsWriteModeConfig_Q

C Syntax

ViStatus hpe5022_stabilityAbsWritePatternConfig_Q(ViSession id, ViPInt16 writeMode);

Visual Basic Syntax

hpe5022_stabilityAbsWritePatternConfig_Q(ByVal id As Long, ByRef writeMode As Integer) As Long

Description

This function returns the specified write current direction at the beginning of write operation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writeMode

  Description: Returns the specified write current direction at the beginning of write operation.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_WRIT_DATA_POL_FIXED	0	The write current direction at the beginning of each write operation is always the same.
hpe5022_WRIT_DATA_POL_CONTINUOUS	1	The write current direction at the beginning of write operation is the same as the last current direction of the previous write operation.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writeMode’ is invalid.

See Also “hpe5022_stabilityAbsWriteModeConfig” on page 543

hpe5022_stabilityAbsSequenceConfig

C Syntax

ViStatus hpe5022_stabilityAbsSequenceConfig(ViSession id, ViInt16 seqType);

Visual Basic Syntax

hpe5022_stabilityAbsSequenceConfig(ByVal id As Long, ByVal seqType As Integer) As Long

Description

This function specifies the type of sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- seqType

  Description: Specifies the type of sequence. When the “hpe5022_SEQ_M” is selected, the erase and write sequences are not executed.

  Direction: IN

  Preset value: hpe5022_SEQ_ER_WR_M

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Excitation and Measurement
hpe5022_SEQ_M	2	Excitation and Measurement

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘seqType’ is out of range.

See Also

“hpe5022_measureStabilityAbs” on page 549

“hpe5022_setupStabilityAbs” on page 554

hpe5022_stabilityAbsSequenceConfig_Q

C Syntax

ViStatus hpe5022_stabilityAbsSequenceConfig_Q(ViSession id, ViPInt16 seqType);

Visual Basic Syntax

hpe5022_stabilityAbsSequenceConfig_Q(ByVal id As Long, ByRef seqType As Integer) As Long

Description

This function returns the type of sequence.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Returns the type of sequence.

  Direction — OUT

  Values — Same as the ‘seqType’ of “hpe5022_stabilitySequenceConfig”

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_stabilityAbsSequenceConfig” on page 547

hpe5022_measureStabilityAbs

C Syntax

ViStatus hpe5022_measureStabilityAbs(ViSession id, ViInt16 measFunc, ViReal64 writeTime, ViReal64 delay, ViReal64 readTime, ViInt16 counts);

Visual Basic Syntax

hpe5022_measureStabilityAbs(ByVal id As Long, ByVal measFunc As Integer, ByVal writeTime As Double, ByVal delay As Double, ByVal readTime As Double, ByVal counts As Integer) As Long

Description

This function measures the stability. This allows you to divide a track into some segments to get many write-read iterations. The sequence of this function is as follows:

1. Calculate the number of segments (segment) and revolutions (rev) from the specified write gate time, delay, read gate time and number of measurement counts.
2. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
3. If ‘seqType’ of “hpe5022_stabilityAbsSequenceConfig” function is set at “hpe5022_SEQ_ER_WR_M”, erase (same as the “hpe5022_erase” function) for an entire track.
4. Write the data pattern as specified by the “hpe5022_stabilityAbsWritePatternConfig” function.
5. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
6. Set i to zero. (i = 0): a number of revolutions.
7. Set j to zero. (j = 0): a number of segments.
8. Write the data pattern as specified by the “hpe5022_stabilityAbsWritePatternConfig” function during the write time period in the segment [j].
9. Wait for delay time to elapse.
10. Measure the specified parameter (specified by ‘measFunc’) for the read ratio period in the segment [j].
11. Increment j by 1. (j = j+1)
12. If j is less than ‘segment’ (see parameters), go to the step 8.
13. Increment i by 1. (i = i +1)
14. If j is less than ‘rev’ (see parameters), go to the step 7.

The “hpe5022_stabilityAbsData_Q” function returns the measurement results.

Parameters

- id

  Description Specifies the system identifier. This is given by the

Function Reference

"hpe5022_init" function.

Direction IN

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_PW	2	Measure PW
hpe5022_MEAS_NB_TAA	5	Measure Narrow Band TAA

- writeTime

  Description Specifies the time to write.

  Direction IN

  Unit Second

  Values

Name	Value
hpe5022_STABILITY_WRIT_TIME_MIN	1×10^{-6}
hpe5022_STABILITY_WRIT_TIME_MAX	10×10^{-3}

- delay

  Description Specifies the delay time. Delay time is defined as the time distance between write and read operation.

  Direction IN

  Unit Second

  Values

Name	Value
hpe5022_STABILITY_DEL_MIN	0
hpe5022_STABILITY_DEL_MAX	10×10^{-3}

- readTime

  Description Specifies the time to read. This parameter is dependent on

the selected “measFunc”.

Direction IN

Unit Second

Values

Name	Value
When the selected ‘MeasFunc’ is “hpe5022_MEAS_TAA” or “hpe5022_MEAS_PW”	
hpe5022_STABILITY_READ_TIME_MIN	20×10^{-6}
hpe5022_STABILITY_READ_TIME_MAX	10×10^{-3}
When the selected ‘MeasFunc’ is “hpe5022_MEAS_NB_TAA”	
hpe5022_STABILITY_NB_TAA_READ_TIME_MIN	800×10^{-6}
hpe5022_STABILITY_NB_TAA_READ_TIME_MAX	10×10^{-3}

- counts

  Description Specifies the number of test counts

  Direction IN

  Values

Number of Counts	Value
hpe5022_STABILITY_COUN_MIN ≤ **segment** × **rev** ≤ hpe5022_STABILITY_COUN_MAX	
When the selected ‘MeasFunc’ is “hpe5022_MEAS_TAA” or “hpe5022_MEAS_PW”	
hpe5022_STABILITY_COUN_MIN	1
hpe5022_STABILITY_COUN_MAX	1000
When the selected ‘MeasFunc’ is “hpe5022_MEAS_NB_TAA”	
hpe5022_STABILITY_NB_TAA_COUN_MIN	1
hpe5022_STABILITY_NB_TAA_COUN_MAX	100

NOTE There are certain gate timing restrictions that the user must compensate when the spindle speed (RPM) is set up. When RPM is set below 2400[rpm], the system will

automatically set the spindle speed to 2400[rpm].

Gate Time Limitations
track time = (60/rpm) -aper (aper is specified by the hpe5022_trackOffsetComPattern function.) aper = 0 when the stat of the hpe5022_trackOffset CompPattenState is set at VI_FALSE.
rpm = spindle speed (where spindle speed ≥ 2400 rpm) = 2400 rpm (where spindle speed < 2400)
(for TAA and PW Measurement) gateTime = writeTime +delay + readTime + 30 µsec.
(for Narrow Band TAA Measurement) gateTime = writeTime +delay + readTime + 100 µsec.
gateTime = (writeTime + 100 µsec) (Where gateTime < (writeTime + 100 µsec) gateTime = 150 µsec (Where gateTime < 150 µsec)

Return Values

Completion Code	**Description**
VI_SUCCESS	No Error

Error Code	**Description**
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘measFunc’, ‘writeTime’, ‘delay’, ‘readTime’and/or ‘counts’ are out of range.

Error Code	Description
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive is turned off. Turn it on before this function is executed. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when an overflow is detected while the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_driveState” on page 209

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_stabilityAbsSenseModeConfig” on page 537

“hpe5022_stabilityAbsSequenceConfig” on page 547

“hpe5022_stabilityAbsWritePatternConfig” on page 540

“hpe5022_stabilityAbsWriteModeConfig” on page 543

“hpe5022_stabilityAbsData_Q” on page 560

hpe5022_setupStabilityAbs

C Syntax

ViStatus hpe5022_setupStabilityAbs(ViSession id, ViInt16 measFunc, ViReal64 writeTime, ViReal64 delay, ViReal64 readTime, ViInt16 counts, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupStabilityAbs(ByVal id As Long, ByVal measFunc As integer, ByVal writeTime As Double, ByVal delay As Double, ByVal readTime As Double, ByVal counts As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the stability measurement sequence to the specified test identifier. Refer to the “hpe5022_measureStabilityAbs” function for details about the test sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function. The “hpe5022_stabilityAbsData_Q” function returns the measurement results.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_PW	2	Measure PW
hpe5022_MEAS_NB_TAA	5	Measure Narrow Band TAA

- writeTime

  Description Specifies the time to write.

  Direction IN

  Unit Second

  Values

Name	Value
hpe5022_STABILITY_WRIT_TIME_MIN	1×10^{-6}

Name	Value
hpe5022_STABILITY_WRIT_TIME_MAX	10×10^{-3}

- delay

  Description Specifies the delay time. Delay time is defined as the time distance between write and read operation.

  Direction IN

  Unit Second

  Values

Name	Value
hpe5022_STABILITY_DEL_MIN	0
hpe5022_STABILITY_DEL_MAX	10×10^{-3}

- readTime

  Description Specifies the time to read. This parameter is dependent on the selected “measFunc”.

  Direction IN

  Unit Second

  Values

Name	Value
When the selected ‘MeasFunc’ is “hpe5022_MEAS_TAA” or “hpe5022_MEAS_PW”	
hpe5022_STABILITY_READ_TIME_MIN	20×10^{-6}
hpe5022_STABILITY_READ_TIME_MAX	10×10^{-3}
When the selected ‘MeasFunc’ is “hpe5022_MEAS_NB_TAA”	
hpe5022_STABILITY_NB_TAA_READ_TIME_MIN	800×10^{-6}
hpe5022_STABILITY_NB_TAA_READ_TIME_MAX	10×10^{-3}

- counts

  Description Specifies the number of test counts

  Direction IN

Values

Number of Counts	**Value**
hpe5022_STABILITY_COUN_MIN ≤ **segment** × **rev** ≤ hpe5022_STABILITY_COUN_MAX	
When the selected ‘MeasFunc’ is “hpe5022_MEAS_TAA” or “hpe5022_MEAS_PW”	
hpe5022_STABILITY_COUN_MIN	1
hpe5022_STABILITY_COUN_MAX	1000
When the selected ‘MeasFunc’ is “hpe5022_MEAS_NB_TAA”	
hpe5022_STABILITY_NB_TAA_COUN_MIN	1
hpe5022_STABILITY_NB_TAA_COUN_MAX	100

- testHndl

  Description Returns the test identifier. This identifier is used to execute the stability measurement by the “hpe5022_measure” function.

  Direction OUT

Return Values

Completion Code	**Description**
VI_SUCCESS	No Error

Error Code	**Description**
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘measFunc’, ‘writeTime’, ‘delay’, ‘readTime’ and/or ‘counts’ are out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureStabilityAbs” on page 549

“hpe5022_stabilityAbsData_Q” on page 560

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_stabilityAbsDataSize_Q

C Syntax

ViStatus hpe5022_stabilityAbsDataSize_Q(ViSession id, ViPInt16 segment,ViPInt16 rev);

Visual Basic Syntax

hpe5022_stabilityAbsDataSize_Q(ByVal id As Long, ByRef segment As Integer, ByRef rev As Integer) As Long

Description

This function returns the number of segments and revolutions based on the calculated write gate time, delay, read gate time and number of data counts. The total number of data counts is computed by *segment x rev.*

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- segment

  Description: Returns the number of segments per revolution.

  Direction: OUT

- rev

  Description: Returns the number of revolutions.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The stability data is corrupt.

See Also

“hpe5022_measureStabilityAbs” on page 549

hpe5022_stabilityAbs_Q

C Syntax

ViStatus hpe5022_stabilityAbs_Q(ViSession id, ViInt16 segment_no, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_stabilityAbs_Q(ByVal id As Long, ByVal segment_no As Integer, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (i.e, mean, minimum, maximum and standard deviation) of the specified segment number.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- segment_no

  Description — Specifies the sector number to report the measurement results.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- mean

  Description Returns the mean value of the specified segment number.

  Direction OUT

- min

  Description Returns the minimum value of the specified segment number.

  Direction OUT

- max

  Description Returns the maximum value of the specified segment number.

  Direction OUT

- stdDev

  Description Returns the standard deviation of the specified segment number.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘segment_no’ and/or ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_stabilityData_Q” on page 531

hpe5022_stabilityAbsData_Q

C Syntax

ViStatus hpe5022_stabilityAbsData_Q(ViSession id, ViInt16 segment_no, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_stabilityAbsData_Q(ByVal id As Long, ByVal segment_no As Integer, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the data results in array order of the specified segment.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- segment_no

  Description: Specifies the segment number from where to get the data. The number of segment is reported by “hpe5022_stabilityAbsDataSize_Q” function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- data

  Description — Returns the data of the specified segment. Data size is equal to the number of revolutions, which is reported by “hpe5022_stabilityAbsDataSize_Q”.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameters ‘segment_no’ and/or ‘dataType’ are out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_measureStabilityAbs” on page 549

“hpe5022_setupStabilityAbs” on page 554

hpe5022_stabilityAbsStatistic_Q

C Syntax

ViStatus hpe5022_stabilityAbsStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_stabilityAbsStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics data of the stability measurement. Total data size is equal to *segment x rev.* The number of segments and revolutions are reported by the“hpe5022_stabilityAbsDataSize_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported for statistics analysis.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- mean

  Description: Returns the mean value of the stability measurement.

  Direction: OUT

- min

  Description Returns the minimum value of the stability measurement.

  Direction OUT

- max

  Description Returns the maximum value of the stability measurement.

  Direction OUT

- stdDev

  Description Returns the standard deviation value of the stability measurement.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_stabilityData_Q” on page 531

Narrow Band TAA Measurement

This section describes the functions related with narrow band track average amplitude (TAA) measurement. Narrow band TAA is measured by a spectrum analyzer.

hpe5022_narrowBandTaaFrequencyMode

C Syntax ViStatus hpe5022_narrowBandTaaFrequencyMode(ViSession id,ViInt16 mode);

VB Syntax hpe5022_narrowBandTaaFrequencyMode(ByVal id As Long, ByVal mode As Integer) As Long

Description This function specifies the narrow band TAA measurement frequency mode. When auto mode is selected, the frequency of the TAA measurement in the spectrum analyzer is set to the fundamental frequency of the selected data pattern. See chapter 5 in the Operaion Manual.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Specifies the narrow band TAA measurement mode.

  When “hpe5022_NB_TAA_FREQ_AUTO” is selected, the measurement frequency of the spectrum analyzer is automatically set to the fundamental frequency of the data pattern.

  When “hpe5022_NB_TAA_FREQ_FIXED” is selected, the measurement frequency is set to the frequency specified by the “hpe5022_narrowBandTaaFrequency” function.

  Direction: IN

  Preset Value: 0 (hpe5022_NB_TAA_FREQ_AUTO)

  Values

Name	Value	Description
hpe5022_NB_TAA_FREQ_AUTO	0	Auto mode.
hpe5022_NB_TAA_FREQ_FIXED	1	Fixed mode.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘mode’ is out of range.

See Also

“hpe5022_narrowBandTaaFrequency” on page 567

“hpe5022_narrowBandTaaFrequencyMode_Q” on page 566

hpe5022_narrowBandTaaFrequencyMode_Q

C Syntax

ViStatus hpe5022_narrowBandTaaFrequencyMode_Q(ViSession id,ViPInt16 mode);

VB Syntax

hpe5022_narrowBandTaaFrequencyMode_Q(ByVal id As Long, ByRef mode As Integer) As Long

Description

This function returns specified frequency mode of the narrow band TAA measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Returns the frequency mode of the narrow band TAA measurement.

  Direction: OUT

  Values: Same as ‘mode’ in the “hpe5022_narrowBandTaaFrequencyMode” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_narrowBandTaaFrequencyMode” on page 564

hpe5022_narrowBandTaaFrequency

C Syntax

ViStatus hpe5022_narrowBandTaaFrequency(ViSession id,ViReal64 center);

VB Syntax

hpe5022_narrowBandTaaFrequency(ByVal id As Long, ByVal center As Double) As Long

Description

This function specifies the measurement frequency of the spectrum analyzer for narrow band TAA measurement. The frequency is set automatically when the frequency mode is set to auto in the “hpe5022_narrowBandTaaFrequencyMode” function. The value is changed automatically depending on the data pattern and the channel bit rate when auto mode is selected.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- center

  Description — Specifies the center frequency to measure narrow band TAA.

  Direction — IN

  Unit — Hz

  Preset Value — 2.5×10^6

  Values — The upper and lower limit values are queryable by hpe5022_narrowBandTaaFrequencyRange_Q function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘center’ is out of range.

See Also

“hpe5022_narrowBandTaaFrequencyMode” on page 564

“hpe5022_narrowBandTaaFrequency_Q” on page 568

hpe5022_narrowBandTaaFrequency_Q

C Syntax

ViStatus hpe5022_narrowBandTaaFrequency_Q(ViSession id,ViPReal64 center);

VB Syntax

hpe5022_narrowBandTaaFrequency_Q(ByVal id As Long, ByRef center As Double) As Long

Description

This function returns the specified center frequency of the narrow band TAA.

This value is automatically overwritten depending on the data pattern and the channel bit rate when the frequency mode is set to auto in the “hpe5022_narrowBandTaaFrequencyMode” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- center

  Description — Returns the specified center frequency.

  Direction — OUT

  Unit — Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_narrowBandTaaFrequency”

“hpe5022_narrowBandTaaFrequencyMode”

hpe5022_narrowBandTaaFrequencyRange_Q

C Syntax

ViStatus _ VI_FUNC hpe5022_narrowBandTaaFrequencyRange_Q(ViSession id,ViPReal64 freqMin, ViPReal64 freqMAx);

Visual Basic Syntax

Declare Function _ hpe5022_narrowBandTaaFrequencyRange_Q Lib “hpe5022_32.dll” (_ By Val id As Long, _ ByRef freqMin As Double, _ ByRef freqMin As Double _) As Long

Description

This function returns the frequency range of the narrow band TAA measurement.

This value is automatically overwritten depending on the data pattern and the channel bit rate when the frequency mode is set to auto in the “hpe5022_narrowBandTaaFrequencyMode” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- freqMin — Returns the minimum frequency in Hz.
- freqMax — Returns the maximum frequency in Hz.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

hpe5022_narrowBandTaaBandWidth

C Syntax ViStatus hpe5022_narrowBandTaaBandWidth(ViSession id, ViReal64 bw);

VB Syntax hpe5022_narrowBandTaaBandWidth(ByVal id As Long, ByVal bw As Double) As Long

Description This function specifies the resolution band width of the spectrum analyzer for narrow band TAA measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bw

  Description: Specifies the resolution band width. Narrowing this value, will widen the dynamic range of the spectrum analyzer.

  Direction: IN

  Unit: Hz

  Preset Value: 10×10^3

  Values

Name	Value
hpe5022_NB_TAA_BW_MIN	3×10^3
hpe5022_NB_TAA_BW_MAX	100×10^3

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘bw’ is out of range.

See Also “hpe5022_narrowBandTaaBandWidth_Q” on page 571

hpe5022_narrowBandTaaBandWidth_Q

C Syntax

ViStatus hpe5022_narrowBandTaaBandWidth_Q(ViSession id, ViPReal64 bw);

VB Syntax

hpe5022_narrowBandTaaBandWidth(ByVal id As Long, ByRef bw As Double) As Long

Description

This function returns the resolution band width of the spectrum analyzer for narrow band TAA measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bw

  Description: Returns the resolution band width.

  Direction: IN

  Unit: Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_narrowBandTaaBandWidth” on page 570

hpe5022_measureNarrowBandTaa

C Syntax

ViStatus hpe5022_measureNarrowBandTaa(ViSession id,ViInt16 seqType, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureNarrowBandTaa(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer) As Long

Description

This function measures the narrow band TAA parameter according to the specified sequence type. There are three sequence types: 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The sequence is as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If seqType is set to “hpe5022_SEQ_ER_WR_M”, erase the entire track.
3. If seqType is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write the data pattern specified by the “hpe5022_selectPattern” function.
4. Move the head to read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Repeat the steps 3 and 4‘ave’ (see parameters) times if it is more than 1.

The “hpe5022_narrowBandTaa_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measurement
hpe5022_SEQ_WR_M	1	Write->Measurement
hpe5022_SEQ_M	2	Measurement

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

Values

Name	Value
hpe5022_NB_TAA_COUN_MIN	1
hpe5022_NB_TAA_COUN_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_narrowBandTaa_Q” on page 577

“hpe5022_driveState” on page 209

hpe5022_setupNarrowBandTaa

C Syntax

ViStatus hpe5022_setupNarrowBandTaa(ViSession id, ViInt16 seqType, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupNarrowBandTaa(ByVal id As Long, ByVal seqType As Integer, ByVal ave As Integer, ByVal testHndl As long) As Long

Description

This function assigns the narrow band TAA measurement sequence to the specified test identifier. See the “hpe5022_measureNarrowBandTaa” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function

The “hpe5022_narrowBandTaa_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measureNarrowBandTaa” function.

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as the ‘ave’ in the “hpe5022_measureNarrowBandTaa” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the narrow band TAA measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureNarrowBandTaa” on page 572

“hpe5022_narrowBandTaa_Q” on page 577

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_calibrateNarrowBandTaa

C Syntax

ViStatus_VI_FUNC hpe5022_calibrateNarrowBandTaa(ViSession id);

Visual Basic Syntax

Declare Function _ hpe5022_calibrateNarrowBandTaa Lib “hpe5022_32.dll (_
ByVal id As Long _) As Long

Description

This function performs the narrow band TAA measurement calibration.

This function is supported only when the SA module is the hybrid system of E5040 and 3GHz Spectrum Analyzer (E4402B).

Connect a cable between front panel connector AMPTD REF OUT and the INPUT connector of the E4402B before this function is called. If the two connectors are connected when this function is called, hpe5022_ERROR_CAL_FAILED error is generated.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_NSUP_FUNC	The SA module in use does not support this function.
hpe5022_ERROR_CAL_FAILED	Calibration failed.

hpe5022_narrowBandTaa_Q

C Syntax

ViStatus hpe5022_narrowBandTaa_Q(ViSession id,ViPReal64 taa);

Visual Basic Syntax

hpe5022_narrowBandTaa_Q(ByVal id As Long, ByRef taa As Double) As Long

Description

This function returns the measurement results of narrow band TAA.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- taa

  Description: Returns the result of narrow band TAA.

  Direction: OUT

  Unit: RMS Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_DATA_CORRUPT	The narrow band TAA data is corrupt. Check if your measurement sequence is correct.

See Also

"hpe5022_measureNarrowBandTaa" on page 572

"hpe5022_setupNarrowBandTaa" on page 574

hpe5022_narrowBandTaaDataSize_Q

C Syntax

ViStatus hpe5022_narrowBandTaaDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_narrowBandTaaDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the data size of the narrow band TAA from the “hpe5022_narrowBandTaaData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- size

  Description: Returns the data size of the narrow band TAA.

  The size will be the same as the number of measurement points. as the value of ‘ave’ specified by the “hpe5022_measureNarrowBandTaa” or “hpe5022_setupNarrowBandTaa” function if the measurement is done properly.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The narrow band TAA data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_narrowBandTaaData_Q”

hpe5022_narrowBandTaaData_Q

C Syntax

ViStatus hpe5022_narrowBandTaaData_Q(ViSession id,ViReal64 data[]);

Visual Basic Syntax

hpe5022_narrowBandTaaData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the TAA data for each revolution. When the ‘ave’ parameter in the “hpe5022_measureNarrowBandTaa” or the “hpe5022_setupNarrowBandTaa” function is set to more than one, the function allows you to get the TAA data for each revolution. If the ‘ave’ is set to 1, the output of this function is the same as one of the “hpe5022_narrowBandTaa_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the data array of the parameter specified by the ‘dataType’. The array size is returned by the “hpe5022_narrowBandTaaDataSize_Q” function.

  Direction: OUT

  Unit: RMS Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The narrow band TAA data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_narrowBandTaaDataSize_Q” on page 578

“hpe5022_narrowBandTaa_Q” on page 577

“hpe5022_measureNarrowBandTaa” on page 572

“hpe5022_setupNarrowBandTaa” on page 574

hpe5022_narrowBandTaaStatistic_Q

C Syntax

ViStatus hpe5022_narrowBandTaaStatistic_Q(ViSession id,ViPReal64 mean, ViPReal64 min,ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_narrowBandTaaStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the narrow band TAA measurement. The data returned by the “hpe5022_narrowBandTaa_Q” function is used for statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

  Unit: RMS Volt

- mean

  Description: Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_narrowBandTaa_Q” function.

  Direction: OUT

  Unit: RMS Volt

- min

  Description: Returns the minimum value.

  Direction: OUT

  Unit: RMS Volt

- max

  Description: Returns the maximum value.

  Direction: OUT

  Unit: RMS Volt

- stdDev

  Description: Returns the standard deviation value.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The narrow band TAA data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_narrowBandTaaData_Q” on page 579

hpe5022_narrowBandTaaRawDataSize_Q

C Syntax

ViStatus hpe5022_narrowBandTaaRawDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_narrowBandTaaRawDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the data returned by “hpe5022_narrowBandTaaRawData_Q” function

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the array size of narrow band TAA raw data.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_NSUP_FUNC	The 4395A is used in your system. This function is only for E5040A.

See Also

“hpe5022_narrowBandTaaRawData_Q” on page 583

hpe5022_narrowBandTaaRawData_Q

C Syntax

ViStatus hpe5022_narrowBandTaaRawData_Q(ViSession id, ViReal64 sec[], ViReal64 data[]);

Visual Basic Syntax

hpe5022_narrowBandTaaRawData_Q(ByVal id As Long, ByRef sec As Double, ByRef data As Double) As Long

Description

This function returns the narrow band TAA raw data.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- sec

  Description — Returns the time point for each narrow band TAA raw data. The size of array is returned by the “hpe5022_narrowBandTaaRawDataSize_Q”.

  Direction — OUT

  Unit — Second

- data

  Description — Returns the narrow band TAA raw data. The size of array is returned by the “hpe5022_narrowBandTaaRawDataSize_Q”. Each value in array is the measurement result relative to the time point in the sec array.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

Error Code	Description
hpe5022_ERROR_NSUP_FUNC	The 4395A is used in your system. This function is only for E5040A.

See Also “hpe5022_narrowBandTaaRawDataSize_Q” on page 582

hpe5022_narrowBandTaaRawStatistic_Q

C Syntax

ViStatus hpe5022_narrowBandTaaRawStatistic_Q(ViSession id, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_narrowBandTaaRawStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function returns the narrow band TAA raw data.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- mean

  Description — Returns the mean value of the narrow band TAA raw data returned by the “hpe5022_narrowBandTaaRawData_Q”.

  Direction — OUT

- min

  Description — Returns the minimum value of the narrow band TAA raw data returned by the “hpe5022_narrowBandTaaRawData_Q”.

  Direction — OUT

- max

  Description — Returns the maximum value of the narrow band TAA raw data returned by the “hpe5022_narrowBandTaaRawData_Q”.

  Direction — OUT

- stdDev

  Description — Returns the standard deviation value of the narrow band TAA raw data returned by the “hpe5022_narrowBandTaaRawData_Q”.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_NSUP_FUNC	The 4395A is used in your system. This function is only for E5040A.

See Also “hpe5022_narrowBandTaaRawData_Q” on page 583

Absolute Polarity Function

This test is used to determine the output polarity and asymmetry of the isolated pulse TAA and 1-3 bit patterns. Absolute polarity test is used to determine the pinned layer reversal of the GMR head.

hpe5022_absolutePolarityPatternMode

C Syntax

ViStatus hpe5022_absolutePolarityPatternMode(ViSession id, ViInt16 mode);

Visual Basic Syntax

hpe5022_absolutePolarityPatternMode(ByVal id As Long, ByVal mode As Integer) As Long

Description

This function controls the configuration of the data pattern for absolute polarity measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

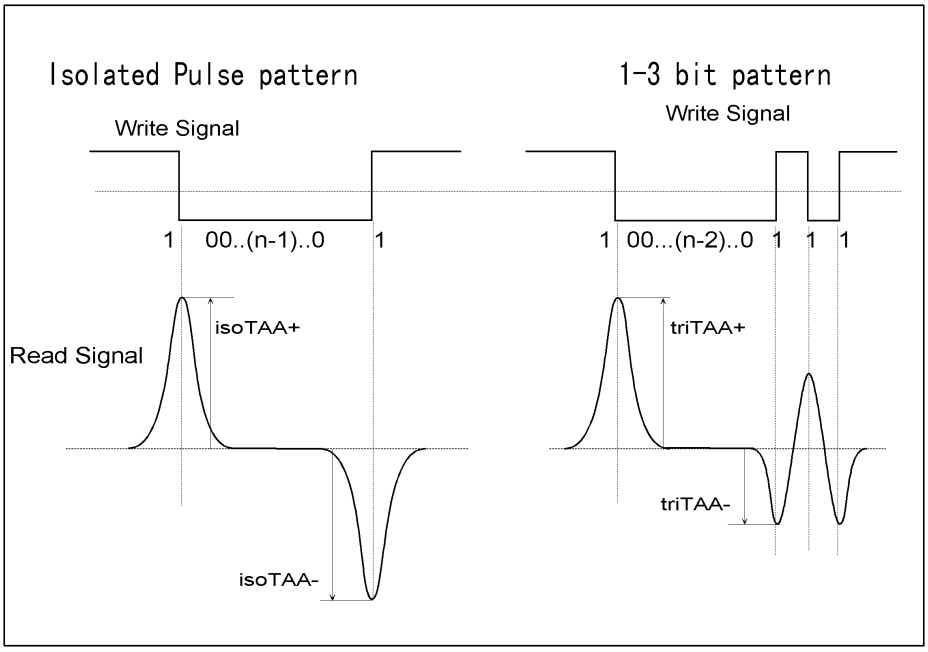
  Description: Specifies the configuration type of the data pattern.

  Direction: IN

  Values

Name	Value	Description
hpe5022_ABS_POL_PAT_MODE_AUTO	0	1-3 bit pattern
hpe5022_ABS_POL_PAT_MODE_USER	1	user-defined pattern

Figure 3-19 1-3 bit pattern



When “hpe5022_ABS_POL_PAT_MODE_AUTO” is selected, the absolute polarity pattern will be defined automatically as the combination of isolated pulse and tribit. As shown in the figure above 1-3 bit pattern is the combination of isolated pulse and three consecutive bits (i.e, tribit). Where n denotes the repetitive period specified in the “hpe5022_isolatedPulsePattern” function. When “hpe5022_ABS_POL_PAT_MODE_USER” is selected, the absolute polarity pattern will be defined by “hpe5022_absolutePolarityPattern” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘mode’ is out of range.

See Also

“hpe5022_absolutePolarityPatternMode_Q” on page 589

hpe5022_absolutePolarityPatternMode_Q

C Syntax

ViStatus hpe5022_absolutePolarityPatternMode_Q(ViSession id, ViPInt16 mode);

Visual Basic Syntax

hpe5022_absolutePolarityPatternMode_Q(ByVal id As Long, ByRef mode As Integer) As Long

Description

This function returns the specified configuration of the data pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mode

  Description: Returns the type of data pattern for absolute polarity measurement.

  Direction: OUT

  Values

Name	Value	Description
hpe5022_ABS_POL_PAT_MODE_AUTO	0	1-3 bit pattern
hpe5022_ABS_POL_PAT_MODE_USER	1	user-defined pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘mode’ is out of range.

See Also

“hpe5022_absolutePolarityPatternMode” on page 587

hpe5022_absolutePolarityPattern

C Syntax

ViStatus hpe5022_absolutePolarityPattern(ViSession id, ViInt16 dataForm, ViInt32 bitLength, const ViChar data []);

Visual Basic Syntax

hpe5022_absolutePolarityPattern(ByVal id As Long, ByVal dataForm As Integer, ByVal bitLength As Long, ByVal data As String) As Long

Description

This function specifies the user data pattern for absolute polarity measurement. When the absolute polarity pattern mode is “hpe5022_ABS_POL_PAT_MODE_AUTO”, this function is of no effect. If you want to use the user-defined pattern for absolute polarity test, change the absolute polarity pattern mode to “USER”, then call this function to define the user-defined pattern.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataForm

  Description — Specifies the data format.

  Direction — IN

  Values

Name	Value	Description
hpe5022_USER_DATA_HEX	0	Hexadecimal format (0-F or 0-f)
hpe5022_USER_DATA_BIN	1	Binary format (0 or 1)

- bitLength

  Description — Specifies the bit length of the user data pattern.

  Direction — IN

  Values

Name	Value
hpe5022_PAT_LENGTH_MIN	0
hpe5022_PAT_LENGTH_MAX	32,768

- data

  Description — Specifies the user data pattern. When hexadecimal format is used, the bit sequence of the user data pattern is generated as

MSB (most significant bit) first and LSB (least significant bit) last.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataForm’, ‘bitLength’ and/or ‘data’ is out of range.
hpe5022_ERROR_INV_DATA_SIZE	The specified user data pattern is shorter than the use data bit length.
hpe5022_ERROR_INV_DATA_TYPE	The use data is specified by an invalid character.

See Also

“hpe5022_absolutePolarityPattern_Q” on page 592

hpe5022_absolutePolarityPattern_Q

C Syntax

ViStatus hpe5022_absolutePolarityPattern_Q(ViSession id, ViInt16 dataForm, ViPInt32 bitLength, const ViChar data []);

Visual Basic Syntax

hpe5022_absolutePolarityPattern_Q(ByVal id As Long, ByVal dataForm As Integer, ByRef bitLength As Long, ByVal data As String) As Long

Description

This function returns the specified user data pattern for absolute polarity measurement. When the selected pattern mode is “hpe5022_ABS_POL_PAT_MODE_AUTO” this function will return the 1-3 bit pattern.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataForm

  Description Specifies the data format.

  Direction IN

  Values

Name	Value	Description
hpe5022_USER_DATA_HEX	0	Hexadecimal format (0-F or 0-f)
hpe5022_USER_DATA_BIN	1	Binary format (0 or 1)

- bitLength

  Description Returns the bit length of the user data pattern.

  Direction OUT

  Values

Name	Value
hpe5022_PAT_LENGTH_MIN	0
hpe5022_PAT_LENGTH_MAX	32,768

- data

  Description Returns the user data pattern.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataForm’ is out of range.

See Also

“hpe5022_absolutePolarityPattern” on page 590

hpe5022_measureAbsolutePolarity

C Syntax

ViStatus hpe5022_measureAbsolutePolarity(ViSession id, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureAbsolutePolarity(ByVal id As Long, ByVal ave As Integer) As Long

Description

This function measures absolute polarity.

Parameters

1. Move the head to the write track offset as specified by the “hpe5022_writeTrackOffset” function.
2. Execute a erase (same as the “hpe5022_erase” function) for an entire track.
3. Write an isolated pulse pattern as specified by the “hpe5022_isolatedPulsePattern” function.
4. Move the head to the read track offset as specified by “hpe5022_readTrackOffset” function.
5. Measure the TAA of the isolated pulse.
6. Move the head to the write track offset as specified by the “hpe5022_writeTrackOffset” function.
7. Execute a erase (same as the “hpe5022_erase” function) for an entire track.
8. Write the user-defined polarity pattern or 1-3 bit pattern.
9. Move the head to the read track offset as specified by “hpe5022_readTrackOffset” function.
10. Measure Tribit-TAA.
11. Iterate steps 1 to 10 for averaging.

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- ave

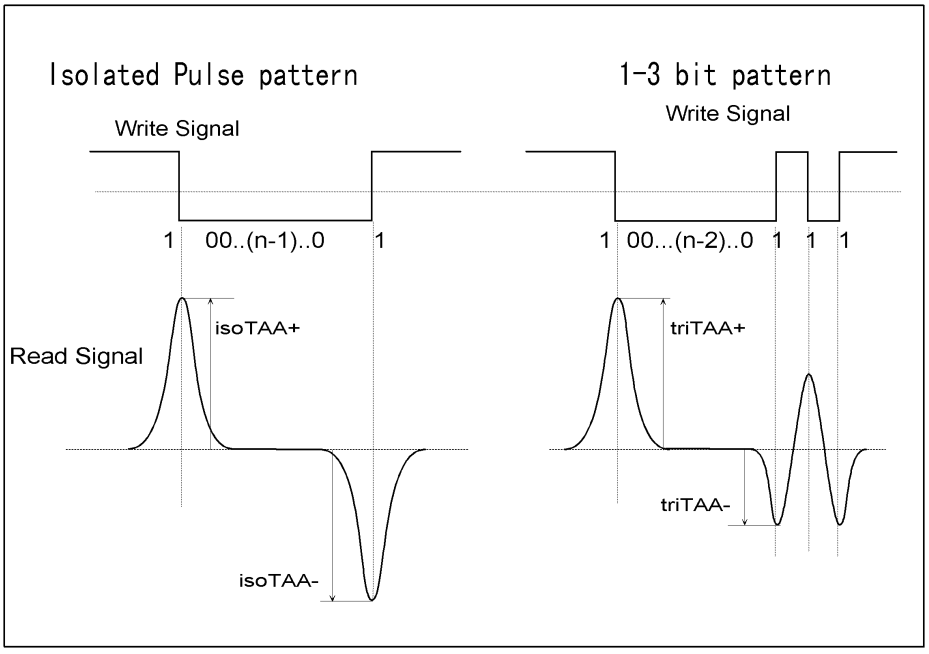
  Description Specifies the number of average counts.

  Direction IN

  Values

Name	Value
hpe5022_ABS_POL_COUN_MIN	1
hpe5022_ABS_POL_COUN_MAX	50

Figure 3-20 Absolute Polarity Pattern



Absolute polarity pattern is a combination of isolated pulse and three consecutive bits (tribit) as shown in the figure above. Three consecutive bits are written over the isoTAA-. Where n denotes the number of repetitive periods specified in the “hpe5022_isolatedPulsePattern” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘sector’ and/or ‘rev’ is out of range.

Error Code	Description
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before this function is executed. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_setupAbsolutePolarity” on page 597

“hpe5022_absolutePolarity_Q” on page 599

“hpe5022_absolutePolarityIsolatedPulseTaa_Q” on page 601

“hpe5022_absolutePolarityIsolatedPulseTaaDataSize_Q” on page 603

“hpe5022_absolutePolarityIsolatedPulseTaaData_Q” on page 604

hpe5022_setupAbsolutePolarity

C Syntax

ViStatus hpe5022_setupAbsolutePolarity(ViSession id, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupAbsolutePolarity(ByVal id As Long, ByVal ave As Integer, ByRef testHndl as Long) As Long

Description

This function assigns the absolute polarity measurement sequence to test identifier. Refer to the “hpe5022_measureAbsolutePolarity” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function. Moreover, the measurement result is returned by the “hpe5022_absolutePolarity_Q”.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- ave

  Description: Specifies the number of measurement revolutions to be averaged.

  Direction: IN

  Values

Name	Value
hpe5022_ABS_POL_COUN_MIN	1
hpe5022_ABS_POL_COUN_MAX	50

- testHndl

  Description: Returns the test handle.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Can not allocate enough memory.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_absolutePolarity_Q” on page 599

hpe5022_absolutePolarity_Q

C Syntax

ViStatus hpe5022_absolutePolarity_Q(ViSession id, ViPReal64 abs_pol, ViPReal64 iso_taa);

Visual Basic Syntax

hpe5022_absolutePolarity_Q(ByVal id As Long, ByRef abs_pol As Double, ByRef iso_taa as Double) As Long

Description

This function returns the measured data of the absolute polarity.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- abs_pol

  Description: Returns the measured absolute polarity.

  Direction: OUT

  Values

Equation 3-2 Absolute Polarity

$$AbsPol = \frac{\left(\frac{TribitTAA_{Pos}}{IsoTAA_{Pos}}\right) - \left(\frac{TribitTAA_{Neg}}{IsoTAA_{Neg}}\right)}{\left(\frac{TribitTAA_{Pos}}{IsoTAA_{Pos}}\right) + \left(\frac{TribitTAA_{Neg}}{IsoTAA_{Neg}}\right)}$$

- iso_taa

  Description: Returns the measured isolated TAA.

  Unit: Volts

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The returned abs_pol and iso_taa data are corrupt.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_setupAbsolutePolarity” on page 597

hpe5022_absolutePolarityIsolatedPulseTaa_Q

C Syntax

ViStatus hpe5022_absolutePolarityIsolatedPulseTaa_Q(ViSession id, ViPReal64 taa, ViPReal64 taaPos, ViPReal64 taaNeg, ViPReal64 taaAsym);

Visual Basic Syntax

hpe5022_absolutePolarityIsolatedPulseTaa_Q(ByVal id As Long, ByRef taa As Double, ByRef taaPos as Double, ByRef taaNeg As Double, ByRef taaAsym As Double) As Long

Description

This function returns the measurement data of the Isolated Pulse TAA.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- taa

  Description — Returns the TAA of isolated pulse.

  Unit — Volts

  Direction — OUT

- taaPos

  Description — Returns the TAA+ of isolated pulse.

  Unit — Volts

  Direction — OUT

- taaNeg

  Description — Returns the TAA- of isolated pulse.

  Unit — Volts

  Direction — OUT

- taaAsym

  Description — Returns the TAA Asymmetry of isolated pulse.

  Unit — Volts

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The taa, taaPos, taaNeg and taaAsym data are corrupt.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_absolutePolarityIsolatedPulseTaaDataSize_Q” on page 603

“hpe5022_absolutePolarityIsolatedPulseTaaData_Q” on page 604

“hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q” on page 606

hpe5022_absolutePolarityIsolatedPulseTaaDataSize_Q

C Syntax

ViStatus hpe5022_absolutePolarityIsolatedPulseTaaDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_absolutePolarityIsolatedPulseTaaDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the TAA data size of the Isolated Pulse.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the TAA data size of the Isolated Pulse.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The returned data size is corrupt.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_absolutePolarityIsolatedPulseTaa_Q” on page 601

“hpe5022_absolutePolarityIsolatedPulseTaaData_Q” on page 604

“hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q” on page 606

hpe5022_absolutePolarityIsolatedPulseTaaData_Q

C Syntax

ViStatus hpe5022_absolutePolarityIsolatedPulseTaaData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_absolutePolarityIsolatedPulseTaaData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the specified data type of the Isolated Pulse TAA.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the TAA data type of the Isolated Pulse. Four data types can be selected from the table below.

  Values

Name	Value	Description
hpe5022_DATA_TAA	0	TAA
hpe5022_DATA_TAA_POS	1	TAA +
hpe5022_DATA_TAA_NEG	2	TAA -
hpe5022_DATA_TAA_ASYM	3	TAA Asymmetry

  Direction: IN

- data

  Description: Returns the TAA data of the Isolated Pulse.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The returned data is corrupt.

See Also

“hpe5022_absolutePolarityIsolatedPulseTaa_Q” on page 601

“hpe5022_absolutePolarityIsolatedPulseTaaDataSize_Q” on page 603

“hpe5022_absolutePolarityIsolatedPulseTaaData_Q” on page 604

“hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q” on page 606

hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q

C Syntax

ViStatus hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics of the resulting TAA data of the Isolated Pulse.

Parameters

-

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies TAA data of the Isolated Pulse. Four data types can be selected from the table below.

  Values

Name	Value	Description
hpe5022_DATA_TAA	0	TAA
hpe5022_DATA_TAA_POS	1	TAA +
hpe5022_DATA_TAA_NEG	2	TAA -
hpe5022_DATA_TAA_ASYM	3	TAA Asymmetry

  Direction: OUT

- mean

  Description: Returns the mean value of the selected TAA data of the Isolated Pulse.

  Direction: OUT

- min

  Description: Returns the min value of the selected TAA data of the Isolated Pulse.

  Direction: OUT

- max

  Description: Returns the max value of the selected TAA data of the Isolated Pulse.

Direction OUT

- stdDev

  Description Returns the standard deviation value of the selected TAA data of the Isolated Pulse.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The returned mean, min, max and stdDev data are corrupt.

See Also

“hpe5022_absolutePolarityIsolatedPulseTaa_Q” on page 601

“hpe5022_absolutePolarityIsolatedPulseTaaDataSize_Q” on page 603

“hpe5022_absolutePolarityIsolatedPulseTaaData_Q” on page 604

hpe5022_absolutePolarityTribitTaa_Q

C Syntax

ViStatus hpe5022_absolutePolarityTribitTaa_Q(ViSession id, ViPReal64 taa, ViPReal64 taaPos, ViPReal64 taaNeg, ViPReal64 taaAsym);

Visual Basic Syntax

hpe5022_absolutePolarityTribitTaa_Q(ByVal id As Long, ByRef taa As Double, ByRef taaPos As Double, ByRef taaNeg As Double, ByRef taaAsym As Double) As Long

Description

This function returns the TAA measurement data of Tribit.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- taa

  Description: Returns the TAA data of Tribit.

  Unit: Volt

  Direction: OUT

- taaPos

  Description: Returns the positive TAA data of Tribit.

  Unit: Volt

  Direction: OUT

- taaNeg

  Description: Returns the negative TAA data of Tribit.

  Unit: Volt

  Direction: OUT

- taaAsym

  Description: Returns the TAA Asymmetry data of Tribit.

  Unit: Volt

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The returned taa, taaPos, taaNeg and taaAsym data are corrupt.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_absolutePolarityTribitTaaDataSize_Q” on page 610

“hpe5022_absolutePolarityTribitTaaData_Q” on page 611

“hpe5022_absolutePolarityTribitTaaStatistic_Q” on page 613

hpe5022_absolutePolarityTribitTaaDataSize_Q

C Syntax

ViStatus hpe5022_absolutePolarityTribitTaaDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_absolutePolarityTribitTaaDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the TAA data size of Tribit.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- size

  Description: Returns the TAA data size of Tribit.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The returned absolute polarity data is corrupt.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_setupAbsolutePolarity” on page 597

“hpe5022_absolutePolarityTribitTaa_Q” on page 608

“hpe5022_absolutePolarityTribitTaaData_Q” on page 611

“hpe5022_absolutePolarityTribitTaaStatistic_Q” on page 613

hpe5022_absolutePolarityTribitTaaData_Q

C Syntax

ViStatus hpe5022_absolutePolarityTribitTaaData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_absolutePolarityTribitTaaData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the specified TAA data type of Tribit.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataType

  Description Specifies the type of TAA data of Tribit. Four data types can be selected from the table below.

  Values

Name	Value	Description
hpe5022_DATA_TAA	0	TAA
hpe5022_DATA_TAA_POS	1	TAA +
hpe5022_DATA_TAA_NEG	2	TAA -
hpe5022_DATA_TAA_ASYM	3	TAA Asymmetry

  Direction IN

- data

  Description Returns the TAA data of Tribit.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The returned absolute polarity data is corrupt.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_setupAbsolutePolarity” on page 597

“hpe5022_absolutePolarityTribitTaa_Q” on page 608

“hpe5022_absolutePolarityTribitTaaDataSize_Q” on page 610

“hpe5022_absolutePolarityTribitTaaStatistic_Q” on page 613

hpe5022_absolutePolarityTribitTaaStatistic_Q

C Syntax

ViStatus hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q(ViSession id, ViInt16 dataType, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_absolutePolarityIsolatedPulseTaaStatistic_Q(ByVal id As Long, ByVal dataType As Integer, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function returns the statistics of the TAA data of Tribit.

Parameters

-

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of TAA data of Tribit. Four data types can be selected from the table below.

  Values

Name	Value	Description
hpe5022_DATA_TAA	0	TAA
hpe5022_DATA_TAA_POS	1	TAA +
hpe5022_DATA_TAA_NEG	2	TAA -
hpe5022_DATA_TAA_ASYM	3	TAA Asymmetry

  Direction: IN

- mean

  Description: Returns the mean value of the selected TAA data of Tribit.

  Direction: OUT

- min

  Description: Returns the min value of the selected TAA data of Tribit.

  Direction: OUT

- max

  Description: Returns the max value of the selected TAA data of Tribit.

  Direction: OUT

- stdDev

Description Returns the standard deviation value of the selected TAA data of Tribit.

Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The returned absolute polarity data is corrupt.

See Also

“hpe5022_measureAbsolutePolarity” on page 594

“hpe5022_setupAbsolutePolarity” on page 597

“hpe5022_absolutePolarityTribitTaa_Q” on page 608

“hpe5022_absolutePolarityTribitTaaDataSize_Q” on page 610

“hpe5022_absolutePolarityTribitTaaData_Q” on page 611

Track Profile Measurement Function

This section describes the functions of the track profile measurement. The track profile measurement observes the read track offset capability of the head. This measurement is performed by a parametric module, the narrow band TAA measurement is executed by the spectrum analyzer.

hpe5022_measureTrackProfile

C Syntax

ViStatus hpe5022_measureTrackProfile(ViSession id, ViInt16 seqType, ViInt16 mode, ViInt16 measFunc, ViInt16 points, const ViReal64 offset[]);

Visual Basic Syntax

hpe5022_measureTrackProfile(ByVal id As Long, ByVal seqType As Integer, ByVal mode As Integer, ByVal measFunc As Integer, ByVal points As Integer, ByRef offset As Double) As Long

Description

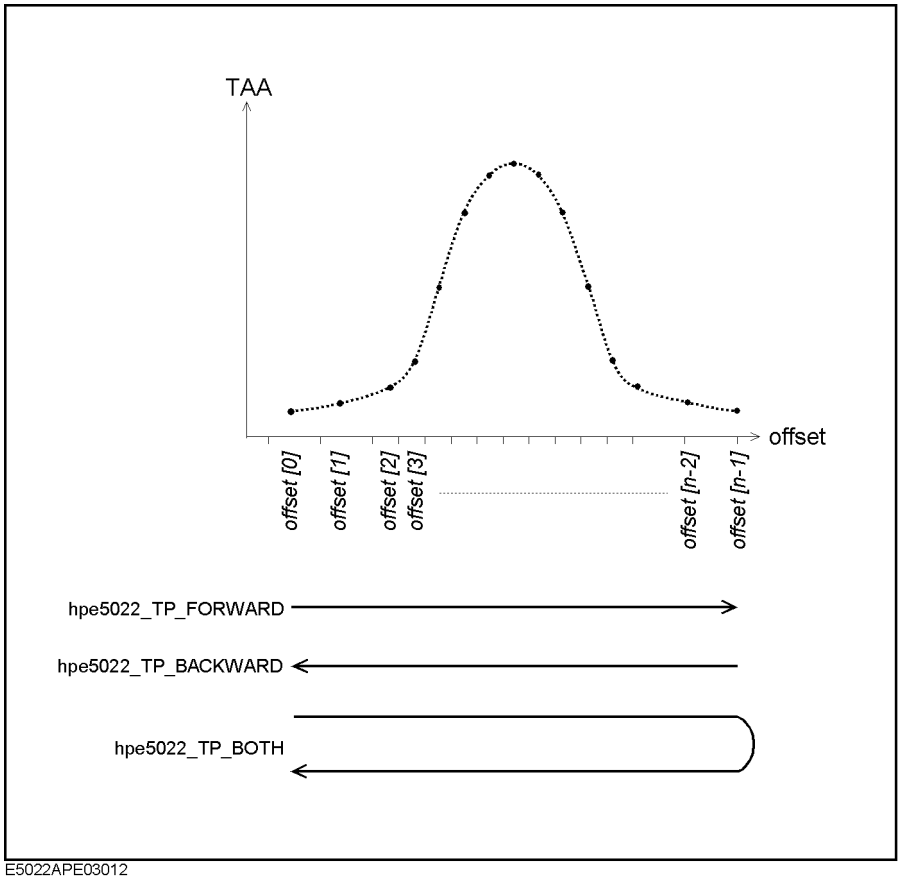
This function measures the TAA parameters according to the specified sequence type. There are three sequence types: 1) erase, write a data, then measure 2) not erase, write a data, then measure 3) only measure. The detailed sequence are as follows:

1. If seqType is set to “hpe5022_SEQ_ER_WR_M”, perform the three-track-erase. (the entire track erase for 5 positions, the track, its both side tracks, and the center between them.)
2. If seqType is set to “hpe5022_SEQ_WR_M”, write the data pattern at the track center. The data pattern is specified by the “hpe5022_selectPattern” function.
3. Set i to zero (i = 0)
4. Set the read track offset from the track center at ‘offset[i]’ (see parameters).
5. Move the head to the read track offset.
6. Measure a specified parameter.
7. Increment i by 1. (i = i+1).
8. If i is less than ‘points’ (see parameters), go to the step 4.
9. Set i to ‘points’ (see parameters)
10. Decrement i by 1. (i = i-1).
11. Set the read track offset from the track center to ‘offset[i]’
12. Move the head to the read track offset.
13. Measure a specified parameter.
14. If i is not equal to zero, go to the step 10.

If ‘mode’ (see parameters) is set to “hpe5022_TP_FORWARD”, skip the steps from 10 to 15. If it is set to “hpe5022_TP_BACKWARD”, skip the steps from 4 to

9.

Figure 3-21 Track Profile Measurement



The “hpe5022_trackProfileData_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measurement
hpe5022_SEQ_WR_M	1	Write->Measurement

Name	Value	Description
hpe5022_SEQ_M	2	Measurement

- mode

  Description Specifies the direction of the read offset list array. See Figure 3-21.

  Direction IN

  Values

Name	Value	Description
hpe5022_TP_FORWARD	1	The argument of offset array increases from 0 to ‘points-1’.
hpe5022_TP_BACKWARD	2	The argument of offset array decreases from ‘points-1’ to 0.
hpe5022_TP_BOTH	3	The argument of offset array increases from 0 to ‘points-1’ and then decreases from ‘points-1’ to 0.

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the read offset list.

  Direction IN

  Values

Name	Value
hpe5022_TRACK_PROFILE_SIZE_MIN	1

Name	Value
hpe5022_TRACK_PROFILE_SIZE_MAX	201

- offset

  Description — Specifies the read offset as the data array. The read offset will be set to the data in array order. The direction depends on the parameter ‘mode’.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction — IN

  Unit — Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_AUTO_RANGE_FAIL	The autoranging in the track profile measurement is failed. Check if the paramter setting and head is correct.
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’, ‘mode’, ‘measFunc’, points’ and/or ‘offset’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q ” function. If it is not 0, the range is narrowed.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_trackProfileData_Q” on page 623

“hpe5022_selectPattern” on page 119

“hpe5022_trackOffsetCompValue” on page 984

hpe5022_setupTrackProfile

C Syntax

ViStatus hpe5022_setupTrackProfile(ViSession id, ViInt16 seqType, ViInt16 mode, ViInt16 measFunc,ViInt16 points, const ViReal64 offset[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupTrackProfile(ByVal id As Long, ByVal seqType As Integer, ByVal mode As Integer, ByVal measFunc As Integer, ByVal points As Integer, ByRef offset As Double, ByRef testHndl As Long) As Long

Description

This function assigns the track profile measurement sequence to the specified test identifier. Refer to the “hpe5022_measureTrackProfile” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_trackProfileData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measureTrackProfile” function.

- mode

  Description — Specifies the direction of the argument of the read offset list array.

  Direction — IN

  Values — Same as the ‘mode’ in the “hpe5022_measureTrackProfile” function.

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measureTrackProfile” function.

- points

  Description — Specifies the number of measurement points. This number must be the same as the array size of the read offset list.

  When the ‘mode’ is not ”hpe5022_TP_BOTH”, the maximum value of this parameter is “hpe5022_TRACK_PROFILE_SIZE_MAX”.

  When the ‘mode’ is “hpe5022_TP_BOTH” , the maximum value of this parameter is “hpe5022_TRACK_PROFILE_SIZE_MAX” / 2.

  Direction — IN

  Values — Same as the ‘points’ in the “hpe5022_measureTrackProfile” function.

- offset

  Description — Specifies the read offset as the data array. The read offset will be set to the data array order. The direction depends on the parameter ‘mode’.

  Direction — IN

  Unit — Meter

  Values — Same as the ‘offset’ in the “hpe5022_measureTrackProfile” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the track profile measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’, ‘mode’, ‘measFunc’, points’ and/or ‘offset’ is out of range.

Error Code	Description
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureTrackProfile” on page 615

“hpe5022_trackProfileData_Q” on page 623

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_trackProfileData_Q

C Syntax

ViStatus hpe5022_trackProfileData_Q(ViSession id, ViInt16 tpDataType,ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_trackProfileData_Q(ByVal id As Long, ByVal tpDataType As Integer, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the track profile measurement. The returned data is set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and read track offset list are specified by the “hpe5022_measureTrackProfile” function and “hpe5022_setupTrackProfile” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- tpDataType

  Description — Specifies the type of track profile result.

  When the “hpe5022_TP_BOTH” is selected in the “hpe5022_measureTrackProfile” or ““hpe5022_setupTrackProfile” function, selecting the “hpe5022_TP_DATA_AVE” returns an averaging result of the forward and backward results.

  When the “hpe5022_TP_FORWARD” is selected in that function, selecting the “hpe5022_TP_DATA_AVE” returns the same result of “hpe5022_TP_DATA_FOR”. If the hpe5022_TP_DATA_BACK” is selected, an error of “hpe5022_ERROR_INV_PARAMETER” occurs.

  When the “hpe5022_TP_BACKWARD” is selected in that function, selecting the “hpe5022_TP_DATA_AVE” returns the same result of “hpe5022_TP_DATA_BACK”. If the hpe5022_TP_DATA_FOR” is selected, an error of “hpe5022_ERROR_INV_PARAMETER” occurs.

  Direction — IN

  Values

Name	Value	Description
hpe5022_TP_DATA_AVE	1	Averaged data
hpe5022_TP_DATA_FOR	2	Forward data
hpe5022_TP_DATA_BACK	3	Backward data

- dataType

  Description Specifies the type of data to be reported.

  Direction IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description Returns the data (set in array) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureTrackProfile” or “hpe5022_setupTrackProfile” function. Each value in array is the measurement result relative to the read offset that corresponds to the same order value in the array of the read offset list.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘tpDataType’ and/or ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The Track Profile data is corrupt. Check if your measurement sequence is correct. Also, check if ‘tpDataType’ is correct.

See Also

“hpe5022_measureTrackProfile” on page 615

“hpe5022_setupTrackProfile” on page 620

hpe5022_calculateTrackWidth_Q

C Syntax

ViStatus hpe5022_calculateTrackWidth_Q(ViSession id, ViInt16 dataType,ViPReal64 trackWidth);

Visual Basic Syntax

hpe5022_calculateTrackWidth_Q(ByVal id As Long, ByVal dataType As Integer, ByRef trackWidth As Double) As Long

Description

This function calculates the equivalent write track width for the track profile measurement and the equivalent read track width for the micro track profile. Before this function is executed, the track profile measurement must be performed. See chapter 5 in the Operaion Manual for the definition of the track width.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- trackWidth

  Description: Returns the track width.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The track profile data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_PEAK_NOT_FOUND	The peak can’t be found in the track profile result. Check if the track profile is performed properly.

See Also

“hpe5022_measureTrackProfile” on page 615

“hpe5022_setupTrackProfile” on page 620

“hpe5022_trackProfileData_Q” on page 623

hpe5022_calculateReadWriteOffset_Q

C Syntax

ViStatus hpe5022_calculateReadWriteOffset_Q(ViSession id, ViInt16 dataType,ViPReal64 rwOffset);

Visual Basic Syntax

hpe5022_calculateReadWriteOffset_Q(ByVal id As Long, ByVal dataType As Integer, ByRef rwOffset As Double) As Long

Description

This function calculates the read/write offset. Before this function is executed, the track profile measurement must be performed. See chapter 5 in the Operaion Manual for the definition of the read/write offset.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- rwOffset

  Description: Returns the read write offset.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The track profile data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_PEAK_NOT_FOUND	The peak can not be found in the track profile result. Check if the track profile is performed properly.

See Also

“hpe5022_measureTrackProfile” on page 615

“hpe5022_trackProfileData_Q” on page 623

hpe5022_calculateRWOffsetAndWidth_Q

C Syntax

ViStatus hpe5022_calculateRWOffsetAndWidth_Q(ViSession id, ViInt16 dataType, ViReal64 lev0pct, ViInt16 rwOffsPos, ViReal64 thresLow, ViReal64 thresHigh, ViPReal64 rwOffset, ViPReal64 readTW, ViPReal64 writeTW);

Visual Basic Syntax

hpe5022_calculateRWOffsetAndWidth_Q(ByVal id As Long, ByVal dataType As Integer, ByVal lev0pct As Double, ByVal rwOffsPos As Integer, ByVal thresLow As Double, ByVal thresHigh As Double, ByRef rwOffset As Double, ByRef readTW As Double, ByRef writeTW As Double) As Long

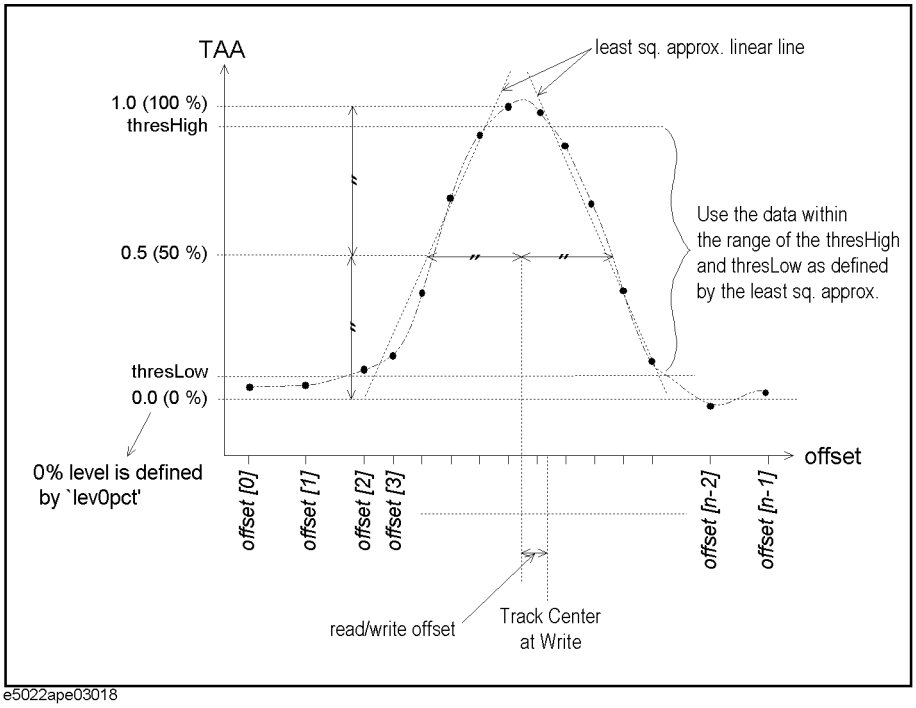
Description

This function allows you to calculate the read/write offset by various definition. Before this function is executed, the track profile measurement must be performed first. There are three types of definition for read/write offset calculation as described below.

- Center Point of 50 % level

  Figure 3-22 shows the definition of this method. Using the data within the range of ‘thresHigh’ and ‘thresLow’ you can calculate the least square approximation linear line. The center point is defined at the midpoint of the 50 % level between the two lines. The read/write offset is defined as the horizontal distance from the center point to the track center at write sequence.

Figure 3-22 Center Point of 50% Level

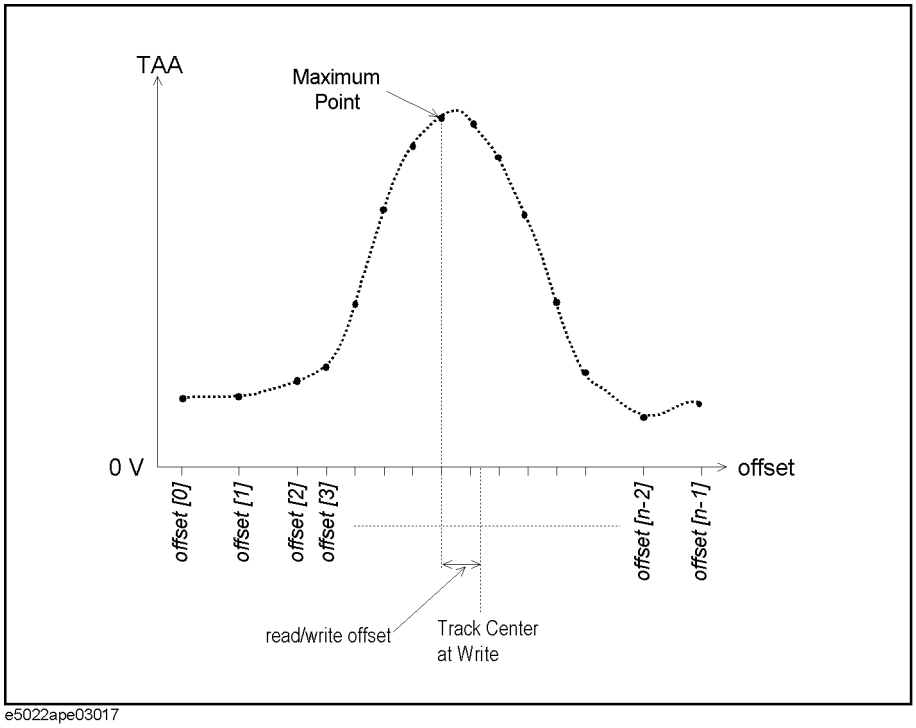


- Maximum Point of Track Profile Result

  Figure 3-23 shows the definition of this method. The write/read offset is defined as the horizontal distance between the maximum point of the track

profile result and the track center at write sequence.

Figure 3-23 Maximum Point of Track Profile Result

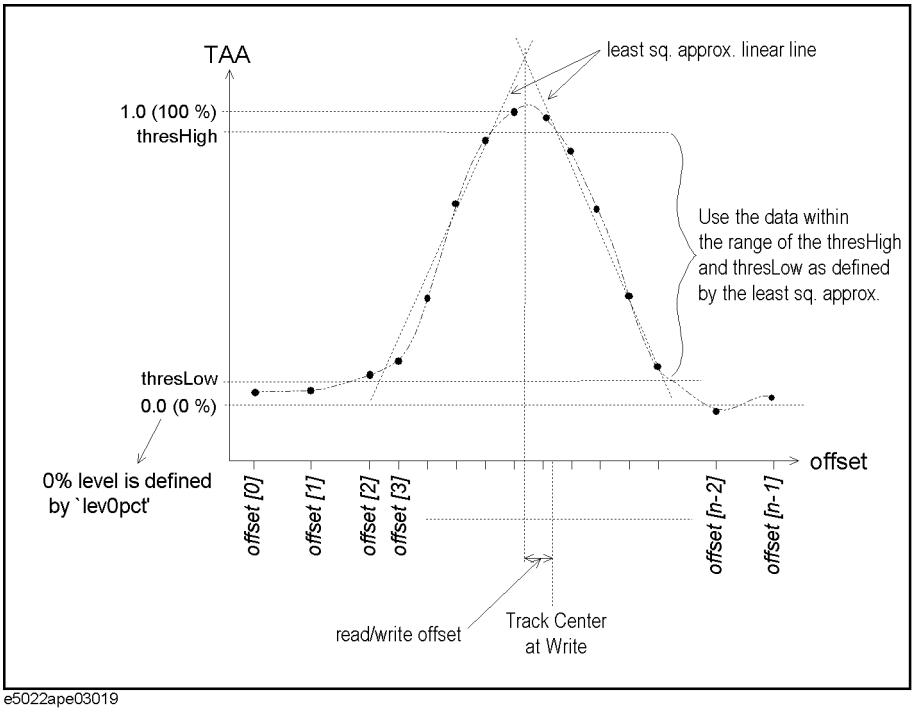


e5022ape03017

- Least Square Approximation Linear Line Cross Point

  Figure 3-24 shows the definition of this method. The write/read offset is defined as the horizontal distance from the intersection of two Least Square Approximation Linear Lines and the track center at write sequence.

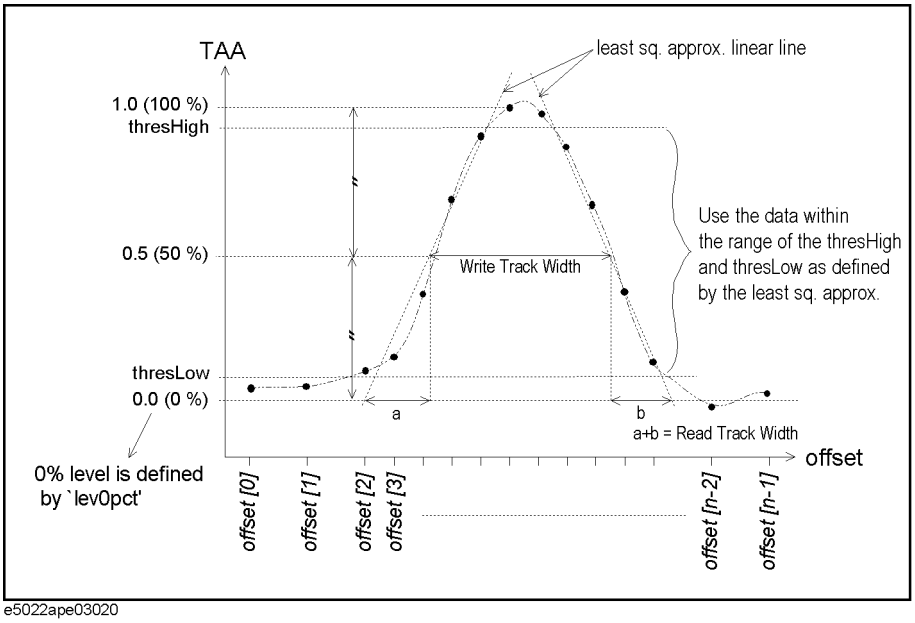
Figure 3-24 Least Square Approximation Linear Line Cross Point



e5022ape03019

The definitions of the calculated write track width and read track width are shown in Figure 3-25.

Figure 3-25 Write Track Width and Read Track Width Definitions



e5022ape03020

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

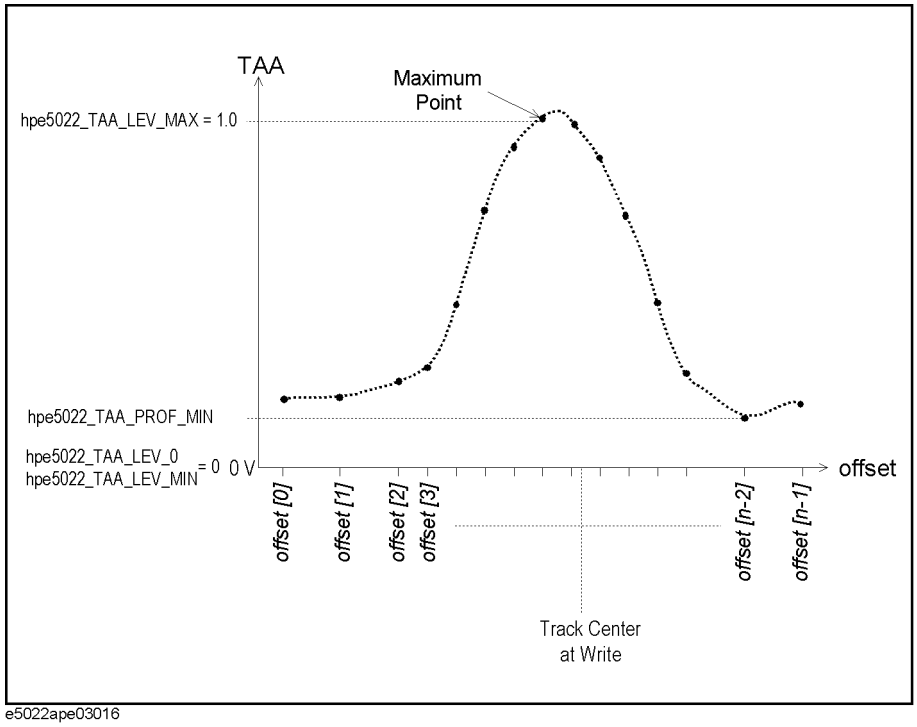
  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- lev0pct

  Description — Specifies 0% level to calculate the read/write offset. This level is defined as the percentage of the maximum level of the track profile result. If you want to set 0 V level as the 0 % level for calculation, set to 0.0 or “hpe5022_TAA_LEV_0” for this parameter. When “hpe5022_TAA_LEV_PROF_MIN” is selected, the minimum level of the track profile result is set as the 0% level. Also, you can set various levels between 0 and the maximum voltage of TAA. The level is set by a ratio of the maximum TAA level.

Figure 3-26 ‘lev0pct’ definition



e5022ape03016

Direction IN

Values

Name	Value
hpe5022_TAA_LEV_PROF_MIN	-1
hpe5022_TAA_LEV_0	0.0
hpe5022_TAA_LEV_MIN	0.0
hpe5022_TAA_LEV_MAX	1.0

- rwOffsPos

  Description Specifies the calculation method of the read write offset.

  Direction IN

  Values

Name	Value	Description
hpe5022_RWOF_POS_50PERC_CENT	0	Center Point at 50% level
hpe5022_RWOF_POS_PROF_MAX	1	Maximum point of track profile

Name	Value	Description
hpe5022_RWOF_POS_CROS_INT_LINE	2	Least Sq. Approx. Linear Line Cross Point

- thresLow

  Description Specifies the low threshold for least square root approximation. The measurement result points in the range between the thresLow and thresHigh is used to calculate the least root approximation line.

  Direction IN

  Values

Name	Value
hpe5022_INT_THR_MIN	0.0
hpe5022_INT_THR_MAX	1.0

- thresHigh

  Description Specifies the high threshold for the least square root approximation. This value must be more than the value specified by ‘thresLow’.

  Direction IN

  Values

Name	Value
hpe5022_INT_THR_MIN	0.0
hpe5022_INT_THR_MAX	1.0

- rwOffset

  Description Returns the read write offset.

  Direction OUT

  Unit Meter

- readTW

  Description Returns the equivalent read track width.

  Direction OUT

  Unit Meter

- writeTW

  Description Returns the equivalent write track width.

  Direction OUT

  Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’, ‘lev0pct’. ‘rwOffsPos’, ‘thresLow’ and/or ‘thresHigh’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The track profile data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_PEAK_NOT_FOUND	The peak can not be found in the track profile result. Check if the track profile is performed properly.

See Also

“hpe5022_measureTrackProfile” on page 615

“hpe5022_trackProfileData_Q” on page 623

Micro Track Function

This section describes the functions related with micro track profile measurement. There are two types of micro track profile measurement.

One method of creating micro track is to erase both sides of the track with a monitoring the level. After creating a micro track, then measure the track profile using the “hpe5022_measureTrackProfile” functions.

Another method in creating micro track profile is through the use of fast micro track profile.This method creates micro track by erasing both sides of the track at a specified width. There is no level monitoring during erase, it includes the sequence such that allows for faster measurement.

hpe5022_microTrackConfig

C Syntax

ViStatus hpe5022_microTrackConfig(ViSession id,ViReal64 start, ViReal64 step, ViReal64 ratio);

Visual Basic Syntax

hpe5022_microTrackConfig(ByVal id As Long, ByVal start As Double, ByVal step As Double, ByVal ratio As Double) As Long

Description

This function controls the micro track configurations. This function does not execute the creation of micro track. The “hpe5022_createMicroTrack” function creates the micro track. The detailed sequence is as follows:

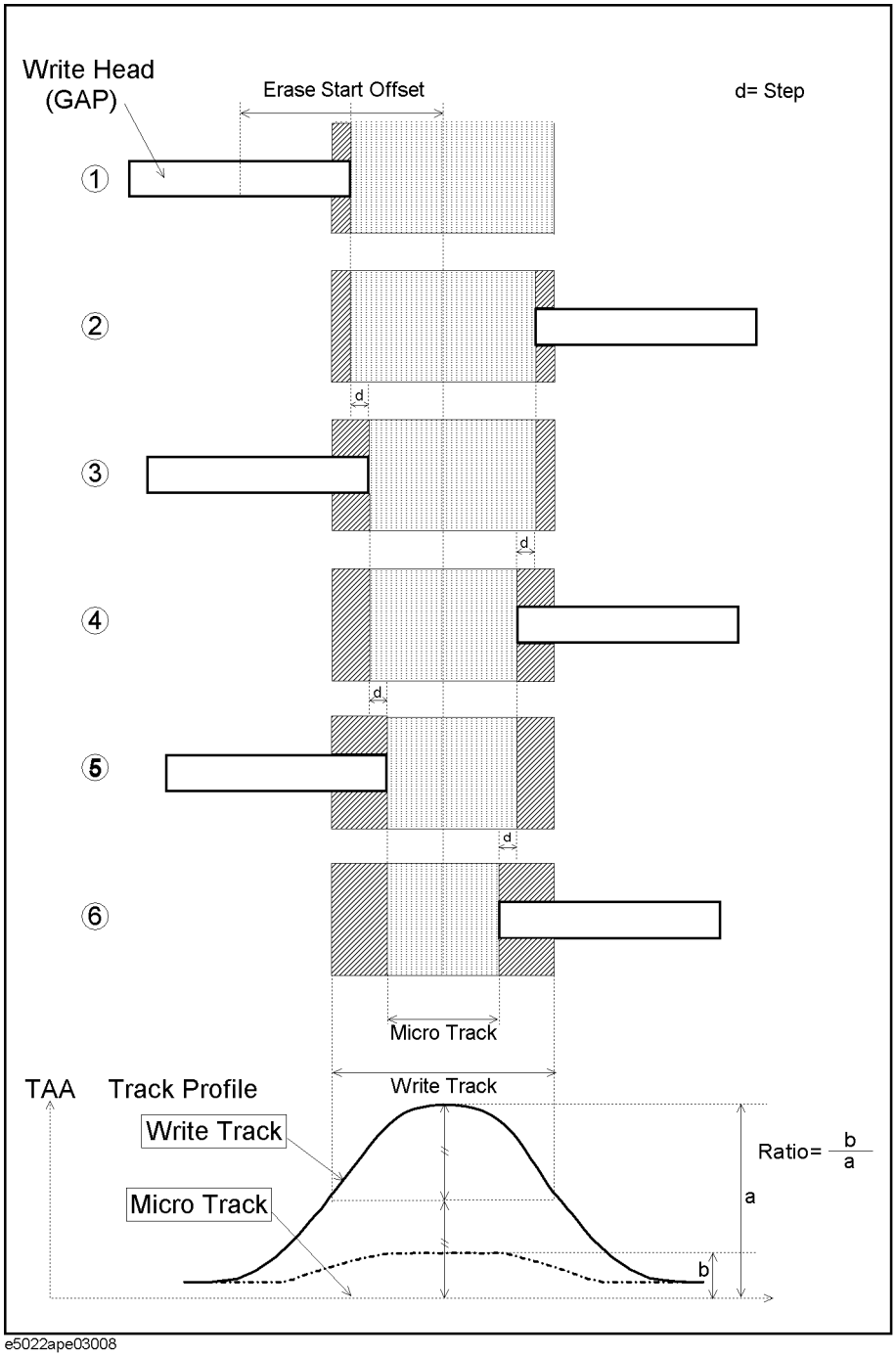
1. Move the head to the write offset position specified by the “hpe5022_writeTrackOffset” function.
2. Erase (same as the “hpe5022_erase” function) for an entire track.
3. Write the data pattern specified by the “hpe5022_selectPattern” function.
4. Move the head to the read offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure the narrow band TAA and memorize the result (a).
6. Set the ‘squashPos’ at ‘start’.
7. Move the head at the position of (write offset + squashPos).
8. Erase (same as the “hpe5022_erase” function) for an entire track.
9. Move the head at the position of (write offset - squashPos).
10. Erase (same as the “hpe5022_erase” function) for an entire track.
11. Move the head to read offset position.
12. Measure the narrow band TAA.
13. If the measured value is less than the value of (ratio × a), then the procedure is end.

14. Set the ‘squashPos’=’SquashPos’-’step’.

15. Go to the step 7 until the squashPos is less than zero.

When the track offset compensation state (“hpe5022_trackOffsetCompPatternState”) is turned on, the position is compensated automatically at the time before the step 7. The “hpe5022_executeTrackOffsetComp” is not necessary to execute.

Figure 3-27 **Micro Track Parameters**



Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- start

  Description Specifies the start offset of erase sequence.

  Direction IN

  Unit Meter

  Preset value 3.0×10^{-6}

  (same as hpe5022_MICRO_TRACK_START_MAX / 2.0)

  Values

Name	**Value**
hpe5022_MICRO_TRACK_START_MIN	0
hpe5022_MICRO_TRACK_START_MAX	6.0×10^{-6} (same as “hpe5022_TRACK_OFFSET_MAX”)

- step

  Description Specifies the step size.

  Direction IN

  Unit Meter

  Preset value 6.0×10^{-7} (same as the absolute value of (hpe5022_MICRO_TRACK_START_MAX / 10))

  Values

Name	**Value**
hpe5022_MICRO_TRACK_STEP_MIN[a]	1×10^{-9}
hpe5022_MICRO_TRACK_STEP_MAX	6.0×10^{-6} (same as hpe5022_TRACK_OFFSET_MAX)

  a. This value is a software limitation. Actual minimum resolution of piezo depends on its hardware.

- ratio

  Description Specifies the amplitude ratio of the TAA.When the measured TAA reaches this level which is the initial TAA level multiplied by this ratio, the procedure is done.

  Direction IN

  Preset value 0.1

Values

Name	Value
hpe5022_MICRO_TRACK_RATIO_MIN	0.01
hpe5022_MICRO_TRACK_RATIO_MAX	1.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘start’. ‘step’ and/or ‘ratio’ is out of range.

See Also

“hpe5022_createMicroTrack” on page 644

“hpe5022_microTrackConfig_Q” on page 642

hpe5022_microTrackConfig_Q

C Syntax

ViStatus hpe5022_microTrackConfig_Q(ViSession id,ViPReal64 start, ViPReal64 step, ViPReal64 ratio);

Visual Basic Syntax

hpe5022_microTrackConfig_Q(ByVal id As Long, ByRef start As Double, ByRef step As Double, ByRef ratio As Double) As Long

Description

This function returns the micro track configuration.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- start

  Description: Returns the start offset of erase sequence.

  Direction: OUT

  Values: Same as the ‘start’ in the “hpe5022_microTrackConfig” function.

- step

  Description: Returns the step size.

  Direction: OUT

  Values: Same as the ‘step’ in the “hpe5022_microTrackConfig” function.

- ratio

  Description: Returns the amplitude ratio of TAA.

  Direction: OUT

  Values: Same as the ‘ratio’ in the “hpe5022_microTrackConfig” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also “hpe5022_microTrackConfig” on page 637

hpe5022_createMicroTrack

C Syntax

ViStatus hpe5022_createMicroTrack(ViSession id);

Visual Basic Syntax

hpe5022_createMicroTrack(ByVal id As Long) As Long

Description

This function executes the creation of micro track. See the “hpe5022_microTrackConfig” function for the sequence. This function doesn’t measure track profile, the track profile measurement should be done before creating a micro track. See the “Micro Track Profile Measurement” on page 52.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_MICRO_TRACK_FAILED	Fail to create a micro track. The measured TAA level does not reach a specified level. Check if the parameters of the “hpe5022_microTrackConfig” function.

See Also

“hpe5022_microTrackConfig” on page 637

hpe5022_measureMicroTrackProfile

C Syntax

ViStatus hpe5022_measureMicroTrackProfile(ViSession id, ViInt16 seqType, ViInt16 mode, ViInt16 erasMode, ViReal64 erasPos, ViInt16 points, const ViReal64 offset[]);

Visual Basic Syntax

hpe5022_measureMicroTrackProfile(ByVal id As Long, ByVal seqType As Integer, ByVal mode As Integer, ByVal erasMode As Integer, ByVal erasPos As Double, ByVal points As Integer, ByRef offset As Double) As Long

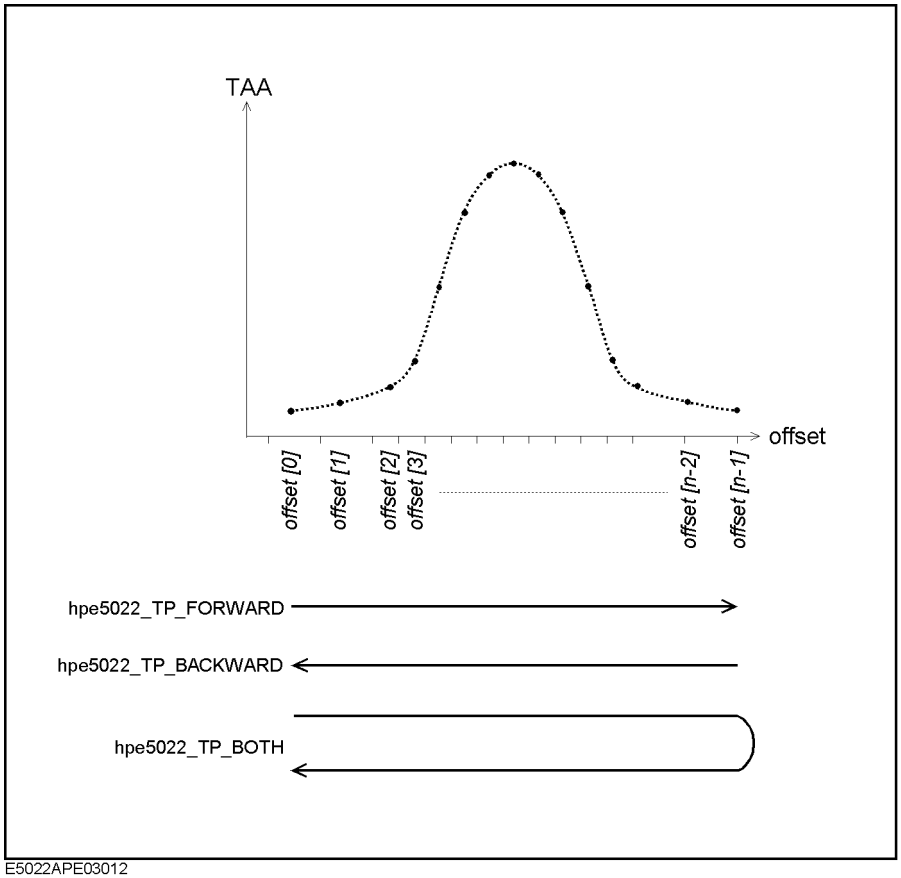
Description

This function measures a fast micro track profile. This function includes not only the creation of micro track but also the measurement of track profile. The detailed sequence is as follows:

1. If ‘seqType’ is set to “hpe5022_SEQ_ER_WR_M”, perform the three-track-erase. (the entire track erase for 5 positions, the track, its both side tracks, and the center between them.)
2. Write the data pattern specified by the “hpe5022_selectPattern” function.
3. Move the head to the offset position specified by the ‘erasePos’
4. Perform erase with the method specified by “hpe5022_eraseType”.
5. If ‘erasMode’ is set to “hpe5022_MTP_ERAS_BOTH_SIDE”, Move the head to the offset position specified by the -‘erasePos’ then perform an erase.
6. Set i to zero (i = 0)
7. Set the read track offset from the track center at ‘offset[i]’.
8. Move the head to the read track offset.
9. Measure a narrow band TAA.
10. Increment i by 1. (i = i+1).
11. If i is not equal to ‘points’, go to the step 7.
12. Set i to ‘points’
13. Decrement i by 1. (i = i-1).
14. Set the read track offset from the track center at ‘offset[i]’
15. Move the head to the read track offset.
16. Measure narrow band TAA.
17. If i is not equal to zero, go to the step 13.

If ‘mode’ is set to “hpe5022_TP_FORWARD”, skip the steps from 12 to 17. If it is set to “hpe5022_TP_BACKWARD”, skip the steps from 6 to 11.

Figure 3-28 **Track Profile Measurement**



The “hpe5022_microTrackProfileData_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Three Track Erase ->Write/Erase Sides of Track ->Measurement

Name	Value	Description
hpe5022_SEQ_WR_M	1	Write/Erase Sides of Track->Measurement

- mode

  Description Specifies the direction of the read offset list array. See Figure 3-28.

  Direction IN

  Values

Name	Value	Description
hpe5022_TP_FORWARD	1	The argument of offset array increases from 0 to ‘points-1’.
hpe5022_TP_BACKWARD	2	The argument of offset array decreases from ‘points-1’ to 0.
hpe5022_TP_BOTH	3	The argument of offset array increases from 0 to ‘points-1’ and then decreases from ‘points-1’ to 0.

- erasMode

  Description Specifies the type of erase.

  Direction IN

  Values

Name	Value	Description
hpe5022_MTP_ERAS_SING_SIDE	1	Erase single side only
hpe5022_MTP_ERAS_BOTH_SIDE	2	Erase both sides

- erasPos

  Description Specifies the side erase position.

  Direction IN

  Unit Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}

Name	Value
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the read offset list.

  Direction IN

  Values

Name	Value
hpe5022_TRACK_PROFILE_SIZE_MIN	1
hpe5022_TRACK_PROFILE_SIZE_MAX	201

- offset

  Description Specifies the read offset list as the data array. The read offset is set to the data in array order. The direction depends on the parameter ‘mode’

  Direction IN

  Unit Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’, ‘mode’. ‘erasMode’, ‘erasPos’, points’ and/or ‘offset’ is out of range.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_selectPattern” on page 119

“hpe5022_eraseType” on page 355

“hpe5022_microTrackProfileData_Q” on page 655

hpe5022_setupMicroTrackProfile

C Syntax

ViStatus hpe5022_setupMicroTrackProfile(ViSession id, ViInt16 seqType, ViInt16 mode, ViInt16 erasMode, ViReal64 erasPos, ViInt16 points, const ViReal64 offset[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupMicroTrackProfile(ByVal id As Long, ByVal seqType As Integer, ByVal mode As Integer, ByVal erasMode As Integer, ByVal erasPos As Double, ByVal points As Integer, ByRef offset As Double, ByRef testHndl As Long) As Long

Description

This function assigns a fast micro track profile measurement sequence to the specified test identifier. Refer to the “hpe5022_measureMicroTrackProfile” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_microTrackProfileData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the sequence type.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measureMicroTrackProfile” function.

- mode

  Description — Specifies the direction of the read offset list array.

  Direction — IN

  Values — Same as the ‘mode’ in the “hpe5022_measureMicroTrackProfile” function.

- erasMode

  Description — Specifies the type of erase.

  Direction — IN

  Values — Same as the ‘erasMode’ in the “hpe5022_measureMicroTrackProfile” function.

- erasPos

  Description — Specifies the side erase position.

Direction IN

Unit Meter

Values Same as the ‘erasPos’ in the “hpe5022_measureMicroTrackProfile” function.

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the read offset list.

  Direction IN

  Values Same as the ‘points’ in the “hpe5022_measureMicroTrackProfile” function.

- offset

  Description Specifies the read offset list as the data array. The read offset will be set to the data in array order. The direction is depending on the parameter ‘mode’

  Direction IN

  Unit Meter

  Values Same as the ‘offset’ in the “hpe5022_measureMicroTrackProfile” function.

- testHndl

  Description Returns the test identifier. This identifier is used to execute the micro track profile measurement by the “hpe5022_measure” function.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

“hpe5022_measureMicroTrackProfile” on page 645

“hpe5022_microTrackProfileData_Q” on page 655

hpe5022_modifyMicroTrackProfileErasePosition

C Syntax

ViStatus hpe5022_modifyMicroTrackProfileErasePosition(ViSession id, ViObject testHndl, ViReal64 erasPos);

Visual Basic Syntax

hpe5022_modifyMicroTrackProfileErasePosition(ByVal id As Long, ByVal testHndl As Long, ByVal erasPos As Double) As Long

Description

This function modifies the side erase position of the micro track profile sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- testHndl

  Description: Specifies the test identifier. This identifier is used to execute the micro track profile measurement by the “hpe5022_measure” function.

  Direction: IN

- erasPos

  Description: Specifies the modified position of side erase.

  Direction: IN

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘erasPos’ is out of range.

See Also

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

“hpe5022_measureMicroTrackProfile” on page 645

“hpe5022_microTrackProfileData_Q” on page 655

hpe5022_microTrackProfileData_Q

C Syntax

ViStatus hpe5022_microTrackProfileData_Q(ViSession id, ViInt16 tpDataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_microTrackProfileData_Q(ByVal id As Long, ByVal tpDataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of fast micro track profile measurement. The returned data is set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and read track offset list are specified by the “hpe5022_measureMicroTrackProfile” function and “hpe5022_setupMicroTrackProfile” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- tpDataType

  Description — Specifies the type of track profile result.

  When the “hpe5022_TP_BOTH” is selected in the “hpe5022_measureMicroTrackProfile” or “hpe5022_setupMicroTrackProfile” function, selecting the “hpe5022_TP_DATA_AVE” returns an averaging result of the forward and backward results.

  When the “hpe5022_TP_FORWARD” is selected in that function, selecting the “hpe5022_TP_DATA_AVE” returns the same result of “hpe5022_TP_DATA_FOR”. If the hpe5022_TP_DATA_BACK” is selected, an error of “hpe5022_ERROR_INV_PARAMETER” occurs.

  When the “hpe5022_TP_BACKWARD” is selected in that function, selecting the “hpe5022_TP_DATA_AVE” returns the same result of “hpe5022_TP_DATA_BACK”. If the hpe5022_TP_DATA_FOR” is selected, an error “hpe5022_ERROR_INV_PARAMETER” will occur.

  Direction — IN

  Values

Name	Value	Description
hpe5022_TP_DATA_AVE	1	Averaged data
hpe5022_TP_DATA_FOR	2	Forward data
hpe5022_TP_DATA_BACK	3	Backward data

- data

  Description — Returns the TAA data (set in array). The size of array is specified by the ‘points’ parameter in the “hpe5022_measureMicroTrackProfile” or “hpe5022_setupMicroTrackProfile” function. Each value in array is the measurement result relative to the read offset that corresponds to the same order value in the array of the read offset list.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘tpDataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The Track Profile data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureMicroTrackProfile” on page 645

“hpe5022_setupMicroTrackProfile” on page 650

hpe5022_microTrackProfileTrackWidth_Q

C Syntax

ViStatus hpe5022_microTrackProfileTrackWidth_Q(ViSession id, ViPReal64 rw);

Visual Basic Syntax

hpe5022_microTrackProfileTrackWidth_Q(ByVal id As Long, ByRef rw As Double) As Long

Description

This function calculates the equivalent read track width based on the result of the fast micro track profile measurement. The read track width is defined as the width at half amplitude level of the track profile result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rw

  Description: Returns the read track width.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The track profile data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_PEAK_NOT_FOUND	The peak can’t be found in the track profile result. Check if the result of “hpe5022_microTrackProfileData_Q ” is proper.

See Also

“hpe5022_measureMicroTrackProfile” on page 645

“hpe5022_setupMicroTrackProfile” on page 650

“hpe5022_microTrackProfileData_Q” on page 655

Write Current Sweep Measurement Function

This section describes the functions related with the write current sweep measurement. This measurement allows you to observe the write current characteristics of the parameters such as TAA and PW.

hpe5022_measureWriteCurrentSweep

C Syntax

ViStatus hpe5022_measureWriteCurrentSweep(ViSession id,ViInt16 measFunc, ViInt16 points, const ViReal64 iw[]);

Visual Basic Syntax

hpe5022_measureWriteCurrentSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef iw As Double) As Long

Description

This function measures the specified parameter while changing the write current. The sequence of this function is as follows:

- TAA, PW and Narrow Band TAA
  1. Set i to zero. (i = 0)
  2. Erase (same as the “hpe5022_erase” function) for an entire track.
  3. Set the write current at the value of iw[i].
  4. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  5. Increment i by 1. (i = i+1)
  6. If i is not equal to ‘points’, go to step 2.
- Overwrite, and NLTS 5th
  1. Set i to zero. (i = 0)
  2. Set the write current at the value of iw[i].
  3. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  4. Increment i by 1. (i = i+1)
  5. If i is not equal to ‘points’, go to step 2.

The “hpe5022_writeCurrentSweepData_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- measFunc

  Description Specifies the measurement type.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_PW	2	Measure PW
hpe5022_MEAS_OVERWRITE	3	Measure Overwrite
hpe5022_MEAS_NLTS_5TH	4	Measure NLTS 5th
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the write current list.

  Direction IN

  Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- iw

  Description Specifies the write current list as the data array. The write current will be set to the data in the array order. The range of current value is returned by the “hpe5022_writeCurrentRange_Q” function.

  Direction IN

  Unit Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘point’ and/or ‘iw’ is out of range.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_writeCurrentSweepData_Q” on page 663

“hpe5022_writeCurrentRange_Q” on page 325

hpe5022_setupWriteCurrentSweep

C Syntax

ViStatus hpe5022_setupWriteCurrentSweep(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 iw[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupWriteCurrentSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef iw As Double, ByRef testHndl As Long) As Long

Description

This function assigns the write current sweep measurement sequence to the specified test identifier. Refer to the “hpe5022_measureWriteCurrentSweep” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_writeCurrentSweepData_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- measFunc

  Description: Specifies the measurement type

  Direction: IN

  Values: Same as the ‘measFunc’ in the “hpe5022_measureWriteCurrentSweep” function.

- points

  Description: Specifies the number of measurement points. This number must be the same as the array size of the write current list.

  Direction: IN

  Values: Same as the ‘points’ in the “hpe5022_measureWriteCurrentSweep” function.

- iw

  Description: Specifies the write current list as the data array. The write current will be set to data in the array in order.

  Direction: IN

  Unit: Ampere

  Values: Same as the ‘iw’ in the “hpe5022_measureWriteCurrentSweep” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the write current sweep measurement by the "hpe5022_measure" function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'measFunc', 'points' and/or 'iw' is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measureWriteCurrentSweep" on page 658

"hpe5022_writeCurrentSweepData_Q" on page 663

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_writeCurrentSweepData_Q

C Syntax

ViStatus hpe5022_writeCurrentSweepData_Q(ViSession id,ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_writeCurrentSweepData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the write current sweep measurement. The returned data are set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and write current list are specified by the “hpe5022_measureWriteCurrentSweep” function or “hpe5022_setupWriteCurrentSweep” function.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataType

  Description Specifies the type of data to be reported.

  Direction IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_OVERWRITE	13	Overwrite
hpe5022_DATA_NLTS_5TH	14	NLTS 5th
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description — Returns the data (set in array) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureWriteCurrentSweep” or “hpe5022_setupWriteCurrentSweep” function. Each value in the array order is the measurement result measured at each write current list.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’, ‘points’ and/or ‘iw’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The write current sweep data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureWriteCurrentSweep” on page 658

“hpe5022_setupWriteCurrentSweep” on page 661

Sense Current Sweep Measurement Function

This section describes the functions related with the sense current sweep measurement. This measurement allows you to observe a sense current characteristics of the parameters such as TAA and PW.

hpe5022_measureSenseCurrentSweep

C Syntax

ViStatus hpe5022_measureSenseCurrentSweep(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 ib[]);

Visual Basic Syntax

hpe5022_measureSenseCurrentSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef ib As Double) As Long

Description

This function measures the specified parameter while changing the sense current. The sequences of this function are as follows:

- TAA, PW and Narrow Band TAA
  1. Set i to zero. (i = 0)
  2. Erase (same as the “hpe5022_erase” function) for an entire track.
  3. Set the sense current at the value of is[i].
  4. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  5. Increment i by 1. (i = i+1)
  6. If i is not equal to ‘points’, go to step 2.
- Overwrite, and NLTS 5th
  1. Set i to zero. (i = 0)
  2. Set the sense current at the value of is[i].
  3. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  4. Increment i by 1. (i = i+1)
  5. If i is not equal to ‘points’, go to step 2.

The “hpe5022_writeCurrentSweepData_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_PW	2	Measure PW
hpe5022_MEAS_OVERWRITE	3	Measure Overwrite
hpe5022_MEAS_NLTS_5TH	4	Measure NLTS 5th
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the sense current list.

  Direction IN

  Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- ib

  Description Specifies the sense current list as the data array. The sense current will be set to the data in array order. The range of current value is returned by the “hpe5022_senseCurrentRange_Q” function.

  Direction IN

  Unit Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘points’ and/or ‘ib’ is out of range.
hpe5022_ERROR_INV_SETUP	The sense stimulus mode in the “hpe5022_senseStimulusMode” is not set at hpe5022_SENSE_STIMULUS_CURRENT.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_senseCurrentSweepData_Q” on page 670

“hpe5022_senseCurrentRange_Q” on page 339

hpe5022_setupSenseCurrentSweep

C Syntax

ViStatus hpe5022_setupSenseCurrentSweep(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 ib[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSenseCurrentSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef ib As Double, ByRef testHndl As Long) As Long

Description

This function assigns the sense current sweep measurement sequence to the specified test identifier. Refer to the “hpe5022_measureSenseCurrentSweep” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_senseCurrentSweepData_Q” function returns the measurement result

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measureSenseCurrentSweep”“hpe5022_measure SenseCurrentSweep” function.

- points

  Description — Specifies the number of measurement points. This number must be the same as the array size of the sense current list.

  Direction — IN

  Values — Same as the ‘points’ in the “hpe5022_measureSenseCurrentSweep”“hpe5022_measure SenseCurrentSweep” function.

- ib

  Description — Specifies the sense current list as the data array. The sense current will be set to the data in array order.

  Direction — IN

  Unit — Ampere

  Values — Same as the ‘ib’ in the

"hpe5022_measureSenseCurrentSweep""hpe5022_measure SenseCurrentSweep" function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the sense current sweep measurement by the "hpe5022_measure" function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'measFunc', 'point', and/or 'ib' is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measureSenseCurrentSweep" on page 665

"hpe5022_senseCurrentSweepData_Q" on page 670

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_senseCurrentSweepData_Q

C Syntax

ViStatus hpe5022_senseCurrentSweepData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_senseCurrentSweepData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the sense current sweep measurement. The returned data is set in array order, whose size is specified by the ‘points’ parameter. The ‘points’ and sense current list are specified by the “hpe5022_measureSenseCurrentSweep” function or “hpe5022_setupSenseCurrentSweep” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_OVERWRITE	13	Overwrite
hpe5022_DATA_NLTS_5TH	14	NLTS 5th
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description — Returns the data (set in array order) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureSenseCurrentSweep” or “hpe5022_setupSenseCurrentSweep” function. Each value in the array is the measurement result measured from each sense current list.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The sense current sweep data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSenseCurrentSweep” on page 665

“hpe5022_setupSenseCurrentSweep” on page 668

Sense Stimulus Sweep Measurement Function

This section describes the functions related with the sense stimulus sweep measurement. This measurement allows you to observe a sense stimulus characteristics of the parameters such as TAA and PW.

hpe5022_measureSenseStimulusSweep

C Syntax

ViStatus hpe5022_measureSenseStimulusSweep(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 ib[]);

Visual Basic Syntax

hpe5022_measureSenseStimulusSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef ib As Double) As Long

Description

This function measures the specified parameter while changing the sense stimulus. The stimulus type is selected by the “hpe5022_senseStimulus” function. When the hpe5022_SENSE_STIMULUS_CURRENT is selected in the “hpe5022_senseStimulus” function, this function is exactly same as the “hpe5022_setupSenseCurrentSweep” function.

The sequences of this function are as follows:

- TAA, PW and Narrow Band TAA
  1. Set i to zero. (i = 0)
  2. Erase (same as the “hpe5022_erase” function) for an entire track.
  3. Set the sense current at the value of ib[i].
  4. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  5. Increment i by 1. (i = i+1)
  6. If i is not equal to ‘points’, go to step 2.
- Overwrite, and NLTS 5th
  1. Set i to zero. (i = 0)
  2. Set the sense current at the value of ib[i].
  3. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  4. Increment i by 1. (i = i+1)
  5. If i is not equal to ‘points’, go to step 2.

The “hpe5022_senseStimulusSweepData_Q” function returns the measurement result.

Parameters

- id

Description Specifies the system identifier. This is given by the "hpe5022_init" function.

Direction IN

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_PW	2	Measure PW
hpe5022_MEAS_OVERWRITE	3	Measure Overwrite
hpe5022_MEAS_NLTS_5TH	4	Measure NLTS 5th
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the sense stimulus list.

  Direction IN

  Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- ib

  Description Specifies the sense stimulus list as the data array. The sense stimulus will be set to the data in array order. The range of value is returned by the “hpe5022_senseStimulusRange_Q” function.

  Direction IN

  Unit Depends on the mode of “hpe5022_senseStimulusMode”

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘points’ and/or ‘ib’ is out of range.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_senseStimulusSweepData_Q” on page 677

“hpe5022_senseStimulusRange_Q” on page 348

hpe5022_setupSenseStimulusSweep

C Syntax

ViStatus hpe5022_setupSenseStimulusSweep(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 ib[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSenseStimulusSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef ib As Double, ByRef testHndl As Long) As Long

Description

This function assigns the sense stimulus sweep measurement sequence to the specified test identifier. Refer to the “hpe5022_measureSenseStimulusSweep” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_senseStimulusSweepData_Q” function returns the measurement result

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- measFunc

  Description: Specifies the type of measurement.

  Direction: IN

  Values: Same as the ‘measFunc’ in the “hpe5022_measureSenseStimulusSweep”“hpe5022_measureSenseCurrentSweep” function.

- points

  Description: Specifies the number of measurement points. This number must be the same as the array size of the sense stimulus list.

  Direction: IN

  Values: Same as the ‘points’ in the “hpe5022_measureSenseCurrentSweep”“hpe5022_measureSenseStimulusSweep” function.

- ib

  Description: Specifies the sense stimulus list as the data array. The sense stimulus will be set to the data in array order.

  Direction: IN

  Unit: Same as the “hpe5022_measureSenseStimulusSweep”“hpe5022_measureSenseCurrentSweep” function.

Values — Same as the ‘ib’ in the “hpe5022_measureSenseStimulusSweep”“hpe5022_measureSenseCurrentSweep” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the sense stimulus sweep measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘point’, and/or ‘ib’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureSenseStimulusSweep” on page 672

“hpe5022_senseStimulusSweepData_Q” on page 677

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_senseStimulusSweepData_Q

C Syntax

ViStatus hpe5022_senseStimulusSweepData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_senseStimulusSweepData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the sense stimulus sweep measurement. The returned data is set in array order, whose size is specified by the ‘points’ parameter. The ‘points’ and sense stimulus list are specified by the “hpe5022_measureSenseStimulusSweep” function or “hpe5022_setupSenseStimulusSweep” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_OVERWRITE	13	Overwrite
hpe5022_DATA_NLTS_5TH	14	NLTS 5th
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description — Returns the data (set in array order) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureSenseStimulusSweep” or “hpe5022_setupSenseStimulusSweep” function. Each value in the array is the measurement result measured from each sense stimulus list.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The sense stimulus sweep data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSenseStimulusSweep” on page 672

“hpe5022_setupSenseStimulusSweep” on page 675

Channel Bit Rate Sweep Measurement

This section describes the functions related with the channel bit rate sweep measurement. This measurement allows you to observe the channel bit rate characteristics of the parameter such as the TAA and the PW.

hpe5022_measureChannelBitRateSweep

C Syntax

ViStatus hpe5022_measureChannelBitRateSweep(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 chanBitRate[]);

Visual Basic Syntax

hpe5022_measureChannelBitRateSweep (ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef chanBitRate As Double) As Long

Description

This function measures the channel bit rate sweep measurement. The sequences of this function are as follows:

- TAA, PW and Narrow Band TAA
  1. Set i to zero. (i = 0)
  2. Erase (same as the “hpe5022_erase” function) for an entire track.
  3. Set the write current at the value of iw[i].
  4. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  5. Increment i by 1. (i = i+1)
  6. If i is not equal to ‘points’, go to step 2.
- Overwrite, and NLTS 5th
  1. Set i to zero. (i = 0)
  2. Set the write current at the value of iw[i].
  3. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
  4. Increment i by 1. (i = i+1)
  5. If i is not equal to ‘points’, go to step 2.

The “hpe5022_channelBitRateSweepData_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

Type ViSession

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_PW	2	Measure PW
hpe5022_MEAS_OVERWRITE	3	Measure overwrite
hpe5022_MEAS_NLTS_5TH	4	Measure NLTS 5th
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the channel bit rate list.

  Direction IN

  Value

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- chanBitRate

  Description Specifies the channel bit rate as the data array. The channel bit rate will be set to data in array order. The range of channel bit rate is returned by the “hpe5022_channelBitRateRange_Q” function.

  Direction IN

  Unit bps (bit per second)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘points’ and/or ‘chanbitRate’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_setupChannelBitRateSweep” on page 682

“hpe5022_channelBitRateSweepData_Q” on page 684

“hpe5022_channelBitRateRange_Q” on page 118

hpe5022_setupChannelBitRateSweep

C Syntax

ViStatus hpe5022_setupChannelBitRateSweep(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 chanBitRate[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupChannelBitRateSweep (ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef chanBitRate As Double, ByRef testHndl As Long) As Long

Description

This function assigns the channel bit rate measurement sequence to the specified test identifier. Refer to the “hpe5022_measureChannelBitRateSweep” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_channelBitRateSweepData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measureChannelBitRateSweep” function.

- points

  Description — Specifies the number of measurement points. This number must be the same as the array size of the channel bit rate list.

  Direction — IN

  Values — Same as the ‘points’ in the “hpe5022_measureChannelBitRateSweep” function.

- chanBitRate

  Description — Specifies the channel bit rate as the data array. The channel bit rate will be set to data in the array order.

  Direction — IN

  Unit — bps (bit per second)

  Values — Same as the ‘chanBitRate’ in the “hpe5022_measureChannelBitRateSweep” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the channel bit rate sweep measurement by the "hpe5022_measure" function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by "id" is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'measFunc', 'points' and/or 'chanbitRate' is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measureChannelBitRateSweep" on page 679

"hpe5022_channelBitRateSweepData_Q" on page 684

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_channelBitRateSweepData_Q

C Syntax

ViStatus hpe5022_channelBitRateSweepData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_channelBitRateSweepData_Q (ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the channel bit rate sweep measurement. The returned data are set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and the channel bit rate list are specified by the “hpe5022_measureChannelBitRateSweep” function or “hpe5022_setupChannelBitRateSweep” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_PW	5	PW
hpe5022_DATA_PW_POS	6	PW positive
hpe5022_DATA_PW_NEG	7	PW negative
hpe5022_DATA_PW_ASYM	8	PW asymmetry
hpe5022_DATA_OVERWRITE	13	Overwrite
hpe5022_DATA_NLTS_5TH	14	NLTS 5th
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description — Returns the data (set in array) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureChannelBitRateSweep” or “hpe5022_setupChannelBitRateSweep” function. Each value in the array is the measurement result at each channel bit rate list.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureChannelBitRateSweep” on page 679

“hpe5022_setupChannelBitRateSweep” on page 682

Roll Off Measurement Function

This section describes the functions related with the roll off measurement. The roll off measurement is a flux frequency sweep measurement. This measurement allows you to observe a flux frequency characteristics of the parameters TAA and narrow band TAA.

hpe5022_measureRollOff

C Syntax

ViStatus hpe5022_measureRollOff(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 fluxFreq[]);

Visual Basic Syntax

hpe5022_measureRollOff(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef fluxFreq As Double) As Long

Description

This function measures the roll off characteristic curve. The sequences of this function are as follows:

1. Set i to zero. (i = 0)
2. Erase (same as the “hpe5022_erase” function) for an entire track.
3. Set the flux frequency at ‘fluxFreq[i]’ (see parameters).
4. Write a repetitive pattern (1T) at the specified flux frequency for an entire track.
5. Move the head to the read track offset specified by the “hpe5022_readTrackOffset” function.
6. Make measurement as specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
7. Increment i by 1. (i = i+1)
8. If i is not equal to ‘points’ (see parameters), go to the step 2.

The “hpe5022_rollOffData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the flux frequency list.

  Direction IN

  Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- fluxFreq

  Description Specifies the flux frequency data as the data array. The flux frequency will be set to data in the array order. The maximum limit of this value is returned by the “hpe5022_channelBitRateRange_Q” function.

  Direction IN

  Unit flux per second

  Values

Name	Value
hpe5022_ROLLOFF_FREQ_MIN	4×10^{6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘points’ and/or ‘fluxFreq’ is out of range.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_rollOffData_Q” on page 691

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_driveState” on page 209

“hpe5022_channelBitRateRange_Q” on page 118

hpe5022_setupRollOff

C Syntax

ViStatus hpe5022_setupRollOff(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 fluxFreq[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupRollOff(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef fluxFreq As Double, ByRef testHndl As Long) As Long

Description

This function assigns the roll off measurement sequence to the specified test identifier. Refer to the “hpe5022_measureRollOff” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_rollOffData_Q” function returns the measurement result

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measureRollOff” function.

- points

  Description — Specifies the number of measurement points. This number must be the same as the array size of the flux frequency list.

  Direction — IN

  Values — Same as the ‘points’ in the “hpe5022_measureRollOff” function.

- fluxFreq

  Description — Specifies the flux frequency data as the data array. The flux frequency will be set to data in the array order.

  Direction — IN

  Unit — flux per second

  Values — Same as the ‘fluxFreq’ in the “hpe5022_measureRollOff” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the roll off measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’. ‘points’ and/or ‘fluxFreq’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureRollOff” on page 686

“hpe5022_rollOffData_Q” on page 691

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_rollOffData_Q

C Syntax

ViStatus hpe5022_rollOffData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_rollOffData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the roll off measurement. The returned data are set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and the flux frequency list are specified by the “hpe5022_measureRollOff” function or “hpe5022_setupRollOff” function

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_TAA_ASYM	4	TAA asymmetry
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description — Returns the data (set in array) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureRollOff” or “hpe5022_setupRollOff” function. Each value in the array is the measurement result relative to the flux frequency that corresponds to the same order value in the array of flux frequency list.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureRollOff” on page 686

“hpe5022_setupRollOff” on page 689

hpe5022_calculateD50_Q

C Syntax

ViStatus hpe5022_calculateD50_Q(ViSession id, ViInt16 dataType, ViPReal64 d50);

Visual Basic Syntax

hpe5022_calculateD50_Q(ByVal id As Long, ByVal dataType As Integer, ByRef d50 As Double) As Long

Description

This function calculates D_{50} (Roll Off Density) in flux per second. D_{50} is a flux frequency value at half of the peak level in the data returned by the “hpe5022_rollOffData_Q” function. When you want to convert an unit from flux per second to flux per meter use the “hpe5022_calculateTimeToLength_Q” function.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataType

  Description Specifies the type of data to be reported.

  Direction IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_TAA_POS	2	TAA positive
hpe5022_DATA_TAA_NEG	3	TAA negative
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- d50

  Description Returns the D50.

  Direction OUT

  Unit flux per second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_D50_NOT_FOUND	Can’t find the D50 parameter point. Check the data returned by the “hpe5022_rollOffData_Q” function.

See Also

“hpe5022_measureRollOff” on page 686

“hpe5022_setupRollOff” on page 689

“hpe5022_rollOffData_Q” on page 691

“hpe5022_calculateTimeToLength_Q” on page 695

hpe5022_calculateTimeToLength_Q

C Syntax

ViStatus hpe5022_calculateTimeToLength_Q(ViSession id,ViReal64 time,ViPReal64 length);

Visual Basic Syntax

hpe5022_calculateTimeToLength_Q(ByVal id As Long, ByVal time As Double, ByRef length As Double) As Long

Description

This function converts time to length by using the current spindle speed and head position radius.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- time

  Description: Specifies time to be converted to length.

  Direction: IN

  Unit: Second

- length

  Description: Returns the length.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_driveState” on page 209

Precompensation Value Sweep Measurement Function

This section describes the functions related with precompensation value sweep measurement. This measurement allows you to observe a precompensation characteristics of the NLTS. NLTS is measured by the 5th harmonic method. See chapter 5 in the Operaion Manual.

hpe5022_measurePrecompSweep

C Syntax

ViStatus hpe5022_measurePrecompSweep(ViSession id, ViInt16 measFunc,ViInt16 points, const ViReal64 precomp[]);

Visual Basic Syntax

hpe5022_measurePrecompSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal measFunc As Integer, ByVal points As Integer, ByRef precomp As Double) As Long

Description

This function measures the precompensation value measurement. The sequences of this function are as follows:

1. Set i to zero. (i = 0)
2. Set precompensation value of which number is 1 at ‘precomp[i]’ (see parameters).
3. Erase (same as the “hpe5022_erase” function) for an entire track.
4. Measure NLTS (same as the “hpe5022_measureNlts5th” function).
5. Increment i by 1. (i = i+1)
6. If i is not equal to ‘points’ (see parameters), go to the step 2.

The “hpe5022_precompSweepData_Q” function returns the measurement result. Before the precompensation is performed, the “001” must be set as one of the prebit of precompensation pattern and the precompensation status of NLTS 5th pattern must be turned on. See “Precompensation Delay Sweep Measurement” on page 59.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

Values

Name	Value	Description
hpe5022_MEAS_NLTS_5TH	4	Measure NLTS 5th

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the precompensation value list.

  Direction IN

  Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- precomp

  Description Specifies the precompensation value data as the data array. The precompensation value will be set to the data in the array order.

  Direction IN

  Unit Second

  Values

Name	Value
hpe5022_PREC_VAL_MIN	-1900×10^{-12}
hpe5022_PREC_VAL_MAX	1900×10^{-12}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘points’ and/or ‘precomp’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_precompSweepData_Q” on page 701

“hpe5022_measureNlts5th” on page 810

“hpe5022_driveState” on page 209

hpe5022_setupPrecompSweep

C Syntax

ViStatus hpe5022_setupPrecompSweep(ViSession id, ViInt16 measFunc,ViInt16 points, const ViReal64 precomp[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupPrecompSweep(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef precomp As Double, ByRef testHndl As Long) As Long

Description

This function assigns the precompensation value sweep measurement sequence to the specified test identifier. Refer to the “hpe5022_measurePrecompSweep” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_precompSweepData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measurePrecompSweep” function.

- points

  Description — Specifies the number of measurement points. This number must be the same as the array size of the precompensation value list.

  Direction — IN

  Values — Same as the ‘points’ in the “hpe5022_measurePrecompSweep” function.

- precomp

  Description — Specifies the precompensation value data as the data array. The precompensation value will be set to data in the array order.

  Direction — IN

  Unit — Second

  Values — Same as the ‘precomp’ in the “hpe5022_measurePrecompSweep” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the precompensation sweep measurement by the "hpe5022_measure" function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'measFunc', 'points' and/or 'precomp' is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measurePrecompSweep" on page 696

"hpe5022_precompSweepData_Q" on page 701

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_precompSweepData_Q

C Syntax

ViStatus hpe5022_precompSweepData_Q(ViSession id, ViInt16 dataType,ViReal64 data[]);

Visual Basic Syntax

hpe5022_precompSweepData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the precompensation value sweep measurement. The returned data are set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and the precompensation value list are specified by the “hpe5022_measurePrecompSweep” or “hpe5022_setupPrecompSweep” functions.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_NLTS_5TH	14	NLTS 5th

- data

  Description: Returns the data (set in array) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measurePrecompSweep” or “hpe5022_setupPrecompSweep” function. Each value in the array is the measurement result relative to the precompensation value that corresponds to the same order value in the array of precompensation value list.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measurePrecompSweep” on page 696

“hpe5022_setupPrecompSweep” on page 699

Standard Triple Track Test Measurement Function

This section describes the functions related with the standard triple track test measurement.

hpe5022_tripleTrackWindowConfig

C Syntax

ViStatus hpe5022_tripleTrackWindowConfig(ViSession id, ViBoolean bestFitWin, ViReal64 winSize, ViReal64 high, ViReal64 low);

Visual Basic Syntax

hpe5022_tripleTrackWindowConfig(ByVal id As Long, ByVal bestFitWin As Integer, ByVal winSize As Double, ByVal high As Double, ByVal low As Double) As Long

Description

This function controls the window configuration of the triple track test in order to calculate the regression lines.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bestFitWin

  Description: Specifies the use of the Best Fit Window.

  Direction: IN

  Values

Name	Value	Description
VI_FALSE	0	Don’t use the best fit window.
VI_TRUE	1	Use the best fit window.

  Preset Value: VI_TRUE

- winSize

  Description: Specifies the size of the Best Fit Window when “bestFitWin” is set to VI_TRUE.

  Direction: IN

  Preset Value: 0.75

- high

  Description: Specifies the high threshold limit to calculate the regression lines.

Direction IN

Preset Value 1.0

- low

  Description Specifies the low threshold limit to calculate the regression lines. The floor level is defined as 0.0 in case of best fit window enable.

  Direction IN

  Preset Value 0.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘bestFitWin’, ‘winSize’, ‘high’ and/or ‘low’ is out of range

See Also

“hpe5022_measureTripleTrack” on page 714

“hpe5022_setupTripleTrack” on page 719

“hpe5022_tripleTrackWindowConfig_Q” on page 705

hpe5022_tripleTrackWindowConfig_Q

C Syntax

ViStatus hpe5022_tripleTrackWindowConfig_Q(ViSession id, ViPBoolean bestFitWin, ViPReal64 winSize, ViPReal64 high, ViPReal64 low);

Visual Basic Syntax

hpe5022_tripleTrackWindowConfig_Q(ByVal id As Long, ByRef bestFitWin As Integer, ByRef winSize As Double, ByRef high As Double, ByRef low As Double) As Long

Description

This function returns the current setting of the window configuration for triple track test.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bestFitWin

  Description: Returns the use of the Best Fit Window.

  Direction: OUT

  Values: Same as the ‘bestFitWin’ in the “hpe5022_tripleTrackWindowConfig” function.

- winSize

  Description: Returns the size of the Best Fit Window.

  Direction: OUT

- high

  Description: Returns the high interpolation limit of the window.

  Direction: OUT

- low

  Description: Returns the low interpolation limit of the window.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also “hpe5022_tripleTrackWindowConfig” on page 703

hpe5022_tripleTrackPositionConfig

C Syntax

ViStatus hpe5022_tripleTrackPositionConfig(ViSession id, ViReal64 squeezePos, ViReal64 otrcWrPos);

Visual Basic Syntax

hpe5022_tripleTrackPositionConfig(ByVal id As Long, ByVal squeezePos As Double, ByVal otrcWrPos As Double) As Long

Description

This function controls the position of the adjacent track and data track.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- squeezePos

  Description: Specifies the position of the adjacent track.

  Direction: IN

  Unit: Meter

  Preset Value: 3×10^{-6}

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- otrcWrPos

  Description: Specifies the write position of the data track. This value is specified as the ratio of the write width.

  Direction: IN

  Unit: Meter

  Preset Value: 0.65

  Values

Name	Value
hpe5022_OTRC_WRIT_POS_MIN	0
hpe5022_OTRC_WRIT_POS_MAX	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘squeezePos’ and/or ‘otrcWrPos’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue _Q” function. If it is not 0, the ‘squeezePos’ range is narrowed.

See Also

“hpe5022_measureTripleTrack” on page 714

“hpe5022_setupTripleTrack” on page 719

“hpe5022_tripleTrackPositionConfig_Q” on page 709

hpe5022_tripleTrackPositionConfig_Q

C Syntax

ViStatus hpe5022_tripleTrackPositionConfig(ViSession id, ViPReal64 squeezePos, ViPReal64 otrcWrPos);

Visual Basic Syntax

hpe5022_tripleTrackPositionConfig(ByVal id As Long, ByRef squeezePos As Double, ByRef otrcWrPos As Double) As Long

Description

This function returns the specified “squeezePos” and “otrcWrPos”.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- squeezePos

  Description: Returns the squeeze position.

  Direction: IN

  Unit: Meter

- otrcWrPos

  Description: Returns the write position of data track.

  Direction: IN

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_tripleTrackPositionConfig” on page 707

hpe5022_tripleTrackAdjacentTrackConfig

C Syntax

ViStatus hpe5022_tripleTrackAdjacentTrackConfig(ViSession id, ViInt16 pat);

Visual Basic Syntax

hpe5022_tripleTrackAdjacentTrackConfig(ByVal id As Long, ByVal pat As Integer) As Long

Description

This function controls the data pattern of the adjacent track.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pat

  Description: Specifies the data pattern of the adjacent track.

  Direction: IN

  Preset value: hpe5022_PAT_ERASE

  Value

Name	Value	Description
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_USER	20	User-defined Pattern
hpe5022_PAT_ERASE	101	Default Erase Pattern specified by the “hpe5022_eraseType” function.
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dat’ is out of range.

See Also

“hpe5022_tripleTrackAdjacentTrackConfig_Q” on page 712

hpe5022_tripleTrackAdjacentTrackConfig_Q

C Syntax

ViStatus hpe5022_tripleTrackAdjacentTrackConfig_Q(ViSession id, ViPInt16 pat);

Visual Basic Syntax

hpe5022_tripleTrackAdjacentTrackConfig_Q(ByVal id As Long, ByRef pat As Integer) As Long

Description

This function returns the data pattern of the adjacent track.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pat

  Description: Returns the data pattern of the adjacent track.

  Direction: IN

  Value

Name	Value	Description
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_USER	20	User-defined Pattern
hpe5022_PAT_ERASE	101	Default Erase Pattern specified by the “hpe5022_eraseType” function.
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_tripleTrackAdjacentTrackConfig” on page 710

hpe5022_measureTripleTrack

C Syntax

ViStatus hpe5022_measureTripleTrack(ViSession id, ViInt16 seqType, ViInt16 measFunc, ViReal64 offsStart, ViReal64 offsStop, ViInt16 points);

Visual Basic Syntax

hpe5022_measureTripleTrack(ByVal id As Long, ByVal seqType As Integer, ByRef measFunc As Integer, ByRef offsStart As Double, ByRef offsStop As Double, ByVal points As Integer) As Long

Description

This function performs the triple track measurement.

The detailed sequences of this function is as follows:

1. When the “seqType” is set to “hpe5022_SEQ_ER_WR_M”, perform the three-track-erase. (the entire track erase for 5 positions, the track, its both side tracks, and the center between them.)
2. When the “seqType” is set to “hpe5022_SEQ_ER_WR_M”, Move the head to the center of the track.
3. When the “seqType” is set to “hpe5022_SEQ_ER_WR_M”, write the data pattern specified by the “hpe5022_selectPattern” function.
4. Perform the track profile measurement (seqType: hpe5022_SEQ_M, mode: hpe5022_TP_FORWARD, Start Offset: offsStart, Stop Offset: offsStop, Number of Points: points).
5. Calculate a write width, read width, read-write offset and maximum TAA of the track profile measurement.
6. Set the read track offset to the calculated read-write offset value.
7. Calculate Squeezed OTRC.
8. Move the head to the ‘squeezePos’ position specified by the “hpe5022_tripleTrackPositionConfig” function.
9. Write the data pattern on the adjacent track as specified by the “hpe5022_tripleTrackAdjacentTrackConfig” function.
10. Move the head to the -‘squeezePos’ position as specified by the “hpe5022_tripleTrackPositionConfig” function.
11. Write the data pattern on the adjacent track as specified by the “hpe5022_tripleTrackAdjacentTrackConfig” function.
12. Move the head to the same point where the maximum TAA was measured at step 5.
13. Measure TAA
14. Calculate squash.
15. Move the head to the ‘otrcWrPos’ position specified by the “hpe5022_tripleTrackPositionConfig” function.
16. Write the data pattern specified by the “hpe5022_selectPattern” function.

17. Move the head to the -‘otrcWrPos’ position specified by the “hpe5022_tripleTrackPositionConfig” function.
18. Write the data pattern specified by the “hpe5022_selectPattern” function.
19. Move the head to the center of the track.
20. Write the adjacent data pattern as specified by the “hpe5022_tripleTrackAdjacentTrackConfig” function.
21. Perform the OTRC track profile measurement (seqType: hpe5022_SEQ_M, mode: hpe5022_TP_FORWARD, Start Offset: offsStart, Stop Offset: offsStop, Number of Points: points)
22. Calculate OTRC (off-track read capability).

The “hpe5022_tripleTrack_Q” and “hpe5022_tripleTrackProfile_Q” functions return the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	3 Track erase-->Write-->Measure Triple Track Test
hpe5022_SEQ_WR_M	1	Measure triple track test only

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA
hpe5022_MEAS_NB_TAA	5	Measure Narrow Band TAA

- offsStart

Description — Specifies the start offset position for track profile measurement of the Triple Track Test.

If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

Direction — IN

Unit — Meter

Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- offsStop

Description — Specifies the stop offset position for track profile measurement of the Triple Track Test.

If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

Direction — IN

Unit — Meter

Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- points

Description — Specifies the number of measurement points for track profile measurement.

Direction — IN

Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_TRACK_PROFILE_SIZE_MAX	201

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_AUTO_RANGE_FAIL	The autoranging in the track profile measurement is failed. Check if the paramter setting and head is correct.
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘Offsstart’, ‘Offsstop’, and/or ‘points’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q” function. If it is not 0, the ‘Offsstart’ or ‘Offsstop’ range is narrowed.

See Also

“hpe5022_tripleTrackWindowConfig” on page 703

“hpe5022_tripleTrackPositionConfig” on page 707

“hpe5022_tripleTrackAdjacentTrackConfig” on page 710

“hpe5022_tripleTrack_Q” on page 722

“hpe5022_tripleTrackProfile_Q” on page 725

“hpe5022_measureTrackProfile” on page 615

“hpe5022_selectPattern” on page 119

“hpe5022_driveState” on page 209

hpe5022_setupTripleTrack

C Syntax

ViStatus hpe5022_setupTripleTrack(ViSession id, ViInt16 seqType, ViInt16 measFunc, ViReal64 start, ViReal64 stop, ViInt16 points, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupTripleTrack(ByVal id As Long, ByVal seqType As Integer, ByRef measFunc As Integer, ByRef start As Double, ByRef stop As Double, ByVal points As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the triple track measurement sequence to the specified test identifier. Refer to the “hpe5022_measureTripleTrack” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_tripleTrack_Q” and “hpe5022_tripleTrackProfile_Q” functions return the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- seqType

  Description: Specifies the type of measurement sequence.

  Direction: IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	3 Track erase-->Write-->Measure Triple Track Test
hpe5022_SEQ_WR_M	1	Measure triple track test only

- measFunc

  Description: Specifies the type of measurement.

  Direction: IN

  Values: Same as the ‘measFunc’ in the “hpe5022_measureTripleTrack” function.

- start

  Description: Specifies the start offset position for track profile measurement of the Triple Track Test.

  Direction: IN

Unit Meter

Values Same as the ‘offsStart’ in the “hpe5022_measureTripleTrack” function.

- stop

  Description Specifies the stop offset position for track profile measurement of the Triple Track Test.

  Direction IN

  Unit Meter

  Values Same as the ‘offsStop’ in the “hpe5022_measureTripleTrack” function.

- points

  Description Specifies the number of measurement points for track profile measurement.

  Direction IN

  Values Same as the ‘points’ in the “hpe5022_measureTripleTrack” function.

- testHndl

  Description Returns the test identifier. This identifier is used to execute the triple track measurement by the “hpe5022_measure” function.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘Offstart’, ‘Offstop’, and/or ‘points’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureTripleTrack” on page 714

“hpe5022_tripleTrack_Q” on page 722

“hpe5022_tripleTrackProfile_Q” on page 725

“hpe5022_tripleTrackWindowConfig” on page 703

“hpe5022_tripleTrackPositionConfig” on page 707

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_tripleTrack_Q

C Syntax

ViStatus hpe5022_tripleTrack_Q(ViSession id, ViInt16 dataType, ViInt16 rwOffsPos, ViPReal64 ww, ViPReal64 rw, ViPReal64 rwOfst, ViPReal64 maxTaa, ViPReal64 squash, ViPReal64 otrc, ViPReal64 sqOtrc);

Visual Basic Syntax

hpe5022_tripleTrack_Q Lib(ByVal id As Long, ByVal dataType As Integer, ByVal rwOffsPos As Integer, ByRef ww As Double, ByRef rw As Double, ByRef rwOffset As Double, ByRef maxTaa As Double, ByRef squash As Double, ByRef otrc As Double, ByRef sqOtrc As Double) As Long

Description

This function returns the result of the triple track measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be queried.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- rwOffsPos

  Description: Specifies the calculation method of the read write offset. This is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function.

  Direction: IN

  Values

Name	Value	Description
hpe5022_RWOF_POS_50PERC_CENT	0	Center Point at 50% level
hpe5022_RWOF_POS_PROF_MAX	1	Maximum point of track profile
hpe5022_RWOF_POS_CROS_INT_LINE	2	Least Sq. Approx. Linear Line Cross Point

- ww

  Description — Returns the write track width. The definition of the write track width is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function. The floor level is set at 0% in that function.

  Direction — OUT

  Unit — Meter

- rw

  Description — Returns the read track width. The definition of the write track width is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function. The floor level is set at 0% in that function.

  Direction — OUT

  Unit — Meter

- rwOfst

  Description — Returns the read-write offset according to the setting of ‘rwOffsPos’. The definition of the read/write offset is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function. The floor level is set at 0% in that function.

  Direction — OUT

  Unit — Meter

- maxTaa

  Description — Returns the maximum TAA value of the track profile result.

  Direction — OUT

  Unit — Volt

- squash

  Description — Returns the squash ratio.

  Direction — OUT

- otrc

  Description — Returns the OTRC(Off-Track Read Capability).

  Direction — OUT

  Unit — Meter

- sqOtr

  Description — Returns the squeezed OTRC.

  Direction — OUT

Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The triple track test data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureTripleTrack” on page 714

“hpe5022_setupTripleTrack” on page 719

“hpe5022_calculateReadWriteOffset_Q” on page 628

hpe5022_tripleTrackProfile_Q

C Syntax

ViStatus hpe5022_tripleTrackProfile_Q(ViSession id, ViInt16 dataType, ViInt16 profileType, ViReal64 profile[], ViPReal64 interceptL, ViPReal64 slopeL, ViPReal64 interceptR, ViPReal64 slopeR);

Visual Basic Syntax

hpe5022_tripleTrackProfile_Q(ByVal id As Long, ByVal dataType As Integer, ByVal profileType As Integer, ByRef profile As Double, ByRef interceptL As Double, ByRef slopeL As Double, ByRef interceptR As Double, ByRef slopeR As Double) As Long

Description

This function returns the result of the track profile data and the regression line data.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataType

  Description Specifies the type of data to get the results.

  Direction IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA
hpe5022_DATA_NB_TAA	15	Narrow Band TAA

- profileType

  Description Specifies type of track profile.

  Direction IN

  Values

Name	Value	Description
hpe5022_TRIPLE_TRACK_SINGLE	1	Single Profile
hpe5022_TRIPLE_TRACK_SQUEEZE	2	Squeezed Profile
hpe5022_TRIPLE_TRACK_OTRC	3	OTRC Profile

- profile

  Description Returns the data (set in array) of the parameter specified by the 'dataType' and ‘profileType’. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureTripleTrack” function or “hpe5022_setupTripleTrack” function.

  Direction OUT

- interceptL

  Description Returns the intercept of the left regression line.

  Direction OUT

  Unit Meter

- slopeL

  Description Returns the slope of the left regression line.

  Direction OUT

- interceptR

  Description Returns the intercept of the right regression line.

  Direction OUT

  Unit Meter

- slopeR

  Description Returns the slope of the right regression line.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The triple track test data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureTripleTrack” on page 714

“hpe5022_setupTripleTrack” on page 719

SNR Triple Track Test Measurement Function

This section describes the functions related with the SNR triple track test measurement.

hpe5022_snrTripleTrackWindowConfig

C Syntax

ViStatus hpe5022_snrTripleTrackWindowConfig(ViSession id, ViBoolean bestFitWin, ViReal64 winSize, ViReal64 high, ViReal64 low);

Visual Basic Syntax

hpe5022_snrTripleTrackWindowConfig(ByVal id As Long, ByVal bestFitWin As Integer, ByVal winSize As Double, ByVal high As Double, ByVal low As Double) As Long

Description

This function controls the window configuration of the SNR triple track test in order to calculate the regression lines.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- bestFitWin

  Description: Specifies the use of the best fit window.

  Direction: IN

  Values

Name	Value	Description
VI_FALSE	0	Don’t use the best fit window.
VI_TRUE	1	Use the best fit window.

  Preset Value: VI_TRUE

- winSize

  Description: Specifies the size of the best fit window when “bestFitWin” is set to VI_TRUE.

  Direction: IN

  Preset Value: 0.75

- high

  Description: Specifies the high threshold limit to calculate the regression lines.

Direction IN

Preset Value 1.0

- low

  Description Specifies the low threshold limit to calculate the regression lines.

  Direction IN

  Preset Value 0.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘bestFitWin’, ‘winSize’, ‘high’ and/or ‘low’ is out of range

See Also

“hpe5022_measureSnrTripleTrack” on page 757

“hpe5022_setupSnrTripleTrack” on page 766

“hpe5022_snrTripleTrackWindowConfig_Q” on page 730

hpe5022_snrTripleTrackWindowConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackWindowConfig_Q(ViSession id, ViPBoolean bestFitWin, ViPReal64 winSize, ViPReal64 high, ViPReal64 low);

Visual Basic Syntax

hpe5022_snrTripleTrackWindowConfig_Q(ByVal id As Long, ByRef bestFitWin As Integer, ByRef winSize As Double, ByRef high As Double, ByRef low As Double) As Long

Description

This function returns the current setting of the window configuration for SNR triple track test.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- bestFitWin

  Description — Returns the use of the best fit window.

  Direction — OUT

  Values — Same as the ‘bestFitWin’ in the “hpe5022_tripleTrackWindowConfig” function.

- winSize

  Description — Returns the size of the best fit window.

  Direction — OUT

- high

  Description — Returns the high interpolation limit of the window.

  Direction — OUT

- low

  Description — Returns the low interpolation limit of the window.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also “hpe5022_snrTripleTrackWindowConfig” on page 728

hpe5022_snrTripleTrackPositionConfig

C Syntax

ViStatus hpe5022_snrTripleTrackPositionConfig(ViSession id, ViReal64 squeezePos, ViReal64 otrcWrPos);

Visual Basic Syntax

hpe5022_snrTripleTrackPositionConfig(ByVal id As Long, ByVal squeezePos As Double, ByVal otrcWrPos As Double) As Long

Description

This function controls the position of the adjacent track and data track.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- squeezePos

  Description: Specifies the position of the adjacent track.

  Direction: IN

  Unit: Meter

  Preset Value: 3×10^{-6}

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- otrcWrPos

  Description: Specifies the write position of the data track. This value is specified as the ratio of the track pitch specified by hpe5022_dataArea.

  Direction: IN

  Unit: (none)

  Preset Value: 0.65

  Values

Name	Value
hpe5022_OTRC_WRIT_POS_MIN	0
hpe5022_OTRC_WRIT_POS_MAX	1

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘squeezePos’ and/or ‘otrcWrPos’ is out of range.

See Also

“hpe5022_measureSnrTripleTrack” on page 757

“hpe5022_setupSnrTripleTrack” on page 766

“hpe5022_snrTripleTrackPositionConfig” on page 732

“hpe5022_dataArea” on page 180

hpe5022_snrTripleTrackPositionConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackPositionConfig(ViSession id, ViPReal64 squeezePos, ViPReal64 otrcWrPos);

Visual Basic Syntax

hpe5022_snrTripleTrackPositionConfig(ByVal id As Long, ByRef squeezePos As Double, ByRef otrcWrPos As Double) As Long

Description

This function returns the specified “squeezePos” and “otrcWrPos”.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- squeezePos

  Description — Returns the position of the adjacent track.

  Direction — OUT

  Unit — Meter

- otrcWrPos

  Description — Returns the write position of the data track.

  Direction — OUT

  Unit — (none)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackPositionConfig” on page 732

hpe5022_snrTripleTrackAdjacentTrackConfig

C Syntax

ViStatus hpe5022_snrTripleTrackAdjacentTrackConfig(ViSession id, ViInt16 pat);

Visual Basic Syntax

hpe5022_snrTripleTrackAdjacentTrackConfig(ByVal id As Long, ByVal pat As Integer) As Long

Description

This function controls the data pattern of the adjacent track.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- pat

  Description Specifies the data pattern of the adjacent track.

  Direction IN

  Preset value hpe5022_PAT_ERASE

  Value

Name	Value	Description
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_USER	20	User-defined Pattern
hpe5022_PAT_ERASE	101	Default Erase Pattern specified by the “hpe5022_eraseType” function.
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘pat’ is out of range.

See Also

“hpe5022_snrTripleTrackAdjacentTrackConfig_Q” on page 737

hpe5022_snrTripleTrackAdjacentTrackConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackAdjacentTrackConfig_Q(ViSession id, ViPInt16 pat);

Visual Basic Syntax

hpe5022_snrTripleTrackAdjacentTrackConfig_Q(ByVal id As Long, ByRef pat As Integer) As Long

Description

This function returns the data pattern of the adjacent track.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- pat

  Description Returns the data pattern of the adjacent track.

  Direction OUT

  Value

Name	Value	Description
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_USER	20	User-defined Pattern
hpe5022_PAT_ERASE	101	Default Erase Pattern specified by the “hpe5022_eraseType” function.
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackAdjacentTrackConfig” on page 735

hpe5022_snrTripleTrackEraseBandConfig

C Syntax

ViStatus hpe5022_snrTripleTrackEraseBandConfig(ViSession id, ViReal64 range, ViReal64 pitch);

Visual Basic Syntax

hpe5022_snrTripleTrackEraseBandConfig(ByVal id As Long, ByVal range As Double, ByVal pitch As Double) As Long

Description

This function specifies parameters for band erase in the SNR triple track test. Band erase is performed from −(specified range) to +(specified range) with specified pitch.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- range

  Description: Specifies the erase range. Minimum and maximum values are 0 and 6.0×10^{-6} respectively.

  Direction: IN

  Unit: Meter

  Preset Value: 3.0×10^{-6}

  Values

Name	Value
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- pitch

  Description: Specifies the erasing pitch.

  Direction: IN

  Unit: Meter

  Preset Value: 0.3×10^{-6}

  Values

Name	Value
hpe5022_ERASE_BAND_PITCH_MIN	50×10^{-9}
hpe5022_ERASE_BAND_PITCH_MAX	6.0×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘range’ or ‘pitch’ is out of range.

See Also

“hpe5022_snrTripleTrackEraseBandConfig_Q” on page 741

hpe5022_snrTripleTrackEraseBandConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackEraseBandConfig_Q(ViSession id, ViPReal64 range, ViPReal64 pitch);

Visual Basic Syntax

hpe5022_snrTripleTrackEraseBandConfig_Q(ByVal id As Long, ByRef range As Double, ByRef pitch As Double) As Long

Description

This function returns parameters for band erase in the SNR triple track test.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- range

  Description: Returns the erase range.

  Direction: OUT

  Unit: Meter

- pitch

  Description: Returns the erasing pitch.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackPositionConfig” on page 732

hpe5022_snrTripleTrackSnrThresholdConfig

C Syntax

ViStatus hpe5022_snrTripleTrackSnrThresholdConfig(ViSession id, ViReal64 threshold);

Visual Basic Syntax

hpe5022_snrTripleTrackSnrThresholdConfig(ByVal id As Long, ByVal threshold As Double) As Long

Description

This function sets the SNR threshold configuration for SNR triple track test.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- threshold

  Description — Specifies the SNR threshold.

  Direction — IN

  Unit — dB

  Preset value — 0.0

  Value

Name	Value
hpe5022_SNR_TRIPLE_TRACK_SNR_THR_MIN	0.0
hpe5022_SNR_TRIPLE_TRACK_SNR_THR_MAX	50.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘threshold’ is out of range.

See Also

“hpe5022_snrTripleTrackSnrThresholdConfig_Q” on page 743

hpe5022_snrTripleTrackSnrThresholdConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackSnrThresholdConfig_Q(ViSession id, ViPReal64 threshold);

Visual Basic Syntax

hpe5022_snrTripleTrackSnrThresholdConfig_Q(ByVal id As Long, ByRef threshold As Double) As Long

Description

This function queries the SNR threshold configuration for SNR triple track test.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- threshold

  Description — Returns the SNR threshold.

  Direction — OUT

  Unit — dB

  Value

Name	Value
hpe5022_SNR_TRIPLE_TRACK_SNR_THR_MIN	0.0
hpe5022_SNR_TRIPLE_TRACK_SNR_THR_MAX	50.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackSnrThresholdConfig” on page 742

hpe5022_snrTripleTrackNoiseConstantConfig

C Syntax

ViStatus hpe5022_snrTripleTrackNoiseConstantConfig(ViSession id, ViBoolean addNoise, ViReal64 ng, ViReal64 ni);

Visual Basic Syntax

hpe5022_snrTripleTrackNoiseConstantConfig(ByVal id As Long, ByVal addNoise As Integer, ByVal ng As Double, ByVal ni As Double) As Long

Description

This function sets the noise constant configuration for SNR triple track test.

These values are used for SNR calculation.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- addNoise

  Description — Specifies the flag which is used to calculate *"OTRC SNR"* or *"Squeezed SNR."* See step 8 on page 760 for details.

  Direction — IN

  Preset Value — VI_TRUE

  Values

Name	Description
VI_TRUE	Average noise floor is added to signal term in SNR calcuration.
VI_FALSE	Average noise floor is not added to signal term in SNR calculation.

- ng

  Description — Specifies the Gausian noise constant which is used to calculate *"OTRC SNR"* or *"Squeezed SNR."*. See step 8 on page 760 for details.

  Direction — IN

  Preset Value — 1

  Values

Name	Value
hpe5022_SNR_TRIPLE_TRACK_NOISE_CONST_MIN	0
hpe5022_SNR_TRIPLE_TRACK_NOISE_CONST_MAX	10

- ni

  Description — Specifies the interference noise constant which is used to calculate *"OTRC SNR"* or *"Squeezed SNR."*. See step 8 on page 760 for details.

  Direction — IN

  Preset Value — 1

  Values

Name	Value
hpe5022_SNR_TRIPLE_TRACK_NOISE_CONST_MIN	0
hpe5022_SNR_TRIPLE_TRACK_NOISE_CONST_MAX	10

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'addNoise', 'ng', and/or 'ni' is out of range.

See Also

"hpe5022_snrTripleTrackNoiseConstantConfig_Q" on page 746

hpe5022_snrTripleTrackNoiseConstantConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackNoiseConstantConfig_Q(ViSession id, ViPBoolean addNoise, ViPReal64 ng, ViPReal64 ni);

Visual Basic Syntax

hpe5022_snrTripleTrackNoiseConstantConfig_Q(ByVal id As Long, ByRef addNoise As Integer, ByRef ng As Double, ByRef ni As Double) As Long

Description

This function queries the noise constant configuration for SNR triple track test.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- addNoise

  Description: Returns the flag which is used to calculate *"OTRC SNR"* or *"Squeezed SNR."* See step 8 on page 760 for details.

  Direction: OUT

  Values

Name	Description
VI_TRUE	Average noise floor is added to signal term in SNR calcuration.
VI_FALSE	Average noise floor is not added to signal term in SNR calculation.

- ng

  Description: Returns the Gausian noise constant which is used to calculate *"OTRC SNR"* or *"Squeezed SNR."*. See step 8 on page 760 for details.

  Direction: OUT

  Values

Name	Value
hpe5022_SNR_TRIPLE_TRACK_NG_COEF_MIN	0
hpe5022_SNR_TRIPLE_TRACK_NG_COEF_MAX	10

- ni

  Description: Returns the interference noise constant which is used to calculate *"OTRC SNR"* or *"Squeezed SNR."*. See step 8 on page 760 for details.

Direction OUT

Values

Name	Value
hpe5022_SNR_TRIPLE_TRACK_NI_COEF_MIN	0
hpe5022_SNR_TRIPLE_TRACK_NI_COEF_MAX	10

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackNoiseConstantConfig” on page 744.

hpe5022_snrTripleTrackNoiseThresholdConfig

C Syntax

ViStatus hpe5022_snrTripleTrackNoiseThresholdConfig(ViSession id, ViReal64 threshold);

Visual Basic Syntax

hpe5022_snrTripleTrackNoiseThresholdConfig(ByVal id As Long, ByVal threshold As Double) As Long

Description

This function sets the noise threshold configuration for SNR triple track test.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- threshold

  Description — Specifies the noise threshold ratio for the SNR triple track test.

  Direction — IN

  Preset value — 0.0

  Value

Name	Value
hpe5022_SNR_TRIPLE_TRACK_NOISE_THR_MIN	0.0
hpe5022_SNR_TRIPLE_TRACK_NOISE_THR_MAX	1.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘threshold’ is out of range.

See Also

“hpe5022_snrTripleTrackNoiseThresholdConfig_Q” on page 749

hpe5022_snrTripleTrackNoiseThresholdConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackNoiseThresholdConfig_Q(ViSession id, ViPReal64 threshold);

Visual Basic Syntax

hpe5022_snrTripleTrackNoiseThresholdConfig_Q(ByVal id As Long, ByRef threshold As Double) As Long

Description

This function queries the noise threshold configuration for SNR triple track test.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- threshold

  Description: Returns the noise threshold ratio for the SNR triple track test.

  Direction: OUT

  Value

Name	Value
hpe5022_SNR_TRIPLE_TRACK_NOISE_THR_MIN	0.0
hpe5022_SNR_TRIPLE_TRACK_NOISE_THR_MAX	1.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackNoiseThresholdConfig” on page 748

hpe5022_snrTripleTrackFilterConfig

C Syntax

ViStatus hpe5022_snrTripleTrackFilterConfig(ViSession id, ViBoolean state, ViInt16 filterFactor);

Visual Basic Syntax

hpe5022_snrTripleTrackFilterConfig(ByVal id As Long, ByVal state As Integer, ByVal filterFactor As Integer) As Long

Description

This function sets the FIR filter configuration for SNR triple track test.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- state

  Description — Specifies the FIR filter state.

  Direction — IN

  Preset value — VI_FALSE

  Value

Name	Description
VI_TRUE	FIR filter is enabled.
VI_FALSE	FIR filter is disabled.

- filterFactor

  Description — Specifies the filter factor.

  Direction — IN

  Preset value — hpe5022_SNR_TRIPLE_TRACK_FILTER_MAX

  Value

Name	Description
hpe5022_SNR_TRIPLE_TRACK_FILTER_FACTOR_MIN	1
hpe5022_SNR_TRIPLE_TRACK_FILTER_FACTOR_MAX	31

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘state’ and/or ‘filterFactor’ is out of range.

See Also

“hpe5022_snrTripleTrackAdjacentTrackConfig_Q” on page 737

“hpe5022_snrTripleTrackFilterConfig_Q” on page 752

hpe5022_snrTripleTrackFilterConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackFilterConfig_Q(ViSession id, ViPBoolean state, ViPInt16 filterFactor);

Visual Basic Syntax

hpe5022_snrTripleTrackFilterConfig_Q(ByVal id As Long, ByVal state As Integer, ByRef filterFactor As Integer) As Long

Description

This function queries the FIR filter configuration for the SNR triple track test.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- state

  Description Returns the fitter state.

  Direction OUT

  Value

Name	Description
VI_TRUE	FIR filter is enabled.
VI_FALSE	FIR filter is disabled.

- filterFactor

  Description Returns the filter factor.

  Direction OUT

  Value

Name	Description
hpe5022_SNR_TRIPLE_TRACK_FILTER_FACTOR_MIN	1
hpe5022_SNR_TRIPLE_TRACK_FILTER_FACTOR_MAX	31

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also “hpe5022_snrTripleTrackFilterConfig” on page 750

hpe5022_snrTripleTrackSectorNormalizeConfig

C Syntax

ViStatus hpe5022_snrTripleTrackSectorNormalizeConfig(ViSession id, ViBoolean normOn);

Visual Basic Syntax

hpe5022_snrTripleTrackSectorNormalizeConfig(ByVal id As Long, ByVal normOn As Integer) As Long

Description

This function sets the sector normalization mode of SNR triple track test. When sector normalization mode is enabled, the following sequence is executed after the main sequence of SNR triple track test (track profile of *"Data," "SquashedData," "CockpitNoise," "SideNoise,"* and *"SideNoiseOD"*) and each track profile data are normalized.

Sequence:

1. Erase at track center.
2. Read noise level of each sector at maximum TAA position using the track profile data of *"Data"* pattern which is already acquired in main sequence. The obtained data is called *"noise [ ]"*.
3. Write at track center.
4. Read TAA level of each sector at maximum TAA position using the track profile data of *"Data"* pattern which is already aquired in main sequence. The obtained data is called *"taa [ ]."*
5. Calculate average of noise as follows:

   $avgNoise = \frac{\sum noise}{5}$

   (4: number of sectors)
6. Calculate average of TAA as follows:

   $avgTaa = \frac{\sum taa}{5}$

   (4: number of sectors)
7. Calculate coefficient of normalization for each sector as follows:

   $coef[\ ] = \frac{avgTaa - avgNoise}{taa[\ ] - avgNoise}$
8. Normalize track profile data of each sector as follows:

   $trkProf[\ ] = coef[\ ] \times (trkProf[\ ] - avgNoise) + avgNoise$

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- normOn

Description Specifies the sector normalization mode.

Direction IN

Preset value VI_FALSE

Value

Name	Description
VI_TRUE	Sector normalization is enabled.
VI_FALSE	Sector normalization is disabled.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackSectorNormalizeConfig_Q” on page 756

hpe5022_snrTripleTrackSectorNormalizeConfig_Q

C Syntax

ViStatus hpe5022_snrTripleTrackSectorNormalizeConfig_Q(ViSession id, ViPBoolean normOn);

Visual Basic Syntax

hpe5022_snrTripleTrackSectorNormalizeConfig_Q(ByVal id As Long, ByRef normOn As Integer) As Long

Description

This function returns the sector normalization mode of SNR triple track test.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- normOn

  Description: Returns the sector normalization mode.

  Direction: OUT

  Value

Name	Description
VI_TRUE	Sector normalization is enabled.
VI_FALSE	Sector normalization is disabled.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_snrTripleTrackSectorNormalizeConfig” on page 754

hpe5022_measureSnrTripleTrack

C Syntax

ViStatus hpe5022_measureSnrTripleTrack(ViSession id, ViInt16 seqType, ViInt16 measFunc, ViReal64 offsStart, ViReal64 offsStop, ViInt16 points);

Visual Basic Syntax

hpe5022_measureSnrTripleTrack(ByVal id As Long, ByVal seqType As Integer, ByVal measFunc As Integer, ByVal offsStart As Double, ByVal offsStop As Double, ByVal points As Integer) As Long

Description

This function performs the SNR triple track measurement.

The detailed sequences of this function is as follows:

Note: In this description, data name with [ ] shows that the data type is an array.

1. Track Profile Measurement
   a. When the "seqType" is set to "hpe5022_SEQ_ER_WR_M," perform band erase to eliminate noise from the media. Band erase parameters are specified by "hpe5022_snrTripleTrackEraseBandConfig" on page 739.
   b. Write data and adjacent data pattern at four different sectors of the track as follows:

Sector #	Profile Name	Write Pattern
0	*"Data[ ]"*	Data pattern at track center
1	*"Squashed Data[ ]"*	Data pattern at track center
		Adjacent data pattern at + squeezePos
		Adjacent data pattern at – squeezePos
2	*"Side Noise ID[ ]"*	Data pattern at + otrcWrPos
		Adjacent data pattern at track center
3	*"Cockpit Noise[ ]"*	Data pattern at + otrcWrPos
		Data pattern at – otrcWrPos
		Adjacent data pattern at track center
		Adjacent data pattern at + squeezePos
		Adjacent data pattern at – squeezePos
4	*"Side Noise OD[ ]"*	Data pattern at – otrcWrPos
		Adjacent pattern at track center

NOTE

Data pattern is specified by "hpe5022_selectPattern" on page 119.

Adjacent data pattern is specified by "hpe5022_snrTripleTrackAdjacentTrackConfig" on page 735.
squeezedPos and otrcWrPos are specified by "hpe5022_snrTripleTrackPositionConfig" on page 732.

Squeeze position is described as a distance in meter from the test track. OTRC write position is described as a ratio to the track pitch.

c. Measure the track profile of the TAA for each sector (seqType: hpe5022_SEQ_M, mode: hpe5022_TP_FORWARD).
Profile data for sector 0, 1, 2, 3, 4 are named *"Data," "Squashed Data," "SideNoseID," "Cockpit Noise,"* and *"Side NoiseOD"* respectively.

2. Sector Normalization

   If sector normalization mode (specified by "hpe5022_snrTripleTrackSectorNormalizeConfig" on page 754) is enabled, execute the sequence for sector normalization and all profile data are normalized. Refer to "hpe5022_snrTripleTrackSectorNormalizeConfig" on page 754 for details of the sector normalization.

3. *"Noise Floor"* Calculation

   Calculate noise floor level using *"Data[ ]"* and *"Squashed Data[ ]."* See NOTE on page 762 for the noise floor calculation algorithm. The obtained noise floor level is called *"Noise Floor."*

4. *"OTRC Signal[ ]"* and *"Normalized OTRC Signal[ ]"* Calculation

   a. Remove *"Noise Floor"* from *"Data[ ]"* using the following equation:

   $$Data1[\,] = \sqrt{Data[\,]^2 - NoiseFloor^2}$$

   b. Perform smoothing for *"Data1[ ]."* In this calculation, filtering parameter (specified by "hpe5022_snrTripleTrackFilterConfig" on page 750) is used. The derived trace is named *"Data2[ ]."*

   c. *"OTRC Signal[ ]"* is derived from the following equation:

   $$OTRC\ Signal[\,] = \sqrt{Data2[\,]^2 + NoiseFloor^2}$$

   d. Normalize *"OTRC Signal[ ]"* as the maximum and minimum values of *"OTRC Signal[ ]"* are converted to 1 and 0 respectively. The obtained data is called *"Normalized OTRC Signal[ ]."*

5. *"Squeezed Signal[ ]"* and *"Normalized Squeezed Signal[ ]"* Calculation

   a. Remove *"Noise Floor"* from *"Squashed Data[ ]"* using the following equation:

   $$Squashed\ Data1[\,] = \sqrt{SquashedData[\,]^2 - NoiseFloor^2}$$

   b. Perform smoothing for *"Squashed Data1[ ]."* In this calculation, filtering parameter (specified by "hpe5022_snrTripleTrackFilterConfig" on page 750) is used. The result profile is called *"Squashed Data2[ ]."*

   c. *"Squeezed Signal[ ]"* is derived from the following equation:

   $$Squeezed\ Signal[\,] = \sqrt{SquashedData2[\,]^2 + NoiseFloor^2}$$

d. Normalize *"Squeezed Signal[ ]"* as the maximum and minimum values of *"OTRC Signal[ ]"* are converted to 1 and 0 respectively. The obtained data is called *"Normalized Squeezed Signal[ ]."*

6. *"OTRC Noise[ ]"* and *"Normalized OTRC Noise[ ]"* Calculation

   a. Remove *"Noise Floor"* from *"Side NoiseID[ ]"* using the following equation:

   $$\text{Side Noise ID1[ ]} = \sqrt{SideNoiseID[\ ]^2 - NoiseFloor^2}$$

   b. Perform smoothing for *"Side Noise ID1[ ]."* In this calculation, filtering parameter (specified by "hpe5022_snrTripleTrackFilterConfig" on page 750) is used. The result is called *"Side Noise ID2[ ]."*

   c. Remove *"Noise Floor"* from *"Side Noise OD[ ]"* using the following equation:

   $$\text{Side Noise OD1[ ]} = \sqrt{SideNoiseOD[\ ]^2 - NoiseFloor^2}$$

   d. Perform smoothing for *"Side Noise OD1[ ]."* In this calculation, filtering parameter (specified by "hpe5022_snrTripleTrackFilterConfig" on page 750) is used. The result is called *"Side Noise OD2[ ]."*

   e. Compose *"SideNoiseID2"* and *"Side Noise OD2"* using the following equation (the composite data is called *"Side Noise2[ ]"*):

   If *Side Noise ID2[] > Side Noise OD2[]*

   Side Noise 2[ ] = Side Noise ID2[ ]

   Else,

   Side Noise2[ ] = Side Noise OD2[ ]

   f. *"OTRC Noise[ ]"* is derived from the following equation:

   $$\text{OTRC Noise[ ]} = \sqrt{SideNoise2[\ ]^2 + NoiseFloor^2}$$

   g. Normalize *"OTRC Noise[ ]"* as the maximum and minimum values of *"OTRC Signal[ ]"* are converted to 1 and 0 respectively. The obtained data are called *"Normalized OTRC Noise[ ]."*

7. *"Squeezed Noise[ ]"* and *"Normalized Squeezed Noise[ ]"* Calculation

   a. Obtain *"Squeezed Noise[ ]"* by combining two *"OTRC Signal[ ]"* profiles plotted at ± squeezePos.

   b. Remove *"Noise Floor"* from *"Cockpit Noise[ ]"* using the following equation:

   $$\text{Cockpit Noise1[ ]} = \sqrt{CockpitNoise[\ ]^2 - NoiseFloor^2}$$

   c. Perform smoothing for *"Cockpit Noise1[ ]."* In this calculation, filtering parameter (specified by "hpe5022_snrTripleTrackFilterConfig" on page 750) is used. The result is called *"Cockpit Noise2[ ]."*

d. *"Squeezed Noise [ ]"* is derived from the following equation:

$$Squeezed\ Noise\ [\ ] = \sqrt{SqueezedNoise[\ ]^2 + CockpitNoise2[\ ]^2}$$

e. Normalize *"Squeezed Noise[ ]"* as the maximum and minimum values of *"OTRC Signal[ ]"* are converted to 1 and 0 respectively. The obtained data is called *"Normalized Squeezed Noise[ ]."*

8. *"OTRC SNR[ ],"* *"Squeezed SNR[ ],"* *"OTRC (Unsqueezed OTRC),"* and *"Squeezed OTRC"* Calculation

a. Calculate *"OTRC SNR[ ]"* profile from *"OTRC Signal[ ]"* and *"OTRC Noise[ ]."*

If *OTRCSignal'[ ] > OTRC Noise' [ ]*
and *OTRC Noise'[ ] + Ng* ≠ 0
and *Ng + Ni* ≠ 0, then

$$OTRC\ SNR[\ ] = 10 \times \log \frac{OTRCSignal'[\ ]^2 + addNoise \times SNg^2}{(OTRCNoise'[\ ] \times Ni)^2 + (NNg \times Ng)^2}$$

Else,

OTRC SNR[ ] = 0

where,

$$OTRCSignal'[\ ] = \sqrt{OTRCSignal[\ ]^2 - SNg^2}$$

$$OTRCNoise'[\ ] = \sqrt{OTRCNoise[\ ]^2 - NNg^2}$$

Ng: Gaussian noise constant (user-defined)
Ni: Interference constant (user-defined)
SNg: Gaussian noise of *"OTRC Signal[ ]"* profile
NNg: Gaussian noise of *"OTRC Noise[ ]"* profile
addNoise: 1 or 0 (specified by "hpe5022_snrTripleTrackNoiseConstantConfig" on page 744)

See NOTE on page 762 for the Gaussian noise calculation algorithm.

Noise constants and noise threshold parameters are specified by "hpe5022_snrTripleTrackNoiseConstantConfig" on page 744 and "hpe5022_snrTripleTrackNoiseThresholdConfig" on page 748 respectively.

b. Calculate *"Squeezed SNR[ ]"* profile from *"Squeezed Signal[ ]"* and *"Squeezed Noise[ ]."*

If *Squeezed Signal'[ ] > Squeezed Noise' [ ]*
and *Squeezed Noise'[ ] + Ng* ≠ 0
and *Ng + Ni* ≠ 0, then

$$Squeezed\ SNR[\ ] = 10 \times \log \frac{SqueezedSignal'[\ ]^2 + addNoise \times SNg^2}{(SqueezedNoise'[\ ] \times Ni)^2 + (NNg \times Ng)^2}$$

Else,

Squeezed SNR[ ] = 0

where,

$$Squeezed\ Signal'[\ ] = \sqrt{SqueezedSignal[\ ]^2 - SNg^2}$$

$$Squeezed\ Noise'[\ ] = \sqrt{SqueezedNoise[\ ]^2 - NNg^2}$$

where,
Ng: Gaussian noise constant (user-defined)
Ni: Interference constant (user-defined)
SNg: Gaussian noise of *"Squeezed Signal[ ]"* profile
NNg: Gaussian noise of *"Squeezed Noise[ ]"* profile
addNoise: 1 or 0 (user-defined)
Noise constants and noise threshold parameters are specified by "hpe5022_snrTripleTrackNoiseConstantConfig" on page 744 and "hpe5022_snrTripleTrackNoiseThresholdConfig" on page 748 respectively.

See NOTE on page 762 for the Gaussian noise calculation algorithm.

c. Calculate *"OTRC (Unsqueezed OTRC)"* value from *"OTRC SNR[ ]"* profile. SNR threshold parameter used in this calculation is specified by "hpe5022_snrTripleTrackSnrThresholdConfig" on page 742.

d. Calculate *"Squeezed OTRC"* value from *"Squeezed SNR[ ]"* profile. SNR threshold parameter used in this calculation is specified by "hpe5022_snrTripleTrackSnrThresholdConfig" on page 742.

9. *"WW," "RW," "RW Offset,"* and *"Squash"* Calculation

a. For the *"Normalized OTRC Signal[ ],"* specify the interpolation range which includes the threshold and window size settings.

b. Calculate the distance between the regression lines at 50% threshold and denote the measured distance as the *"Write Track Width (WW)."*

c. Calculate the distance between the regression lines at 0% threshold (i.e., floor level). Subtract the write track width from this result and denote the new width as the *"Read Track Width (RW)."*

d. Calculate the distance between the center of the track (i.e, at zero offset) and the center of the write track width. And denote the calculated distance as *"Read/Write Offset (RW offset)."*

e. Calculate *"Squash"* using the following equation:

$$Squash = \frac{TAAsquash}{TAAmax}$$

where,
TAAmax: *"OTRC Signal[ ]"* value at maximum TAA position
TAAsquash: *"Squeezed Signal[ ]"* value at *"TAAmax"* position

The "hpe5022_snrTripleTrack_Q" on page 769 and "hpe5022_snrTripleTrackProfile_Q" on page 772 functions return the measurement result.

NOTE

Gaussian Noise Level Calculation Method

If a track profile data array and noise threshold ratio are described as *"profileData[ ]"* and *"noiseThr"* respectively, the Gaussian noise level (*"gaussianNoiseLevel"*) is calculated as follows:

1. Obtain maximum value (*"max"*) and minimum value (*"min"*) of *"profileData[ ]."*
2. Calculate noise threshold level (*"noiseThrLevel"*) using the following equation:

   $noiseThrLevel = (max - min) \times noiseThr + min$

3. Calculate Gaussian noise level using the following equation:

   $gaussianNoiseLevel = \frac{sum}{count}$

   where,
   count: number of elements whose values are below *"noiseThrLevel"*
   sum: sum of element's values below *"noiseThrLevel."*

Noise Floor Calculation Method

Noise Floor is calculated as follows:

1. Calculate noise level (*"noiseLevelD"*) from *"Data[ ]"* and noise threshold ratio (*"noiseThr"*) in accordance with "Gaussian Noise Level Calculation Method."
2. Calculate noise level (*"noiseLevelS"*) from *"SquashedData[ ]"* and noise threshold ratio (*"noiseThr"*) in accordance with "Gaussian Noise Level Calculation Method."
3. Calculate noise floor (*"NoiseFloor"*) using the following equation:

   $NoiseFloor = \frac{noiseLevelD + noiseLevelS}{2}$

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

Description Specifies the type of measurement sequence.

Direction IN

Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Perform band erase → Perform SNR triple track test
hpe5022_SEQ_WR_M	1	Perform SNR triple track test only

- measFunc

  Description Specifies the type of measurement.

  Direction IN

  Values

Name	Value	Description
hpe5022_MEAS_TAA	1	Measure TAA

- offsStart

  Description Specifies the start offset position for track profile measurement of the SNR triple track test.

  Direction IN

  Unit Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- offsStop

  Description Specifies the stop offset position for track profile measurement of the SNR triple track test.

  Direction IN

  Unit Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}

Name	Value
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

- points

  Description Specifies the number of measurement points for track profile measurement.

  Direction IN

  Values

Name	Value
hpe5022_TRACK_PROFILE_SIZE_MIN	1
hpe5022_TRACK_PROFILE_SIZE_MAX	201

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_AUTO_RANGE_FAIL	The autoranging in the track profile measurement is failed. Check if the paramter setting and head is correct.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘OffsStart’, ‘OffsStop’, or ‘points’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q” function. If it is not 0, the ‘Offsstart’ or ‘Offsstop’ range is narrowed.

See Also

"hpe5022_snrTripleTrackWindowConfig" on page 728

"hpe5022_snrTripleTrackPositionConfig" on page 732

"hpe5022_snrTripleTrackAdjacentTrackConfig" on page 735

"hpe5022_snrTripleTrack_Q" on page 769

"hpe5022_snrTripleTrackProfile_Q" on page 772

"hpe5022_measureTrackProfile" on page 615

"hpe5022_selectPattern" on page 119

"hpe5022_driveState" on page 209

hpe5022_setupSnrTripleTrack

C Syntax

ViStatus hpe5022_setupSnrTripleTrack(ViSession id, ViInt16 seqType, ViInt16 measFunc, ViReal64 start, ViReal64 stop, ViInt16 points, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupTripleTrack(ByVal id As Long, ByVal seqType As Integer, ByVal measFunc As Integer, ByVal start As Double, ByVal stop As Double, ByVal points As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the SNR triple track measurement sequence to the specified test identifier. Refer to the “hpe5022_measureTripleTrack” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_tripleTrack_Q” and “hpe5022_tripleTrackProfile_Q” functions return the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Band Erase → Write → Measure Triple Track Test
hpe5022_SEQ_WR_M	1	Measure SNR triple track test only

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measureTripleTrack” function.

- start

  Description — Specifies the start offset position for track profile measurement of the Triple Track Test.

  Direction — IN

Unit Meter

Values Same as the ‘offsStart’ in the “hpe5022_measureTripleTrack” function.

- stop

  Description Specifies the stop offset position for track profile measurement of the Triple Track Test.

  Direction IN

  Unit Meter

  Values Same as the ‘offsStop’ in the “hpe5022_measureTripleTrack” function.

- points

  Description Specifies the number of measurement points for track profile measurement.

  Direction IN

  Values Same as the ‘points’ in the “hpe5022_measureTripleTrack” function.

- testHndl

  Description Returns the test identifier. This identifier is used to execute the SNR triple track measurement by the “hpe5022_measure” function.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘Offstart’, ‘Offstop’, and/or ‘points’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureSnrTripleTrack” on page 757

“hpe5022_snrTripleTrack_Q” on page 769

“hpe5022_snrTripleTrackProfile_Q” on page 772

“hpe5022_snrTripleTrackWindowConfig” on page 728

“hpe5022_snrTripleTrackPositionConfig” on page 732

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_snrTripleTrack_Q

C Syntax

ViStatus hpe5022_snrTripleTrack_Q(ViSession id, ViInt16 dataType, ViInt16 rwOffsPos, ViPReal64 ww, ViPReal64 rw, ViPReal64 rwOfst, ViPReal64 maxTaa, ViPReal64 squash, ViPReal64 otrc, ViPReal64 sqOtrc);

Visual Basic Syntax

hpe5022_snrTripleTrack_Q Lib(ByVal id As Long, ByVal dataType As Integer, ByVal rwOffsPos As Integer, ByRef ww As Double, ByRef rw As Double, ByRef rwOffset As Double, ByRef maxTaa As Double, ByRef squash As Double, ByRef otrc As Double, ByRef sqOtrc As Double) As Long

Description

This function returns the result of the SNR triple track measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the data type to be querried.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA

- rwOffsPos

  Description: Specifies the calculation method of the read write offset. This is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function.

  Direction: IN

  Values

Name	Value	Description
hpe5022_RWOF_POS_50PERC_CENT	0	Center Point at 50% level
hpe5022_RWOF_POS_PROF_MAX	1	Maximum point of track profile
hpe5022_RWOF_POS_CROS_INT_LINE	2	Least Sq. Approx. Linear Line Cross Point

- ww

Description	Returns the write track width. The definition of the write track width is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function. The floor level is set at 0% in that function.
Direction	OUT
Unit	Meter

- rw

Description	Returns the read track width. The definition of the write track width is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function. The floor level is set at 0% in that function.
Direction	OUT
Unit	Meter

- rwOfst

Description	Returns the read-write offset according to the setting of ‘rwOffsPos’. The definition of the read/write offset is the same as the “hpe5022_calculateRWOffsetAndWidth_Q” function. The floor level is set at 0% in that function.
Direction	OUT
Unit	Meter

- maxTaa

Description	Returns the maximum TAA value of the track profile result.
Direction	OUT
Unit	Volt

- squash

Description	Returns the squash ratio.
Direction	OUT

- otrc

Description	Returns the unsqueezed OTRC (Off-Track Read Capability).
Direction	OUT
Unit	Meter

- sqOtr

Description	Returns the squeezed OTRC.
Direction	OUT
Unit	Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The triple track test data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSnrTripleTrack” on page 757

“hpe5022_calculateReadWriteOffset_Q” on page 628

hpe5022_snrTripleTrackProfile_Q

C Syntax

ViStatus hpe5022_snrTripleTrackProfile_Q(ViSession id, ViInt16 dataType, ViInt16 profileType, ViReal64 profile[], ViPReal64 interceptL, ViPReal64 slopeL, ViPReal64 interceptR, ViPReal64 slopeR);

Visual Basic Syntax

hpe5022_snrTripleTrackProfile_Q(ByVal id As Long, ByVal dataType As Integer, ByVal profileType As Integer, ByRef profile As Double, ByRef interceptL As Double, ByRef slopeL As Double, ByRef interceptR As Double, ByRef slopeR As Double) As Long

Description

This function returns the result of the SNR track profile data and the regression line data.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- dataType

  Description Specifies the type of data to get the results.

  Direction IN

  Values

Name	Value	Description
hpe5022_DATA_TAA	1	TAA

- profileType

  Description Specifies type of SNR track profile.

  Direction IN

  Values

Name	Value	Description
hpe5022_SNR_TRIPLE_TRACK_OTRC_SIGNAL	1	OTRC Signal Profile
hpe5022_SNR_TRIPLE_TRACK_SQUEEZE_SIGNAL	2	Squeezed Signal Profile
hpe5022_SNR_TRIPLE_TRACK_OTRC_NOISE	3	OTRC Noise Profile
hpe5022_SNR_TRIPLE_TRACK_SQUEEZE_NOISE	4	Squeezed Noise Profile
hpe5022_SNR_TRIPLE_TRACK_OTRC_SNR	5	OTRC SNR Profile
hpe5022_SNR_TRIPLE_TRACK_SQUEEZE_SNR	6	Squeezed SNR Profile

Name	Value	Description
hpe5022_SNR_TRIPLE_TRACK_OTRC_SIGNAL_NORM	7	Normalized OTRC Signal Profile
hpe5022_SNR_TRIPLE_TRACK_SQUEEZE_SIGNAL_NORM	8	Normalized Squeezed Signal Profile
hpe5022_SNR_TRIPLE_TRACK_OTRC_NOISE_NORM	9	Normalized OTRC Noise Profile
hpe5022_SNR_TRIPLE_TRACK_SQUEEZE_NOISE_NORM	10	Normalized Squeezed Noise Profile

- profile

  Description — Returns the data (set in array) of the parameter specified by the 'dataType' and ‘profileType’. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureSnrTripleTrack” function or “hpe5022_setupSnrTripleTrack” function.

  Direction — OUT

- interceptL

  Description — Returns the intercept of the left regression line when profileType is hpe5022_SNR_TRIPLE_TRACK_OTRC_SIGNAL_NORM, hpe5022_SNR_TRIPLE_TRACK_OTRC_SNR, or hpe5022_SNR_TRIPLE_TRACK_SQUEEZE_SNR. Returns 0 when profileType is another value.

  Direction — OUT

  Unit — Meter

- slopeL

  Description — Returns the slope of the left regression line when profileType is hpe5022_SNR_TRIPLE_TRACK_OTRC_SIGNAL_NORM. If profileType is another value, this parameter is invalid (always 0).

  Direction — OUT

- interceptR

  Description — Returns the intercept of the right regression line when profileType is hpe5022_SNR_TRIPLE_TRACK_SIGNAL_NORM, hpe5022_SNR_TRIPLE_TRACK_OTRC_SNR, or hpe5022_SNR_TRIPLE_TRACK_SQUEEZE_SNR. Returns 0 when profileType is another value.

Direction OUT

Unit Meter

- slopeR

  Description Returns the slope of the right regression line when profileType is hpe5022_SNR_TRIPLE_TRACK_OTRC_SIGNAL_NORM. Returns 0 when profileType is another value.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The triple track test data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSnrTripleTrack” on page 757

“hpe5022_setupSnrTripleTrack” on page 766

Sub-Harmonic Noise Ratio (SHNR)

This section describes the functions related with sub-harmonic noise ratio measurement.

hpe5022_shnrConfig

C Syntax

ViStatus hpe5022_shnrConfig(ViSession id, ViReal64 freq,ViReal64 wrTime, ViReal64 delay, ViReal 64 rdTime);

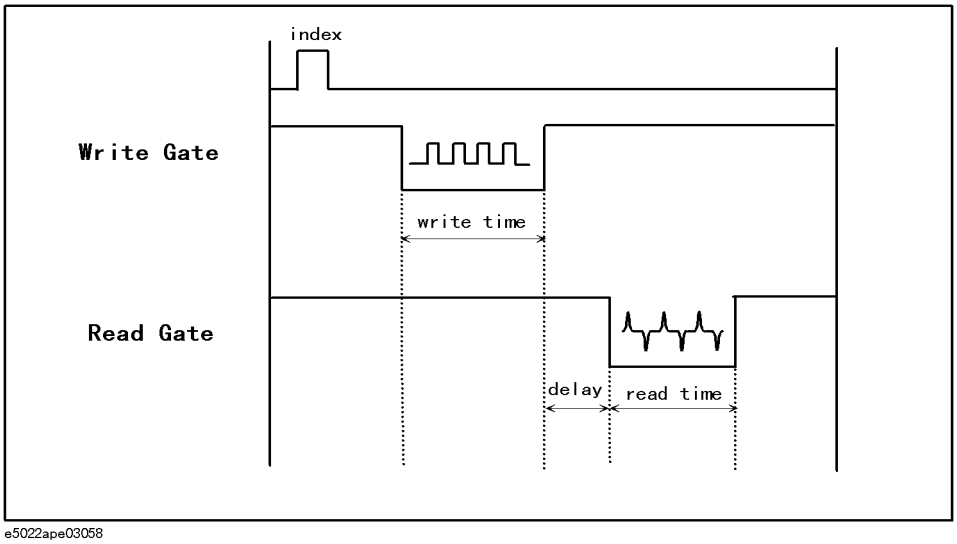
Visual Basic Syntax

hpe5022_shnrConfig(ByVal id As Long, ByVal freq As Double, ByVal wrTime As Double, ByVal delay As Double, By Val rdTime As Double) As Long

Description

This function controls the configuration of SHNR measurement using the spectrum analyzer. Write excitation is performed for the time specified by “wrTime”. Then after the delay time has elapsed the read gate opens and the input signal is read for noise detection at a specified frequency by the spectrum analyzer. See the illustration below.

Figure 3-29 Gate Diagram



e5022ape03058

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- freq

  Description — Specifies the frequency at which the read signal is measured by the spectrum analyzer.

Direction IN

Unit Hz

Preset Value 25×10^{6}

Values The upper and lower limit values are queryable by “hpe5022_narrowBandTaaFrequencyRange_Q” function.

- wrTime

  Description Specifies the time to execute write excitation.

  Direction IN

  Unit Second

  Preset Value hpe5022_SHNR_WRIT_TIME_MIN(1×10^{-3})

  Values

Name	Value
hpe5022_SHNR_WRIT_TIME_MIN	1×10^{-3}
hpe5022_SHNR_WRIT_TIME_MAX	10×10^{-3}

- delay

  Description Specifies the delay time. Delay time is defined as the time distance between write excitation and read operation.

  Direction IN

  Unit Second

  Preset Value hpe5022_SHNR_DEL_MIN

  Values

Name	Value
hpe5022_SHNR_DEL_MIN	0
hpe5022_SHNR_DEL_MAX	10×10^{-3}

- rdTime

  Description Specifies the read time.

  Direction IN

  Unit Second

  Preset Value hpe5022_SHNR_READ_TIME_MIN

Values

Name	Value
hpe5022_SHNR_READ_TIME_MIN	800×10^{-6}
hpe5022_SHNR_READ_TIME_MAX	10×10^{-3}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’freq’, ‘wrTime’, ‘delay’ and/or ‘rdTime’ is out of range.

See Also

“hpe5022_shnrConfig_Q” on page 778

“hpe5022_setupShnr” on page 793

“hpe5022_measureShnr” on page 788

hpe5022_shnrConfig_Q

C Syntax

ViStatus hpe5022_shnrConfig_Q(ViSession id, ViPReal64 freq,ViPReal64 wrTime, ViPReal64 delay, ViPReal 64 rdTime);

Visual Basic Syntax

hpe5022_shnrConfig_Q(ByVal id As Long, ByRef freq As Double, ByRef wrTime As Double, ByRef delay As Double, ByRef rdTime As Double) As Long

Description

This function returns the specified configuration of SHNR measurement.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- freq

  Description — Returns the measurement frequency of the spectrum analyzer.

  Direction — OUT

- wrTime

  Description — Returns the time to execute write excitation.

  Direction — OUT

- delay

  Description — Returns the delay time. Delay time is defined as the time distance between write excitation and read operation.

  Direction — OUT

- rdTime

  Description — Returns the read time.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_shnrConfig” on page 775

hpe5022_shnrWritePatternConfig

C Syntax

ViStatus hpe5022_shnrWritePatternConfig(ViSession id, ViInt16 pat);

Visual Basic Syntax

hpe5022_shnrWritePatternConfig(ByVal id As Long, ByVal pat As Integer) As Long

Description

This function controls the data pattern of the write excitation for SHNR measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pat

  Description: Specifies the data pattern of the write excitation. When this parameter is set to hpe5022_PAT_DEFAULT, write excitation data pattern is the same as the pattern specified by “hpe5022_selectPattern” function.

  Direction: IN

  Preset value: hpe5022_PAT_DEFAULT

  Values

Name	Value	Description
hpe5022_PAT_DEFAULT	-1	Default pattern
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_REP_2	8	Repetition Pattern2
hpe5022_PAT_REP_3	9	Repetition Pattern3
hpe5022_PAT_REP_4	10	Repetition Pattern4
hpe5022_PAT_USER	20	User-defined Pattern

Name	Value	Description
hpe5022_PAT_USER_2	21	User-defined Pattern2
hpe5022_PAT_USER_3	22	User-defined Pattern3
hpe5022_PAT_USER_4	23	User-defined Pattern4
hpe5022_PAT_ERASE	101	Default Erase Pattern selected by the “hpe5022_eraseType” function
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter “pat” is out of range.

See Also

“hpe5022_shnrWritePatternConfig_Q” on page 781

hpe5022_shnrWritePatternConfig_Q

C Syntax

ViStatus hpe5022_shnrWritePatternConfig_Q(ViSession id, ViPInt16 pat);

Visual Basic Syntax

hpe5022_shnrWritePatternConfig_Q(ByVal id As Long, ByRef pat As Integer) As Long

Description

This function returns the specified data pattern of the write excitation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pat

  Description: Returns the specified data pattern of the write excitation.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_shnrWritePatternConfig” on page 779

hpe5022_shnrWriteCurrentConfig

C Syntax

ViStatus hpe5022_shnrWriteCurrentConfig(ViSession id, ViReal64 current);

Visual Basic Syntax

hpe5022_shnrWriteCurrentConfig(ByVal id As Long, ByVal current As Double) As Long

Description

This function controls the write excitation current for SHNR measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Specifies the write excitation current for SHNR measurement. The limit of write excitation current is dependent on the installed head amplifier. The maximum and minimum values that the write current can be set to are queriable by the " hpe5022_writeCurrentRange_Q" function.

  Direction: IN

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter "current" is out of range.

See Also

"hpe5022_measureShnr" on page 788

"hpe5022_setupShnr" on page 793

"hpe5022_shnrWriteModeConfig_Q" on page 787

hpe5022_shnrWriteCurrentConfig_Q

C Syntax

ViStatus hpe5022_shnrWriteCurrentConfig_Q(ViSession id, ViPReal64 current);

Visual Basic Syntax

hpe5022_shnrWriteCurrentConfig_Q(ByVal id As Long, ByRef current As Double) As Long

Description

This function returns specified write excitation current for SHNR measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- current

  Description: Returns the specified write excitation current for SHNR measurement.

  Direction: OUT

  Unit: Ampere

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_shnrWriteCurrentConfig” on page 782

hpe5022_shnrWriteModeConfig

C Syntax

ViStatus hpe5022_shnrWriteModeConfig(ViSession id, ViInt16 writeMode);

Visual Basic Syntax

hpe5022_shnrWriteModeConfig(ByVal id As Long, ByVal writeMode As Integer) As Long

Description

This function controls the write current direction at the beginning of write operation.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- writeMode

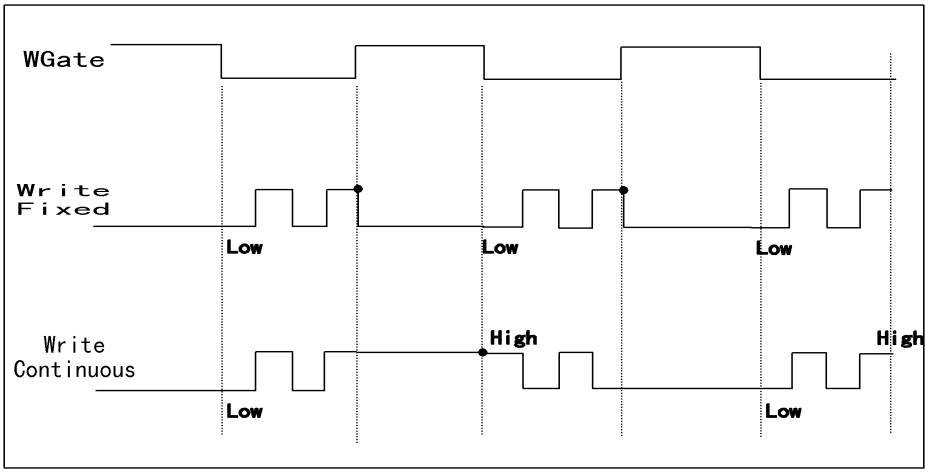
  Description Specifies the direction of the write excitation current.

  Direction IN

  Values

Name	Value	Description
hpe5022_WRIT_DATA_POL_FIXED	0	The write excitation current direction at the beginning of each write operation is always the same.
hpe5022_WRIT_DATA_POL_CONTINUOUS	1	The write excitation current direction at the beginning of write operation is the same as the last current direction of the previous write operation.

Figure 3-30 Write Mode



When [Write Mode] is set to “fixed”, the write excitation current direction at the beginning of each write operation is always the same. On the other hand, when [Write Mode] is set to “continuous” the write current direction is the same as the last current direction of the previous write operation. Each time the write excitation current alternates it allows you to check the polarity of the output pulse at the end of the previous write. You must set the repetitive period of the data pattern T to 1T, 2T, 4T and 8T due to the system’s internal constraints and it is also dependent on the installed head amplifier. If you use other data patterns, E5022A/B system can not guarantee the change of direction of the write excitation current. This function makes it possible for SHNR measurement to achieve an acceptable level of the measured signal under similar usage of the drive.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writeMode’ is out of range.
hpe5022_ERROR_NSUP_CONF	The installed head amplifier has no capability to control the write current direction.

See Also

“hpe5022_measureShnr” on page 788

“hpe5022_setupShnr” on page 793

“hpe5022_shnrWriteModeConfig_Q” on page 787

hpe5022_shnrWriteModeConfig_Q

C Syntax

ViStatus hpe5022_shnrWriteModeConfig_Q(ViSession id, ViPInt16 writeMode);

Visual Basic Syntax

hpe5022_shnrWriteModeConfig_Q(ByVal id As Long, ByRef writeMode As Integer) As Long

Description

This function returns the configuration specified by “hpe5022_shnrWriteModeConfig” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- writeMode

  Description — Returns the specified direction of the write excitation current.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_shnrWriteModeConfig” on page 784

hpe5022_measureShnr

C Syntax

ViStatus hpe5022_measureShnr(ViSession id, ViInt16 ib_points, const ViReal64 ib[], ViInt16 offtrk_pos_points, const ViReal64 offtrk_pos[], ViInt16 average);

Visual Basic Syntax

hpe5022_measureShnr(ByVal id As Long, ByVal ib_points As Integer, ByRef ib As Double, ByVal offtrk_pos_points As Integer, ByRef offtrk_pos As Double, ByVal average As Integer) As Long

Description

This function measures SHNR (sub-harmonic noise ratio). When this function is executed the following sequence of events take place.

1. Perform the three-track-erase. (the entire track erase for 5 positions, the track, its both side tracks, and the center between them.)
2. Move the head to the read offset.
3. For sense stimulus = ib[0] to ib[ib_points-1].
4. Measure the signal and denote the result as Noise_Ref[i]. This noise reference is measured at a frequency specified by “hpe5022_shnrConfig” function.
5. Next sense stimulus go to **step 3**.
6. Write data at the track center specified by “hpe5022_write” function.
7. For read offset = ‘offtrk_pos[0]’ + ‘readOffset’ to offtrk_pos[offtrk_pos_points-1] + readOffset.
8. For sense stimulus = ib[0] to [ib_points-1].
9. Let Sum TAA = 0
10. For average = 1 to n
11. Perform write excitation during wrTime. (‘wrTime’ is specified by the “hpe5022_shnrConfig” function.)
12. Wait for delay time to elapse. (‘delay’ is specified by the “hpe5022_shnrConfig” function.)
13. Measure the Narrow Band TAA (i.e, TAA) during rdTime. (‘rdTime’ is specified by the “hpe5022_shnrConfig” function). Narrow Band TAA is measured at a frequency specified by “hpe5022_shnrConfig” function similar to step 4.
14. Get the summation of the measured TAA (i.e, Sum_TAA= Sum TAA + TAA).
15. Next average go to **step 10.**
16. Calculate the average TAA = Sum_TAA/ n
17. Next sense stimulus go to **step 8**.
18. Next read offset go to **step 7**.
19. For ib_index = 0 to (ib_points-1)

20. Calculate the SHNR data as;

    SHNR_DATA(ib_index) = 20log (Max (Aver_TAA[ib_index]) / Noise_Ref[ib_index]). The result is returned by “hpe5022_shnrData_Q” function.

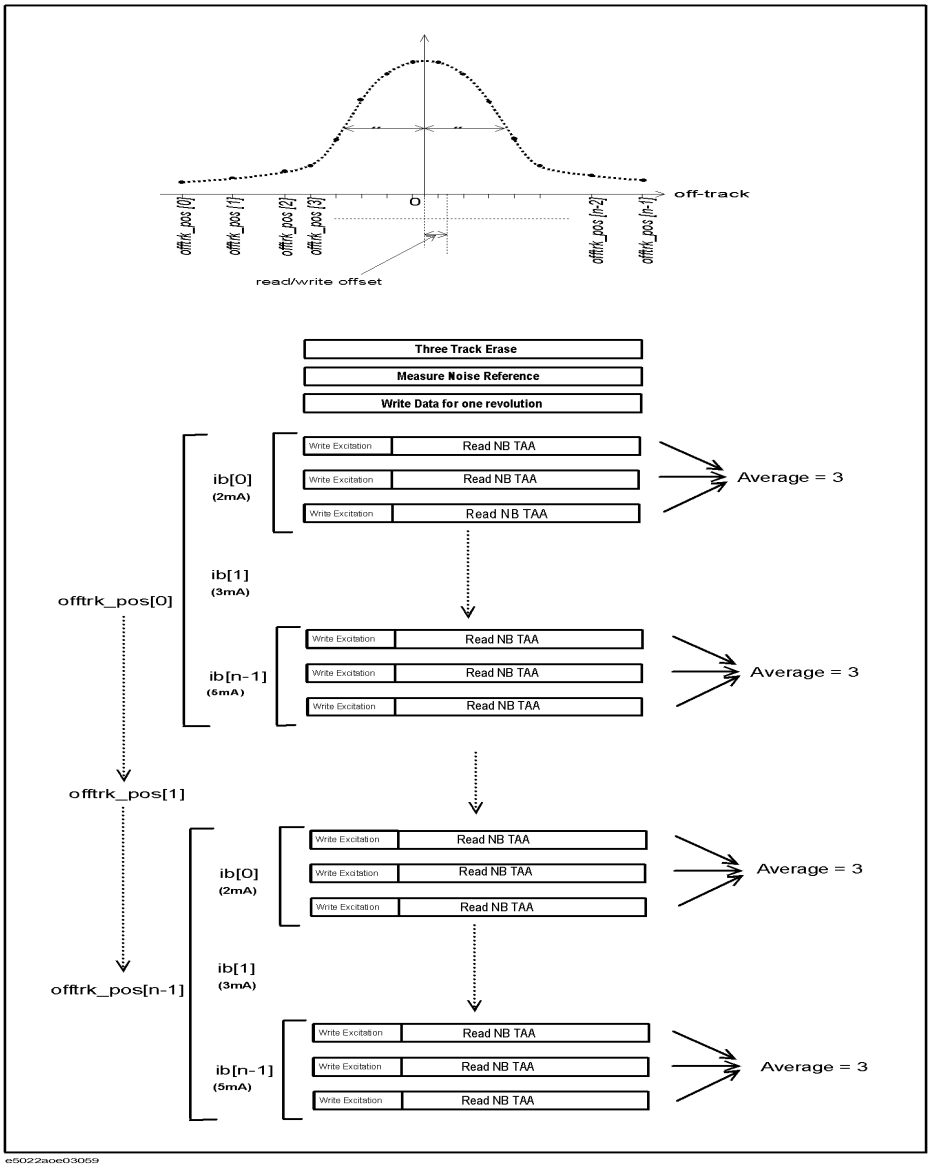
Equation 3-3 Calculation of SHNR

$$SHNRdata = 20\log\left(\frac{MaxAverTAA}{NoiseRef}\right)$$

21. Next sense stimulus go to **step 19**.

22. Denote SHNR as the maximum SHNR from the data array (i.e, SHNR = Max

(SHNR_DATA[]). The result is returned by “hpe5022_shnr_Q” function.



Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ib_points

  Description — Specifies the number of sense stimulus points.

Direction IN

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- ib

  Description Specifies the sense stimulus list as the data array. The sense stimulus will be set to the data in array order. The “hpe5022_senseStimulusMode” function selects a sense stimulus type. The range of value is returned by the “hpe5022_senseStimulusRange_Q” function.

  Direction IN

  Unit Depends on the mode of “hpe5022_senseStimulusMode”

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

- offtrk_pos_points

  Description Specifies the number of off-track positions.

  Direction IN

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- offtrk_pos

  Description Specifies the off-track position list.

  Direction IN

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

  Unit Meter

- average

  Description Specifies the number of measurement points to be averaged.

Direction IN

Name	Value
hpe5022_SHNR_AVERAGE_MIN	1
hpe5022_SHNR_AVERAGE_MAX	50

NOTE The “ib_points”, “offtrk_pos_points” and “average’” parameters must be set within the limits of “hpe5022_SHNR_COUN_MIN ≤ ib_points × offtrk_pos_points × average ≤ hpe5022_SHNR_COUN_MAX

Name	Value
hpe5022_SHNR_COUN_MIN	1
hpe5022_SHNR_COUN_MAX	500

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ib_points’, ‘ib’, ‘offtrk_pos_points’, ‘offtrk_pos’, and/or ‘average’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The drive status is “OFF”.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.

See Also

“hpe5022_shnrConfig” on page 775

“hpe5022_shnrWritePatternConfig” on page 779

“hpe5022_shnrWriteCurrentConfig” on page 782

“hpe5022_shnrWriteModeConfig” on page 784

“hpe5022_shnr_Q” on page 796

“hpe5022_senseStimulusMode” on page 343

hpe5022_setupShnr

C Syntax

ViStatus hpe5022_setupShnr(ViSession id, ViInt16 ib_points, const ViReal64 ib[], ViInt16 offtrk_pos_points, const ViReal64 offtrk_pos[], ViInt16 average, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupShnr(ByVal id As Long, ByVal ib_points As Integer, ByRef ib As Double, ByVal offtrk_pos_points As Integer, ByRef offtrk_pos As Double, ByVal average As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the shnr measurement sequence to the test identifier. Refer to the “hpe5022_measureShnr” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ib_points

  Description — Specifies the number of sense stimulus points. This number must be the same as the array size of the sense stimulus list “ib”.

  Direction — IN

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- ib

  Description — Specifies the sense stimulus list as the data array. The sense stimulus will be set to the data in array order. The “hpe5022_senseStimulusMode” function selects a sense stimulus type. The range of value is returned by the “hpe5022_senseStimulusRange_Q” function.

  Direction — IN

  Unit — Depends on the mode of “hpe5022_senseStimulusMode”

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

- offtrk_pos_points

Description Specifies the number of off-track position points. This number must be the same as the array size of the off-track position list “offtrk_pos”.

Direction IN

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- offtrk_pos

  Description Specifies the off-track position list.

  Direction IN

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- average

  Description Specifies the number of measurement points to be averaged for each sense stimulus “ib”.

  Direction IN

Name	Value
hpe5022_SHNR_AVERAGE_MIN	1
hpe5022_SHNR_AVERAGE_MAX	50

- testHndl

  Description Returns the test identifier. This identifier is used to execute the SHNR measurement by the “hpe5022_measure” function.

  Direction OUT

NOTE The “ib_points”, “offtrk_pos_points” and “average’” parameters must be set within the limits of “hpe5022_SHNR_COUN_MIN ≤ ib_points ×

offtrk_pos_points × average ≤ hpe5022_SHNR_COUN_MAX

Name	Value
hpe5022_SHNR_COUN_MIN	1
hpe5022_SHNR_COUN_MAX	500

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ib_points’, ‘ib’, ‘offtrk_pos_points’, ‘offtrk_pos’, and/or ‘average’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Can not allocate enough memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureShnr” on page 788

“hpe5022_measure” on page 387

“hpe5022_shnr_Q” on page 796

“hpe5022_shnrData_Q” on page 798

hpe5022_shnr_Q

C Syntax

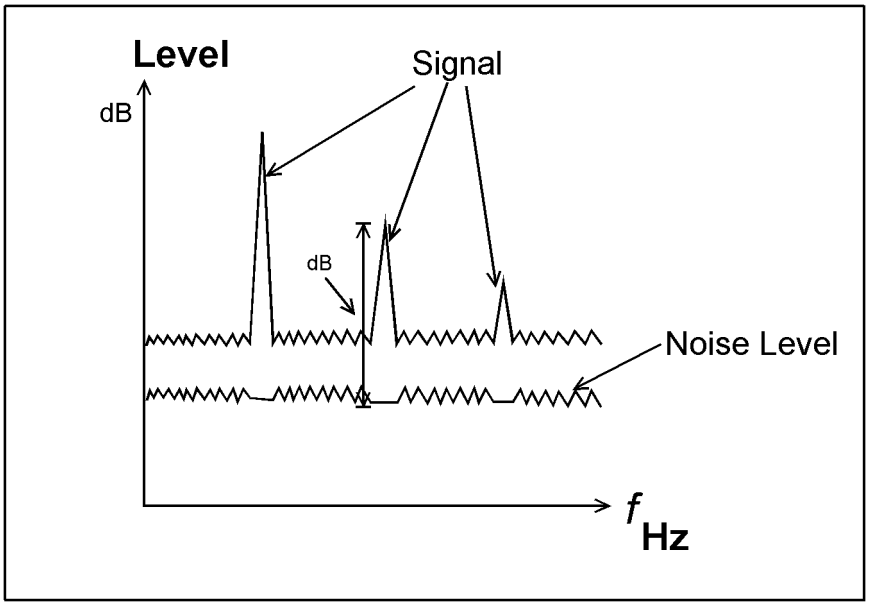
ViStatus hpe5022_shnr_Q(ViSession id, ViPReal64 harm, ViPReal64 noise, ViPReal64 shnr);

Visual Basic Syntax

hpe5022_shnr_Q(ByVal id As Long, ByRef harm As Double, ByRef noise As Double, ByRef shnr As Double) As Long

Description

This function returns the maximum calculated results of the SHNR measurement.



e5022ape03052

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- harm

  Description — Returns the harmonic level in volts. This signal is the maximum average TAA.

  Direction — OUT

- noise

  Description — Returns the noise level in volts. This signal is the measured TAA after performing three track erase (i.e, erase TAA).

  Direction — OUT

- shnr

  Description — Returns the maximum sub-harmonic noise ratio in dB. This

result is the maximum shnr calculated from the maximum average TAA (harm) and erase TAA (noise). Refer to Equation 3-3 of “hpe5022_measureShnr”.

Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The “shnr” data is corrupt.

See Also

“hpe5022_measureShnr” on page 788

hpe5022_shnrData_Q

C Syntax

ViStatus hpe5022_shnrData_Q(ViSession id, ViInt16 ib_index, ViReal64 harm[], ViPReal64 noise, ViReal64 shnr[]);

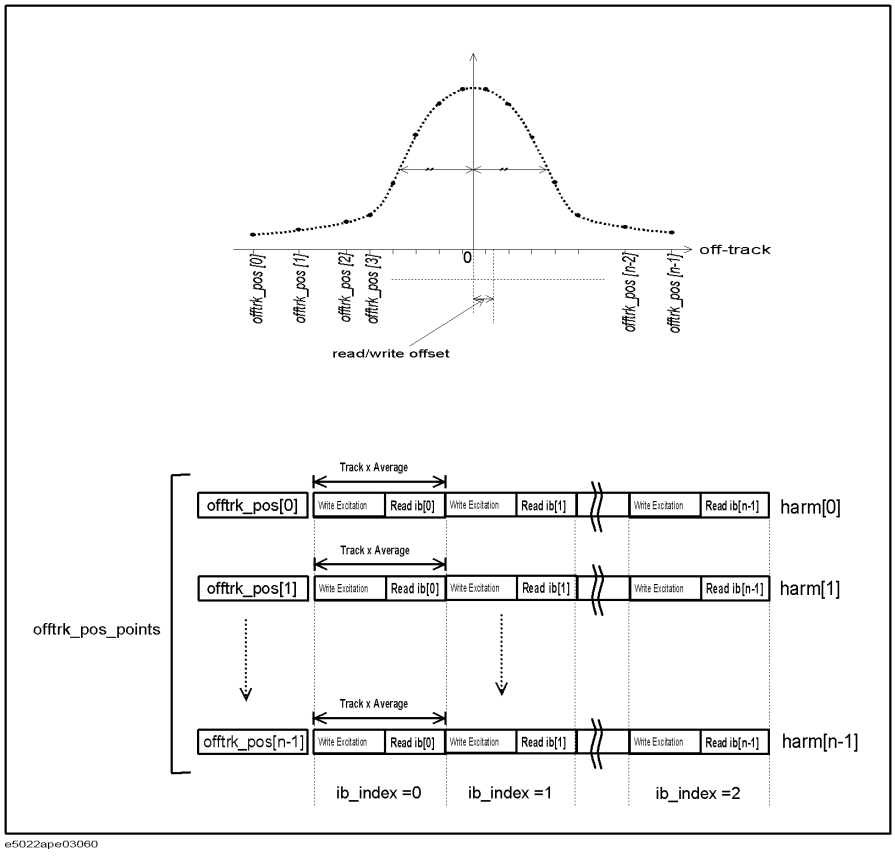
Visual Basic Syntax

hpe5022_shnrData_Q(ByVal id As Long, ByVal ib_index As Integer, ByRef harm As Double, ByRef noise As Double, ByRef shnr As Double) As Long

Description

This function returns the read data from all offtrack positions as measured by the specified sense stimulus “ib”.

Figure 3-31 SHNR Data



Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ib_index

  Description — Specifies the sense stimulus index where the read data of a specific sense stimulus “ib” is to be queried. The parameter represents the location of the SHNR raw data measured

from all “offtrk_pos_points” as defined by “hpe5022_measureShnr” function for a particular value of sense stimulus “ib”.

Direction IN

Parameter Limit
0 ≤ ib_index ≤ ib_points -1

- harm

  Description Returns the sub-harmonic level measured from each off-track position. This data is returned in array form. Refer to Figure 3-31.

  Direction OUT

  Unit Volt

- noise

  Description Returns the noise level.

  Direction OUT

  Unit Volt

- shnr

  Description Returns the calculated SHNR from each off-track position. This data is returned in array form.

  Direction OUT

  Unit dB

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter “ib_index” is out of range.
hpe5022_ERROR_DATA_CORRUPT	The “shnr” data is corrupt.

See Also

“hpe5022_measureShnr” on page 788

Baseline Popping Measurement Function

This section describes the functions related with the baseline popping measurement.

hpe5022_baselinePoppingConfig

C Syntax

ViStatushpe5022_baselinePoppingConfig(ViSession id, ViReal64 freq);

Visual Basic Syntax

hpe5022_baselinePoppingConfig(ByVal id As Long, ByVal freq As Double) As Long

Description

This function specifies the measurement frequency for baseline popping test. Frequency in this function is the sub harmonics of the data pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Specifies the measurement frequency.

  Direction: IN

  Unit: Hz

  Preset Value: 25×10^6

  Values

Name	Value
hpe5022_NB_TAA_FREQ_MIN	0
hpe5022_NB_TAA_FREQ_MAX	500×10^6

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freq’ is out of range

See Also

“hpe5022_measureBaselinePopping” on page 803

“hpe5022_setupBaselinePopping” on page 806

“hpe5022_baselinePoppingConfig_Q” on page 802

hpe5022_baselinePoppingConfig_Q

C Syntax

ViStatushpe5022_baselinePoppingConfig_Q(ViSession id, ViPReal64 freq);

Visual Basic Syntax

hpe5022_baselinePoppingConfig_Q(ByVal id As Long, ByRef freq As Double) As Long

Description

This function returns the measurement frequency for baseline popping test.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Returns the measurement frequency.

  Direction: OUT

  Unit: Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_baselinePoppingConfig” on page 800

hpe5022_measureBaselinePopping

C Syntax

ViStatus hpe5022_measureBaselinePopping(ViSession id, ViInt16 points, const ViReal64 ib[]);

Visual Basic Syntax

hpe5022_measureBaselinePopping(ByVal id As Long, ByVal points As Integer, ByRef ib As Double) As Long

Description

This function measures the baseline popping.

The detailed sequences of this function are as follows:

1. Set i to zero. (i = 0)
2. Move the head to the center of the track.
3. Erase (same as the “hpe5022_erase” function) for an entire track.
4. Set the sense stimulus value of which number is 0 at ‘ib[i]’ (see parameters).
5. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
6. Measure a narrow band TAA (same as the “hpe5022_measureNarrowBandTaa” function).
7. Move the head to the center of the track.
8. Write the data pattern specified by the “hpe5022_selectPattern” function.
9. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
10. Set the sense stimulus value at ‘ib[i]’ (see parameters).
11. Measure a narrow band TAA (same as the “hpe5022_measureNarrowBandTaa” function).
12. Increment i by 1. (i = i+1)
13. go to the step 10.

The “hpe5022_baselinePoppingData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- points

  Description — Specifies the number of measurement points for baseline popping measurement. This number must be the same as the array size of the sense stimulus value list.

  Direction — IN

Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- ib

  Description — Specifies the sense stimulus list as the data array. The sense stimulus will be set to the data in array order. The “hpe5022_senseStimulusMode” function selects a sense stimulus type. The range of value is returned by the “hpe5022_senseStimulusRange_Q” function.

  Direction — IN

  Unit — Depends on the mode of “hpe5022_senseStimulusMode”

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘points’, and/or ‘ib’ is out of range.

See Also

“hpe5022_baselinePoppingConfig” on page 800

“hpe5022_baselinePoppingData_Q” on page 808

“hpe5022_readTrackOffset” on page 219

“hpe5022_measureNarrowBandTaa” on page 572

“hpe5022_selectPattern” on page 119

“hpe5022_driveState” on page 209

“hpe5022_senseStimulusMode” on page 343

hpe5022_setupBaselinePopping

C Syntax

ViStatus hpe5022_setupBaselinePopping(ViSession id, ViInt16 points, const ViReal64 ib[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupBaselinePopping(ByVal id As Long, ByVal points As Integer, ByRef ib As Double, ByRef testHndl As Long) As Long

Description

This function assigns the baseline popping measurement sequence to the specified test identifier. Refer to the “hpe5022_measureBaselinePopping” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_baselinePoppingData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- points

  Description — Specifies the number of measurement points for baseline popping measurement. This number must be the same as the array size of the sense stimulus value list.

  Direction — IN

  Values — Same as the ‘points’ in the “hpe5022_measureBaselinePopping” function.

- ib

  Description — Specifies the sense stimulus list as the data array. The sense stimulus will be set to the data in array order. The “hpe5022_senseStimulusMode” function selects a sense stimulus type. The range of value is returned by the “hpe5022_senseStimulusRange_Q” function.

  Direction — IN

  Unit — Depends on the mode of “hpe5022_senseStimulusMode”

  hpe5022_SENSE_STIMULUS_CURRENT : Ampere

  hpe5022_SENSE_STIMULUS_VOLTAGE : Volt

  pe5022_SENSE_STIMULUS_POWER : Watt

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the baseline popping measurement by the

"hpe5022_measure" function.

Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'points', and/or 'ib' is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measureBaselinePopping" on page 803

"hpe5022_baselinePoppingData_Q" on page 808

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_baselinePoppingData_Q

C Syntax

ViStatus hpe5022_baselinePoppingData_Q(Visession id, ViReal64 blp[]);

Visual Basic Syntax

hpe5022_baselinePoppingData_Q(ByVal id As Long, ByRef blp As Double) As Long

Description

This function returns the result of the baseline popping measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- blp

  Description: Returns the baseline popping data set in array. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureBaselinePopping” function or “hpe5022_setupBaselinePopping” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The baseline popping data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureBaselinePopping” on page 803

“hpe5022_setupBaselinePopping” on page 806

hpe5022_baselinePoppingMax_Q

C Syntax

ViStatus hpe5022_baselinePoppingMax_Q(Visession id, ViPReal64 blpMax);

Visual Basic Syntax

hpe5022_baselinePoppingMax_Q(ByVal id As Long, ByRef blpMax As Double) As Long

Description

This function returns the maximum data of baseline popping measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- blpMax

  Description: Returns the maximum data of baseline popping. The maximum value in the data returned by the “hpe5022_baselinePoppingData_Q” function is returned.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The baseline popping data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureBaselinePopping” on page 803

“hpe5022_setupBaselinePopping” on page 806

“hpe5022_baselinePoppingData_Q” on page 808

NLTS (5th Harmonics) Measurement Function

This section describes the functions related with NLTS measurement. The E5022A/B can measure NLTS from the following three methods.

- 5th Harmonic Method
- Dipulse Extraction Method
- Time Correlation Method

The NLTS by 5th harmonic method is measured by a spectrum analyzer, the dipulse extraction and the time correlation methods are measured by an oscilloscope. NLTS by dipulse extraction and time correlation methods are described in “Setup Function” .

hpe5022_measureNlts5th

C Syntax

ViStatus hpe5022_measureNlts5th(ViSession id, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureNlts5th(ByVal id As Long, ByVal ave As Integer) As Long

Description

This function measures NLTS by the 5th harmonic method.

The function sequences are as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. Perform a erase (same as the hpe5022_erase function) for an entire track.
3. Write NLTS-5th-Elimination data pattern (11000000100000011000000100000) for an entire track at the write track offset position.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure the 5th-harmonic level.
6. Move the head to the write track offset position.
7. Perform a erase (same as the hpe5022_erase function) for an entire track.
8. Write a reference data pattern whose repetitive period is 15T for an entire track.
9. Move the head to the read track offset position.
10. Measure the 5th-harmonic reference level..
11. Repeat the steps 1 to 10 ‘ave’ (see parameters) times if it is more than 1.

The data patterns for this measurement are selected automatically. The data pattern specified by the “hpe5022_selectPattern” function is not used for this measurement.

The “hpe5022_nlts5th_Q” function returns the measurement result.

See chapter 5 in the Operaion Manual for the calculation formula of NLTS measurement.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_NLTS_5TH_COUN_MIN	1
hpe5022_NLTS_5TH_COUN_MAX	25

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

Error Code	Description
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module.Check if the parameter setting is correct.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module.Check if the parameter setting is correct.

See Also

“hpe5022_nlts5th_Q” on page 815

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_driveState” on page 209

hpe5022_setupNlts5th

C Syntax

ViStatus hpe5022_setupNlts5th(ViSession id, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupNlts5th(ByVal id As Long, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the NLTS by the 5th harmonic method measurement sequence to the specified test identifier. Refer to the “hpe5022_measureNlts5th” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with test identifier specified in this function.

The “hpe5022_nlts5th_Q” function returns the measurement result

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as the ‘ave’ in the “hpe5022_measureNlts5th” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the NLTS measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureNlts5th” on page 810

“hpe5022_nlts5th_Q” on page 815

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_nlts5th_Q

C Syntax

ViStatus hpe5022_nlts5th_Q(ViSession id,ViPReal64 nlts);

Visual Basic Syntax

hpe5022_nlts5th_Q(ByVal id As Long, ByRef nlts As Double) As Long

Description

This function returns the result of NLTS by the 5th harmonic method measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- nlts

  Description: Returns the results of NLTS by 5th harmonic method.

  Direction: OUT

  Unit: (Ratio of bit cell time)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The NLTS data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureNlts5th” on page 810

“hpe5022_setupNlts5th” on page 813

hpe5022_nlts5thDataSize_Q

C Syntax

ViStatus hpe5022_nlts5thDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_nlts5thDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of NLTS by the 5th harmonic method data returned by the “hpe5022_nlts5th_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the NLTS data.

  The size will be the same as the value of ‘ave’ specified by “hpe5022_measureNlts5th” or “hpe5022_setupNlts5th” functions if the measurement done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The NLTS data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureNlts5th” on page 810

hpe5022_nlts5thData_Q

C Syntax

ViStatus hpe5022_nlts5thData_Q(ViSession id,ViReal64 data[]);

Visual Basic Syntax

hpe5022_nlts5thData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the NLTS data of the measurement for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureNlts5th” or the “hpe5022_setupNlts5th” function is set for more than one, the function allows you to get the NLTS data for each revolution. If the ‘ave’ is set to 1, the output of this function is the same as one of the “hpe5022_nlts5th_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the data array of NLTS. The array size is returned by the “hpe5022_nlts5thDataSize_Q” function.

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The NLTS data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_nlts5thDataSize_Q” on page 816

hpe5022_nlts5thStatistic_Q

C Syntax

ViStatus hpe5022_nlts5thStatistic_Q(ViSession id,ViPReal64 mean, ViPReal64 min,ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_nlts5thStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistic (mean, minimum, maximum and standard deviation) of NLTS by the 5th harmonic method data. The data returned by the “hpe5022_nlts5thData_Q” function is used for this statistic analysis.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- mean

  Description — Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_nlts5th_Q” function.

  Direction — OUT

  Unit — (Ratio of bit cell time)

- min

  Description — Returns the minimum value.

  Direction — OUT

  Unit — (Ratio of bit cell time)

- max

  Description — Returns the maximum value.

  Direction — OUT

  Unit — (Ratio of bit cell time)

- stdDev

  Description — Returns the standard deviation value.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The NLTS data is corrupt. Check if your measurement sequence is correct.

See Also “hpe5022_nlts5thData_Q” on page 817

SNR Measurement

This section describes the functions related with SNR (Signal to Noise Ratio) measurement. SNR is measured by a parametric module.

hpe5022_noiseReference

C Syntax

ViStatus hpe5022_noiseReference(ViSession id, ViReal64 noiseRef);

Visual Basic Syntax

hpe5022_noiseReference(ByVal id As Long, ByVal noiseRef As Double) As Long

Description

This function sets the noise reference value in order to measure noise accurately.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- noiseRef

  Description: Specifies the noise reference value.

  Direction: IN

  Unit: Volt (rms)

  Preset Value: 100×10^{-6}

  Values

Name	Value
hpe5022_NOISE_REF_MIN	0
hpe5022_NOISE_REF_MAX	100×10^{-3}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘noiseRef’ is out of range.

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also “hpe5022_measureSnr” on page 823

hpe5022_noiseReference_Q

C Syntax

ViStatus hpe5022_noiseReference_Q(ViSession id, ViPReal64 noiseRef);

Visual Basic Syntax

hpe5022_noiseReference_Q(ByVal id As Long, ByRef noiseRef As Double) As Long

Description

This function returns the specified value of the noise reference.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- noiseRef

  Description: Returns the noise reference value.

  Direction: OUT

  Unit: Volt (rms)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_noiseReference” on page 820

hpe5022_measureSnr

C Syntax

ViStatus hpe5022_measureSnr(ViSession id, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureSnr(ByVal id As Long, ByVal ave As Integer) As Long

Description

This function measures the signal and noise level, then calculates the SNR (Signal to Noise Ratio) and crest factor. The measurement sequence of this function are as follows:

1. Erase an entire track.
2. Move the head to the read offset position as specified by “hpe5022_readTrackOffset”.
3. Set the input range to measure noise, the input range level is defined by “hpe5022_noiseReference”.
4. Measure noise.
5. Resume the input range.
6. Move the head to the write offset position as specified by “hpe5022_writeTrackOffset” function.
7. Write the selected data pattern as specified by the “hpe5022_selectPattern” function.
8. Move the head to the read offset position as specified by “hpe5022_readTrackOffset”.
9. Measure the TAA.
10. Repeat steps 1-8 as specified by average count.

The measurement result is returned by the “hpe5022_snr_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- ave

  Description: Specifies the average count for measurement.

  Direction: IN

  Values

Name	Value
hpe5022_SNR_COUN_MIN	1

Name	Value
hpe5022_SNR_COUN_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_setupSnr” on page 825

“hpe5022_snr_Q” on page 827

hpe5022_setupSnr

C Syntax

ViStatus hpe5022_setupSnr(ViSession id, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSnr(ByVal id As Long, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function sets the SNR measurement sequence.

This function assigns the SNR measurement sequence to the specified test identifier. Refer to the “hpe5022_measureSnr” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier given by this function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- ave

  Description: Specifies the average count for measurement.

  Direction: IN

  Values

Name	Value
hpe5022_SNR_COUN_MIN	1
hpe5022_SNR_COUN_MAX	50

- testHndl

  Description: Returns the test identifier. This identifier is used to execute SNR measurement by the “hpe5022_measure”function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_release_Setup” function.

See Also

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

“hpe5022_measureSnr” on page 823

hpe5022_snr_Q

C Syntax

ViStatus hpe5022_snr_Q(ViSession id, ViPReal64 signal, ViPReal64 noise, ViPReal64 snr, ViPReal64 crest);

Visual Basic Syntax

hpe5022_snr_Q(ByVal id As Long, ByRef signal As Double, ByRef noise As Double, ByRef snr As Double, ByRef crest As Double) As Long

Description

This function returns the results of the measured SNR and crest factor. The returned parameters are signal, noise and crest factor. Crest factor is the ratio of the maximum voltage value of the noise to the peak to peak value of the TAA.

See chapter 5 in the Operaion Manual for the definition of this parameter.

Each result is the mean value of the corresponding parameter of the “hpe5022_snrData_Q” and “hpe5022_crestFactorData_Q” functions.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- signal

  Description: Returns the measured signal level.

  Direction: OUT

  Unit: Volt

- noise

  Description: Returns the measured noise level.

  Direction: OUT

  Unit: Volt (rms)

- snr

  Description: Returns the measured SNR level.

  Direction: OUT

  Unit: dB

- crest

  Description: Returns the measured crest factor.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The SNR data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_snrData_Q” on page 830

“hpe5022_measureSnr” on page 823

“hpe5022_setupSnr” on page 825

“hpe5022_crestFactorData_Q” on page 834

hpe5022_snrDataSize_Q

C Syntax

ViStatus hpe5022_snrDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_snrDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the size of the SNR data as returned by the “hpe5022_snrData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the SNR data.

  The size is the same as the value of ‘ave’ specified by the “hpe5022_measureSnr” or “hpe5022_setupSnr” function if measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The SNR data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_setupSnr” on page 825

“hpe5022_snrData_Q” on page 830

“hpe5022_measureSnr”

hpe5022_snrData_Q

C Syntax

ViStatus hpe5022_snrData_Q(ViSession id, ViReal64 data[]);

Visual Basic Syntax

hpe5022_snrData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the measurement data of the SNR for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureSnr” or the “hpe5022_setupSnr” function is set for more than one, this function will allow you to get the SNR data for each revolution. (If the ‘ave‘ is set to 1, the output of this function will be the same as the “hpe5022_snr_Q” function.)

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the SNR data. The data size is returned by the “hpe5022_snrDataSize_Q” function.

  Direction: OUT

  Unit: dB

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The SNR data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_measureSnr” on page 823

“hpe5022_setupSnr” on page 825

“hpe5022_snrDataSize_Q” on page 829

hpe5022_snrStatistic_Q

C Syntax

ViStatus hpe5022_snrStatistic_Q(ViSession id, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_snrStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the SNR measurement. The data returned by the “hpe5022_snrData_Q” function are used for the statistic analysis of this function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mean

  Description: Returns the mean value.

  Direction: OUT

- min

  Description: Returns the minimum value.

  Direction: OUT

- max

  Description: Returns the maximum value.

  Direction: OUT

- stdDev

  Description: Returns the standard deviation value.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The SNR data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_measureSnr” on page 823

“hpe5022_setupSnr” on page 825

“hpe5022_snrData_Q” on page 830

hpe5022_crestFactorDataSize_Q

C Syntax

ViStatus hpe5022_crestFactorDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_crestFactorDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of crest factor data as returned by the “hpe5022_crestFactorData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the crest factor data.

  The size will be the same as the value of ‘ave’ specified by the “hpe5022_measureSnr” or “hpe5022_setupSnr” function if measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The crest factor data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_measureSnr” on page 823

“hpe5022_setupSnr” on page 825

“hpe5022_crestFactorData_Q” on page 834

hpe5022_crestFactorData_Q

C Syntax

ViStatus hpe5022_crestFactorData_Q(ViSession id, ViReal64 data[]);

Visual Basic Syntax

hpe5022_crestFactorData_QBy(Val id As Long, ByRef data As Double) As Long

Description

This function returns the crest factor data of the measurement at each revolution.

When ‘ave’ parameter in the “hpe5022_measureSnr” or the “hpe5022_setupSnr” function is set for more than one, the function will allow you to get the crest factor data for each revolution. (If the ‘ave‘is set to 1, then the output of this function becomes the same as the “hpe5022_snr_Q” function.)

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the crest factor data. The data size is returned by the “hpe5022_crestFactorDataSize_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The crest factor data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_measureSnr” on page 823

“hpe5022_setupSnr” on page 825

“hpe5022_snr_Q” on page 827

“hpe5022_crestFactorDataSize_Q” on page 833

hpe5022_crestFactorStatistic_Q

C Syntax

ViStatus hpe5022_crestFactorStatistic_Q(ViSession id, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_crestFactorStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the SNR (crest factor) measurement. The data returned by the “hpe5022_crestFactorData_Q” function are used for this statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mean

  Description: Returns the mean value of the measurement result. The returned value is the same as the corresponding values returned by the “hpe5022_snr_Q” function.

  Direction: OUT

- min

  Description: Returns the minimum value of measurement result.

  Direction: OUT

- max

  Description: Returns the maximum value of the measurement result.

  Direction: OUT

- stdDev

  Description: Returns the standard deviation value calculated from the measurement result.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The crest factor data is corrupt. Check if the measurement sequence is correct.

See Also

“hpe5022_measureSnr” on page 823

“hpe5022_setupSnr” on page 825

“hpe5022_snr_Q” on page 827

“hpe5022_crestFactorData_Q” on page 834

Spectral SNR Measurement

This section describes the functions related with spectral SNR measurement. The spectral SNR measurement is measured by a spectrum analyzer.

hpe5022_spectralSnrFrequency

C Syntax

ViStatus hpe5022_spectralSnrFrequency(ViSession id, ViReal64 startFreq, ViReal64 stopFreq);

VB Syntax

hpe5022_spectralSnrFrequency(ByVal id As Long, ByVal startFreq As Double, ByVal stopFreq As Double) As Long

Description

This function specifies the starting and stopping frequencies of a spectrum analyzer for spectral SNR measurement.

Parameters

- id

Description	Specifies the system identifier. This is given by the "hpe5022_init" function.
Direction	IN

- startFreq

Description	Specifies the starting frequency.
Direction	IN
Unit	Hz
Preset Value	Set to the minimum value of the frequency range.
Values	The upper and lower limit values are queryable by hpe5022_spectrumFrequencyRange_Q function.

- stopFreq

Description	Specifies the stopping frequency.
Direction	IN
Unit	Hz
Preset Value	Set to the maximum value of the frequency range.
Values	The upper and lower limit values are queryable by hpe5022_spectrumFrequencyRange_Q function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘startFreq’ or/and ‘stopFreq’ is out of range.

See Also

“hpe5022_spectralSnrFrequency_Q” on page 839

hpe5022_spectralSnrFrequency_Q

C Syntax

ViStatus hpe5022_spectralSnrFrequency_Q(ViSession id, ViPReal64 startFreq, ViPReal64 stopFreq);

VB Syntax

hpe5022_spectralSnrFrequency_Q(ByVal id As Long, ByRef startFreq As Double, ByRef stopFreq As Double) As Long

Description

This function returns the starting and stopping frequencies of a spectrum analyzer for the spectral SNR measurement.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- startFreq

  Description — Returns the starting frequency.

  Direction — OUT

  Unit — Hz

- stopFreq

  Description — Returns the stopping frequency.

  Direction — OUT

  Unit — Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectralSnrFrequency” on page 837

hpe5022_spectralSnrBandWidth

C Syntax

ViStatus hpe5022_spectralSnrBandWidth(ViSession id, ViReal64 rbw, ViReal64 vbw);

VB Syntax

hpe5022_spectralSnrBandWidth(ByVal id As Long, ByVal rbw As Double, ByVal vbw As Double) As Long

Description

This function specifies the resolution band width and video band width of the spectrum analyzer for the spectral SNR measurement. Resolution band width allows you to improve the resolution of frequency, i.e, selectivity of signal, and also to reduce the display noise. On the other hand, video band width reduces the noise variations and makes the signal clearly visible.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rbw

  Description: Specifies the resolution band width. Narrowing this, will widen the dynamic range. Widening this, makes measurement speed faster. This can be set to 3×10^3, 10×10^3, 30×10^3, 100×10^3, 300×10^3, 1×10^6, 3×10^6 and 5×10^6. The maximum rbw is 1×10^6 when SA module is 4395A or E5040A, and 5×10^6 when 3GHz Spectrum Analyzer is used. When the value other than these values is set, the closest and lower value is set automatically.

  Direction: IN

  Unit: Hz

  Preset Value: 10×10^3

  Values

Name	Value
hpe5022_SPEC_BW_MIN	3×10^3
hpe5022_SPEC_BW_MAX	5×10^6

- vbw

  Description: Specifies the video band width. When 4395A or 3GHz Spectrum analyzer is used, this value is clipped to a largest (and not greater than it) value within the range of ‘rbw’/300 - ‘rbw’ and among 10, 30, 100, 300, 1×10^3, 3×10^3, $10e\times10^3$,

30×10^3, 100×10^3, 300×10^3, 1×10^6, and 3×10^6. The maximum vbw is 1×10^6 when SA module is 4395A, and 3×10^6 when 3GHz Spectrum analyzer is used,respectively. In case of E5040A, this value is ignored and set automatically at the same value as the rbw.

Narrowing this, will reduce the noise variation. Widening this, will speed up measurement.The value of 1/1, 1/3, 1/30, 1/100 or 1/300 of the resolution band width can be set as a video band width. When other value is entered, the value is set automatically to the closest and lower value of the setting.

Direction IN

Unit Hz

Preset Value 10×10^3

Values

Name	Value
hpe5022_SPEC_VBW_MIN	10
hpe5022_SPEC_VBW_MAX	3×10^6

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘rbw’ and/or ‘vbw’ is out of range.

See Also

“hpe5022_spectralSnrBandWidth_Q” on page 842

hpe5022_spectralSnrBandWidth_Q

C Syntax

ViStatus hpe5022_SpectralSnrBandWidth_Q(ViSession id, ViPReal64 rbw, ViPReal64 vbw);

VB Syntax

hpe5022_SpectralSnrBandWidth_Q(ByVal id As Long, ByRef rbw As Double, ByRef vbw As Double) As Long

Description

This function returns the resolution band width and the video band width of the spectrum analyzer for the spectral SNR measurement.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- rbw

  Description — Returns the resolution band width.

  Direction — OUT

  Unit — Hz

- vbw

  Description — Returns the video band width.

  Direction — OUT

  Unit — Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectralSnrBandWidth” on page 840

hpe5022_measureSpectralSnr

C Syntax

ViStatus hpe5022_measureSpectralSnr(ViSession id,ViInt16 seqType);

Visual Basic Syntax

hpe5022_measureSpectralSnr(ByVal id As Long, ByVal seqType As Integer) As Long

Description

This function measures the spectral SNR according to the specified sequence type. There are three sequence types: 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The sequence is as follows:

(For 4395A and E5040A Spectrum Analyzer.)

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If seqType is set to “hpe5022_SEQ_ER_WR_M”, erase an entire track.
3. If seqType is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write data pattern specified by the “hpe5022_selectPattern” function.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure noise level. (Same as follows)
    a. “hpe5022_measureSpectrum” (seqType: hpe5022_SEQ_M, detMode: hpe5022_SPEC_DET_SAMP)
    b. The data returned from “hpe5022_spectrumNoise_Q” becomes the noise level.
6. Measure the peak signal level.(Same as follows)
    a. “hpe5022_measureSpectrum” (seqType: hpe5022_SEQ_M, detMode: “hpe5022_SPEC_DET_POS_PEAK”.
    b. The maximum data returned from “hpe5022_spectrumDataAll_Q” becomes the peak signal level.

(For 3GHz Spectrum Analyzer.)

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If seqType is set to “hpe5022_SEQ_ER_WR_M”, the full-erase is performed.
3. If seqType is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write data on the nearly full track with the data pattern specified by the “hpe5022_selectPattern” function.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure the peak signal level. (Same as follows)
    a. “hpe5022_measureSpectrum” (seqType: hpe5022_SEQ_M, detMode:

hpe5022_SPEC_DET_POST_PEAK) VBW is automatically set to 30kHz if the specified RBW is 100kHz or 300kHz and VBW > 30kHz, and set to 1MHz if the specified RBW is 1MHz and VBW > 300kHz, respectively.)

b. The amount of the amplitude levels at fundamental and harmonic frequencies become the peak signal level.

6. Measure noise level. (Same as follows)

a. “hpe5022_measureSpectrum” (seqType: hpe5022_SEQ_M, detMode: “hpe5022_SPEC_DET_SAMP”. RBW and VBW are set to the specified values by the “hpe5022_spectralSnrBandWidth()” function.

b. The data returned from “hpe5022_spectrumDataAll_Q” becomes the peak noise level.

The “hpe5022_spectralSnr_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measurement
hpe5022_SEQ_WR_M	1	Write->Measurement
hpe5022_SEQ_M	2	Measurement

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_spectralSnr_Q” on page 848

“hpe5022_driveState” on page 209

hpe5022_setupSpectralSnr

C Syntax

ViStatus hpe5022_setupSpectralSnr(ViSession id, ViInt16 seqType, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSpectralSnr(ByVal id As Long, ByVal seqType As Integer, ByVal testHndl As long) As Long

Description

This function assigns the spectral SNR measurement sequence to the specified test identifier. See the “hpe5022_measureSpectralSnr” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function

The “hpe5022_spectralSnr_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measureSpectralSnr” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the spectrum measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ is out of range.

Error Code	Description
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureSpectralSnr” on page 843

“hpe5022_spectralSnr_Q” on page 848

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_spectralSnr_Q

C Syntax

ViStatus hpe5022_spectralSnr_Q(ViSession id, ViPReal64 signal, ViPReal64 noise, ViPReal64 snr);

Visual Basic Syntax

hpe5022_spectralSnr_Q(ByVal id As Long, ByRef signal As Double, ByRef noise As Double, ByRef snr As Double) As Long

Description

This function returns the spectral SNR measurement results. The signal level, the noise level, and the SNR are returned.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- signal

  Description: Returns the measured signal level.

  Direction: OUT

  Unit: rms Volt

- noise

  Description: Returns the measured noise level.

  Direction: OUT

  Unit: rms Volt

- snr

  Description: Returns the measured SNR.

  Direction: OUT

  Unit: dB

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The spectrum data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectralSnr” on page 843

“hpe5022_setupSpectralSnr” on page 846

Spectrum Measurement

This section describes the functions related with spectrum measurement. The spectrum measurement measure a power level using a spectrum analyzer.

hpe5022_spectrumFrequency

C Syntax

ViStatus hpe5022_spectrumFrequency(ViSession id, ViReal64 startFreq, ViReal64 stopFreq);

VB Syntax

hpe5022_spectrumFrequency(ByVal id As Long, ByVal startFreq As Double, ByVal stopFreq As Double) As Long

Description

This function controls the starting and stopping frequencies of a spectrum analyzer for spectrum measurement.

Parameters

- id

Description	Specifies the system identifier. This is given by the "hpe5022_init" function.
Direction	IN

- startFreq

Description	Specifies the starting frequency.
Direction	IN
Unit	Hz
Preset Value	Set to the minimum value of the frequency range.
Values	The upper and lower limit values are queryable by hpe5022_spectrumFrequencyRange_Q function.

- stopFreq

Description	Specifies the stopping frequency.
Direction	IN
Unit	Hz
Preset Value	Set to the maximum value of the frequency range.
Values	The upper and lower limit values are queryable by hpe5022_spectrumFrequencyRange_Q function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘startFreq’ or/and ‘stopFreq’ is out of range.

See Also

“hpe5022_spectrumFrequency_Q” on page 852

hpe5022_spectrumFrequency_Q

C Syntax

ViStatus hpe5022_spectrumFrequency_Q(ViSession id, ViPReal64 startFreq, ViPReal64 stopFreq);

VB Syntax

hpe5022_spectrumFrequency_Q(ByVal id As Long, ByRef startFreq As Double, ByRef stopFreq As Double) As Long

Description

This function returns the starting and stopping frequencies of a spectrum analyzer for spectrum measurement.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- startFreq

  Description — Returns the starting frequency.

  Direction — OUT

  Unit — Hz

- stopFreq

  Description — Returns the stopping frequency.

  Direction — OUT

  Unit — Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectrumFrequency” on page 850

hpe5022_spectrumFrequencyRange_Q

C Syntax

ViStatus _ VI_FUNC hpe5022_spectrumFrequencyRange_Q (ViSession id, ViPReal64 freqMin, ViPReal64 freqMax);

VB Syntax

Dcleare Function _hpe5022_spectrumFrequencyRange_Q Lib “hpe5022_32.dll” (_ ByVal id As Long, _ ByRef freqMin As Doule, _ ByRef freqMax As Double _) As Long

Description

This function queries the frequency range of a spectrum analyzer for spectrum measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freqMin — Returns the minimum frequency in Hertz.
- freqMax — Returns the maximum frequency in Hertz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

hpe5022_spectrumBandWidth

C Syntax

ViStatus hpe5022_spectrumBandWidth(ViSession id, ViReal64 rbw, ViReal64 vbw);

VB Syntax

hpe5022_spectrumBandWidth(ByVal id As Long, ByVal rbw As Double, ByVal vbw As Double) As Long

Description

This function specifies the resolution band width and the video band width of the spectrum analyzer for the spectrum measurement. Resolution band width allows you to improve resolution of frequency, in other words, selectivity of signal, and also to reduce the display noise. On the other hand, video band width reduces the noise variations and makes the signal clearly visible.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- rbw

  Description — Specifies the resolution band width. Narrowing this, will widen the dynamic range. Widening this, will make a measurement speed fast. This can be set to 3×10^3, 10×10^3, 30×10^3, 100×10^3, 300×10^3, 1×10^6, 3×10^6 and 5×10^6. The maximum rbw is 1×10^6 when SA module is 4395A or E5040A, and 5×10^6 when 3GHz Spectrum Analyzer is used. When rbw is set other than these values, then rbw is set to the closest and lower value from the above selection.

  Direction — IN

  Unit — Hz

  Preset Value — 10×10^3

  Values

Name	Value
hpe5022_SPEC_BW_MIN	3×10^3
hpe5022_SPEC_BW_MAX	5×10^6

- vbw

  Description — Specifies the video band width. When 4395A or 3GHz Spectrum analyzer is used, this value is clipped to a largest (and not greater than it) value within the range of ‘rbw’/300 - ‘rbw’ and among 10, 30, 100, 300, 1×10^3, 3×10^3, 10×10^3,

30×10^3, 100×10^3, 300×10^3, 1×10^6, and 3×10^6.

The maximum vbw is 1e6 when SA module is 4395A, and 3×10^6 when 3GHz Spectrum analyzer is used, respectively. In case of E5040A, this value is ignored and set automatically at the same value as the rbw.

Narrowing this, will reduce noise variation. Widening this, will speed up the measurement. The values 1/1, 1/3, 1/30, 1/100 or 1/300 of the resolution band width can be set as video band width. When other values are used, the value is set automatically at nearest and lower value of vbw.

Direction IN

Unit Hz

Preset Value 10×10^3

Values

Name	Value
hpe5022_SPEC_VBW_MIN	10
hpe5022_SPEC_VBW_MAX	3×10^6

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘rbw’ and/or ‘vbw’ is out of range.

See Also

“hpe5022_spectrumBandWidth_Q” on page 856

hpe5022_spectrumBandWidth_Q

C Syntax

ViStatus hpe5022_spectrumBandWidth_Q(ViSession id, ViPReal64 rbw, ViPReal64 vbw);

VB Syntax

hpe5022_spectrumBandWidth_Q(ByVal id As Long, ByRef rbw As Double, ByRef vbw As Double) As Long

Description

This function returns the resolution band width and the video band width of the spectrum analyzer for spectrum measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rbw

  Description: Returns the resolution band width.

  Direction: OUT

  Unit: Hz

- vbw

  Description: Returns the video band width.

  Direction: OUT

  Unit: Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectrumBandWidth” on page 854

hpe5022_measureSpectrum

C Syntax

ViStatus hpe5022_measureSpectrum(ViSession id,ViInt16 seqType, ViInt16 detMode);

Visual Basic Syntax

hpe5022_measureSpectrum(ByVal id As Long, ByVal seqType As Integer, ByVal detMode As Integer) As Long

Description

This function measures the power level according to the specified sequence type. There are three sequence types: 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The sequence is as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.

   (Spectrum Analyzer is 4395A or E5040A)

2a. If seqType is set to “hpe5022_SEQ_ER_WR_M”, erase an entire track.

3a. If seqType is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write data pattern specified by the “hpe5022_selectPattern” function.

   (3GHz Spectrum Analyzer)

2b. If seqType is set to “hpe5022_SEQ_ER_WR_M”, the full-erase is performed.

3b. If seqType is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write data on the nearly full track with the data pattern specified by the “hpe5022_selectPattern” function.

   (3GHz Spectrum Analyzer)

4. Move the head to read track offset position specified by the “hpe5022_readTrackOffset” function.

5. Measure the power level using a spectrum analyzer.

The “hpe5022_spectrumData_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measurement

Name	Value	Description
hpe5022_SEQ_WR_M	1	Write->Measurement
hpe5022_SEQ_M	2	Measurement

- detMode

  Description Specifies the detection mode of the spectrum analyzer. Select sample mode when the noise is measured. Select positive peak mode when the signal is measured.

  Direction IN

  Values

Name	Value	Description
hpe5022_SPEC_DET_SAMP	1	Sample Mode
hpe5022_SPEC_DET_POS_PEAK	2	Positive Peak Mode

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘detMode’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_spectrumData_Q” on page 861

“hpe5022_spectrumDataAll_Q” on page 863

“hpe5022_driveState” on page 209

hpe5022_setupSpectrum

C Syntax

ViStatus hpe5022_setupSpectrum(ViSession id, ViInt16 seqType, ViInt16 detMode, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSpectrum(ByVal id As Long, ByVal seqType As Integer, ByVal detModeAs Integer, ByVal testHndl As long) As Long

Description

This function assigns the spectrum measurement sequence to the specified test identifier. See the “hpe5022_measureSpectrum” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function

The “hpe5022_spectrumData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measureSpectrum” function.

- detMode

  Description — Specifies the detection mode of the spectrum analyzer. Select sample mode when the noise is measured. Select positive peak mode when the signal is measured.

  Direction — IN

  Values — Same as the ‘detMode’ in the “hpe5022_measureSpectrum” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the spectrum measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘detMode’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureSpectrum” on page 857

“hpe5022_spectrumData_Q” on page 861

“hpe5022_spectrumDataAll_Q” on page 863

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_spectrumData_Q

C Syntax

ViStatus hpe5022_spectrumData_Q(ViSession id, ViReal64 freq, ViPReal64 data);

Visual Basic Syntax

hpe5022_spectrumData_Q(ByVal id As Long, ByVal freq As Double, ByRef data As Double) As Long

Description

This function returns a power level for specified frequency. As the spectrum analyzer measures only at the measurement frequencies returned by the “hpe5022_spectrumDataAll_Q” function, the returned data is a interpolated value from the measurement result returned by the “hpe5022_spectrumDataAll_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Specifies the center frequency.

  Direction: IN

  Unit: Hz

  Values

Name	Value
hpe5022_SPEC_FREQ_MIN	0
hpe5022_SPEC_FREQ_MAX	500×10^{6}

- data

  Description: Returns the power level at specified frequency.

  Direction: OUT

  Unit: dBm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freq’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The spectrum data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectrum” on page 857

“hpe5022_setupSpectrum” on page 859

“hpe5022_spectrumDataAll_Q” on page 863

hpe5022_spectrumDataAll_Q

C Syntax

ViStatus hpe5022_spectrumDataAll_Q (ViSession id, ViInt16 points, ViReal64 freq[], ViReal64 data[]);

Visual Basic Syntax

hpe5022_spectrumDataAll_Q(ByVal id As Long, ByRef points As Integer, ByRef freq As Double, ByRef data As Double) As Long

Description

This function returns data of power level for all measurement points.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- points

  Description — Returns the number of measurement points.

  Direction — OUT

- freq

  Description — Returns the data array of the frequency. The array size is same as the number of the ‘point’ parameter. The frequency points determined from the start, the stop frequencies and the resolution band width.

  Direction — OUT

  Unit — Hz

- data

  Description — Returns the data array of the power level. The array size is same as the ‘point’ parameter.

  Direction — OUT

  Unit — dBm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The spectrum measurement data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectrum” on page 857

“hpe5022_setupSpectrum” on page 859

“hpe5022_spectrumData_Q” on page 861

hpe5022_spectrumNoise_Q

C Syntax

hpe5022_spectrumNoise_Q (ViSession id, ViReal64 freqLow, ViReal64 freqHigh, ViPReal64 noise);

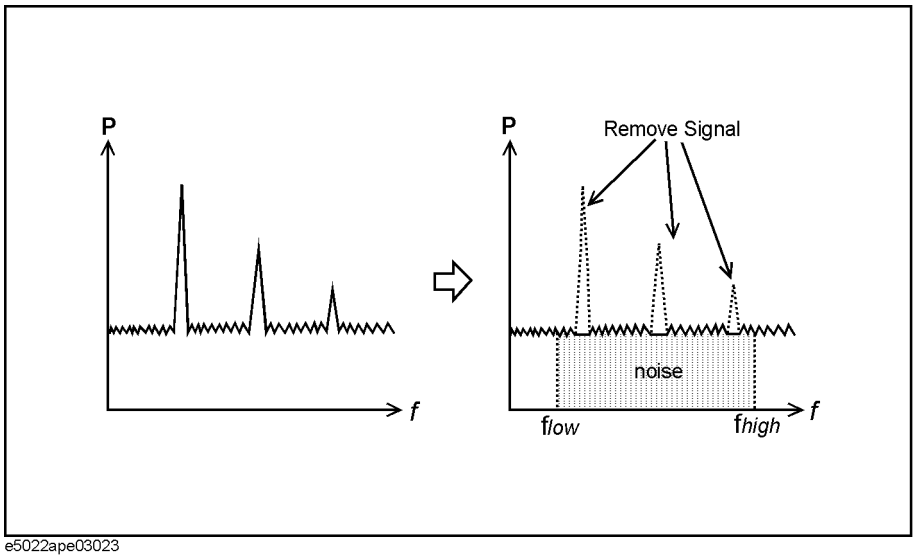
Visual Basic Syntax

hpe5022_spectrumNoise_Q(ByVal id As Long, ByVal freqLow As Double, ByVal freqHigh As Double, ByRef noise As Double) As Long

Description

This function returns the calculated noise level from spectrum data. The portion of signal peak is removed and noise level is calculated.

Figure 3-32 Noise Level Measurement



e5022ape03023

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- freqLow

  Description — Specifies the low integration range of frequency.

  Direction — IN

  Unit — Hz

  Values

Name	Value
hpe5022_SPEC_FREQ_MIN	0

Name	Value
hpe5022_SPEC_FREQ_MAX	500×10^6

- freqHigh

  Description Specifies the high integration range of frequency.

  Direction IN

  Unit Hz

  Values

Name	Value
hpe5022_SPEC_FREQ_MIN	0
hpe5022_SPEC_FREQ_MAX	500×10^6

- noise

  Description Returns the noise level.

  Direction OUT

  Unit rms Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freqLow’ or/and ‘freqHigh’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The spectrum measurement data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectrum” on page 857

“hpe5022_setupSpectrum” on page 859

“hpe5022_spectrumDataAll_Q” on page 863

Popcorn Noise Measurement Function (Sampling Method)

This section describes the functions related with popcorn noise measurement using E5038A parametric module. The popcorn noise is monitered by the voltage at each 5µsecond.

hpe5022_popcornNoiseConfig

C Syntax

ViStatushpe5022_popcornNoiseConfig(ViSession id, ViReal64 threshold, ViReal64 wrTime, ViReal64 delay, ViReal 64 rdTime);

Visual Basic Syntax

hpe5022_popcornNoiseConfig(ByVal id As Long, ByVal threshold As Double, ByVal wrTime As Double, ByVal delay As Double, By Val rdTime As Double) As Long

Description

This function controls the configuration of popcorn noise test measured by E5038A/B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- threshold

  Description: Specifies the threshold level of the popcorn noise. Popcorn noise is counted when the input signal or read-back signal exceeds this level.

  Direction: IN

  Unit: Volt

  Preset Value: 100×10^{-6}

  Values

Name	Value
hpe5022_POPCORN_THR_MIN	0
hpe5022_POPCORN_THR_MAX	100×10^{-3}

- wrTime

  Description: Specifies the time to write.

  Direction: IN

Unit Second

Preset Value hpe5022_POPCORN_WRIT_TIME_MIN(10×10^{-6})

Values

Name	Value
hpe5022_POPCORN_WRIT_TIME_MIN	10×10^{-6}
hpe5022_POPCORN_WRIT_TIME_MAX	10×10^{-3}

- delay

  Description Specifies the delay time. Delay time is defined as the time distance between write and read operation.

  Direction IN

  Unit Second

  Preset Value 0 (hpe5022_POPCORN_DEL_MIN)

  Values

Name	Value
hpe5022_POPCORN_DEL_MIN	0
hpe5022_POPCORN_DEL_MAX	10×10^{-3}

- rdTime

  Description Specifies the read time.

  Direction IN

  Unit Second

  Preset Value 0 (hpe5022_POPCORN_READ_TIME_MIN)

  Values

Name	Value
hpe5022_POPCORN_READ_TIME_MIN	1×10^{-6}
hpe5022_POPCORN_READ_TIME_MAX	10×10^{-3}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’threshold’, ‘wrTime’, ‘delay’ and/or ‘rdTime’ is out of range.

See Also “hpe5022_measurePopcornNoise” on page 879

hpe5022_popcornNoiseConfig_Q

C Syntax

ViStatushpe5022_popcornNoiseConfig_Q(ViSession id, ViPReal64 threshold, ViPReal64 wrTime, ViPReal64 delay, ViPReal64 rdTime);

Visual Basic Syntax

hpe5022_popcornNoiseConfig_Q(ByVal id As Long, ByRef threshold As Double, ByRef wrTime As Double, ByRef delay As Double, ByRef rdTime As Double) As Long

Description

This function returns the specified configuration of the popcorn noise test.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- threshold

  Description: Returns the threshold level of the popcorn noise. Popcorn noise is counted when the input signal exceeds this level.

  Direction: OUT

  Unit: Volt

- wrTime

  Description: Returns the time to write.

  Direction: OUT

  Unit: Second

- delay

  Description: Returns the delay time between write and read operation.

  Direction: OUT

  Unit: Second

- rdTime

  Description: Returns the read time.

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also “hpe5022_popcornNoiseConfig” on page 867

hpe5022_popcornNoiseWriteModeConfig

C Syntax

ViStatushpe5022_popcornNoiseConfig(ViSession id, ViInt16 writeMode);

Visual Basic Syntax

hpe5022_popcornNoiseConfig(ByVal id As Long, ByVal writeMode As Integer) As Long

Description

This function controls the write current direction at the beginning of write operation.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- writeMode

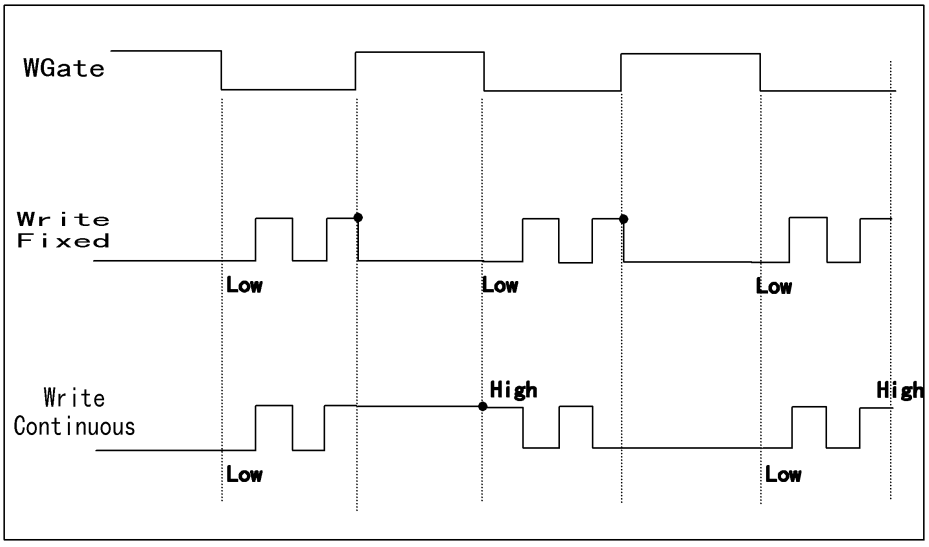
  Description Specifies the write current direction at the beginning of write operation.

  Direction IN

  Values

Name	Value	Description
hpe5022_WRIT_DATA_POL_FIXED	0	The write current direction at the beginning of each write operation is always the same.
hpe5022_WRIT_DATA_POL_CONTINUOUS	1	The write current direction at the beginning of write operation is the same as the last current direction of the previous write operation.

Figure 3-33 Write Mode



When [Write Mode] is set to “fixed”, the write current direction at the beginning of each write operation is always the same. On the other hand, when [Write Mode] is set to “continuous” the write current direction is the same as the last current direction of the previous write operation. Each time the write current alternates it allows you to check the polarity of the output pulse at the end of the previous write. You must set the repetitive period of the data pattern T to 1T, 2T, 4T and 8T due to the system’s internal constraints. If you use other repetitive periods of the data pattern, E5022A/B system can not guarantee the change of write current polarity. This function makes it possible for popcorn noise measurement (i.e. failcount) to achieve an acceptable level of TAA under similar usage of the drive. Before this function was introduced, the write current direction always starts at a constant direction, thus you can only observe one side of the output.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writeMode’ is out of range.
hpe5022_ERROR_NSUP_CONF	The installed head amplifier has no capability to control the write current direction.

See Also “hpe5022_popcornNoiseWriteModeConfig_Q” on page 875

hpe5022_popcornNoiseWriteModeConfig_Q

C Syntax

ViStatushpe5022_popcornNoiseWriteModeConfig_Q(ViSession id, ViPInt16 writeMode);

Visual Basic Syntax

hpe5022_popcornNoiseWriteModeConfig_Q(ByVal id As Long, ByRef writeMode As Integer) As Long

Description

This function returns the specified write current direction at the beginning of write operation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writeMode

  Description: Returns the specified write current direction at the beginning of write operation.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_popcornNoiseWriteModeConfig” on page 872

hpe5022_popcornNoiseWritePatternConfig

C Syntax

ViStatus hpe5022_popcornNoiseWritePatternConfig(ViSession id, ViInt16 writePat);

Visual Basic Syntax

hpe5022_popcornNoiseWritePatternConfig(ByVal id As Long, ByVal writePat As Integer) As Long

Description

This function selects the data pattern for the write excitation. This allows you to select the different data patterns for write excitation from the standard data pattern independently. This also allows you to select the erase patterns as a write pattern..

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- writePat

  Description — Specifies the data pattern of the write. When this parameter is set to hpe5022_PAT_DEFAULT, write data pattern is the same as the pattern specified by “hpe5022_selectPattern” function.

  Direction — IN

  Preset value — hpe5022_PAT_DEFAULT

  Values

Name	Value	Description
hpe5022_PAT_DEFAULT	-1	Default pattern
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_REP_2	8	Repetition Pattern2
hpe5022_PAT_REP_3	9	Repetition Pattern3
hpe5022_PAT_REP_4	10	Repetition Pattern4

Name	Value	Description
hpe5022_PAT_USER	20	User-defined Pattern
hpe5022_PAT_USER_2	21	User-defined Pattern2
hpe5022_PAT_USER_3	22	User-defined Pattern3
hpe5022_PAT_USER_4	23	User-defined Pattern4
hpe5022_PAT_ERASE	101	Default Erase Pattern selected by the “hpe5022_eraseType” function
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writePat’ is out of range.

See Also

“hpe5022_popcornNoiseWritePatternConfig_Q” on page 878

“hpe5022_selectPattern” on page 119

hpe5022_popcornNoiseWritePatternConfig_Q

C Syntax

ViStatus hpe5022_popcornNoiseWritePatternConfig_Q(ViSession id, ViPInt16 writePat);

Visual Basic Syntax

hpe5022_popcornNoiseWritePatternConfig_Q(ByVal id As Long, ByRef writePat As Integer) As Long

Description

This function returns the data pattern of the write sequence.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- writePat

  Description — Returns the data pattern of the write. When this parameter is set to hpe5022_PAT_DEFAULT, write data pattern is the same as the pattern specified by “hpe5022_selectPattern” function.

  Direction — OUT

  Values — Same as the ‘writePat’ in the “hpe5022_popcornNoiseWritePatternConfig” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_popcornNoiseWritePatternConfig” on page 876

hpe5022_measurePopcornNoise

C Syntax

ViStatus hpe5022_measurePopcornNoise(ViSession id, ViInt32 counts);

Visual Basic Syntax

hpe5022_measurePopcornNoise(ByVal id As Long, ByVal counts As Long) As Long

Description

This function measures the popcorn noise. This function operates when the head unloads from the media after write. After the delay time has elapsed it then measures the head noise. When this function is executed, the following sequence are performed.

The sequence of this function are as follows:

1. Write the data during wrTime. (wrTime is specified by “hpe5022_popcornNoiseConfig”).
2. Wait for the delay time to elapse. (delay is specified by “hpe5022_popcornNoiseConfig”).
3. Measure popcorn noise during the read time. (rdTime is specified by “hpe5022_popcornNoiseConfig”).
4. Repeat steps 1-3, for a specified segment number of counts.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- counts

  Description: Specifies the number of test counts

  Direction: IN

  Values

Name	Value
hpe5022_POPCORN_COUN_MIN	1
hpe5022_POPCORN_COUN_MAX	1,000,000

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘counts’ is out of range.

See Also

“hpe5022_popcornNoiseConfig” on page 867

“hpe5022_setupPopcornNoise” on page 881

hpe5022_setupPopcornNoise

C Syntax

ViStatus hpe5022_setupPopcornNoise(ViSession id, ViInt32 counts, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupPopcornNoise(ByVal id As Long, ByVal counts As Long, ByRef testHndl As Long) As Long

Description

This function assigns the popcorn noise sequence to the specified test identifier. Refer to the “hpe5022_measurePopcornNoise” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- counts

  Description — Specifies the number of test counts.

  Direction — IN

Name	Value
hpe5022_POPCORN_COUN_MIN	1
hpe5022_POPCORN_COUN_MAX	1,000,000

- testHndl

  Description — Returns the test identifier. This identifier is used to execute popcorn noise by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘counts’ is out of range.

Error Code	Description
hpe5022_ERROR_MEM_ALLOC	Can not allocate enough memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_popcornNoiseConfig” on page 867

“hpe5022_measurePopcornNoise” on page 879

“hpe5022_measure” on page 387

hpe5022_popcornNoise_Q

C Syntax

ViStatus hpe5022_popcornNoise_Q(Visession id, ViPInt32 failCount);

Visual Basic Syntax

hpe5022_popcornNoise_Q(ByVal id As Long, ByRef failCount As Long) As Long

Description

This function returns the number of fail count occurrence. Fail count represents the number of detected head noise that has exceeded the threshold limits.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- failCount

  Description — Returns the fail count of the popcorn noise measurement.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The popcorn noise data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measurePopcornNoise” on page 879

“hpe5022_setupPopcornNoise” on page 881

hpe5022_popcornNoiseHistogram_Q

C Syntax

ViStatus hpe5022_popcornNoiseHistogram_Q(Visession id, ViReal64 volt[], ViInt32 count []);

Visual Basic Syntax

hpe5022_popcornNoiseHistogram_Q(ByVal id As Long, ByRef volt As Double, ByRef count As Long) As Long

Description

This function returns the histogram results of the popcorn noise measurement. The returned histogram is generated in such a way that the measurement range is equally divided into 256 parts, and these parts are used as range of each category in histogram. All measurement values that are used to calculate the popcorn noise is classified into each category to make the histogram.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- volt

  Description — Returns the center value of the range for 256 categories in the histogram.

  Direction — OUT

  Unit — Volt

- count

  Description — Returns the count for the corresponding volt range.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The popcorn noise data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measurePopcornNoise” on page 879

“hpe5022_setupPopcornNoise” on page 881

“hpe5022_popcornNoise_Q” on page 883

Popcorn Noise Measurement Function (Threshold Method)

This section describes the measurement functions used for Popcorn Noise Measurement when Agilent E5041A Dual Counter Module is used. The popcorn noise is counted by monitoring a level exceeded the specified threshold level.

hpe5022_popcornNoiseCtConfig

C Syntax

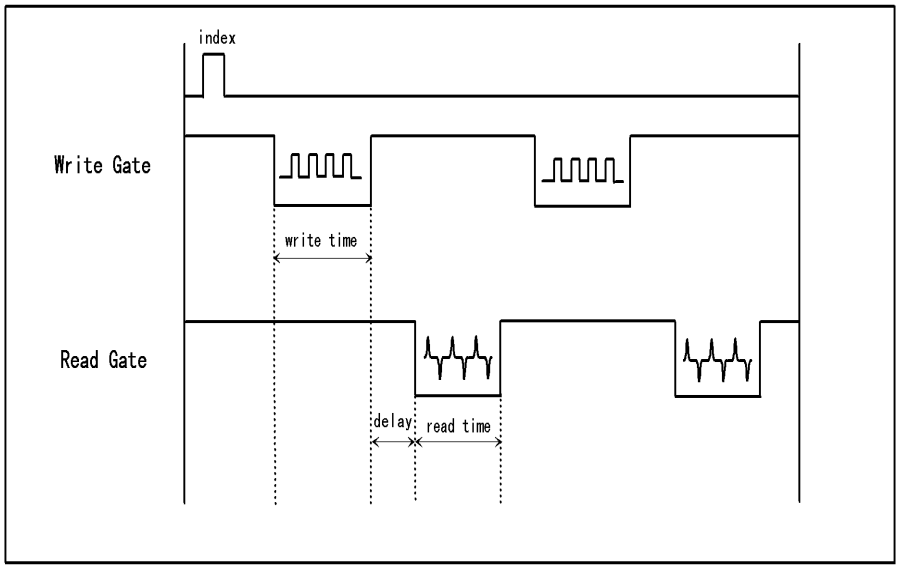
ViStatus hpe5022_popcornNoiseCtConfig(ViSession id, ViReal64 thrPos, ViReal64 thrNeg,ViReal64 wrTime, ViReal64 delay, ViReal 64 rdTime);

Visual Basic Syntax

hpe5022_popcornNoiseCtConfig(ByVal id As Long, ByVal thrPos As Double, ByVal thrNeg As Double, ByVal wrTime As Double, ByVal delay As Double, By Val rdTime As Double) As Long

Description

This function controls the configuration of the popcorn noise test when E5041A is used. In general, data is written at a specified time defined by ‘wrTime’. Then after the delay time has elapsed the read gate opens and the input signal is read for popcorn noise detection. Refer to the illustration below.



Parameters

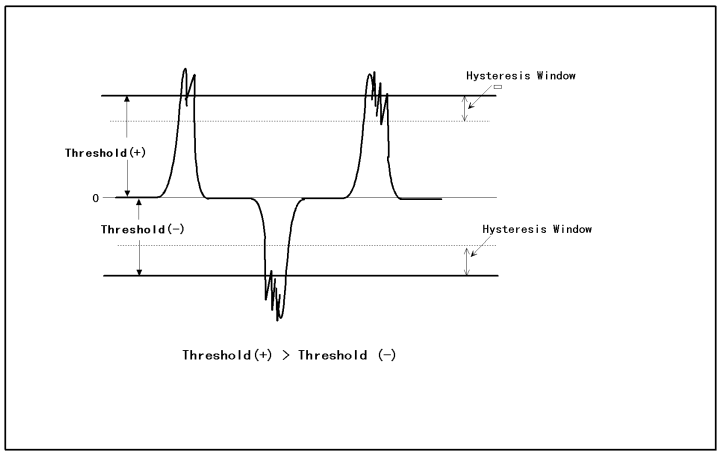
- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- thrPos

Description Specifies the positive threshold level for popcorn noise test. Popcorn noise is detected when the input signal exceeds this level.



The limit value of this parameter depends on the negative threshold and amplifier gain setting.

Limit Value	Description
| thrPos / thrNeg | ≤ 10	When | thrPos | > | thrNeg |
| thrNeg / thrPos | ≤ 10	When | thrPos | < | thrNeg |
$5 \times 10^{-3} \leq thrPos \times ampGain \leq 800 \times 10^{-3}$	ampGain is queriable by the hpe5022_ampGain_Q function

Direction IN

Unit Volt

Preset Value 100×10^{-6}

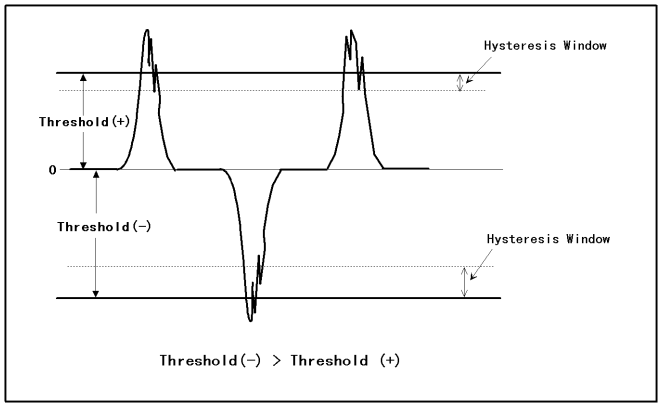
Values

Name	Value
hpe5022_POPCORN_CT_THR_POS_MIN	10×10^{-6}
hpe5022_POPCORN_CT_THR_POS_MAX	10×10^{-3}

- thrNeg

Description Specifies the negative threshold level for popcorn noise test. Popcorn noise is detected when the input signal goes below

this level.



The limit value of this parameter depends on the positive threshold and amplifier gain setting.

Limit Value	**Description**
| thrNeg / thrPos | ≤ 10	When | thrNeg | > | thrPos |
| thrPos / thrNeg | ≤ 10	When | thrNeg | < | thrPos |
$-800 \times 10^{-3} \leq thrNeg \times ampGain \leq -5 \times 10^{-3}$	ampGain is queriable by the hpe5022_ampGain_Q function

Direction IN

Unit Volt

Preset Value -100×10^{-6}

Values

Name	**Value**
hpe5022_POPCORN_CT_THR_NEG_MIN	-10×10^{-3}
hpe5022_POPCORN_CT_THR_NEG_MAX	-10×10^{-6}

- wrTime

  Description Specifies the time to write.

  Direction IN

  Unit Second

  Preset Value hpe5022_POPCORN_WRIT_TIME_MIN(10×10^{-6})

Values

Name	Value
hpe5022_POPCORN_CT_WRIT_TIME_MIN	10×10^{-6}
hpe5022_POPCORN_CT_WRIT_TIME_MAX	10×10^{-3}

- delay

  Description Specifies the delay time. Delay time is defined as the time distance between write and read operation.

  Direction IN

  Unit Second

  Preset Value hpe5022_POPCORN_CT_DEL_MIN (100×10^{-9})

  Values

Name	Value
hpe5022_POPCORN_CT_DEL_MIN	100×10^{-9}
hpe5022_POPCORN_CT_DEL_MAX	10×10^{-3}

- rdTime

  Description Specifies the read time.

  Direction IN

  Unit Second

  Preset Value hpe5022_POPCORN_CT_READ_TIME_MIN

  Values

Name	Value
hpe5022_POPCORN_CT_READ_TIME_MIN	1×10^{-6}
hpe5022_POPCORN_CT_READ_TIME_MAX	10×10^{-3}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’threshold’, ‘wrTime’, ‘delay’ and/or ‘rdTime’ is out of range.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also “hpe5022_measurePopcornNoiseCt” on page 903

hpe5022_popcornNoiseCtConfig_Q

C Syntax

ViStatus hpe5022_popcornNoiseCtConfig_Q(ViSession id, ViPReal64 thrPos, ViPReal64 thrNeg, ViPReal64 wrTime, ViPReal64 delay, ViPReal64 rdTime);

Visual Basic Syntax

hpe5022_popcornNoiseCtConfig_Q(ByVal id As Long, ByRef thrPos As Double, ByRef thrNeg As Double, ByRef wrTime As Double, ByRef delay As Double, ByRef rdTime As Double) As Long

Description

This function returns the configuration specified by “hpe5022_popcornNoiseCt Config” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- thrPos

  Description — Returns the positive threshold level. Popcorn noise is detected when the input signal exceeds this level.

  Direction — OUT

  Unit — Volt

- thrNeg

  Description — Returns the negative threshold level. Popcorn noise is detected when the input signal goes below this level.

  Direction — OUT

  Unit — Volt

- wrTime

  Description — Returns the time to write the data.

  Direction — OUT

  Unit — Second

- delay

  Description — Returns the time distance between write and read operation of the popcorn noise measurement.

  Direction — OUT

  Unit — Second

- rdTime

  Description — Returns the time to read.

Direction	OUT
Unit	Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtConfig” on page 886

hpe5022_popcornNoiseCtWriteModeConfig

C Syntax

ViStatus hpe5022_popcornNoiseCtConfig(ViSession id, ViInt16 writeMode);

Visual Basic Syntax

hpe5022_popcornNoiseCtConfig(ByVal id As Long, ByVal writeMode As Integer) As Long

Description

This function controls the write current direction at the beginning of write operation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writeMode

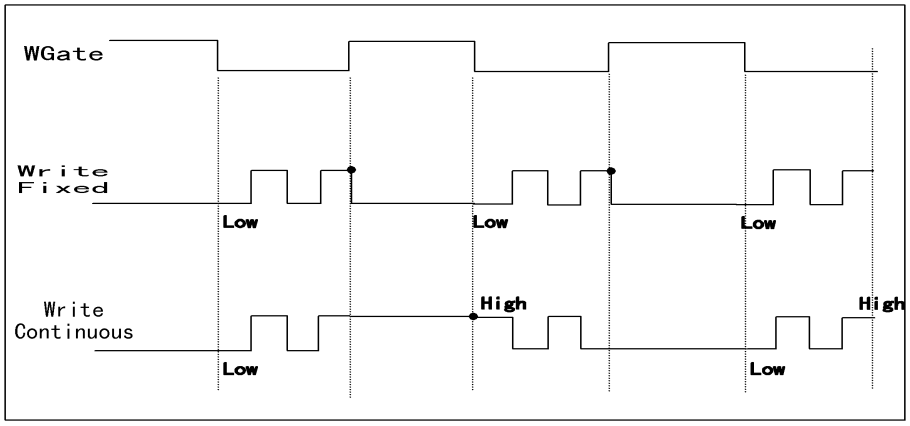
  Description: Specifies the write current direction at the beginning of write operation.

  Direction: IN

  Values

Name	Value	Description
hpe5022_WRIT_DATA_POL_FIXED	0	The write current direction at the beginning of each write operation is always the same.
hpe5022_WRIT_DATA_POL_CONTINUOUS	1	The write current direction at the beginning of write operation is the same as the last current direction of the previous write operation.

Figure 3-34 Write Mode



When [Write Mode] is set to “fixed”, the write current direction at the beginning of each write operation is always the same. On the other hand, when [Write Mode] is set to “continuous” the write current direction is the same as the last current direction of the previous write operation. Each time the write current alternates it allows you to check the polarity of the output pulse at the end of the previous write. You must set the repetitive period of the data pattern T to 1T, 2T, 4T and 8T due to the system’s internal constraints. If you use other repetitive periods of the data pattern, E5022A/B system can not guarantee the change of write current polarity. This function makes it possible for popcorn noise measurement (i.e. failcount) to achieve an acceptable level of TAA under similar usage of the drive. Before this function was introduced, the write current direction always starts at a constant direction, thus you can only observe one side of the output.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writeMode’ is out of range.
hpe5022_ERROR_NSUP_CONF	The installed head amplifier has no capability to control the write current direction.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtWriteModeConfig_Q” on page 895

hpe5022_popcornNoiseCtWriteModeConfig_Q

C Syntax

ViStatus hpe5022_popcornNoiseCtWriteModeConfig_Q(ViSession id, ViPInt16 writeMode);

Visual Basic Syntax

hpe5022_popcornNoiseCtWriteModeConfig_Q(ByVal id As Long, ByRef writeMode As Integer) As Long

Description

This function returns the write current direction at the beginning of write operation.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- writeMode

  Description — Returns the specified write current direction at the beginning of write operation.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtWriteModeConfig” on page 893

hpe5022_popcornNoiseCtWritePatternConfig

C Syntax

ViStatus hpe5022_popcornNoiseCtWritePatternConfig(ViSession id, ViInt16 writePat);

Visual Basic Syntax

hpe5022_popcornNoiseCtWritePatternConfig(ByVal id As Long, ByVal writePat As Integer) As Long

Description

This function specifies the data pattern during the write sequence. This allows you to select the different data pattern for read and write operations independently. This also allows you to select the erase patterns as a write pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- writePat

  Description: Specifies the data pattern of the write. When this parameter is set to hpe5022_PAT_DEFAULT, write data pattern is the same as the pattern specified by “hpe5022_selectPattern” function.

  Direction: IN

  Preset value: hpe5022_PAT_DEFAULT

  Values

Name	Value	Description
hpe5022_PAT_DEFAULT	-1	Default pattern
hpe5022_PAT_HF	0	HF pattern
hpe5022_PAT_LF	1	LF pattern
hpe5022_PAT_ISO	2	Isolated pulse pattern
hpe5022_PAT_PRBS	3	Pseudo random pattern
hpe5022_PAT_REP	4	Repetitive pattern
hpe5022_PAT_NLTS_5TH	5	NLTS 5th pattern
hpe5022_PAT_OWHF	6	Overwrite HF pattern
hpe5022_PAT_OWLF	7	Overwrite LF pattern
hpe5022_PAT_REP_2	8	Repetition Pattern2
hpe5022_PAT_REP_3	9	Repetition Pattern3
hpe5022_PAT_REP_4	10	Repetition Pattern4

Name	Value	Description
hpe5022_PAT_USER	20	User-defined Pattern
hpe5022_PAT_USER_2	21	User-defined Pattern2
hpe5022_PAT_USER_3	22	User-defined Pattern3
hpe5022_PAT_USER_4	23	User-defined Pattern4
hpe5022_PAT_ERASE	101	Default Erase Pattern selected by the “hpe5022_eraseType” function
hpe5022_PAT_ERASE_DC_NEG	102	DC- Erase Pattern
hpe5022_PAT_ERASE_DC_POS	103	DC+ Erase Pattern
hpe5022_PAT_ERASE_AC	104	AC Erase Pattern

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘writePat’ is out of range.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtWritePatternConfig_Q” on page 898

“hpe5022_selectPattern” on page 119

hpe5022_popcornNoiseCtWritePatternConfig_Q

C Syntax

ViStatus hpe5022_popcornNoiseCtWritePatternConfig_Q(ViSession id, ViPInt16 writePat);

Visual Basic Syntax

hpe5022_popcornNoiseCtWritePatternConfig_Q(ByVal id As Long, ByRef writePat As Integer) As Long

Description

This function returns the data pattern of the write sequence.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- writePat

  Description — Returns the data pattern of the write. When this parameter is set to hpe5022_PAT_DEFAULT, write data pattern is the same as the pattern specified by “hpe5022_selectPattern” function.

  Direction — OUT

  Values — Same as the ‘writePat’ in the “hpe5022_popcornNoiseCtWritePatternConfig” function.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtWritePatternConfig” on page 896

hpe5022_popcornNoiseCtHysteresisConfig

C Syntax

ViStatus hpe5022_popcornNoiseCtHysteresisConfig(ViSession id, ViInt16 hystLevel);

Visual Basic Syntax

hpe5022_popcornNoiseCtHysteresisConfig(ByVal id As Long, ByVal hystLevel As Integer) As Long

Description

This function controls the size of the hysteresis level (i.e, hysteresis window) used in detecting popcorn noise.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- hystLevel

  Description — Specifies the hysteresis level. Three hysteresis levels can be selected to detect popcorn noise. Select the hysteresis level that suits your measurement applications.

  Direction — IN

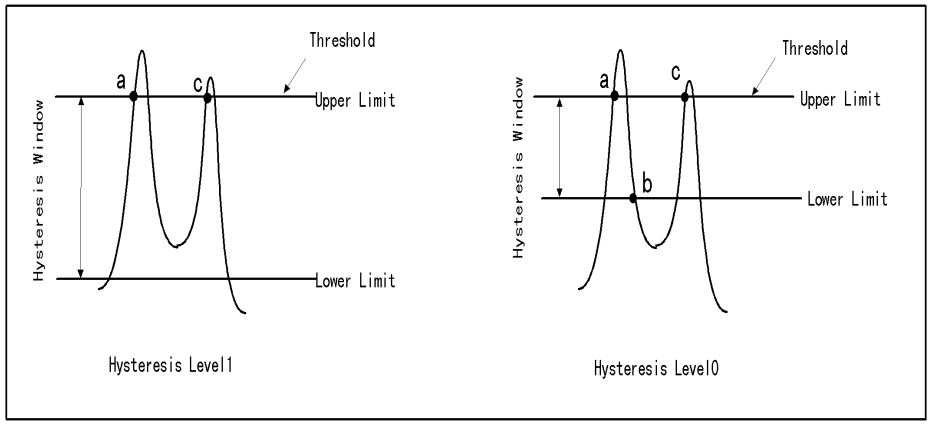
  Preset Value — 1 (hpe5022_COMP_HYST_LEVEL_1)

  Values

Name	Value	Description
hpe5022_COMP_HYST_LEVEL_0	0	1.4% of threshold voltage (nominal).
hpe5022_COMP_HYST_LEVEL_1	1	4.0% of threshold voltage (nominal).
hpe5022_COMP_HYST_LEVEL_2	2	7.0% of threshold voltage (nominal).

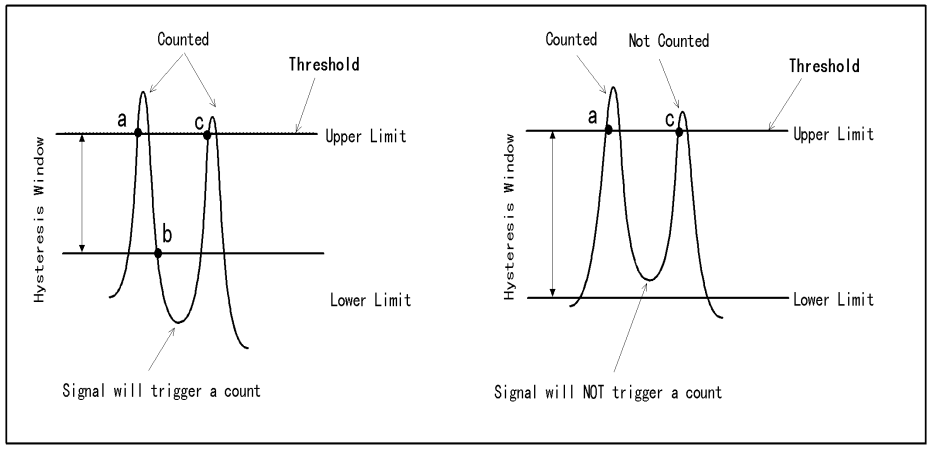
The above percentage value of each hysteresis level is valid only when the specified threshold is between 45 to 450 [mV]. When the threshold level is between 5 to 45[mV], the hysteresis level becomes larger. When the threshold level is between 450 to 800 [mV], the hysteresis level becomes smaller.

Figure 3-35 Hysteresis Level



The level of the hysteresis window can be selected with this function. Optimum hysteresis level may depend on measurement applications and other parameter factors such as signal to noise, popcorn noise level etc. As shown in Figure 3-35 (left figure) a wider hysteresis window translates to less sensitive detection of popcorn noise. Narrowing the hysteresis window allows you to detect popcorn noise at low signal amplitudes as shown in Figure 3-35 (right figure).

Figure 3-36 Counting Popcorn Noise



In order to trigger a count, the input signal must cross both the threshold level and the lower limit of the hysteresis window at points a and b as shown in Figure 3-36 (left figure). If the lower peak of the input signal does not cross the lower limit of the hysteresis window, this will not trigger a count even if the next peak amplitude at point c crosses over the threshold level as shown in Figure 3-36 (right figure). This will not be counted as popcorn noise.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘hystLevel’ is out of range.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtHysteresisConfig_Q” on page 902

hpe5022_popcornNoiseCtHysteresisConfig_Q

C Syntax

ViStatus hpe5022_popcornNoiseCtHysteresisConfig_Q(ViSession id, ViPInt16 hystLevel);

Visual Basic Syntax

hpe5022_popcornNoiseCtHysteresisConfig_Q(ByVal id As Long, ByRef hystLevel As Integer) As Long

Description

This function returns the specified hysteresis level for popcorn noise measurement in the “hpe5022_popcornNoiseCtHysteresisConfig” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- hystLevel

  Description: Returns the specified hysteresis level.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘hystlevel’ is out of range.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtHysteresisConfig” on page 899

hpe5022_measurePopcornNoiseCt

C Syntax

ViStatus hpe5022_measurePopcornNoiseCt(ViSession id, ViInt32 counts);

Visual Basic Syntax

hpe5022_measurePopcornNoiseCt(ByVal id As Long, ByVal counts As Long) As Long

Description

This function measures the popcorn noise. When this function is executed the following sequence of events take place. Data is written during write time, wait for delay time to elapse, then measure popcorn noise during read time. The “hpe5022_setupPopcornNoiseCt” function is used to setup the measurement sequence.

The sequence of this function are as follows:

1. Write the data during wrTime. (wrTime is specified by “hpe5022_popcornNoiseCtConfig”). The data pattern specified by “hpe5022_selectPattern” function will be written during wrTime.
2. Wait for delay time to elapse. (delay is specified by “hpe5022_popcornNoiseCtConfig”).
3. Measure popcorn noise during the read time. (rdTime is specified by “hpe5022_popcornNoiseCtConfig”).
4. Repeat steps 1-3 for specified number of counts.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

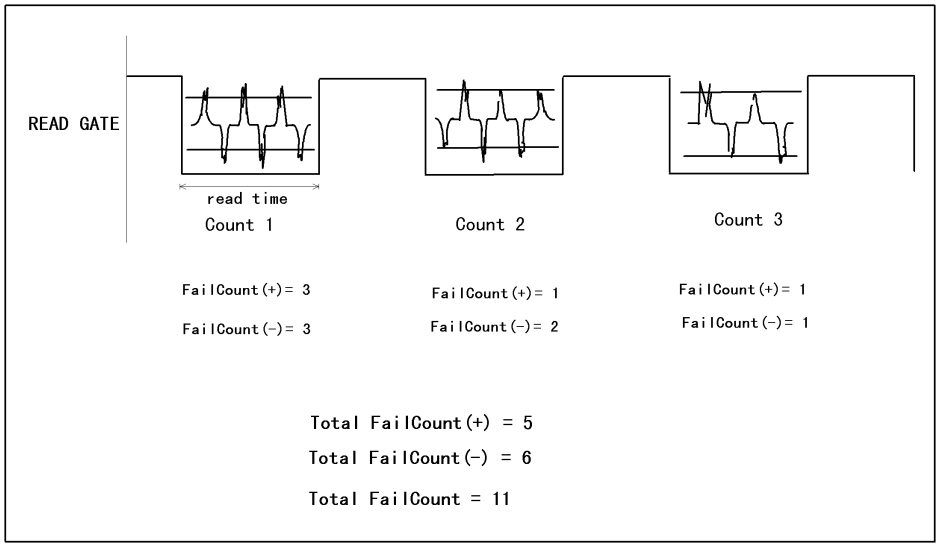
- counts

  Description — Specifies the number of times to count popcorn noise.

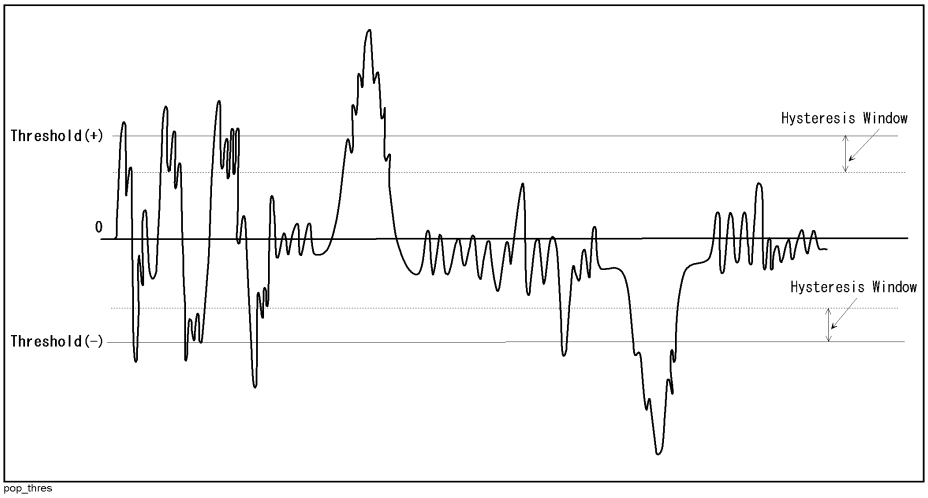
  Direction — IN

  Values

Name	Value
hpe5022_POPCORN_COUN_MIN	1
hpe5022_POPCORN_COUN_MAX	1,000,000



After write operation the read gate will not immediately open. Read operation will only start after the delay time has elapsed. The input signal is evaluated for popcorn noise at specified amount of time defined by ‘read time’. Popcorn noise can only be detected if the input signal cross both the threshold level and the lower limit of the hysteresis window. Refer to Figure 3-36 on page 900. Since E5041A is a dual counter module, it has two threshold levels that detect popcorn noise for both positive and negative amplitudes of the input signal as shown below. Each threshold has its own hysteresis window that triggers the count. As shown in the figure above, for each count the detected popcorn noise for both positive and negative amplitudes are counted as failcount (+) and failcount(-) respectively.



Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘counts’ is out of range.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtConfig” on page 886

“hpe5022_setupPopcornNoiseCt” on page 906

hpe5022_setupPopcornNoiseCt

C Syntax

ViStatus hpe5022_setupPopcornNoiseCt(ViSession id, ViInt32 counts, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupPopcornNoiseCt(ByVal id As Long, ByVal counts As Long, ByRef testHndl As Long) As Long

Description

This function assigns the popcorn noise measurement sequence to the test identifier. Refer to the “hpe5022_measurePopcornNoiseCt” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- counts

  Description — Specifies the number of times to count popcorn noise.

  Direction — IN

Name	Value
hpe5022_POPCORN_COUN_MIN	1
hpe5022_POPCORN_COUN_MAX	1,000,000

- testHndl

  Description — Returns the test identifier. This identifier is used to execute popcorn noise by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘counts’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Can not allocate enough memory. Release the finished setup function using the “hpe5022_releaseSetup” function.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_popcornNoiseCtConfig” on page 886

“hpe5022_measurePopcornNoiseCt” on page 903

“hpe5022_measure” on page 387

hpe5022_popcornNoiseCt_Q

C Syntax

ViStatus hpe5022_popcornNoiseCt_Q(Visession id, ViPInt32 failCountPos, ViPInt32 failCountNeg);

Visual Basic Syntax

hpe5022_popcornNoiseCt_Q(ByVal id As Long, ByRef failCountPos As Long, ByRef failCountNeg As Long) As Long

Description

This function returns the number of counted popcorn noise as detected by both the positive and negative thresholds.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- failCountPos

  Description: Returns the number of counted popcorn noise by the positive threshold. If the actual fail counts exceed the maximum number of allowable failcounts as specified by hpe5022_POPCORN_CT_FAIL_COUN_MAX, the excess number of fail counts will not be counted.

Name	Value
hpe5022_POPCORN_CT_FAIL_COUN_MAX	2,147,483,647

  Direction: OUT

- failCountNeg

  Description: Returns the number counted popcorn noise by the negative threshold. If the actual fail counts exceed the maximum number of allowable failcounts as specified by hpe5022_POPCORN_CT_FAIL_COUN_MAX, the excess number of fail counts will not be counted.

Name	Value
hpe5022_POPCORN_CT_FAIL_COUN_MAX	2,147,483,647

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The popcorn noise data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_NOT_INIT	The E5041A is not installed.

See Also

“hpe5022_measurePopcornNoiseCt” on page 903

“hpe5022_setupPopcornNoiseCt” on page 906

Overwrite Measurement Function

This section describes the functions related with the overwrite measurement. This measurement is done by a spectrum analyzer.

hpe5022_measureOverwrite

C Syntax

ViStatus hpe5022_measureOverwrite(ViSession id, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureOverwrite(ByVal id As Long, ByVal ave As Integer) As Long

Description

This function measures the overwrite ratio. The function sequences are as follows.

1. Perform the three-track-erase. (the entire track erase for five positions, the track, its both side tracks, and the center between them.)
2. Erase at write offset.
3. Write the overwrite LF data pattern specified by the "hpe5022_overwriteLfPattern" function for an entire track.
4. Move the head to the read offset position specified by the "hpe5022_readTrackOffset"function.
5. Measure the LF amplitude level.
6. Perform the band erase as follows:
   a. write offset + erase offset
   b. write offset - erase offset
   c. write offset

   Write offset is specified by the "hpe5022_writeTrackOffset". Erase offset is specified by the service function "hpe5022_overwriteEraseOffset".
7. Write the overwrite LF data pattern specified by the “hpe5022_overwriteLfPattern” function for an entire track.
8. Write the overwrite HF data pattern specified by the “hpe5022_overwriteHfPattern” function for an entire track.
9. Move the head to the read offset position specified by the "hpe5022_readTrackOffset".
10. Measure the residual LF amplitude level.
11. Repeat above 2 through 10 'ave' (see parameters) times if it is set at more than 1.

The data patterns for this measurement are used automatically. The data pattern specified by the “hpe5022_selectPattern” function is not used for this measurement.

The “hpe5022_overwrite_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_OVERWRITE_COUN_MIN	1
hpe5022_OVERWRITE_COUN_MAX	25

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.

Error Code	Description
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

"hpe5022_overwrite_Q" on page 915

"hpe5022_overwriteHfPattern" on page 136

"hpe5022_overwriteLfPattern" on page 138

"hpe5022_readTrackOffset" on page 219

"hpe5022_writeTrackOffset" on page 222

"hpe5022_driveState" on page 209

hpe5022_setupOverwrite

C Syntax

ViStatus hpe5022_setupOverwrite(ViSession id, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupOverwrite(ByVal id As Long, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the overwrite measurement sequence to the specified test identifier. Refer to the “hpe5022_measureOverwrite” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_overwrite_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as the ‘ave’ in the “hpe5022_measureOverwrite” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the overwrite measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureOverwrite” on page 910

“hpe5022_overwrite_Q” on page 915

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_overwrite_Q

C Syntax

ViStatus hpe5022_overwrite_Q(ViSession id,ViPReal64 overwrite);

Visual Basic Syntax

hpe5022_overwrite_Q(ByVal id As Long, ByRef overwrite As Double) As Long

Description

This function returns the result of the overwrite measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- overwrite

  Description: Returns the overwrite ratio.

  Direction: OUT

  Unit: dB

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The overwrite data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureOverwrite” on page 910

“hpe5022_setupOverwrite” on page 913

hpe5022_overwriteDataSize_Q

C Syntax

ViStatus hpe5022_overwriteDataSize_Q(ViSession id,ViPInt32 size);

Visual Basic Syntax

hpe5022_overwriteDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the overwrite data returned by the “hpe5022_overwriteData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of overwrite data.

  The returned size will be the same as the value of 'ave’ specified by the “hpe5022_measureOverwrite” or “hpe5022_setupOverwrite” function if the measurement executed properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The overwrite data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_overwriteData_Q”

“hpe5022_measureOverwrite”

“hpe5022_setupOverwrite”

hpe5022_overwriteData_Q

C Syntax

ViStatus hpe5022_overwriteData_Q(ViSession id,ViReal64 data[]);

Visual Basic Syntax

hpe5022_overwriteData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the overwrite data of the measurement for each revolution.

When the ‘ave’ parameter in the “hpe5022_measureOverwrite” or the “hpe5022_setupOverwrite” function is set for more than one, the function allows you to get the overwrite data for each revolution. If the ‘ave’ is set to 1, the output of this function is the same as one of the “hpe5022_overwrite_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- data

  Description — Returns the data array of the overwrite measurement. The array size is returned by the “hpe5022_overwriteDataSize_Q” function.

  Direction — OUT

  Units — dB

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The overwrite data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_overwriteDataSize_Q” on page 916

hpe5022_overwriteStatistic_Q

C Syntax

ViStatus hpe5022_overwriteStatistic_Q(ViSession id,ViPReal64 mean, ViPReal64 min,ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_overwriteStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistic (mean, minimum, maximum and standard deviation) of overwrite data. The data returned by the “hpe5022_overwriteData_Q” function is used for this statistic analysis.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- mean

  Description — Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_overwrite_Q” function.

  Direction — OUT

  Units — dB

- min

  Description — Returns the minimum value.

  Direction — OUT

  Units — dB

- max

  Description — Returns the maximum value.

  Direction — OUT

  Units — dB

- stdDev

  Description — Returns the standard deviation value.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The overwrite data is corrupt. Check if your measurement sequence is correct.

See Also “hpe5022_overwriteData_Q” on page 917

Sectored Overwrite Measurement Function

This section describes the functions related with the sectored overwrite measurement. This measurement is done by a spectrum analyzer.

hpe5022_measureSectoredOverwrite

C Syntax

ViStatus hpe5022_measureSectoredOverwrite(ViSession id, ViInt16 sectors, ViInt16 gapSectors, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureSectoredOverwrite(ByVal id As Long, ByVal sectors As Integer, ByVal gapSectors As Integer, ByVal ave As Integer) As Long

Description

This function measures the overwrite ratio with the sectoring method.

The data patterns for this measurement are used automatically. The data pattern specified by the “hpe5022_selectPattern” function is not used for this measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- sectors

  Description: Specifies the number of sectors to divide.

Name	Value
hpe5022_SECTORED_OVERWRITE_SECTORS_MIN	2
hpe5022_SECTORED_OVERWRITE_SECTORS_MAX	80

  Sector time must be no less than hpe5022_SECTORED_OVERWRITE_SECTOR_TIME_MIN (100usec). Sector time is 60 sec / (RPM × sectors)

  Direction: IN

- gapSectors

  Description: Specifies the number of sectors to skip after writing one sector. Minimum value is as follows:

Name	Value
hpe5022_SECTORED_OVERWRITE_GAP_SECTORS_MIN	1

Maximum value is “sectors” – 1. “sectors” must be divisible by “gapSectors + 1.” If it is not divisible, “hpe5022_ERROR_INV_SETUP” occurs.

Direction IN

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_SECTORED_OVERWRITE_COUN_MIN	1
hpe5022_SECTORED_OVERWRITE_COUN_MAX	25

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘sectors”, “gapSectors”, or ‘ave’ is out of range.
hpe5022_ERROR_INV_SETUP	The sector time is less than hpe5022_SECTORED_OVERWRITE_SECTOR_TIME_MIN, or the remainder of “sectors / (gapSectors + 1)” is not 0.

See Also

“hpe5022_setupSectoredOverwrite” on page 923

“hpe5022_sectoredOverwrite_Q” on page 925

“hpe5022_sectoredOverwriteDataSize_Q” on page 926

“hpe5022_sectoredOverwriteData_Q” on page 927

“hpe5022_sectoredOverwriteStatistic_Q” on page 928

“hpe5022_sectoredOverwriteRawDataSize_Q” on page 930

“hpe5022_sectoredOverwriteReferenceRawData_Q” on page 931

“hpe5022_sectoredOverwriteResidualRawData_Q” on page 932

hpe5022_setupSectoredOverwrite

C Syntax

ViStatus hpe5022_setupSectoredOverwrite(ViSession id, ViInt16 sectors, ViInt16 gapSectors, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSectoredOverwrite(ByVal id As Long, ByVal sectors As Integer, ByVal gapSectors As Integer, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function sets up the sectored overwrite measurement sequence.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- sectors

  Description — Specifies the number of sectors to divide.

Name	Value
hpe5022_SECTORED_OVERWRITE_SECTORS_MIN	2
hpe5022_SECTORED_OVERWRITE_SECTORS_MAX	80

  Sector time must be no less than hpe5022_SECTORED_OVERWRITE_SECTOR_TIME_MIN (100µsec). Sector time is 60 sec / (RPM × sectors)

  Direction — IN

- gapSectors

  Description — Specifies the number of sectors to skip after writing one sector. Minimum value is as follows:

Name	Value
hpe5022_SECTORED_OVERWRITE_GAP_SECTORS_MIN	1

  Maximum value is “sectors” – 1. “sectors” must be divided by “gapSectors + 1.” If remainder is not 0, hpe5022_ERROR_INV_SETUP occurs.

  Direction — IN

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

Values

Name	Value
hpe5022_SECTORED_OVERWRITE_COUN_MIN	1
hpe5022_SECTORED_OVERWRITE_COUN_MAX	25

- testHndl

  Description Returns test handle.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘sectors”, “gapSectors”, or ‘ave’ is out of range.
hpe5022_ERROR_INV_SETUP	The sector time is less than hpe5022_SECTORED_OVERWRITE_SECTOR_TIME_MIN, or the remainder of “sectors / (gapSectors + 1)” is not 0.

See Also

“hpe5022_measureSectoredOverwrite” on page 920

“hpe5022_sectoredOverwrite_Q” on page 925

“hpe5022_sectoredOverwriteDataSize_Q” on page 926

“hpe5022_sectoredOverwriteData_Q” on page 927

“hpe5022_sectoredOverwriteStatistic_Q” on page 928

“hpe5022_sectoredOverwriteRawDataSize_Q” on page 930

“hpe5022_sectoredOverwriteReferenceRawData_Q” on page 931

“hpe5022_sectoredOverwriteResidualRawData_Q” on page 932

hpe5022_sectoredOverwrite_Q

C Syntax

ViStatus hpe5022_sectoredOverwrite_Q(ViSession id, ViPReal64 overwrite);

Visual Basic Syntax

hpe5022_sectoredOverwrite_Q(ByVal id As Long, ByRef overwrite As Double) As Long

Description

This function returns the overwrite ratio by the sectoring method.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- overwrite

  Description: Returns the overwrite ratio in dB.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_DATA_CORRUPT	The sectored overwrite data is corrupt.

See Also

"hpe5022_measureSectoredOverwrite" on page 920

"hpe5022_setupSectoredOverwrite" on page 923

"hpe5022_sectoredOverwriteDataSize_Q" on page 926

"hpe5022_sectoredOverwriteData_Q" on page 927

"hpe5022_sectoredOverwriteStatistic_Q" on page 928

"hpe5022_sectoredOverwriteRawDataSize_Q" on page 930

"hpe5022_sectoredOverwriteReferenceRawData_Q" on page 931

"hpe5022_sectoredOverwriteResidualRawData_Q" on page 932

hpe5022_sectoredOverwriteDataSize_Q

C Syntax

ViStatus hpe5022_sectoredOverwriteDataSize_Q(ViSession id, ViPReal32 size);

Visual Basic Syntax

hpe5022_sectoredOverwriteDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the size of the sectored overwrite data.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- overwrite

  Description: Returns the size of the sectored overwrite data.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_DATA_CORRUPT	The sectored overwrite data is corrupt.

See Also

"hpe5022_measureSectoredOverwrite" on page 920

"hpe5022_setupSectoredOverwrite" on page 923

"hpe5022_sectoredOverwrite_Q" on page 925

"hpe5022_sectoredOverwriteData_Q" on page 927

"hpe5022_sectoredOverwriteStatistic_Q" on page 928

"hpe5022_sectoredOverwriteRawDataSize_Q" on page 930

"hpe5022_sectoredOverwriteReferenceRawData_Q" on page 931

"hpe5022_sectoredOverwriteResidualRawData_Q" on page 932

hpe5022_sectoredOverwriteData_Q

C Syntax

ViStatus hpe5022_sectoredOverwriteData_Q(ViSession id, ViReal64 data[]);

Visual Basic Syntax

hpe5022_sectoredOverwriteData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the sectored overwrite data.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the sectored overwrite data. The data size is returned by “hpe5022_sectoredOverwriteDataSize_Q”

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The sectored overwrite data is corrupt.

See Also

“hpe5022_measureSectoredOverwrite” on page 920

“hpe5022_setupSectoredOverwrite” on page 923

“hpe5022_sectoredOverwrite_Q” on page 925

“hpe5022_sectoredOverwriteDataSize_Q” on page 926

“hpe5022_sectoredOverwriteStatistic_Q” on page 928

“hpe5022_sectoredOverwriteRawDataSize_Q” on page 930

“hpe5022_sectoredOverwriteReferenceRawData_Q” on page 931

“hpe5022_sectoredOverwriteResidualRawData_Q” on page 932

hpe5022_sectoredOverwriteStatistic_Q

C Syntax

ViStatus hpe5022_sectoredOverwriteStatistic_Q(ViSession id, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_sectoredOverwriteStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function returns the statistics of sectored overwrite data.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mean

  Description: Returns the mean value of the sectored overwrite data.

  Direction: OUT

- min

  Description: Returns the minimum value of the sectored overwrite data.

  Direction: OUT

- max

  Description: Returns the maximum value of the sectored overwrite data.

  Direction: OUT

- stdDev

  Description: Returns the standard deviation value of the sectored overwrite data.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The sectored overwrite data is corrupt.

See Also

“hpe5022_measureSectoredOverwrite” on page 920

“hpe5022_setupSectoredOverwrite” on page 923

“hpe5022_sectoredOverwrite_Q” on page 925

“hpe5022_sectoredOverwriteDataSize_Q” on page 926

“hpe5022_sectoredOverwriteData_Q” on page 927

“hpe5022_sectoredOverwriteRawDataSize_Q” on page 930

“hpe5022_sectoredOverwriteReferenceRawData_Q” on page 931

“hpe5022_sectoredOverwriteResidualRawData_Q” on page 932

hpe5022_sectoredOverwriteRawDataSize_Q

C Syntax

ViStatus hpe5022_sectoredOverwriteRawDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_sectoredOverwriteRawDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the size of the sectored overwrite raw data for the reference and the residual.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the sectored overwrite raw data.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_DATA_CORRUPT	The sectored overwrite data is corrupt.

See Also

"hpe5022_measureSectoredOverwrite" on page 920

"hpe5022_setupSectoredOverwrite" on page 923

"hpe5022_sectoredOverwrite_Q" on page 925

"hpe5022_sectoredOverwriteDataSize_Q" on page 926

"hpe5022_sectoredOverwriteData_Q" on page 927

"hpe5022_sectoredOverwriteStatistic_Q" on page 928

"hpe5022_sectoredOverwriteReferenceRawData_Q" on page 931

"hpe5022_sectoredOverwriteResidualRawData_Q" on page 932

hpe5022_sectoredOverwriteReferenceRawData_Q

C Syntax

ViStatus hpe5022_sectoredOverwriteReferenceRawData_Q(ViSession id, ViReal64 data[]);

Visual Basic Syntax

hpe5022_sectoredOverwriteReferenceRawData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the sectored overwrite raw data for the reference LF.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- data

  Description — Returns the sectored overwrite reference raw data in dBm. Data size is returned by “hpe5022_sectoredOverwriteRawDataSize_Q.”

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The sectored overwrite data is corrupt.

See Also

“hpe5022_measureSectoredOverwrite” on page 920

“hpe5022_setupSectoredOverwrite” on page 923

“hpe5022_sectoredOverwrite_Q” on page 925

“hpe5022_sectoredOverwriteDataSize_Q” on page 926

“hpe5022_sectoredOverwriteData_Q” on page 927

“hpe5022_sectoredOverwriteStatistic_Q” on page 928

“hpe5022_sectoredOverwriteRawDataSize_Q” on page 930

“hpe5022_sectoredOverwriteResidualRawData_Q” on page 932

hpe5022_sectoredOverwriteResidualRawData_Q

C Syntax

ViStatus hpe5022_sectoredOverwriteResidualRawData_Q(ViSession id, ViInt16 rev, ViReal64 data[]);

Visual Basic Syntax

hpe5022_sectoredOverwriteResidualRawData_Q(ByVal id As Long, ByVal rev As Integer, ByRef data As Double) As Long

Description

This function returns the sectored overwrite raw data for the residual LF. The residual LF data is measured average times. The argument “rev” specifies the data to be retrieved. “rev = 0” means the first data, “rev = 1” is for the second, “rev = ave – 1” is for the last.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rev

  Description: Specifies the data to be retrieved. 0 is the first data, 1 for the second, “average – 1” for the last.

  Direction: IN

- data

  Description: Returns the sectored overwrite residual raw data in dBm. Data size is returned by “hpe5022_sectoredOverwriteRawDataSize_Q.”

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘rev” is out of range.
hpe5022_ERROR_DATA_CORRUPT	The sectored overwrite data is corrupt.

See Also

“hpe5022_measureSectoredOverwrite” on page 920

Partial Erasure Measurement

This section describes the functions related with partial erasure measurement. Partial erasure is the amplitude loss of the play-back voltage when two transitions are written at a small separation.

hpe5022_measurePartialErasure

C Syntax

ViStatus hpe5022_measurePartialErasure(ViSession id, ViInt16 ave);

Visual Basic Syntax

hpe5022_measurePartialErasure(ByVal id As Long, ByVal ave As Integer) As Long

Description

This function measures the partial erasure. The function sequence is as follows.

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset”.
2. Perform a erase (same as the hpe5022_erase function) for an entire track.
3. Write the 3T data pattern for an entire track at the write track offset position.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset”.
5. Measure the amplitude level.
6. Move the head to the write track offset position.
7. Perform a erase (same as the hpe5022_erase function) for an entire track.
8. Write the 1T data pattern for an entire track.
9. Move the head to the read track offset position.
10. Measure the amplitude level.
11. Repeat the steps 1 to 10 ‘ave’ (see parameters) times if it is more than 1.

The data patterns for this measurement are used automatically. The data pattern specified by the “hpe5022_selectPattern” function is not used for this measurement.

The “hpe5022_partialErasure_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- ave

  Description Specifies the number of measurements for averaging.

  Direction IN

  Values

Name	Value
hpe5022_PART_ERAS_COUN_MIN	1
hpe5022_PART_ERAS_COUN_MAX	25

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifier.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_partialErasure_Q” on page 939

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_driveState” on page 209

hpe5022_setupPartialErasure

C Syntax

ViStatus hpe5022_setupPartialErasure(ViSession id, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupPartialErasure(ByVal id As Long, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the partial erasure measurement sequence to the specified test identifier. Refer to the “hpe5022_measurePartialErasure” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_partialErasure_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as the ‘ave’ in the “hpe5022_measurePartialErasure” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the partial erasure measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘ave’ is out of range.

Error Code	Description
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measurePartialErasure" on page 934

"hpe5022_partialErasure_Q" on page 939

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_partialErasure_Q

C Syntax

ViStatus hpe5022_partialErasure_Q(ViSession id,ViPReal64 pe);

Visual Basic Syntax

hpe5022_partialErasure_Q(ByVal id As Long, ByRef pe As Double) As Long

Description

This function returns the result of partial erasure measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- pe

  Description: Returns the amplitude loss from partial erasure measurement.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_DATA_CORRUPT	The partial erasure data is corrupt. Check if your measurement sequence is correct.

See Also

"hpe5022_measurePartialErasure" on page 934

"hpe5022_setupPartialErasure" on page 937

"hpe5022_measure" on page 387

hpe5022_partialErasureDataSize_Q

C Syntax

ViStatus hpe5022_partialErasureDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_partialErasureDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the partial erasure data returned by the “hpe5022_partialErasureData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of partial erasure data.

  The size will be the same as the value of 'ave’ specified by the “hpe5022_measurePartialErasure” or “hpe5022_setupPartialErasure” functions if measurement is executed properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The partial erasure data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_partialErasureData_Q” on page 941

“hpe5022_measurePartialErasure” on page 934

“hpe5022_setupPartialErasure” on page 937

hpe5022_partialErasureData_Q

C Syntax

ViStatus hpe5022_partialErasureData_Q(ViSession id,ViReal64 data[]);

Visual Basic Syntax

hpe5022_partialErasureData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the partial erasure data of the measurement for each revolution.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the data array of the partial erasure. The array size is returned by the “hpe5022_partialErasureDataSize_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The partial erasure data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_partialErasureDataSize_Q” on page 940

hpe5022_partialErasureStatistic_Q

C Syntax

ViStatus hpe5022_partialErasureStatistic_Q(ViSession id,ViPReal64 mean, ViPReal64 min,ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_partialErasureStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistic (mean, minimum, maximum and standard deviation) of partial erasure data. The data returned by the “hpe5022_partialErasureData_Q” function is used for this statistic analysis.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- mean

  Description — Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_partialErasure_Q” function.

  Direction — OUT

- min

  Description — Returns the minimum value.

  Direction — OUT

- max

  Description — Returns the maximum value.

  Direction — OUT

- stdDev

  Description — Returns the standard deviation value.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The partial erasure data is corrupt. Check if your measurement sequence is correct.

See Also “hpe5022_partialErasureData_Q” on page 941

Side Reading Measurement Function

This section describes the functions related with side reading measurement.

hpe5022_measureSideReading

C Syntax

ViStatus hpe5022_measureSideReading(ViSession id, ViReal64 wrPos, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureSideReading(ByVal id As Long, ByVal wrPos As Double, ByVal ave As Integer) As Long

Description

This function measures the side reading. The measurement sequences are as follows:

1. Move the head to the ‘wrPos’ position.
2. Write a pattern specified by the “hpe5022_selectPattern” function for an entire track.
3. Move the head to the -‘wrPos’ position.
4. Write a pattern specified by the “hpe5022_selectPattern” function for an entire track.
5. Move the head at the center of track.
6. Erase specified by the “hpe5022_eraseType” function for an entire track.
7. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
8. Measure a Narrow Band TAA ($Erase_{TAA}$)
9. Erase the track and both side of tracks (Gaps between adjacent tracks are also erased.)
10. Move the head at the center of track.
11. Write a pattern specified by the “hpe5022_selectPattern” function for an entire track.
12. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
13. Measure a Narrow Band TAA ($Full_{TAA}$)
14. Repeat the steps 1 to 13 ‘ave’ (see parameters) times if it is more than 1.

The “hpe5022_sideReading_Q” function returns the measurement result.

Equation 3-4 Side Reading Definition

$$SideReading = ave\left(\frac{Erase_{TAA}}{Full_{TAA}}\right)$$

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- wrPos

  Description: Specifies the position for the side write.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction: IN

  Unit: Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- ave

  Description: Specifies the number of measurements for averaging.

  Direction: IN

  Values

Name	Value
hpe5022_SRR_COUN_MIN	1
hpe5022_SRR_COUN_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	A hardware error is detected on the head amplifer.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘wrPos’ or/and ‘ave’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q ” function. If it is not 0, the ‘wrPos’ range is narrowed.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand had been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected. Check Head Amplifier.

See Also

“hpe5022_selectFilter” on page 149

“hpe5022_eraseType” on page 355

“hpe5022_readTrackOffset” on page 219

“hpe5022_sideReading_Q” on page 949

hpe5022_setupSideReading

C Syntax

ViStatus hpe5022_setupSideReading(ViSession id, ViReal64 wrPos, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSideReading(ByVal id As Long, ByVal wrPos As Double, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the side reading measurement sequence to the specified test identifier. See the “hpe5022_measureSideReading” function for details of the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_sideReading_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- wrPos

  Description — Specifies the position for the side write.

  Direction — IN

  Unit — Meter

  Values — Same as ‘wrPos’ in the “hpe5022_measureSideReading” function.

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as ‘ave’ in the “hpe5022_measureSideReading” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the resolution measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'wrPos' and/or 'ave' is out of range. Check if the head offset compensation value by using "hpe5022_trackOffsetCompValue_Q" function. If it is not 0, the 'wrPos' range is narrowed.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measureSideReading" on page 944

"hpe5022_sideReading_Q" on page 949

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_sideReading_Q

C Syntax

ViStatus hpe5022_sideReading_Q(ViSession id, ViPReal64 srr);

Visual Basic Syntax

hpe5022_sideReading_Q(ByVal id As Long, ByRef srr As Double) As Long

Description

This function returns the result of the side reading measurement sequence.

Each result is the mean value of the corresponding parameter of the “hpe5022_sideReadingData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- srr

  Description — Returns the result of the side reading ratio.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSideReading” on page 944

“hpe5022_setupSideReading” on page 947

“hpe5022_sideReadingData_Q” on page 951

hpe5022_sideReadingDataSize_Q

C Syntax

ViStatus hpe5022_sideReadingDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_sideReadingDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of side reading data returned by the “hpe5022_sideReadingData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the side reading data.

  The size is the same as the value of ‘ave’ specified by the “hpe5022_measureSideReading” or “hpe5022_setupSideReading” function if the measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_sideReadingData_Q” on page 951

“hpe5022_measureSideReading” on page 944

“hpe5022_setupSideReading” on page 947

hpe5022_sideReadingData_Q

C Syntax

ViStatus hpe5022_sideReadingData_Q(ViSession id, ViReal64 data[]);

Visual Basic Syntax

hpe5022_sideReadingData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the side reading data of the measurement results.

When the ‘ave’ parameter in the “hpe5022_measureSideReading” or the “hpe5022_setupSideReading” function is set to more than one, the function allows you to get the side reading data for each revolution. If the ‘ave’ is set to 1, the output of this function will be the same as one of the “hpe5022_sideReading_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- data

  Description — Returns the side reading data in array. The array size is returned by the “hpe5022_sideReadingDataSize_Q” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSideReading” on page 944

“hpe5022_setupSideReading” on page 947

“hpe5022_sideReading_Q” on page 949

“hpe5022_sideReadingDataSize_Q” on page 950

hpe5022_sideReadingStatistic_Q

C Syntax

ViStatus hpe5022_sideReadingStatistic_Q(ViSession id, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_sideReadingStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the side reading measurement. The data returned by the “hpe5022_sideReadingData_Q” function is used for statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mean

  Description: Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_sideReading_Q” function.

  Direction: OUT

- min

  Description: Returns the minimum value.

  Direction: OUT

- max

  Description: Returns the maximum value.

  Direction: OUT

- stdDev

  Description: Returns the standard deviation value.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also “hpe5022_sideReadingData_Q” on page 951

Side Erase Measurement Function

This section describes the functions related with side erase measurement.

hpe5022_measureSideErase

C Syntax

ViStatus hpe5022_measureSideErase(ViSession id, ViReal64 erPos, ViInt16 ave);

Visual Basic Syntax

hpe5022_measureSideErase(ByVal id As Long, ByVal erPos As Double, ByVal ave As Integer) As Long

Description

This function measures the side erase. The measurement sequences are as follows:

1. Erase specified by the “hpe5022_eraseType” function for an entire track.
2. Write a pattern specified by the “hpe5022_selectPattern” function for an entire track.
3. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
4. Measure a Narrow Band TAA ($Full_{TAA}$).
5. Move the head to the ‘erPos’ position.
6. Erase an entire track.
7. Move the head to the -‘erPos’ position.
8. Erase an entire track.
9. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
10. Measure a Narrow Band TAA ($Squeeze_{TAA}$)
11. Repeat the steps 1 to 10 ‘ave’ (see parameters) times if it is more than 1.

The “hpe5022_sideErase_Q” function returns the measurement result.

Equation 3-5 Side Erase Definition

$$SideErase = ave\left(\frac{Squeeze_{TAA}}{Full_{TAA}}\right)$$

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- erPos

  Description — Specifies the position for the side erase.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction — IN

  Unit — Meter

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values

Name	Value
hpe5022_SER_COUN_MIN	1
hpe5022_SER_COUN_MAX	50

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	A hardware error is detected on the head amplifer.
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘erPos’ and/or ‘ave’ is out of range. Check if the head offset compensation value by using “hpe5022_trackOffsetCompValue_Q ” function. If it is not 0, the ‘erPos’ range is narrowed.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand had been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_HARD_HAMP	Hardware error is detected. Check Head Amplifier.

See Also

“hpe5022_readTrackOffset” on page 219

“hpe5022_readTrackOffset” on page 219

“hpe5022_userPattern” on page 140

“hpe5022_sideErase_Q” on page 959

hpe5022_setupSideErase

C Syntax

ViStatus hpe5022_setupSideErase(ViSession id, ViReal64 erPos, ViInt16 ave, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSideErase(ByVal id As Long, ByVal erPos As Double, ByVal ave As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the side erase measurement sequence to the specified test identifier. See the “hpe5022_measureSideErase” function for details of the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_sideErase_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- erPos

  Description — Specifies the position for the side erase.

  Direction — IN

  Unit — Meter

  Values — Same as ‘erPos’ in the “hpe5022_measureSideErase” function.

- ave

  Description — Specifies the number of measurements for averaging.

  Direction — IN

  Values — Same as ‘ave’ in the “hpe5022_measureSideErase” function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the resolution measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'erPos' and/or 'ave' is out of range. Check if the head offset compensation value by using "hpe5022_trackOffsetCompValue_Q" function. If it is not 0, the 'erPos' range is narrowed.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the "hpe5022_releaseSetup" function.

See Also

"hpe5022_measureSideErase" on page 954

"hpe5022_sideErase_Q" on page 959

"hpe5022_measure" on page 387

"hpe5022_releaseSetup" on page 390

hpe5022_sideErase_Q

C Syntax

ViStatus hpe5022_sideErase_Q(ViSession id, ViPReal64 ser);

Visual Basic Syntax

hpe5022_sideErase_Q(ByVal id As Long, ByRef ser As Double) As Long

Description

This function returns the result of the side erase measurement sequence.

Each result is the mean value of the corresponding parameter of the “hpe5022_sideEraseData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- ser

  Description — Returns the result of the side erase ratio.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSideReading” on page 944

“hpe5022_setupSideReading” on page 947

“hpe5022_sideEraseData_Q” on page 961

hpe5022_sideEraseDataSize_Q

C Syntax

ViStatus hpe5022_sideEraseDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_sideEraseDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of side erase data returned by the “hpe5022_sideEraseData_Q” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- size

  Description — Returns the size of the side reading data.

  The size is the same as the value of ‘ave’ specified by the “hpe5022_measureSideErase” or “hpe5022_setupSideErase” function if the measurement is done properly.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_sideEraseDataSize_Q” on page 960

“hpe5022_measureSideErase” on page 954

“hpe5022_setupSideErase” on page 957

hpe5022_sideEraseData_Q

C Syntax

ViStatus hpe5022_sideEraseData_Q(ViSession id, ViReal64 data[]);

Visual Basic Syntax

hpe5022_sideEraseData_Q(ByVal id As Long, ByRef data As Double) As Long

Description

This function returns the side erase data of the measurement results.

When the ‘ave’ parameter in the “hpe5022_measureSideErase” or the “hpe5022_setupSideErase” function is set to more than one, the function allows you to get the side reading data for each revolution. If the ‘ave’ is set to 1, the output of this function will be the same as one of the “hpe5022_sideErase_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- data

  Description: Returns the side reading data in array. The array size is returned by the “hpe5022_sideEraseDataSize_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSideErase” on page 954

“hpe5022_setupSideErase” on page 957

“hpe5022_sideErase_Q” on page 959

“hpe5022_sideEraseDataSize_Q” on page 960

hpe5022_sideEraseStatistic_Q

C Syntax

ViStatus hpe5022_sideEraseStatistic_Q(ViSession id, ViPReal64 mean, ViPReal64 min, ViPReal64 max, ViPReal64 stdDev);

Visual Basic Syntax

hpe5022_sideEraseStatistic_Q(ByVal id As Long, ByRef mean As Double, ByRef min As Double, ByRef max As Double, ByRef stdDev As Double) As Long

Description

This function reports the statistics (mean, minimum, maximum and standard deviation) of the side erase measurement. The data returned by the “hpe5022_sideEraseData_Q” function is used for statistic analysis.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- mean

  Description: Returns the mean value. The returned value is the same as the corresponding values returned by the “hpe5022_sideErase_Q” function.

  Direction: OUT

- min

  Description: Returns the minimum value.

  Direction: OUT

- max

  Description: Returns the maximum value.

  Direction: OUT

- stdDev

  Description: Returns the standard deviation value.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The measurement result data is corrupt. Check if your measurement sequence is correct.

See Also “hpe5022_sideEraseData_Q” on page 961

Track Offset Compensation Function

This section describes the programming functions of track offset compensation. This functions compensated the read head position according to the read back signal from burst patterns. See the “Track Offset Compensation” on page 75 for a sample programming.

hpe5022_trackOffsetCompProfile

C Syntax

ViStatus hpe5022_trackOffsetCompProfile(ViSession id, ViInt16 size, ViReal64 rang);

Visual Basic Syntax

hpe5022_trackOffsetCompProfile(ByVal id As Long, ByVal size As Integer, ByVal rang As Double) As Long

Description

This function defines the track profile measurement in the calibration sequence of the track offset compensation. The “hpe5022_writeTrackOffsetCompPattern” function executes the calibration sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- size

  Description: Specifies the number of measurement points for the track profile measurement in the calibration sequence.

  Direction: IN

  Values

Name	Value
hpe5022_TRACK_PROFILE_SIZE_MIN	1
hpe5022_TRACK_PROFILE_SIZE_MAX	201

  Preset Value: 51

- rang

  Description: Specifies the offset range for the track profile measurements in the calibration sequence.

  Direction: IN

Values

Name	Value
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

Preset Value 3×10^{-6}

Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘size’, and/or ‘rang’ is out of range.

See Also

“hpe5022_trackOffsetCompProfile_Q” on page 966

“hpe5022_writeTrackOffsetCompPattern” on page 976

hpe5022_trackOffsetCompProfile_Q

C Syntax

ViStatus hpe5022_trackOffsetCompProfile_Q (ViSession id, ViPInt16 size, ViPReal64 rang);

Visual Basic Syntax

hpe5022_trackOffsetCompProfile_Q(ByVal id As Long, ByRef size As Integer, ByRef rang As Double) As Long

Description

This function defines the track profile measurement in the calibration sequence of the track offset compensation.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- size

  Description: Returns the number of measurement points for the track profile measurement in the calibration sequence.

  Direction: OUT

- rang

  Description: Returns the offset range for the track profile measurements in the calibration sequence.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_trackOffsetCompProfile” on page 964

hpe5022_trackOffsetCompPattern

C Syntax

ViStatus hpe5022_trackOffsetCompPattern(ViSession id, ViReal64 freq, ViReal64 offset, ViReal64 aper);

Visual Basic Syntax

hpe5022_trackOffsetCompPattern(ByVal id As Long, ByVal freq As Double, ByVal offset As Double, ByVal aper As Double) As Long

Description

This function defines the burst pattern. It defined by frequency, the burst offset position and the aperture time. When “hpe5022_trackOffsetCompPatternAuto” function is executed, this parameters are set automatically. The “hpe5022_writeTrackOffsetCompPattern” function writes the burst pattern. See the chapter 5 in the operation manual.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Specifies the frequency of the burst pattern. The range of frequency can be queried by the “hpe5022_channelBitRateRange_Q” function.

  Direction: IN

  Preset Value: 100×10^{-6}

  Unit: flux/sec

- offset

  Description: Specifies the offset position of burst from the track center.

  If the “hpe5022_trackOffsetCompValue” function set the head offset value, the range of the offset is changed. For example, the head offset value in “hpe5022_trackOffsetCompValue” is set at 1.0×10^{-6}, the maximum and the minimum values are 5.0×10^{-6} and -7.0×10^{-6}, respectively.

  Direction: IN

  Values

Name	Value
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

  Preset Value: 1×10^{-6}

Unit Meter

- aper

  Description Specifies the aperture time of burst pattern. The burst pattern is written during this period from the index pulse. When the gate (hpe5022_seqBaseSegGateConfig and hpe5022_seqSaGateConfig) are defined in User Define Sequence, the reference start point is not index pulse but the time at gate open.

  Direction IN

  Values

Name	Value
hpe5022_TRACK_OFFSET_COMP_PAT_APER_MIN	1×10^{-3}
hpe5022_TRACK_OFFSET_COMP_PAT_APER_MAX	10×10^{-3}

  Preset Value hpe5022_TRACK_OFFSET_COMP_PAT_APER_MIN

  Unit Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freq’, ‘offset’ and/or ‘aper’ is out of range.

See Also

“hpe5022_trackOffsetCompPattern_ Q” on page 969

“hpe5022_trackOffsetCompPatternAuto” on page 974

“hpe5022_writeTrackOffsetCompPattern” on page 976

hpe5022_trackOffsetCompPattern_Q

C Syntax

ViStatus hpe5022_trackOffsetCompPattern_Q(ViSession id, ViReal64 freq, ViReal64 offset, ViReal64 aper);

Visual Basic Syntax

hpe5022_trackOffsetCompPattern_Q(ByVal id As Long, ByVal freq As Double, ByVal offset As Double, ByVal aper As Double) As Long

Description

This function returns the burst pattern. The frequency, the burst offset position and the aperture time can be determined.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- freq

  Description — Returns the frequency of the burst pattern.

  Direction — OUT

  Unit — flux/sec

- offset

  Description — Returns the offset position of burst from the track center.

  Direction — OUT

  Unit — Meter

- aper

  Description — Returns the aperture time of burst pattern. The burst pattern is writing during this period from the index pulse.

  Direction — OUT

  Unit — Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_burstPatternConfig_Q” on page 1000

“hpe5022_trackOffsetCompPattern” on page 967

hpe5022_trackOffsetCompPatternPosition

C Syntax

ViStatus hpe5022_trackOffsetCompPatternPosition (ViSession id, ViReal64 compPos);

Visual Basic Syntax

hpe5022_trackOffsetCompPatternPosition(ByVal id As Long, ByVal compPos As Double) As Long

Description

This function specifies the position of the whole burst pattern. See the chapter 5 in the operation manual.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- compPos

  Description: Specifies the offset position of the whole burst pattern from the track center.

  Direction: IN

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

  Preset Value: 0

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘compPos’ is out of range.

See Also

“hpe5022_writeTrackOffsetCompPattern” on page 976

“hpe5022_trackOffsetCompPatternPosition_Q” on page 973

hpe5022_trackOffsetCompPatternPosition_Q

C Syntax

ViStatus hpe5022_trackOffsetCompPatternPosition_Q (ViSession id, ViPReal64 compPos);

Visual Basic Syntax

hpe5022_trackOffsetCompPatternPosition_Q(ByVal id As Long, ByRef compPos As Double) As Long

Description

This function returns the position of the whole burst pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- compPost

  Description: Returns the offset position of the whole burst pattern from the track center.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_trackOffsetCompPatternPosition” on page 971

hpe5022_trackOffsetCompPatternAuto

C Syntax

ViStatus hpe5022_trackOffsetCompPatternAuto(ViSession id);

Visual Basic Syntax

hpe5022_trackOffsetCompPatternAuto(ByVal id As Long) As Long

Description

This function defines the burst pattern automatically, in other words, the parameters in “hpe5022_trackOffsetCompPattern” function is set automatically after this function is executed. The values can be queried by the “hpe5022_trackOffsetCompPattern_ Q” function. This function should be executed after the “hpe5022_trackOffsetCompProfile” function.

1. Set the aperture time for 2m sec. If the aperture time of 2m sec is over half of the track, the aperture time set at half of the track.
2. Perform a narrow band TAA roll off measurement from channel bit rate 50Mbps to 250Mbps (step: 10Mbps) and find the frequency at the peak narrow band TAA. The frequency is set as a burst pattern frequency.
3. Perform a track profile measurement and find the write track width. The offset position of burst pattern is set at the half value of the write track width.

   The parameters of track profile measurement are:
   Range: -rang to +rang defined by “hpe5022_trackOffsetCompProfile”
   Points: size defined by “hpe5022_trackOffsetCompProfile”
   Pattern: The frequency defined by step 2 and its pattern is 1T.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand had been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_trackOffsetCompPattern” on page 967

“hpe5022_trackOffsetCompPattern_ Q” on page 969

“hpe5022_trackOffsetCompProfile” on page 964

hpe5022_writeTrackOffsetCompPattern

C Syntax

ViStatus hpe5022_writeTrackOffsetCompPattern(ViSession id);

Visual Basic Syntax

hpe5022_writeTrackOffsetCompPattern(ByVal id As Long) As Long

Description

This function executes the calibration sequence and writes the burst pattern.

- Calibration Sequence
  1. Reset the offset value (The value is retuned by the “hpe5022_trackOffsetCompValue” function) at 0.
  2. Perform a band erase according to the offset range and points specified by the “hpe5022_trackOffsetCompProfile” function.
  3. Move the head to the -’offset’ position specified by the “hpe5022_trackOffsetCompPattern” function.
  4. Write a 1T pattern at the frequency specified by the “hpe5022_trackOffsetCompPattern” function.
  5. Perform a track profile measurement (TAAa) according to the offset range and points specified by the “hpe5022_trackOffsetCompProfile” function.
  6. Perform a band erase according to the offset range and points specified by the “hpe5022_trackOffsetCompProfile” function.
  7. Move the head to the +’offset’ position specified by the “hpe5022_trackOffsetCompPattern” function.
  8. Write a 1T pattern at the frequency specified by the “hpe5022_trackOffsetCompPattern” function.
  9. Perform a track profile measurement (TAAb) according to the offset range and points specified by the “hpe5022_trackOffsetCompProfile” function.
  10. Calculate the position error signal (PES) from TAAa and TAAb and approximates a PES line.
- Write Burst Pattern Sequence
  1. Perform a band erase according to the offset range and points specified by the “hpe5022_trackOffsetCompProfile” function.
  2. Write burst patterns specified by the “hpe5022_trackOffsetCompPattern” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand had been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_trackOffsetCompProfile” on page 964

“hpe5022_trackOffsetCompPattern” on page 967

“hpe5022_trackOffsetCompPatternAuto” on page 974

hpe5022_trackOffsetCompPatternState

C Syntax

ViStatus hpe5022_trackOffsetCompPatternState(ViSession id, ViBoolean stat);

Visual Basic Syntax

hpe5022_trackOffsetCompPatternState(ByVal id As Long, ByVal stat As Integer) As Long

Description

This function activates the Track offset Compensation function. When this function is set at TURE state, the data pattern is written and read besides the burst pattern area.

A data pattern are written after the burst pattern during a write sequence in a measurement sequence such as hpe5022_measureTAA (If hpe5022_SEQ_ER_WR_M or hpe5022_SEQ_WR_M is selected). And, the burst patterns are ignored in measuring a parameter such as TAA during a read sequence in a measurement sequence.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- Stat

  Description: Selects the track offset compensation pattern state.

  Direction: IN

  Values

Name	Value	Description
VI_TRUE	1	Track Offset Compensation function is activated
VI_FALSE	0	Track Offset Compensation function is inactivated.

  Preset Value: VI_FALSE

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.

See Also

“hpe5022_trackOffsetCompPatternState_Q” on page 980

“hpe5022_executeTrackOffsetComp” on page 983

hpe5022_trackOffsetCompPatternState_Q

C Syntax

ViStatus hpe5022_trackOffsetCompPatternState_Q(ViSession id, ViPBoolean stat);

Visual Basic Syntax

hpe5022_trackOffsetCompPatternState_Q(ByVal id As Long, ByRef stat As Integer) As Long

Description

This function returns the track offset compensation pattern state.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- stat

  Description — Returns the track offset compensation pattern state.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.

See Also

“hpe5022_trackOffsetCompPatternState” on page 978

hpe5022_trackOffsetCompRevolution

C Syntax

ViStatus hpe5022_trackOffsetCompRevolution (ViSession id, ViInt16 rev);

Visual Basic Syntax

hpe5022_trackOffsetCompRevolution(ByVal id As Long, ByVal rev As Integer) As Long

Description

This function specifies the average counts to measure a position error signal (PES) in the “hpe5022_executeTrackOffsetComp” function. The burst pattern is measured for revolutions specified by this parameter and calculates a PES from the averaged result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rev

  Description: Specifies the average counts to measure a position error signal (PES). The burst pattern is measured for revolutions specified by this parameter and calculates a PES from the averaged result.

  Direction: IN

  Values

Name	Value
hpe5022_TRACK_OFFSET_COMP_REV_MIN	1
hpe5022_TRACK_OFFSET_COMP_REV_MAX	50

  Preset Value: hpe5022_TRACK_OFFSET_COMP_REV_MIN

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘rev’ is out of range.

See Also

“hpe5022_executeTrackOffsetComp” on page 983

hpe5022_trackOffsetCompRevolution_Q

C Syntax

ViStatus hpe5022_trackOffsetCompRevolution _Q(ViSession id, ViPInt16 rev);

Visual Basic Syntax

hpe5022_trackOffsetCompRevolution_Q(ByVal id As Long, ByRef rev As Integer) As Long

Description

This function returns the average counts to measure a position error signal (PES) in the “hpe5022_executeTrackOffsetComp” function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- rev

  Description — Returns the average counts to measure a position error signal (PES).

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.

See Also

“hpe5022_trackOffsetCompRevolution” on page 981

hpe5022_executeTrackOffsetComp

C Syntax

ViStatus hpe5022_executeTrackOffsetComp(ViSession id);

Visual Basic Syntax

hpe5022_executeTrackOffsetComp(ByVal id As Long) As Long

Description

This function reads the burst pattern to measure position error signal and changes the track offset value. In other words, the track offset is compensated automatically during a measurement. The “hpe5022_trackOffsetCompValue_Q” function returns the track offset value. Before this function is executed, the “hpe5022_trackOffsetCompPatternState” function must be set at TURE state and “hpe5022_writeTrackOffsetCompPattern” also must be done in order to write the burst patterns.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand had been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function

See Also

“hpe5022_trackOffsetCompPatternState” on page 978

“hpe5022_trackOffsetCompValue_Q” on page 986

hpe5022_trackOffsetCompValue

C Syntax

ViStatus hpe5022_trackOffsetCompValue(ViSession id, ViReal64 offset);

Visual Basic Syntax

hpe5022_trackOffsetCompValue(ByVal id As Long, ByVal offset As Double) As Long

Description

This function gives the offset for the origin of whole data area to compensate a drift of track position. The positions for all tracks are shifted by this compensation. After the “hpe5022_executeTrackOffsetComp” function is executed, the offset value is changed automatically. In other hand, this function allows you to set the track offset manually. It means this function is used only when user wants to change track offset manually. This function affects the range of the track offset. For example, this is set at 1.0×10^{-6}, the maximum and the minimum values in the track offset functions such as “hpe5022_readTrackOffset” are 5.0×10^{-6} and -6.0×10^{-6}, respectively. See the track offset compensation in the chapter 5 of the operation manual.

This offset value is reset at 0 when the “hpe5022_writeTrackOffsetCompPattern” function is executed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Specifies the track compensated position.

  Direction: IN

  Values

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6.0×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6.0×10^{-6}

  Preset Value: 0

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘offset’ is out of range.

See Also

“hpe5022_writeTrackOffsetCompPattern” on page 976

“hpe5022_executeTrackOffsetComp” on page 983

“hpe5022_trackOffsetCompValue_Q” on page 986

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_writeMultiple” on page 330

“hpe5022_eraseMultiple” on page 364

“hpe5022_measureTrackProfile” on page 615

“hpe5022_tripleTrackPositionConfig” on page 707

“hpe5022_measureTripleTrack” on page 714

hpe5022_trackOffsetCompValue_Q

C Syntax

ViStatus hpe5022_trackOffsetCompValue_Q(ViSession id, ViPReal64 offset);

Visual Basic Syntax

hpe5022_trackOffsetCompValue_Q(ByVal id As Long, ByRef offset As Double) As Long

Description

This function returns the position compensation distance. After the “hpe5022_executeTrackOffsetComp” function is executed, this position compensation is executed automatically. This function returns how long it is compensated.

This value is reset at 0 when the “hpe5022_writeTrackOffsetCompPattern” function is executed.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- offset

  Description: Returns the position compensation distance.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.

See Also

“hpe5022_executeTrackOffsetComp” on page 983

“hpe5022_trackOffsetCompValue” on page 984

“hpe5022_writeTrackOffsetCompPattern” on page 976

hpe5022_measureTrackOffsetPositionError

C Syntax

ViStatus hpe5022_measureTrackOffsetPositionError(ViSession id);

Visual Basic Syntax

hpe5022_measureTrackOffsetPositionError(ByVal id As Long) As Long

Description

This function reads the burst pattern to measure the position error signal (PES). The “hpe5022_executeTrackOffsetComp” function measures the PES and compensates the head position according to the values. In the other hand, this function only measures the PES and does not change the head position. Before this function is executed, the “hpe5022_trackOffsetCompPatternState” function must be set at TURE state and “hpe5022_writeTrackOffsetCompPattern” also must be done in order to write the burst patterns.

The “hpe5022_trackOffsetPositionError_Q” function returns the result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_CAL_DATA_CORRUPT	Amplitude calibration for 3GHz Spectrum Analyzer has not been done.
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_CAL_DATA_CORRUPT	The burst pattern is not written. Execute “hpe5022_writeTrackOffsetCompPattern” before this function.
hpe5022_ERROR_PES_DATA	The current head position is far from the center of the burst pattern.

See Also

“hpe5022_trackOffsetPositionError_Q” on page 989

“hpe5022_trackOffsetCompPatternState” on page 978

“hpe5022_writeTrackOffsetCompPattern” on page 976

“hpe5022_executeTrackOffsetComp” on page 983

hpe5022_trackOffsetPositionError_Q

C Syntax

ViStatus hpe5022_trackOffsetPositionError_Q(ViSession id, ViPReal64 position);

Visual Basic Syntax

hpe5022_trackOffsetPositionError_Q(ByVal id As Long, ByRef position As Double) As Long

Description

This function returns the position measured by the “hpe5022_measureTrackOffsetPositionError” function.

If the position error exceeds approximately one-third of the write track width, the “hpe5022_ERROR_DATA_CORRUPT” error may be returned. This is because the position error can monitor within the range that the Position Error Signal (PES) is linear.

When the error is returned though the position error is within the one-third of the write track width, increase the ‘size’ or reduce the ‘rang’ of the “hpe5022_trackOffsetCompProfile” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- position

  Description: Returns the head position from the track center.

  Direction: OUT

  Unit: Meter

Return Values

Completion Code	Description
VI_SUCCESS	No error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified is invalid.
hpe5022_ERROR_DATA_CORRUPT	Data is corrupt. Execute “hpe5022_measureTrackOffsetPositionError” before this function.

See Also

“hpe5022_measureTrackOffsetPositionError” on page 987

“hpe5022_trackOffsetCompPatternPosition” on page 971

“hpe5022_trackOffsetCompProfile” on page 964

Track Search Function

This section describes the functions of a track search. This function searches a track in the specified area by making a track profile measurement.

hpe5022_searchTrack

C Syntax

ViStatus hpe5022_searchTrack(ViSession id, ViReal64 range, ViReal64 pitch);

Visual Basic Syntax

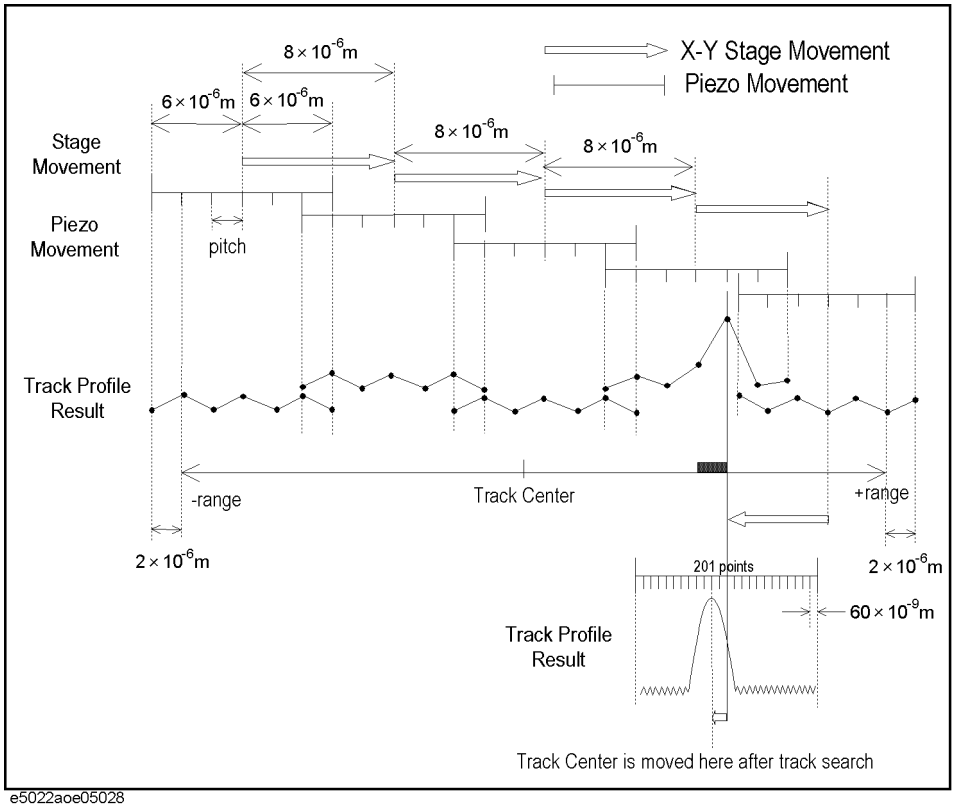
hpe5022_searchTrack(ByVal id As Long, ByVal range As Double, ByVal pitch As Double) As Long

Description

This function searches a track by performing a track profile measurement and move the current head positions at a peak of the track profile waveform.

1. Move a track center position at the -(range)+4×10^{-6}[m] by actuating X-Y stages.
2. Measure TAA track profile
   (same as hpe5022_measureTrackProfile:
   seqType: hpe5022_SEQ_M
   mode: hpe5022_TP_FORWARD
   measFunc: hpe5022_MEAS_TAA
   points: mod(12×10^{-6}/'pitch'+1)
   offset pitch: 'pitch')
3. Move a track center position by +8×10^{-6}[m] by actuating X-Y stages.
4. Repeat the step 2 and 3 until the track center position reaches the (+range).
5. Move the head at the position where the TAA is maximum by actuating X-Y stages.
6. Measure TAA track profile
   (same as hpe5022_measureTrackProfile:
   seqType: hpe5022_SEQ_M
   mode: hpe5022_TP_FORWARD
   measFunc: hpe5022_MEAS_TAA
   points: 201
   offset pitch: 60×10^{-9})
7. Move the head at the position where the TAA is maximum, by actuating X-Y stages.

Figure 3-37 **Track Search**



Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- range

  Description — Specifies a range for searching a peak. When the head position is into the inhibit area or beyond the media size, the error message is displayed.

  Direction — IN

  Values

Name	Value
hpe5022_DISK_RANGE_MAX	47.5×10^{-3}

  Unit — Meter

- pitch

  Description — Specifies an approximate write track width. This value is

used for the pitch of the first rough track profile measurement in order to find the track. See Figure 3-37.

Direction IN

Values

Name	Value
hpe5022_SEARCH_TRACK_PITCH_MIN	60×10^{-9}
hpe5022_SEARCH_TRACK_PITCH_MAX	6×10^{-6}

Unit Meter

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘range’ or/and ‘pitch’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_SEARCH_TRACK_FAILED	The level of input signal in a track profile measurement during the track search is not properly.

See Also

Write Bursts Function

This section describes the programming functions of write-bursts. This function is only used when users wants to only write burst patterns. When user wants to compensate the head position, use “Track Offset Compensation Function”. This function includes to write a burst pattern automatically. See “Track Offset Compensation Function” on page 964.

hpe5022_writeBurst

C Syntax

ViStatus hpe5022_writeBurst(ViSession id, ViInt16 segment, ViReal64 offset);

Visual Basic Syntax

hpe5022_writeBurst(ByVal id As Long, ByVal segment As Integer, ByVal offset As Double) As Long

Description

This function writes the burst pattern. When this function is executed, 1T pattern will be written at a burst length specified by the “hpe5022_burstPatternConfig” and at a frequency specified by “hpe5022_burstFrequencyConfig”.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- segment

  Description — Specifies the number of segments per track.

  Direction — IN

  Values — Specifies the maximum and minimum segments per track.

Name	Value
hpe5022_BURST_SEG_PER_TRACK_MIN	1
hpe5022_BURST_SEG_PER_TRACK_MAX	128

Timing Limitation
segmTime = 60 / RPM/ segment
RPM = spindle speed, when spindle speed ≥ 2400 rpm
RPM = 2400, when spindle speed < 2400 rpm
(segmTime - (gateDelay + gateAper)) ≥ 10×10^{-6}

- offset

  Description Specifies the write offset position from the track center.

  Direction IN

  Unit Meter

  Values Specifies the maximum and minimum write offset values.

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘segment’, and/or ‘offset’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_burstPatternConfig” on page 997

“hpe5022_burstGateConfig” on page 1004

hpe5022_setupWriteBurst

C Syntax

ViStatus hpe5022_setupWriteBurst(ViSession id, ViInt16 segment, ViReal64 offset, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupWriteBurst(ByVal id As Long, ByVal segment As Integer, ByVal offset As Double, ByRef testHndl As Long) As Long

Description

This function sets the write burst sequence to the test identifier. Refer to the “hpe5022_measure” function for details about the sequence. This function does not execute measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- segment

  Description — Specifies the number of segments per track.

  Direction — IN

  Values — Specifies the maximum and minimum segments per track.

Name	Value
hpe5022_BURST_SEG_PER_TRACK_MIN	1
hpe5022_BURST_SEG_PER_TRACK_MAX	128

- offset

  Description — Specifies the write offset position from the track center.

  Direction — IN

  Unit — Meter

  Values — Specifies the maximum and minimum write offset values.

Name	Value
hpe5022_TRACK_OFFSET_MIN	-6×10^{-6}
hpe5022_TRACK_OFFSET_MAX	6×10^{-6}

- testHndl

  Description — Returns the test identifier. This identifier is used to execute write burst using the “hpe5022_measure” function.

Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘segment’, and/or ‘offset’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measure” on page 387

hpe5022_burstPatternConfig

C Syntax

ViStatus hpe5022_burstPatternConfig(ViSession id, ViReal64 burstStart, ViReal64 burstLength);

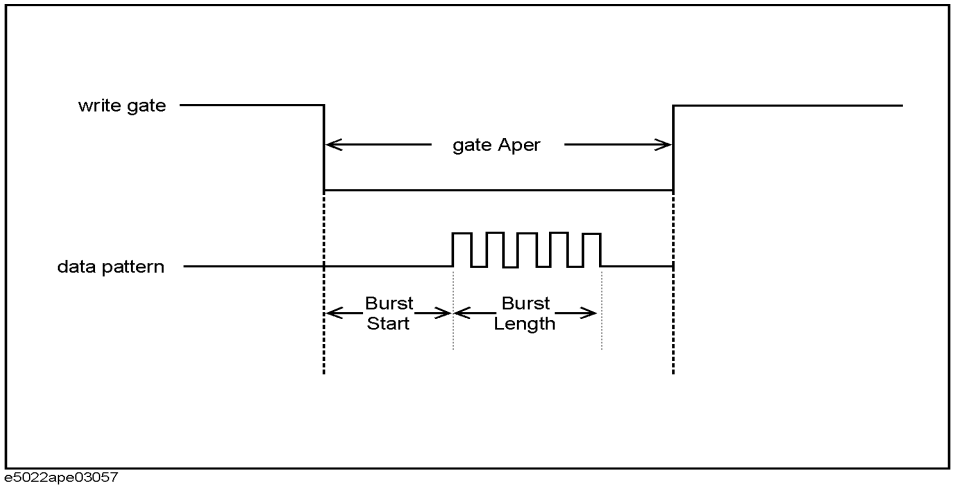
Visual Basic Syntax

hpe5022_burstPatternConfig(ByVal id As Long, ByVal burstStart As Double, ByVal burstLength As Double) As Long

Description

This function controls the timing sequence of the burst pattern.

Figure 3-38 Write Timing of Burst Pattern



NOTE

Due to internal constraints of E5037A (data generator module), data writing is actually delayed by a value of **30× 16T** (T=1/ f, where f is the bit rate of the burst pattern, so T is 1/ bps). Thus, the user must compensate this value when you specify the burst length. Refer to Figure 3-88 of the operation manual for details.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- burstStart

  Description: Specifies the time distance between start of the write gate and burst start.

  Direction: IN

  Preset Value: hpe5022_BURST_START_MIN

  Unit: Second

  Values: Specifies the maximum and minimum time to start writing

the burst pattern.

Name	Value
hpe5022_BURST_START_MIN	0
hpe5022_BURST_START_MAX	100×10^{-6}

- burstLength

  Description — Specifies the length of time to write the burst pattern. The 'burstStart' and 'burstLength' parameters must be specified within the parameter limits of "gateAper" in the "hpe5022_burstGateConfig" function.

Write Burst Limit
hpe5022_BURST_APER_MIN ≤ (**burstStart + burstLength**) ≤ hpe5022_BURST_APER_MAX

  Direction — IN

  Preset Value — hpe5022_BURST_LENGTH_MIN

  Unit — Second

  Values — Specifies the maximum and minimum time to write the burst pattern.

Name	Value
hpe5022_BURST_LENGTH_MIN	0
hpe5022_BURST_LENGTH_MAX	100×10^{-6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'burstStart', and/or 'burstLength' is out of range.

See Also

"hpe5022_burstPatternConfig_Q" on page 1000

"hpe5022_burstGateConfig" on page 1004

hpe5022_burstPatternConfig_Q

C Syntax

ViStatus hpe5022_burstPatternConfig_Q(ViSession id, ViPReal64 burstStart, ViPReal64 burstLength);

Visual Basic Syntax

hpe5022_burstPatternConfig_Q(ByVal id As Long, ByRef burstStart As Double, ByRef burstLength As Double) As Long

Description

This function returns the specified timing sequence of the burst pattern.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- burstStart

  Description — Returns the time distance between the start of the write gate and burst start.

  Unit — Second

  Direction — OUT

- burstLength

  Description — Returns the length of time to write the burst pattern.

  Unit — Second

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_burstPatternConfig” on page 997

“hpe5022_burstGateConfig” on page 1004

hpe5022_burstFrequencyConfig

C Syntax

ViStatus hpe5022_burstFrequencyConfig(ViSession id, ViReal64 freq);

Visual Basic Syntax

hpe5022_burstFrequencyConfig(ByVal id As Long, ByVal freq As Double) As Long

Description

This function controls the frequency of the burst pattern.

1T pattern is always used in Burst write.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Specifies the bit rate of the burst pattern. The upper and lower limits of the frequency can be queried by the function “hpe5022_channelBitRateRange_Q” on page 118.

  Direction: IN

  Preset Value: 100×10^6

  Unit: bps

  Values: Specifies the maximum and minimum frequencies of the burst pattern.

Name	Value
Minimum Frequency	Use the “hpe5022_channelBitRateRange_Q ” function to know the minimum frequency.
Maximum Frequency	Use the “hpe5022_channelBitRateRange_Q ” function to know the maximum frequency.

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freq’ is out of range.

See Also

“hpe5022_burstFrequencyConfig_Q” on page 1003

hpe5022_burstFrequencyConfig_Q

C Syntax

ViStatus hpe5022_burstFrequencyConfig_Q(ViSession id, ViPReal64 freq);

Visual Basic Syntax

hpe5022_burstFrequencyConfig_Q(ByVal id As Long, ByRef freq As Double) As Long

Description

This function returns the specified frequency of the burst pattern.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Returns the specified frequency of the burst pattern.

  Unit: Second

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_burstFrequencyConfig” on page 1001

hpe5022_burstGateConfig

C Syntax

ViStatus hpe5022_burstGateConfig(ViSession id, ViReal64 gateDelay, ViReal64 gateAper);

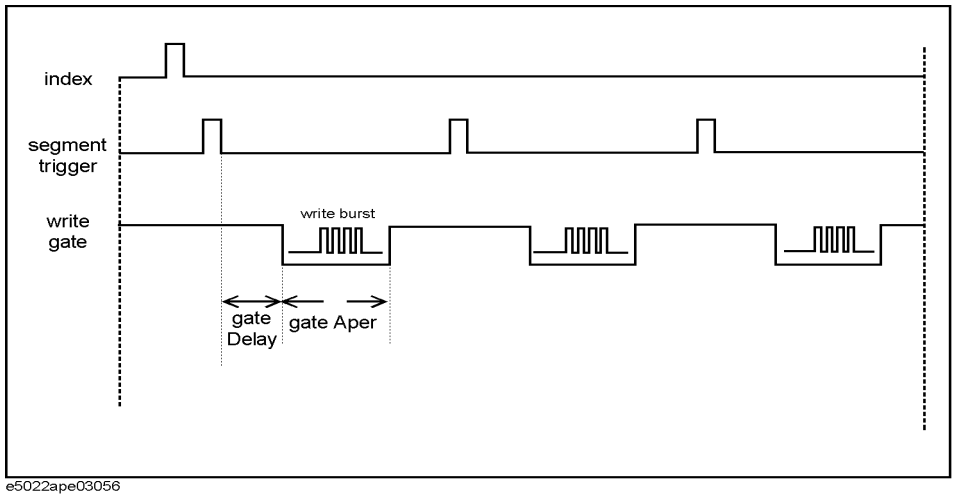
Visual Basic Syntax

hpe5022_burstGateConfig(ByVal id As Long, ByVal gateDelay As Double, ByVal gateAper As Double) As Long

Description

This function controls the timing of the write gate.

Figure 3-39 Write Gate Timing



Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- gateDelay

  Description — Specifies the time distance between segment trigger and write gate.

  Direction — IN

  Preset Value — hpe5022_BURST_DEL_MIN

  Unit — Second

  Values — Specifies the maximum and minimum amount of delay time before opening the write gate.

Name	Value
hpe5022_BURST_DEL_MIN	0
hpe5022_BURST_DEL_MAX	10×10^{-3}

- gateAper

  Description — Specifies the length of time to open the write gate. The 'burstStart' and 'burstLength' parameters of the "hpe5022_burstPatternConfig" function must be specified within the parameter limits of 'gateAper'.

Write Burst Limit
hpe5022_BURST_APER_MIN ≤ (**burstStart + burstLength**) ≤ hpe5022_BURST_APER_MAX

  Direction — IN

  Preset Value — hpe5022_BURST_APER_MIN

  Unit — Second

  Values — Specifies the maximum and minimum length of time to open the write gate.

Name	**Value**
hpe5022_BURST_APER_MIN	1×10^{-6}
hpe5022_BURST_APER_MAX	100×10^{-6}

Timing Limitation
segmTime = 60 / Spindle Speed [rpm] / segment
RPM = spindle speed , when spindle speed ≥ 2400 rpm
RPM = 2400, when spindle speed < 2400 rpm
(segmTime - (gateDelay + gateAper)) ≥ 10×10^{-6}

Return Values

Completion Code	**Description**
VI_SUCCESS	No Error

Error Code	**Description**
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'gateDelay', and/or 'gateLength' is out of range.

See Also “hpe5022_burstGateConfig_Q” on page 1007

hpe5022_burstGateConfig_Q

C Syntax

ViStatus hpe5022_burstGateConfig_Q(ViSession id, ViPReal64 gateDelay, ViPReal64 gateAper);

Visual Basic Syntax

hpe5022_burstGateConfig_Q(ByVal id As Long, ByRef gateDelay As Double, ByRef gateAper As Double) As Long

Description

This function returns the specified timing of the write gate.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- gateDelay

  Description — Returns the length of time between segment trigger and write gate.

  Direction — OUT

  Unit — Second

- gateAper

  Description — Returns the length of time to open the write gate.

  Direction — OUT

  Unit — Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_burstGateConfig” on page 1004

Auto Alignment Function

This section describes the programming functions of auto alignment. Auto alignment function is used to determine the center position of the spindle. Once the auto alignment is completed, the alignment data will be saved in the non volatile memory of the spinstand. This data contains each individual data of the up face and down face HGAs.

After the disk clamp replacement, it is necessary to perform the auto alignment.

hpe5022_autoAlignmentAssumeRadius

C Syntax

ViStatus hpe5022_autoAlignmentAssumeRadius(ViSession id, ViReal64 radius);

Visual Basic Syntax

hpe5022_autoAlignmentAssumeRadius(ByVal id As Long, ByVal radius As Double) As Long

Description

This function configures the radius of the head position when the auto alignment starts. The “hpe5022_autoAlignment” function assumes that the head is on the specified radius. After finishing the rough tune (skew = 0° position search) in “hpe5022_autoAlignment” function, the head is moved to the assumed radius position, then fine tune is performed. See “Measurement Definition” chapter of “Operation Manual” for details about the auto alignment algorithm.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- radius

  Description — Specifies the radius when the auto alignment starts. Default value depends on the disk type.

Disk Type	Default Assumed Radius
1.8 inch	0.018 m
2.5 inch	0.0255 m
3.5 inch, 3 inch	0.036 m

  Valid value is inside the data area specified by “hpe5022_dataArea” function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘radius’ is out of range.

See Also

“hpe5022_autoAlignmentAssumeRadius_Q” on page 1010

hpe5022_autoAlignmentAssumeRadius_Q

C Syntax

ViStatus hpe5022_autoAlignmentAssumeRadius_Q(ViSession id, ViPReal64 radius);

Visual Basic Syntax

hpe5022_autoAlignmentAssumeRadius_Q(ByVal id As Long, ByRef radius As Double) As Long

Description

This function returns the assumed radius when the auto alignment starts. See “Measurement Definision” chapter of “Operation Manual” for detailes about the auto alignment algorithm.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- radius

  Description: Returns the assumed radius when the auto alignment starts.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_autoAlignmentAssumeRadius” on page 1008

hpe5022_autoAlignmentFineTuneSkew

C Syntax

ViStatus hpe5022_autoAlignmentFineTuneSkew(ViSession id, ViInt16 numSkews, const ViReal64 skew1[], const ViReal64 skew2[]);

Visual Basic Syntax

hpe5022_autoAlignmentAssumeFineTuneSkew(ByVal id As Long, ByVal numSkews As Integer, ByRef skew1 as Double, ByRef skew2 As Double) As Long

Description

This function configures a list of skew angle pairs for auto alignment fine tune. The fine tune (except for “fine tune first step” at 0°, +3°, –3°) can be performed by specified times defined by numSkews. For each fine tune step, the coordinate of the spindle center is calculated using the 3 points, skew = 0 [°], skew1[ ] (positive skew angle), skew2[ ] (negative skew angle) on the track. See “Measurement Definition” chapter of “Operation Manual” for details about the auto alignment algorithm.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- numSkews

  Description: Specifies the number of skew angles in skew1 and skew2.

  Direction: IN

  Values

Name	Value
hpe5022_AUTO_ALIGNMENT_SKEW_LIST_MIN	1
hpe5022_AUTO_ALIGNMENT_SKEW_LIST_MAX	10

  Preset Value: 2

- skew1, skew2

  Description: Specifies the skew angle pairs (positive and negative). Skew = 0°, skew1 and skew2 points are used to determine the spindle center coordinates.

  Direction: IN

  Values: .

Parameter	Name	Value
skew1	hpe5022_SKEW_ANGLE_MAX	30.0
	hpe5022_AUTO_ALIGNMENT_SKEW1_MIN	3.0

Parameter	Name	Value
skew2	hpe5022_AUTO_ALIGNMENT_SKEW2_MAX	–3.0
	hpe5022_SKEW_ANGLE_MIN	–60.0

Minimum limit of skew2 angle depend on cassettes specified by “hpe5022_hgaCassette” and HGA specified by “hpe5022_hgaDimension”. See Note in “hpe5022_headPosition” on page 241.

Preset Value

i	skew1	skew2
0	10.0	–10.0
1	20.0	–20.0

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘numSkews,’ ‘skew1,’ or ‘skew2’ is out of range.

See Also

“hpe5022_autoAlignmentFineTuneSkew_Q” on page 1013

hpe5022_autoAlignmentFineTuneSkew_Q

C Syntax

ViStatus hpe5022_autoAlignmentFineTuneSkew_Q(ViSession id, ViPInt16 numSkews, ViReal64 skew1[], ViReal64 skew2[]);

Visual Basic Syntax

hpe5022_autoAlignmentAssumeFineTuneSkew_Q(ByVal id As Long, ByRef numSkews As Integer, ByRef skew1 as Double, ByRef skew2 As Double) As Long

Description

This function returns a list of skew angle pairs for auto alignment fine tune. See “Measurement Definision” chapter of “Operation Manual” for detailes about the auto alignment algorithm.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- numSkews

  Description: Returns the number of skew angles in skew1 and skew2.

  Direction: OUT

- skew1, skew2

  Description: Return the skew angle pairs to calculate the coordinate of spindle center.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

See Also

“hpe5022_autoAlignmentFineTuneSkew” on page 1011

hpe5022_autoAlignment

C Syntax

ViStatus hpe5022_autoAlignment(ViSession id);

Visual Basic Syntax

hpe5022_autoAlignment(ByVal id As Long) As Long

Description

This function executes auto alignment to get the coordinate of spindle center, then the coordinate is stored into the spinstand non-volatile memory.

See “Measurement Definition” chapter of “Operation Manual” for details about the auto alignment algorithm.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_PEAK_NOT_FOUND	Could not find the peak data while searching track.
hpe5022_ERROR_FILE_ACCESS	Failed to make log file.
VI_ERROR_RSRC_NFOUND	Spinstand currently in use is not supported.
hpe5022_ERROR_FUNCTION_NOT_LOADED	Failed to load the DLL files.
hpe5022_ERROR_NSUP_CONF	Spinstand module has not been opened.

See Also

“hpe5022_autoAlignmentAssumeRadius” on page 1008

“hpe5022_autoAlignmentFineTuneSkew” on page 1011

hpe5022_autoAlignmentCurrentState_Q

C Syntax

ViStatus hpe5022_autoAlignmentCurrentState_Q(ViSession id, ViPString state);

Visual Basic Syntax

hpe5022_autoAlignmentCurrentState_Q(ByVal id As Long, ByVal state As String) As Long

Description

This function returns the string that shows “what is going on” inside the auto alignment. The returned string is:

Preprocessing

Rough tune

Fine tune first step

Fine tune at skew = 0, x, y

Storing data

Finished

Error occurred

See “Operation Manual” for details about the auto alignment algorithm and status messages.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- state

  Description Returns the current status of auto alignment.

  Direction OUT

  Values Maximum number of characters:

Name	Value
hpe5022_MESSAGE_LENGTH_MAX	512

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The last measurement is not auto alignment.

See Also “hpe5022_autoAlignment” on page 1014

4 Function Reference for Measurement using Oscillocope

This chapter describes the functions related with the measurement by waveform analysis. Waveform measurement is performed by an oscilloscope. If an oscilloscope is not installed in your system, this function cannot be used.

The following measurements can be measured using waveform analysis.

- TAA (Track Averaged Amplitude)
- PW (Pulse Width)
- Baseline
- Time Asymmetry
- NLTS by the Dipulse Extraction method
- NLTS by the Time Correlation method

In the waveform measurement, the measurement sequence is slightly different from other measurement sequence. Such differences in sequence are as follows:

1. Set up the required parameters
    - Over sampling Rate
    - Delay Time
    - Averaging Factor
2. Acquisition of a waveform
3. Calculate the measurement parameter (listed above)

Setup Function

hpe5022_waveOverSampleRate

C Syntax

ViStatus hpe5022_waveOverSampleRate(ViSession id, ViInt32 rate);

Visual Basic Syntax

hpe5022_waveOverSampleRate(ByVal id As Long, ByVal rate As Long) As Long

Description

This function specifies the over-sample rate for the waveform analysis. The over-sample rate defines the sampling frequency of an oscilloscope. The sampling frequency of the oscilloscope is set at the value of (channel bit rate) × (over-sample rate). When the oscilloscope can not be set at an exact value, the closest and higher value is set as the sampling frequency of the oscilloscope.

For example: the channel bit rate is 100 Mbps and the over-sampling rate is set at 4, the sampling frequency of the oscilloscope is set at 400 MHz.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rate

  Description: Specifies the over-sample rate.

  Direction: IN

  Preset value: 4

  Values

Name	Value
hpe5022_OVER_SAMP_RATE_MIN	4

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘rate’ is out of range.

Error Code	Description
hpe5022_ERROR_NO_OPTION	An option for oscilloscope is not installed. This function can only be used if the system is installed with an oscilloscope.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_measureWave” on page 1028

“hpe5022_setupWave” on page 1031

“hpe5022_waveOverSampleRate_Q” on page 1021

hpe5022_waveOverSampleRate_Q

C Syntax

ViStatus hpe5022_waveOverSampleRate_Q(ViSession id, ViPInt32 rate);

Visual Basic Syntax

hpe5022_waveOverSampleRate_Q(ByVal id As Long, ByRef rate As Long) As Long

Description

This function returns the current over-sample rate setting.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rate

  Description: Returns the over-sample rate.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveOverSampleRate” on page 1019

hpe5022_waveAverage

C Syntax

ViStatus hpe5022_waveAverage(ViSession id, ViInt32 average);

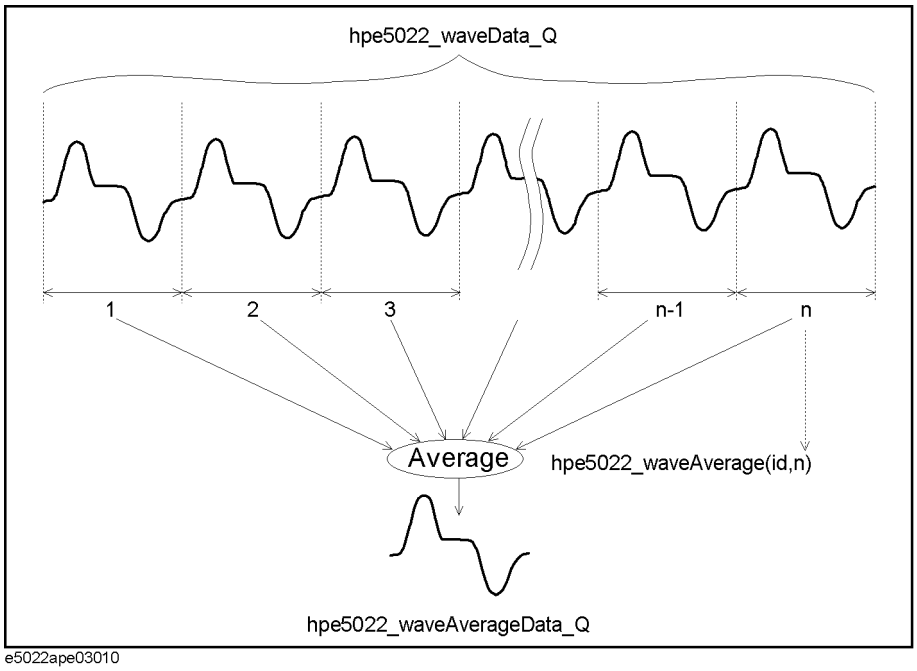
Visual Basic Syntax

hpe5022_waveAverage(ByVal id As Long, ByVal average As Long) As Long

Description

This function specifies the averaging factor of the waveform measurement. The oscilloscope takes a waveform for the number of periods of the data pattern which is specified by this function. Then, it averages the waveform.

Figure 4-1 Waveform Averaging



e5022ape03010

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- average

  Description: Specifies the number of waveform period for measurement averaging.

  Direction: IN

  Preset value: 20

Values

Name	Value
hpe5022_WAVE_AVERAGE_MIN	3

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘average’ is out of range.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_measureWave” on page 1028

“hpe5022_setupWave” on page 1031

“hpe5022_waveAverage_Q” on page 1024

hpe5022_waveAverage_Q

C Syntax

ViStatus hpe5022_waveAverage_Q(ViSession id, ViPInt32 average);

Visual Basic Syntax

hpe5022_waveAverage_Q(ByVal id As Long, ByRef average As Long) As Long

Description

This function returns the number of averaging factor.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- average

  Description: Returns the number of waveform period for measurement averaging.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveAverage” on page 1022

hpe5022_waveDelayTime

C Syntax

ViStatus hpe5022_waveDelayTime(ViSession id, ViReal64 time);

Visual Basic Syntax

hpe5022_waveDelayTime(ByVal id As Long, ByVal time As Double) As Long

Description

This function sets the delay time for the acquisition of waveform from the spinstand index pulse.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- time

  Description: Specifies a delay time from the index pulse.

  Direction: IN

  Preset value: 200×10^{-6}

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_measureWave” on page 1028

“hpe5022_setupWave” on page 1031

“hpe5022_waveDelayTime_Q” on page 1027

hpe5022_waveDelayTime_Q

C Syntax

ViStatus hpe5022_waveDelayTime_Q(ViSession id, ViPReal64 time);

Visual Basic Syntax

hpe5022_waveDelayTime_Q(ByVal id As Long, ByRef time As Double) As Long

Description

This function returns the delay time from the index pulse for the acquisition of waveform.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- time

  Description: Returns a delay time from the index pulse.

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveDelayTime” on page 1025

Data Acquisition Function

hpe5022_measureWave

C Syntax

ViStatus hpe5022_measureWave(ViSession id, ViInt16 seqType);

Visual Basic Syntax

hpe5022_measureWave(ByVal id As Long, ByVal seqType As Integer) As Long

Description

This function digitizes the waveform of a read out signal. Since the required data pattern is different among measurement parameters, the suitable data pattern must be set in advance. The sequences of this function are as follows:

1. Move the head at the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If ‘seqType’ (see parameters) is set at “hpe5022_SEQ_ER_WR_M”, perform a DC erase for the track.
3. If ‘seqType’ (see parameters) is set at “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write the data pattern specified by the “hpe5022_selectPattern” function.
4. Move at the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Digitize and execute re-sampling of the waveform for the number of period which is specified by the “hpe5022_waveAverage” function.

Since the sampling rate of oscilloscope is not continuous, the HP E5022/E5023 makes a re-sampling of the waveform to fit the sampling rate specified by the “hpe5022_waveOverSampleRate” functions

The delay time for acquisition start and over sampling rate is specified by the “hpe5022_waveDelayTime” and “hpe5022_waveOverSampleRate” functions, respectively.

The “hpe5022_waveData_Q” and “hpe5022_waveAverageData_Q” functions returns the waveform data.

The measurement parameters, such as TAA and PW, can be calculated from the digitized waveform. The functions are shown in the “See Also”.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- seqType

Description Selects a measurement sequence type.

Direction IN

Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measure (Digitize)
hpe5022_SEQ_WR_M	1	Write->Measure (Digitize)
hpe5022_SEQ_M	2	Measure (Digitize)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	A hardware error is detected on the head amplifer.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ is out of range.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

Error Code	Description
hpe5022_ERROR_PARAMETER_SET_FAILD	The acquisition parameters are invalid for oscilloscope. Reduce the value of the “hpe5022_waveOverSampleRate”, “hpe5022_waveAverage” or “hpe5022_channelBitRate” functions.

See Also

“hpe5022_waveOverSampleRate” on page 1019

“hpe5022_waveAverage” on page 1022

“hpe5022_waveDelayTime” on page 1025

“hpe5022_waveData_Q” on page 1039

“hpe5022_waveAverageData_Q” on page 1035

“hpe5022_writeTrackOffset” on page 222

“hpe5022_readTrackOffset” on page 219

“hpe5022_selectPattern” on page 119

“hpe5022_waveTaa_Q” on page 1041

“hpe5022_wavePw_Q” on page 1043

“hpe5022_waveBaseline_Q” on page 1045

“hpe5022_waveTimeAsymmetry_Q” on page 1047

“hpe5022_waveAutoCorrelationSnr_Q” on page 1055

“hpe5022_waveDipulseExtraction_Q” on page 1051

“hpe5022_waveCorrelation_Q” on page 1058

hpe5022_setupWave

C Syntax

ViStatus hpe5022_setupWave(ViSession id, ViInt16 seqType, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupWave(ByVal id As Long, ByVal seqType As Integer, ByRef testHndl As Long) As Long

Description

This function assigns the digitalization sequence of the waveform to the specified test identifier. Refer to the “hpe5022_measureWave” function for details about the sequence. This function does not execute the digitalization. Digitalization is executed by the “hpe5022_measure” function with the test identifier specified in this function.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Selects a measurement sequence type.

  Direction — IN

  Values — Same as the ‘seqType’ in the “hpe5022_measureWave” function.

- testHndl

  Description — Specifies the test identifier. This identifier is used to execute the waveform measurement by the “hpe5022_measure” function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ is out of range.

Error Code	Description
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measure” on page 387

“hpe5022_measureWave” on page 1028

“hpe5022_releaseSetup” on page 390

hpe5022_waveAverageDataSize_Q

C Syntax

ViStatus hpe5022_waveAverageDataSize_Q(ViSession id, ViPInt32 aveSize);

Visual Basic Syntax

hpe5022_waveAverageDataSize_Q(ByVal id As Long, ByRef aveSize As Long) As Long

Description

This function returns the array size of the averaged waveform data returned by the “hpe5022_waveAverageData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- aveSize

  Description: Returns the array size of averaged waveform data.

  The size will be the same as the value of (the value of the “hpe5022_channelBitRate” function) × (the value of the “hpe5022_waveOverSampleRate” function) if the acquisition is done properly.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.

Error Code	Description
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_channelBitRate” on page 116

“hpe5022_waveOverSampleRate” on page 1019

“hpe5022_waveAverageData_Q” on page 1035

hpe5022_waveAverageData_Q

C Syntax

ViStatus hpe5022_waveAverageData_Q(ViSession id, ViReal64 aveData[]);

Visual Basic Syntax

hpe5022_waveAverageData_Q(ByVal id As Long, ByRef aveData As Double) As Long

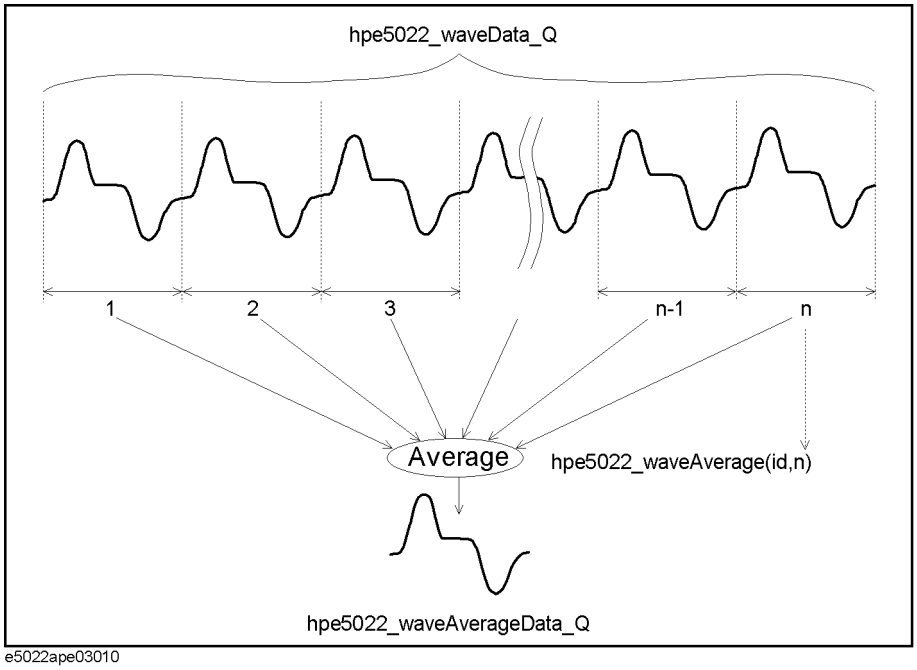
Description

This function returns the averaged waveform data which is digitized by the “hpe5022_measureWave” or the “hpe5022_setupWave” function.

The averaging factor of this function is set by the “hpe5022_waveAverage” function.

The data returned by the “hpe5022_waveData_Q” function is averaged and the averaged data is returned by this function.

Figure 4-2 Averaged Waveform Data



e5022ape03010

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- aveData

  Description: Returns the array data of the averaged waveform.

  The array size is returned by the “hpe5022_waveAverageDataSize_Q” function.

  Direction: OUT

Unit Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An option for oscilloscope is not installed. This function can only be used if the system is installed with an oscilloscope.

See Also

“hpe5022_waveAverage” on page 1022

“hpe5022_measureWave” on page 1028

“hpe5022_setupWave” on page 1031

“hpe5022_waveData_Q” on page 1039

“hpe5022_waveAverageDataSize_Q” on page 1033

hpe5022_waveDataSize_Q

C Syntax

ViStatus hpe5022_waveDataSize_Q(ViSession id, ViPInt32 size);

Visual Basic Syntax

hpe5022_waveDataSize_Q(ByVal id As Long, ByRef size As Long) As Long

Description

This function returns the array size of the digitized waveform data returned by the “hpe5022_waveData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- size

  Description: Returns the array size of waveform data.

  The size will be the same as the value of (the value of the “hpe5022_waveAverageDataSize_Q” function) × (the value of the “hpe5022_waveAverage” function) if the acquisition is done properly.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.

Error Code	Description
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveData_Q” on page 1039

“hpe5022_waveDataSize_Q” on page 1037

“hpe5022_waveAverage” on page 1022

hpe5022_waveData_Q

C Syntax

ViStatus hpe5022_waveData_Q(ViSession id, ViReal64 waveData[]);

Visual Basic Syntax

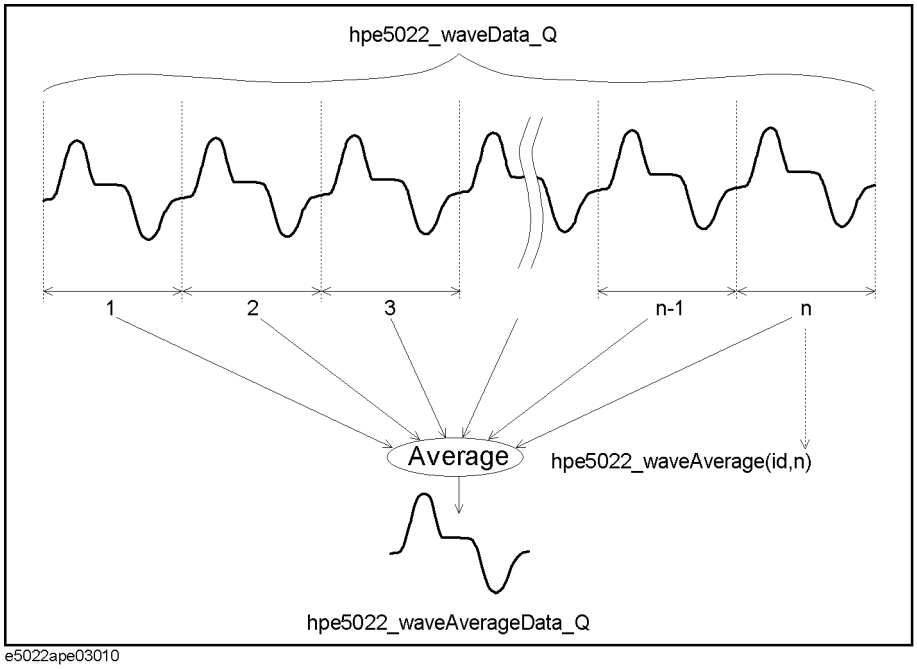
hpe5022_waveData_Q(ByVal id As Long, ByRef waveData As Double) As Long

Description

This function returns all acquisition waveform data which is digitized by the “hpe5022_measureWave” or the “hpe5022_setupWave” function.

This function returns the waveform data. All acquisition data will be returned by this function. The data is for several periods of the test data pattern. The number of period is defined by the “hpe5022_waveAverage” function.

Figure 4-3 Waveform Data



e5022ape03010

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- waveData

  Description: Returns the array data of the waveform data.

  The array size is returned by the “hpe5022_waveDataSize_Q” function.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_measureWave” on page 1028

“hpe5022_waveDataSize_Q” on page 1037

“hpe5022_waveAverageData_Q” on page 1035

Wave Analysis Function

hpe5022_waveTaa_Q

C Syntax

ViStatus hpe5022_waveTaa_Q(ViSession id, ViPReal64 taa, ViPReal64 taaPos, ViPReal64 taaNeg, ViPReal64 taaAsym);

Visual Basic Syntax

hpe5022_waveTaa_Q(ByVal id As Long, ByRef taa As Double, ByRef taaPos As Double, ByRef taaNeg As Double, ByRef taaAsym As Double) As Long

Description

This function calculates the TAA parameters using the averaged waveform data. The data returned by the “hpe5022_waveAverageData_Q” function is used for this calculation. The returned parameters are TAA, TAA positive (TAA+), TAA negative (TAA-) and TAA asymmetry. See chapter 4 in the Operation Manual for the definition of these parameters.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- taa

  Description: Returns value of the TAA.

  Direction: OUT

  Unit: Volt

- taaPos

  Description: Returns value of the TAA positive.

  Direction: OUT

  Unit: Volt

- taaNeg

  Description: Returns value of the TAA negative.

  Direction: OUT

  Unit: Volt

- taaAsym

  Description Returns value of the TAA asymmetry.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveAverageData_Q” on page 1035

hpe5022_wavePw_Q

C Syntax

ViStatus hpe5022_wavePw_Q(ViSession id, ViReal64 threshold, ViPReal64 pw, ViPReal64 pwPos, ViPReal64 pwNeg, ViPReal64 pwAsym);

Visual Basic Syntax

hpe5022_wavePw_Q(ByVal id As Long, ByVal threshold As Double, ByRef pw As Double, ByRef pwPos As Double, ByRef pwNeg As Double, ByRef pwAsym As Double) As Long

Description

This function calculates the PW parameters using the averaged waveform data. The data returned by the “hpe5022_waveAverageData_Q” function is used for this calculation. The returned parameters are PW, PW positive (PW+), PW negative (PW-) and PW asymmetry. See chapter 4 in the Operation Manual for the definition of these parameters.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- threshold

  Description: Specifies the threshold level.

  Direction: IN

  Values

Name	Value
hpe5022_PW_THR_MIN	0.2
hpe5022_PW_THR_MAX	0.8

- pw

  Description: Returns value of the PW.

  Direction: OUT

  Unit: Second

- pwPos

  Description: Returns value of the PW positive.

  Direction: OUT

  Unit: Second

- pwNeg

  Description Returns value of the PW negative.

  Direction OUT

  Unit Second

- pwAsym

  Description Returns value of the PW asymmetry.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘threshold’ is out of range.
hpe5022_ERROR_THRESHOLD_NOT_FOUND	The threshold level is not found.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveAverageData_Q” on page 1035

hpe5022_waveBaseline_Q

C Syntax

ViStatus hpe5022_waveBaseline_Q(ViSession id, ViPReal64 base, ViPReal64 basePos, ViPReal64 baseNeg, ViPReal64 baseSep);

Visual Basic Syntax

hpe5022_waveBaseline_Q(ByVal id As Long, ByRef base As Double, ByRef basePos As Double, ByRef baseNeg As Double, ByRef baseSep As Double) As Long

Description

This function calculates the baseline parameters using the averaged waveform data. The data returned by the “hpe5022_waveAverageData_Q” function is used for this calculation. The returned parameters are baseline, baseline positive (baseline+), baseline negative (baseline-) and Baseline separation. See chapter 4 in the Operation Manual for the definition of these parameters.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- base

  Description: Returns value of the baseline.

  Direction: OUT

  Unit: Volt

- basePos

  Description: Returns value of the baseline positive

  Direction: OUT

  Unit: Volt

- baseNeg

  Description: Returns value of the baseline negative.

  Direction: OUT

  Unit: Volt

- baseSep

  Description: Returns value of the baseline separation.

  Direction: OUT

  Unit: Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveAverageData_Q” on page 1035

hpe5022_waveTimeAsymmetry_Q

C Syntax

ViStatus hpe5022_waveTimeAsymmetry_Q(ViSession id, ViPReal64 timAsym,ViPReal64 tpn,ViPReal64 tnp);

Visual Basic Syntax

hpe5022_waveTimeAsymmetry_Q(ByVal id As Long, ByRef timAsym As Double, ByRef tpn As Double, ByRef tnp As Double) As Long

Description

This function calculates the Time-Asymmetry using the averaged waveform data. The data returned by the “hpe5022_waveAverageData_Q” function is used for this calculation. See chapter 4 in the Operation Manual for the definition of this parameter.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- timAsym

  Description: Returns value of the time asymmetry.

  Direction: OUT

  Unit: Second

- tpn

  Description: Returns the positive peak to negative peak time in second.

  Direction: OUT

  Unit: Second

- tnp

  Description: Returns the negative peak to positive peak time in second.

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also “hpe5022_waveAverageData_Q” on page 1035

Dipulse Extraction NLTS Function

hpe5022_waveDipulseExtractionDataSize_Q

C Syntax

ViStatus hpe5022_waveDipulseExtractionDataSize_Q(ViSession id, ViPInt32 deSize);

Visual Basic Syntax

hpe5022_waveDipulseExtractionDataSize_Q(ByVal id As Long, ByRef deSize As Long) As Long

Description

This function returns the array size of dipulse extraction data which are returned by the “hpe5022_waveDipulseExtraction_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- deSize

  Description: Returns the array size of dipulse extraction data.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.

Error Code	Description
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveDipulseExtraction_Q” on page 1051

hpe5022_waveDipulseExtraction_Q

C Syntax

ViStatus hpe5022_waveDipulseExtraction_Q(ViSession id, ViReal64 locate[], ViReal64 deData[]);

Visual Basic Syntax

hpe5022_waveDipulseExtraction_Q(ByVal id As Long, ByRef locate As Double, ByRef deData As Double) As Long

Description

This function returns the dipulse extraction data calculated from the waveform data. The waveform data returned by the “hpe5022_waveData_Q” function is used for this calculation. The calculation sequence is shown in chapter 4 of the Operation Manual.

The array size of ‘locate’ and ‘deData’ is the same and is returned by the “hpe5022_waveDipulseExtractionDataSize_Q” function. The returned value ‘deData[]’ is a dipulse extraction result for the location returned by ‘locate[]’. For example, the value of ‘deData[1]’ is a dipulse extraction result for the location returned by ‘locate[1]’.

The data pattern must be set at “hpe5022_PAT_PRBS” (Pseudo random pattern) by the “hpe5022_selectPattern” function in advance.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- locate

  Description: Returns the bit location in array. The corresponding dipulse extraction data which has the same argument is returned by ‘deData’ parameter.

  Direction: OUT

- deData

  Description: Returns the dipulse extraction data in array.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveData_Q” on page 1039

“hpe5022_waveDipulseExtractionDataSize_Q” on page 1049

“hpe5022_selectPattern” on page 119

hpe5022_waveDipulseExtractionNlts_Q

C Syntax

ViStatus hpe5022_waveDipulseExtractionNlts_Q(ViSession id, ViReal64 echo, ViPReal64 nlts);

Visual Basic Syntax

hpe5022_waveDipulseExtractionNlts_Q(ByVal id As Long, ByVal echo As Double, ByRef nlts As Double) As Long

Description

This function calculates the NLTS by the dipulse extract method, the second adjacent transition shift and the hard transition shift. The corresponding echo location is required to be inputted. See the “NLTS by Dipulse Extraction Method Measurement” on page 46 for the echo location value. The data used for the calculation can be extracted by the “hpe5022_waveDipulseExtraction_Q” function.

The calculation formula is 2 × (Amplitude at echo position) / (Amplitude at zero position).

The data pattern must be set at “hpe5022_PAT_PRBS” (Pseudo random pattern) by the “hpe5022_selectPattern” function before data acquisition.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- echo

  Description: Specifies the echo location.

  Direction: IN

  Unit: Bit

- nlts

  Description: Returns the value of NLTS by dipulse extract method, the second adjacent transition shift or the hard transition shift. The value is depending on the specified echo location.

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveDipulseExtraction_Q” on page 1051

“hpe5022_selectPattern” on page 119

“hpe5022_prbsPattern” on page 128

Time Correlation NLTS Functions

hpe5022_waveAutoCorrelationSnr_Q

C Syntax

ViStatus hpe5022_waveAutoCorrelationSnr_Q(ViSession id, ViPReal64 snr);

Visual Basic Syntax

hpe5022_waveAutoCorrelationSnr_Q(ByVal id As Long, ByRef snr As Double) As Long

Description

This function calculates the SNR (Signal to Noise Ratio) used in the NLTS measurement by Time Correlation method. SNR is calculated from the waveform data returned by the “hpe5022_waveData_Q” function. See chapter 4 in the Operation Manual for the calculation formula.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- snr

  Description: Returns the SNR.

  Direction: OUT

  Unit: dB

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.

Error Code	Description
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also “hpe5022_waveData_Q” on page 1039

hpe5022_waveCorrelationDataSize_Q

C Syntax

ViStatus hpe5022_waveCorrelationDataSize_Q(ViSession id, ViPInt32 acSize);

Visual Basic Syntax

hpe5022_waveCorrelationDataSize_Q(ByVal id As Long, ByRef acSize As Long) As Long

Description

This function returns the array size of time correlation data which are returned by the “hpe5022_waveCorrelation_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- acSize

  Description: Returns the array size of time correlation data.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An option for oscilloscope is not installed. This function can only be used if the system is installed with an oscilloscope.

See Also

“hpe5022_waveCorrelation_Q” on page 1058

hpe5022_waveCorrelation_Q

C Syntax

ViStatus hpe5022_waveCorrelation_Q(ViSession id, ViReal64 locate[], ViReal64 acData[]);

Visual Basic Syntax

hpe5022_waveCorrelation_Q(ByVal id As Long, ByRef locate As Double, ByRef acData As Double) As Long

Description

This function returns the time correlation data calculated from the waveform data. The data returned by the “hpe5022_waveData_Q” function is used to calculate the correlation. The calculation formula is shown in chapter 4 of the Operation Manual.

The array size of ‘locate’ and ‘acData’ is the same and is returned by the “hpe5022_waveCorrelationDataSize_Q” function. The returned value ‘acData[]’ is a time correlation result for the location returned by ‘locate[]’. For example, the value of ‘acData[1]’ is a time correlation result for the location returned by ‘locate[1]’.

The returned data is the calculated result of the cross correlation coefficient of the readback voltages at t and t+mT +D. The parameter D is the delay at which NLTS echo is expected and the parameter m is a certain integer value.

The data pattern must be set at “hpe5022_PAT_PRBS” (Pseudo random pattern) by the “hpe5022_selectPattern” function in advance.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- locate

  Description: Returns the bit location in array. The corresponding time correlation data which has the same argument is returned by ‘acData’ parameter.

  Direction: OUT

- acData

  Description: Returns the time correlation data in array.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveData_Q” on page 1039

“hpe5022_waveDipulseExtraction_Q” on page 1051

“hpe5022_selectPattern” on page 119

hpe5022_waveTimeCorrelationNlts_Q

C Syntax

ViStatus hpe5022_waveTimeCorrelationNlts_Q(ViSession id, ViReal64 echo, ViPReal64 nlts);

Visual Basic Syntax

hpe5022_waveTimeCorrelationNlts_Q(ByVal id As Long, ByVal echo As Double, ByRef nlts As Double) As Long

Description

This function calculates the NLTS by the Time Correlation method, the second adjacent transition shift and the hard transition shift. The corresponding echo location is required to be inputted. See the “NLTS by Time Correlation Method Measurement” on page 47 for the echo location value. The data used for the calculation can be gotten by the “hpe5022_waveCorrelation_Q” and “hpe5022_waveAutoCorrelationSnr_Q” functions. See chapter 4 in the Operation Manual for the calculation formula.

The data pattern must be set at “hpe5022_PAT_PRBS” (Pseudo random pattern) by the “hpe5022_selectPattern” function before the data acquisition.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- echo

  Description: Specifies the echo location.

  Direction: IN

  Unit: Bit

- nlts

  Description: Returns the value of NLTS by the time correlation method, the second adjacent transition shift or the hard transition. shift

  Direction: OUT

  Unit: Second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The waveform data is corrupt.
hpe5022_ERROR_NO_OPTION	An oscilloscope option is not installed. This function can be used only when the system has oscilloscope. If system has an osilloscope, install HP E5022A software again and make sure that the option number, the serial number and the codeword are correct.
hpe5022_ERROR_NOT_INIT	The resource of oscilloscope is not opened at initialize. Check if the oscilloscope is included in the rsrcArray of the "hpe5022_init" function.

See Also

“hpe5022_waveCorrelation_Q” on page 1058

“hpe5022_waveAutoCorrelationSnr_Q” on page 1055

“hpe5022_selectPattern” on page 119

5 Function Reference for Measurement using Agilent 4396B

This section describes the functions related with the measurement by a spectrum Analzer 4396B. Customers can use 4396B in order to make a high frequency spectrum measurement. See the Appendix D in the operation manual.

Spectrum measurement using 4396B

This section describes the functions related with spectrum measurement using 4396B. This measures a power level using 4396B. Customers can use 4396B in order to make a high frequency spectrum measurement. See the Appendix D in the operation manual.

hpe5022_spectrumOptSaFrequency

C Syntax

ViStatus hpe5022_spectrumOptSaFrequency(ViSession id, ViReal64 startFreq, ViReal64 stopFreq);

VB Syntax

hpe5022_spectrumOptSaFrequency(ByVal id As Long, ByVal startFreq As Double, ByVal stopFreq As Double) As Long

Description

This function controls the starting and stopping frequencies of a spectrum analyzer for spectrum measurement using 4396B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- startFreq

  Description: Specifies the starting frequency.

  Direction: IN

  Unit: Hz

  Preset Value: 0 (hpe5022_SPEC_OPT_SA_FREQ_MIN)

  Values

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MIN	0
hpe5022_SPEC_OPT_SA_FREQ_MAX	1.8×10^{9}

- stopFreq

  Description: Specifies the stopping frequency.

  Direction: IN

  Unit: Hz

  Preset Value: 1.8×10^{9} (hpe5022_SPEC_OPT_SA_FREQ_MAX)

Values

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MIN	0
hpe5022_SPEC_OPT_SA_FREQ_MAX	1.8×10^{9}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘startFreq’ or/and ‘stopFreq’ is out of range.

See Also

“hpe5022_spectrumOptSaFrequency_Q” on page 1066

hpe5022_spectrumOptSaFrequency_Q

C Syntax

ViStatus hpe5022_spectrumOptSaFrequency_Q(ViSession id, ViPReal64 startFreq, ViPReal64 stopFreq);

VB Syntax

hpe5022_spectrumOptSaFrequency_Q(ByVal id As Long, ByRef startFreq As Double, ByRef stopFreq As Double) As Long

Description

This function returns the starting and stopping frequencies of a spectrum analyzer for spectrum measurement using 4396B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- startFreq

  Description: Returns the starting frequency.

  Direction: OUT

  Unit: Hz

- stopFreq

  Description: Returns the stopping frequency.

  Direction: OUT

  Unit: Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectrumOptSaFrequency” on page 1064

hpe5022_spectrumOptSaBandWidth

C Syntax

ViStatus hpe5022_spectrumOptSaBandWidth(ViSession id, ViReal64 rbw, ViReal64 vbw);

VB Syntax

hpe5022_spectrumOptSaBandWidth(ByVal id As Long, ByVal rbw As Double, ByVal vbw As Double) As Long

Description

This function specifies the resolution band width and the video band width of the spectrum analyzer for the spectrum measurement using 4396B. Resolution band width allows you to improve resolution of frequency, in other words, selectivity of signal, and also to reduce the display noise. On the other hand, video band width reduces the noise variations and makes the signal clearly visible.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rbw

  Description: Specifies the resolution band width. Narrowing this, will widen the dynamic range. Widening this, will make a measurement speed fast. This can be set to 10×10^3, 30×10^3, 100×10^3, 300×10^3, 1×10^6 and 3×10^6, When rbw is set other than these values, then rbw is set to the closest and lower value from the above selection.

  Direction: IN

  Unit: Hz

  Preset Value: 10×10^3

  Values

Name	Value
hpe5022_SPEC_OPT_SA_BW_MIN	10×10^3
hpe5022_SPEC_OPT_SA_BW_MAX	3×10^6

- vbw

  Description: Specifies the video band width. Narrowing this, will reduce noise variation. Widening this, will speed up the measurement. The values 1/1, 1/3, 1/30, 1/100 or 1/300 of the resolution band width can be set as video band width. When other values are used, the value is set automatically at nearest and lower value of vbw.

Direction IN

Unit Hz

Preset Value 10×10^3

Values

Name	Value
hpe5022_SPEC_OPT_SA_VBW_MIN	30
hpe5022_SPEC_OPT_SA_VBW_MAX	3×10^6

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘rbw’ and/or ‘vbw’ is out of range.

See Also

“hpe5022_spectrumOptSaBandWidth_Q” on page 1069

hpe5022_spectrumOptSaBandWidth_Q

C Syntax

ViStatus hpe5022_spectrumOptSaBandWidth_Q(ViSession id, ViPReal64 rbw, ViPReal64 vbw);

VB Syntax

hpe5022_spectrumOptSaBandWidth_Q(ByVal id As Long, ByRef rbw As Double, ByRef vbw As Double) As Long

Description

This function returns the resolution band width and the video band width of the spectrum analyzer for spectrum measurement using 4396B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rbw

  Description: Returns the resolution band width.

  Direction: OUT

  Unit: Hz

- vbw

  Description: Returns the video band width.

  Direction: OUT

  Unit: Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectrumOptSaBandWidth” on page 1067

hpe5022_measureSpectrumOptSa

C Syntax

ViStatus hpe5022_measureSpectrumOptSa(ViSession id,ViInt16 seqType, ViInt16 detMode);

Visual Basic Syntax

hpe5022_measureSpectrumOptSa(ByVal id As Long, ByVal seqType As Integer, ByVal detMode As Integer) As Long

Description

This function measures the power level according to the specified sequence type. There are three sequence types: 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The sequence is as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If seqType is set to “hpe5022_SEQ_ER_WR_M”, erase the track.
3. If seqType is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write the data pattern specified by the “hpe5022_selectPattern” function.
4. Move the head to read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure the power level using a spectrum analyzer.

The “hpe5022_spectrumOptSaData_Q” function returns the measurement result.

Parameters

- id

  Description Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction IN

- seqType

  Description Specifies the type of measurement sequence.

  Direction IN

  Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measurement
hpe5022_SEQ_WR_M	1	Write->Measurement
hpe5022_SEQ_M	2	Measurement

- detMode

  Description Specifies the detection mode of the spectrum analyzer. Select sample mode when the noise is measured. Select positive peak mode when the signal is measured.

Direction IN

Values

Name	Value	Description
hpe5022_SPEC_DET_SAMP	1	Sample Mode
hpe5022_SPEC_DET_POS_PEAK	2	Positive Peak Mode

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘detMode’ is out of range.

See Also

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_spectrumOptSaData_Q” on page 1074

“hpe5022_spectrumOptSaDataAll_Q” on page 1076

“hpe5022_driveState” on page 209

hpe5022_setupSpectrumOptSa

C Syntax

ViStatus hpe5022_setupSpectrumOptSa(ViSession id, ViInt16 seqType, ViInt16 detMode, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSpectrumOptSa(ByVal id As Long, ByVal seqType As Integer, ByVal detModeAs Integer, ByRef testHndl As long) As Long

Description

This function assigns the 4396B's spectrum measurement sequence to the specified test identifier. See the "hpe5022_measureSpectrumOptSa" function for details about the sequence. This function does not execute the measurement. Measurement is executed by the "hpe5022_measure" function with the test identifier specified in this function

The "hpe5022_spectrumOptSaData_Q" function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the 'seqType' in the "hpe5022_measureSpectrumOptSa" function.

- detMode

  Description — Specifies the detection mode of the spectrum analyzer. Select sample mode when the noise is measured. Select positive peak mode when the signal is measured.

  Direction — IN

  Values — Same as the 'detMode' in the "hpe5022_measureSpectrumOptSa" function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the spectrum measurement using 4396B by the "hpe5022_measure" function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ and/or ‘detMode’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureSpectrumOptSa” on page 1070

“hpe5022_spectrumOptSaData_Q” on page 1074

“hpe5022_spectrumOptSaDataAll_Q” on page 1076

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_spectrumOptSaData_Q

C Syntax

ViStatus hpe5022_spectrumOptSaData_Q(ViSession id, ViReal64 freq, ViPReal64 data);

Visual Basic Syntax

hpe5022_spectrumOptSaData_Q(ByVal id As Long, ByVal freq As Double, ByRef data As Double) As Long

Description

This function returns a power level for specified frequency. As the spectrum analyzer measures only at the measurement frequencies returned by the “hpe5022_spectrumOptSaDataAll_Q” function, the returned data is a interpolated value from the measurement result returned by the “hpe5022_spectrumOptSaDataAll_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freq

  Description: Specifies the center frequency.

  Direction: IN

  Unit: Hz

  Values

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MIN	0
hpe5022_SPEC_OPT_SA_FREQ_MAX	1.8×10^{9}

- data

  Description: Returns the power level at specified frequency.

  Direction: OUT

  Unit: dBm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freq’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The spectrum data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectrumOptSa” on page 1070

“hpe5022_setupSpectrumOptSa” on page 1072

“hpe5022_spectrumOptSaDataAll_Q” on page 1076

hpe5022_spectrumOptSaDataAll_Q

C Syntax

ViStatus hpe5022_spectrumOptSaDataAll_Q (ViSession id, ViPInt16 points, ViReal64 freq[], ViReal64 data[]);

Visual Basic Syntax

hpe5022_spectrumOptSaDataAll_Q(ByVal id As Long, ByRef points As Integer, ByRef freq As Double, ByRef data As Double) As Long

Description

This function returns data of power level for all measurement points of 4396B.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- points

  Description: Returns the number of measurement points.

  Direction: OUT

- freq

  Description: Returns the data array of the frequency. The array size is same as the number of the ‘point’ parameter. The frequency points determined from the start, the stop frequencies and the resolution band width.

  Direction: OUT

  Unit: Hz

- data

  Description: Returns the data array of the power level. The array size is same as the ‘point’ parameter.

  Direction: OUT

  Unit: dBm

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The spectrum measurement data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectrumOptSa” on page 1070

“hpe5022_setupSpectrumOptSa” on page 1072

“hpe5022_spectrumOptSaData_Q” on page 1074

hpe5022_spectrumOptSaNoise_Q

C Syntax

hpe5022_spectrumOptSaNoise_Q (ViSession id, ViReal64 freqLow, ViReal64 freqHigh, ViPReal64 noise);

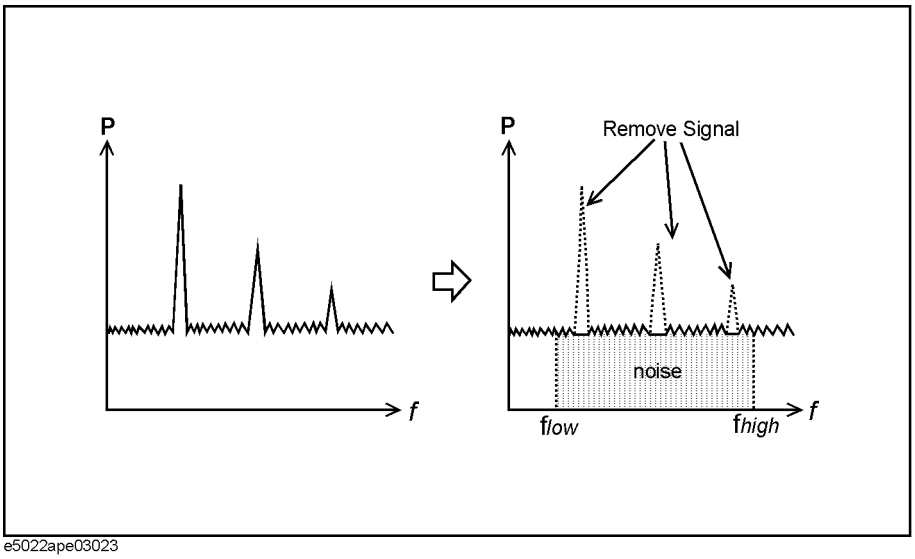
Visual Basic Syntax

hpe5022_spectrumOptSaNoise_Q(ByVal id As Long, ByVal freqLow As Double, ByVal freqHigh As Double, ByRef noise As Double) As Long

Description

This function returns the calculated noise level from spectrum data. The portion of signal peak is removed and noise level is calculated.

Figure 5-1 Noise Level Measurement



e5022ape03023

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- freqLow

  Description: Specifies the low integration range of frequency.

  Direction: IN

  Unit: Hz

  Values

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MIN	0

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MAX	1.8×10^9

- freqHigh

  Description Specifies the high integration range of frequency.

  Direction IN

  Unit Hz

  Values

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MIN	0
hpe5022_SPEC_OPT_SA_FREQ_MAX	1.8×10^9

- noise

  Description Returns the noise level.

  Direction OUT

  Unit rms Volt

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘freqLow’ or/and ‘freqHigh’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The spectrum measurement data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectrumOptSa” on page 1070

“hpe5022_setupSpectrumOptSa” on page 1072

“hpe5022_spectrumOptSaDataAll_Q” on page 1076

Spectral SNR Measurement Using 4396B

This section describes the functions related with spectral SNR measurement using 4396B. The spectral SNR measurement is measured by a 4396B spectrum analyzer.

hpe5022_spectralSnrOptSaFrequency

C Syntax

ViStatus hpe5022_spectralSnrOptSaFrequency(ViSession id, ViReal64 startFreq, ViReal64 stopFreq);

VB Syntax

hpe5022_spectralSnrOptSaFrequency(ByVal id As Long, ByVal startFreq As Double, ByVal stopFreq As Double) As Long

Description

This function specifies the starting and stopping frequencies of a 4396B spectrum analyzer for spectral SNR measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- startFreq

  Description: Specifies the starting frequency.

  Direction: IN

  Unit: Hz

  Preset Value: 0 (hpe5022_SPEC_OPT_SA_FREQ_MIN)

  Values

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MIN	0
hpe5022_SPEC_OPT_SA_FREQ_MAX	1.8×10^{9}

- stopFreq

  Description: Specifies the stopping frequency.

  Direction: IN

  Unit: Hz

  Preset Value: 1.8×10^{9} (hpe5022_SPEC_OPT_SA_FREQ_MAX)

Values

Name	Value
hpe5022_SPEC_OPT_SA_FREQ_MIN	0
hpe5022_SPEC_OPT_SA_FREQ_MAX	1.8×10^{9}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘startFreq’ or/and ‘stopFreq’ is out of range.

See Also

“hpe5022_spectralSnrOptSaFrequency_Q” on page 1082

hpe5022_spectralSnrOptSaFrequency_Q

C Syntax

ViStatus hpe5022_spectralSnrOptSaFrequency_Q(ViSession id, ViPReal64 startFreq, ViPReal64 stopFreq);

VB Syntax

hpe5022_spectralSnrOptSaFrequency_Q(ByVal id As Long, ByRef startFreq As Double, ByRef stopFreq As Double) As Long

Description

This function returns the starting and stopping frequencies of a 4396B spectrum analyzer for the spectral SNR measurement.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- startFreq

  Description — Returns the starting frequency.

  Direction — OUT

  Unit — Hz

- stopFreq

  Description — Returns the stopping frequency.

  Direction — OUT

  Unit — Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectralSnrOptSaFrequency” on page 1080

hpe5022_spectralSnrOptSaBandWidth

C Syntax

ViStatus hpe5022_spectralSnrOptSaBandWidth(ViSession id, ViReal64 rbw, ViReal64 vbw);

VB Syntax

hpe5022_spectralSnrOptSaBandWidth(ByVal id As Long, ByVal rbw As Double, ByVal vbw As Double) As Long

Description

This function specifies the resolution band width and video band width of the 4396B spectrum analyzer for the spectral SNR measurement. Resolution band width allows you to improve the resolution of frequency, i.e, selectivity of signal, and also to reduce the display noise. On the other hand, video band width reduces the noise variations and makes the signal clearly visible.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rbw

  Description: Specifies the resolution band width. Narrowing this, will widen the dynamic range. Widening this, makes measurement speed faster. This can be set to 10×10^3, 30×10^3, 100×10^3, 300×10^3, 1×10^6 and 3×10^3. When the value other than these values is set, the closest and lower value is set automatically.

  Direction: IN

  Unit: Hz

  Preset Value: 10×10^3

  Values

Name	Value
hpe5022_SPEC_OPT_SA_BW_MIN	10×10^3
hpe5022_SPEC_OPT_SA_BW_MAX	3×10^6

- vbw

  Description: Specifies the video band width. Narrowing this, will reduce the noise variation. Widening this, will speed up measurement.The value of 1/1, 1/3, 1/30, 1/100 or 1/300 of the resolution band width can be set as a video band width. When other value is entered, the value is set automatically to the closest and lower value of the setting.

Direction IN

Unit Hz

Preset Value 10×10^3

Values

Name	Value
hpe5022_SPEC_OPT_SA_VBW_MIN	30
hpe5022_SPEC_OPT_SA_VBW_MAX	3×10^6

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘rbw’ and/or ‘vbw’ is out of range.

See Also

“hpe5022_spectralSnrOptSaBandWidth_Q” on page 1085

hpe5022_spectralSnrOptSaBandWidth_Q

C Syntax

ViStatus hpe5022_spectralSnrOptSaBandWidth_Q(ViSession id, ViPReal64 rbw, ViPReal64 vbw);

VB Syntax

hpe5022_spectralSnrOptSaBandWidth_Q(ByVal id As Long, ByRef rbw As Double, ByRef vbw As Double) As Long

Description

This function returns the resolution band width and the video band width of the 4396B spectrum analyzer for the spectral SNR measurement.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- rbw

  Description: Returns the resolution band width.

  Direction: OUT

  Unit: Hz

- vbw

  Description: Returns the video band width.

  Direction: OUT

  Unit: Hz

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.

See Also

“hpe5022_spectralSnrOptSaBandWidth” on page 1083

hpe5022_measureSpectralSnrOptSa

C Syntax

ViStatus hpe5022_measureSpectralSnrOptSa(ViSession id,ViInt16 seqType);

Visual Basic Syntax

hpe5022_measureSpectralSnrOptSa(ByVal id As Long, ByVal seqType As Integer) As Long

Description

This function measures the spectral SNR using 4396B according to the specified sequence type. There are three sequence types: 1) erase, write data, then measure 2) no erase, write data, then measure 3) measure only. The sequence is as follows:

1. Move the head to the write track offset position specified by the “hpe5022_writeTrackOffset” function.
2. If seqType is set to “hpe5022_SEQ_ER_WR_M”, erase the track.
3. If seqType is set to “hpe5022_SEQ_ER_WR_M” or “hpe5022_SEQ_WR_M”, write the data pattern specified by the “hpe5022_selectPattern” function.
4. Move the head to the read track offset position specified by the “hpe5022_readTrackOffset” function.
5. Measure noise level. (Same as follows)
   a. “hpe5022_measureSpectrumOptSa” (seqType: hpe5022_SEQ_M, detMode: hpe5022_SPEC_DET_SAMP)
   b. The data returned from “hpe5022_spectrumOptSaNoise_Q” becomes the noise level.
6. Measure the peak signal level. (Same as follows)
   a. “hpe5022_measureSpectrumOptSa” (seqType: hpe5022_SEQ_M, detMode: hpe5022_SPEC_DET_POS_PEAK)
   b. The maximum data returned from “hpe5022_spectrumOptSaDataAll_Q” becomes the peak signal level.

The “hpe5022_spectralSnrOptSa_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

Values

Name	Value	Description
hpe5022_SEQ_ER_WR_M	0	Erase->Write->Measurement
hpe5022_SEQ_WR_M	1	Write->Measurement
hpe5022_SEQ_M	2	Measurement

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifer.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘seqType’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand is turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.

See Also

“hpe5022_readTrackOffset” on page 219

“hpe5022_writeTrackOffset” on page 222

“hpe5022_spectralSnrOptSa_Q” on page 1090

“hpe5022_driveState” on page 209

hpe5022_setupSpectralSnrOptSa

C Syntax

ViStatus hpe5022_setupSpectralSnrOptSa(ViSession id, ViInt16 seqType, ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupSpectralSnrOptSa(ByVal id As Long, ByVal seqType As Integer, ByRef testHndl As long) As Long

Description

This function assigns the 4396B's spectral SNR measurement sequence to the specified test identifier. See the "hpe5022_measureSpectralSnrOptSa" function for details about the sequence. This function does not execute the measurement. Measurement is executed by the "hpe5022_measure" function with the test identifier specified in this function

The "hpe5022_spectralSnrOptSa_Q" function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- seqType

  Description — Specifies the type of measurement sequence.

  Direction — IN

  Values — Same as the 'seqType' in the "hpe5022_measureSpectralSnrOptSa" function.

- testHndl

  Description — Returns the test identifier. This identifier is used to execute the spectrum measurement using 4396B by the "hpe5022_measure" function.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by 'id' is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter 'seqType' is out of range.

Error Code	Description
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureSpectralSnrOptSa” on page 1086

“hpe5022_spectralSnrOptSa_Q” on page 1090

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_spectralSnrOptSa_Q

C Syntax

ViStatus hpe5022_spectralSnrOptSa_Q(ViSession id, ViPReal64 signal, ViPReal64 noise, ViPReal64 snr);

Visual Basic Syntax

hpe5022_spectralSnrOptSa_Q(ByVal id As Long, ByRef signal As Double, ByRef noise As Double, ByRef snr As Double) As Long

Description

This function returns result of the spectral SNR measurement using 4396B. The signal level, the noise level, and the SNR are returned.

Parameters

- id
  - Description: Specifies the system identifier. This is given by the "hpe5022_init" function.
  - Direction: IN
- signal
  - Description: Returns the measured signal level.
  - Direction: OUT
  - Unit: Volt
- noise
  - Description: Returns the measured noise level.
  - Direction: OUT
  - Unit: rms Volt
- snr
  - Description: Returns the measured SNR.
  - Direction: OUT
  - Unit: dB

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ’id’ is invalid.
hpe5022_ERROR_DATA_CORRUPT	The spectrum data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureSpectralSnrOptSa” on page 1086

“hpe5022_setupSpectralSnrOptSa” on page 1088

Channel Bit Rate Sweep Measurement Using 4396B

This section describes the functions related with the channel bit rate sweep measurement using 4396B.

hpe5022_measureChannelBitRateSweepOptSa

C Syntax

ViStatus hpe5022_measureChannelBitRateSweepOptSa(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 chanBitRate[]);

Visual Basic Syntax

hpe5022_measureChannelBitRateSweepOptSa (ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef chanBitRate As Double) As Long

Description

This function measures the channel bit rate sweep measurement. The sequences of this function are as follows:

1. Set i to zero. (i = 0)
2. Set a channel bit rate to ‘chanBitRate[i]’ (see parameters)
3. Make a measurement specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function.
4. Increment i by 1. (i = i+1)
5. If i is not equal to ‘points’ (see parameter), go to the step 2.

The “hpe5022_channelBitRateSweepOptSaData_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

  Type: ViSession

- measFunc

  Description: Specifies the type of measurement. Only “hpe5022_MEAS_NB_TAA” can be selected.

  Direction: IN

  Values

Name	Value	Description
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the channel bit rate list.

  Direction IN

  Value

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- chanBitRate

  Description Specifies the channel bit rate as the data array. The channel bit rate will be set to data in array order. The range of channel bit rate is returned by the “hpe5022_channelBitRateRange_Q” function.

  Direction IN

  Unit bps (bit per second)

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifer.
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’, ‘points’ and/or ‘chanbitRate’ is out of range.
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.

Error Code	Description
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

"hpe5022_setupChannelBitRateSweepOptSa" on page 1095

"hpe5022_channelBitRateSweepOptSaData_Q" on page 1097

"hpe5022_channelBitRateRange_Q" on page 118

hpe5022_setupChannelBitRateSweepOptSa

C Syntax

ViStatus hpe5022_setupChannelBitRateSweepOptSa(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 chanBitRate[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupChannelBitRateSweepOptSa (ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef chanBitRate As Double, ByRef testHndl As Long) As Long

Description

This function assigns the channel bit rate measurement sequence to the specified test identifier. Refer to the “hpe5022_measureChannelBitRateSweepOptSa” function for details about the sequence. This function does not execute the measurement. Measurement is executed by the “hpe5022_measure” function with the test identifier specified in this function.

The “hpe5022_channelBitRateSweepOptSaData_Q” function returns the measurement result.

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the ‘measFunc’ in the “hpe5022_measureChannelBitRateSweepOptSa” function.

- points

  Description — Specifies the number of measurement points. This number must be the same as the array size of the channel bit rate list.

  Direction — IN

  Values — Same as the ‘points’ in the “hpe5022_measureChannelBitRateSweepOptSa” function.

- chanBitRate

  Description — Specifies the channel bit rate as the data array. The channel bit rate will be set to data in the array order.

  Direction — IN

  Unit — bps (bit per second)

  Values — Same as the ‘chanBitRate’ in the “hpe5022_measureChannelBitRateSweepOptSa” function.

- testHndl

  Description Returns the test identifier. This identifier is used to execute the channel bit rate sweep measurement by the “hpe5022_measure” function.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’ , ‘points’ and/or ‘chanbitRate’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureChannelBitRateSweepOptSa” on page 1092

“hpe5022_channelBitRateSweepOptSaData_Q” on page 1097

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_channelBitRateSweepOptSaData_Q

C Syntax

ViStatus hpe5022_channelBitRateSweepOptSaData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_channelBitRateSweepOptSaData_Q (ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the channel bit rate sweep measurement. The returned data are set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and the channel bit rate list are specified by the “hpe5022_measureChannelBitRateSweepOptSa” function or “hpe5022_setupChannelBitRateSweepOptSa” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported. Only “hpe5022_DATA_NB_TAA” can be selected.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description: Returns the data (set in array) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureChannelBitRateSweepOptSa” or “hpe5022_setupChannelBitRateSweepOptSa” function. Each value in the array is the measurement result at each channel bit rate list.

  Direction: OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by “id” is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureChannelBitRateSweepOptSa” on page 1092

“hpe5022_setupChannelBitRateSweepOptSa” on page 1095

Roll Off Measurement Using 4396B

This section describes the functions related with the roll off measurement using 4396B. The roll off measurement is a flux frequency sweep measurement.

hpe5022_measureRollOffOptSa

C Syntax

ViStatus hpe5022_measureRollOffOptSa(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 fluxFreq[]);

Visual Basic Syntax

hpe5022_measureRollOffOptSa(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef fluxFreq As Double) As Long

Description

This function measures the roll off characteristic curve. The sequences of this function are as follows:

1. Set i to zero. (i = 0)
2. Move the head to the write track offset specified by the “hpe5022_writeTrackOffset” function.
3. Perform DC erase of an entire track.
4. Set the flux frequency at ‘fluxFreq[i]’ (see parameters).
5. Write a repetitive pattern at the specified flux frequency for an entire track.
6. Move the head to the read track offset specified by the “hpe5022_readTrackOffset” function.
7. Make measurement as specified by ‘measFunc’. The detailed measurement sequence is described in each measurement function
8. Increment i by 1. (i = i+1)
9. If i is not equal to ‘points’ (see parameters), go to step 2.

The “hpe5022_rollOffOptSaData_Q” function returns the measurement result.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- measFunc

  Description: Specifies the type of measurement. Only “hpe5022_MEAS_NB_TAA” can be selected.

  Direction: IN

Values

Name	Value	Description
hpe5022_MEAS_NB_TAA	5	Measure narrow band TAA

- points

  Description Specifies the number of measurement points. This number must be the same as the array size of the flux frequency list.

  Direction IN

  Values

Name	Value
hpe5022_LIST_SWEEP_COUN_MIN	1
hpe5022_LIST_SWEEP_COUN_MAX	101

- fluxFreq

  Description Specifies the flux frequency data as the data array. The flux frequency will be set to data in the array order. The maximum limit of this value is returned by the “hpe5022_channelBitRateRange_Q” function.

  Direction IN

  Unit flux per second

  Values

Name	Value
hpe5022_ROLLOFF_FREQ_MIN	4×10^{6}

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_HARD_HAMP	Hardware error is detected in the head amplifer.
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.

Error Code	Description
hpe5022_ERROR_INV_DRIVE_CONDITION	The spinstand drive has been turned off. Turn it on before executing this function. See the “hpe5022_driveState” function.
hpe5022_ERROR_INV_PARAMETER	The parameter ’measFunc’, ‘points’ and/or ‘fluxFreq’ is out of range.
hpe5022_ERROR_OVERFLOW	An overflow is detected in the parametric module. Check if the parameter setting is correct.
hpe5022_ERROR_THERMAL_ASPERITY	Thermal Asperity is detected. Check the head and media. This error occurs when the overflow is detected but the average of measurement is under the limit.
hpe5022_ERROR_UNDERFLOW	An underflow is detected in the parametric module. Check if the parameter setting is correct.

See Also

“hpe5022_rollOffOptSaData_Q” on page 1104

“hpe5022_writeTrackOffset” on page 222

“hpe5022_readTrackOffset” on page 219

“hpe5022_driveState” on page 209

“hpe5022_channelBitRateRange_Q” on page 118

hpe5022_setupRollOffOptSa

C Syntax

ViStatus hpe5022_setupRollOffOptSa(ViSession id, ViInt16 measFunc, ViInt16 points, const ViReal64 fluxFreq[], ViPObject testHndl);

Visual Basic Syntax

hpe5022_setupRollOffOptSa(ByVal id As Long, ByVal measFunc As Integer, ByVal points As Integer, ByRef fluxFreq As Double, ByRef testHndl As Long) As Long

Description

This function assigns the 4396B's roll off measurement sequence to the specified test identifier. Refer to the "hpe5022_measureRollOffOptSa" function for details about the sequence. This function does not execute the measurement. Measurement is executed by the "hpe5022_measure" function with the test identifier specified in this function.

The "hpe5022_rollOffOptSaData_Q" function returns the measurement result

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- measFunc

  Description — Specifies the type of measurement.

  Direction — IN

  Values — Same as the 'measFunc' in the "hpe5022_measureRollOffOptSa" function.

- points

  Description — Specifies the number of measurement points. This number must be the same as the array size of the flux frequency list.

  Direction — IN

  Values — Same as the 'points' in the "hpe5022_measureRollOffOptSa" function.

- fluxFreq

  Description — Specifies the flux frequency data as the data array. The flux frequency will be set to data in the array order.

  Direction — IN

  Unit — flux per second

  Values — Same as the 'fluxFreq' in the "hpe5022_measureRollOffOptSa" function.

- testHndl

  Description Returns the test identifier. This identifier is used to execute the roll off measurement by the “hpe5022_measure” function.

  Direction OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘measFunc’. ‘points’ and/or ‘fluxFreq’ is out of range.
hpe5022_ERROR_MEM_ALLOC	Lack of memory. Release the finished setup function using the “hpe5022_releaseSetup” function.

See Also

“hpe5022_measureRollOffOptSa” on page 1099

“hpe5022_rollOffOptSaData_Q” on page 1104

“hpe5022_measure” on page 387

“hpe5022_releaseSetup” on page 390

hpe5022_rollOffOptSaData_Q

C Syntax

ViStatus hpe5022_rollOffOptSaData_Q(ViSession id, ViInt16 dataType, ViReal64 data[]);

Visual Basic Syntax

hpe5022_rollOffOptSaData_Q(ByVal id As Long, ByVal dataType As Integer, ByRef data As Double) As Long

Description

This function returns the result of the roll off measurement using 4396B. The returned data are set in array, and its size is specified by the ‘points’ parameter. The ‘points’ and the flux frequency list are specified by the “hpe5022_measureRollOffOptSa” function or “hpe5022_setupRollOffOptSa” function

Parameters

- id

  Description — Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction — IN

- dataType

  Description — Specifies the type of data to be reported. Only “hpe5022_DATA_NB_TAA” can be selected.

  Direction — IN

  Values

Name	Value	Description
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- data

  Description — Returns the data (set in array) of the parameter specified by the 'dataType'. The size of array is specified by the ‘points’ parameter in the “hpe5022_measureRollOffOptSa” or “hpe5022_setupRollOffOptSa” function. Each value in the array is the measurement result relative to the flux frequency that corresponds to the same order value in the array of flux frequency list.

  Direction — OUT

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ’dataType’ is out of range.
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.

See Also

“hpe5022_measureRollOffOptSa” on page 1099

“hpe5022_setupRollOffOptSa” on page 1102

hpe5022_calculateD50OptSa_Q

C Syntax

ViStatus hpe5022_calculateD50OptSa_Q(ViSession id, ViInt16 dataType, ViPReal64 d50);

Visual Basic Syntax

hpe5022_calculateD50OptSa_Q(ByVal id As Long, ByVal dataType As Integer, ByRef d50 As Double) As Long

Description

This function calculates D_{50} (Roll Off Density) in flux per second. D_{50} is a flux frequency value at half of the peak level in the data returned by the “hpe5022_rollOffOptSaData_Q” function.

Parameters

- id

  Description: Specifies the system identifier. This is given by the "hpe5022_init" function.

  Direction: IN

- dataType

  Description: Specifies the type of data to be reported. Only “hpe5022_DATA_NB_TAA” can be selected.

  Direction: IN

  Values

Name	Value	Description
hpe5022_DATA_NB_TAA	15	Narrow band TAA

- d50

  Description: Returns the D50.

  Direction: OUT

  Unit: flux per second

Return Values

Completion Code	Description
VI_SUCCESS	No Error

Error Code	Description
hpe5022_ERROR_INV_ID	The handle specified by ‘id’ is invalid.
hpe5022_ERROR_INV_PARAMETER	The parameter ‘dataType’ is out of range.

Error Code	Description
hpe5022_ERROR_DATA_CORRUPT	The data is corrupt. Check if your measurement sequence is correct.
hpe5022_ERROR_D50_NOT_FOUND	Can't find the D50 parameter point. Check the data returned by the "hpe5022_rollOffOptSaData_Q" function.

See Also

"hpe5022_measureRollOffOptSa" on page 1099

"hpe5022_setupRollOffOptSa" on page 1102

"hpe5022_rollOffOptSaData_Q" on page 1104

A Manual Changes

This appendix contains the information of the firmware/software versions and or configurations of the Agilent Technologies E5022/E5023. The information in this manual applies directly to an Agilent Technologies E5022/E5023 whose firmware/software revision is listed on the title page of this manual.

Service Release for Revision B.02.50 from Revision B.02.20

1. The following functions have been added.
    - hpe5022_headLoadType
    - hpe5022_headLoadType_Q
    - hpe5022_headLoadRampConfig
    - hpe5022_headLoadRampConfig_Q

Changes in Revision B.02.20 from Revision B.02.00

1. The following 22 functions have been added.
    - hpe5022_snrTripleTrackWindowConfig
    - hpe5022_snrTripleTrackWindowConfig_Q
    - hpe5022_snrTripleTrackPositionConfig
    - hpe5022_snrTripleTrackPositionConfig_Q
    - hpe5022_snrTripleTrackAdjacentTrackConfig
    - hpe5022_snrTripleTrackAdjacentTrackConfig_Q
    - hpe5022_snrTripleTrackEraseBandConfig
    - hpe5022_snrTripleTrackEraseBandConfig_Q
    - hpe5022_snrTripleTrackSnrThresholdConfig
    - hpe5022_snrTripleTrackSnrThresholdConfig_Q
    - hpe5022_snrTripleTrackNoiseConstantConfig
    - hpe5022_snrTripleTrackNoiseConstantConfig_Q
    - hpe5022_snrTripleTrackNoiseThresholdConfig
    - hpe5022_snrTripleTrackNoiseThresholdConfig_Q
    - hpe5022_snrTripleTrackFilterConfig
    - hpe5022_snrTripleTrackFilterConfig_Q
    - hpe5022_snrTripleTrackSectorNormalizeConfig
    - hpe5022_snrTripleTrackSectorNormalizeConfig_Q
    - hpe5022_measureSnrTripleTrack
    - hpe5022_setupSnrTripleTrack
    - hpe5022_snrTripleTrack_Q
    - hpe5022_snrTripleTrackProfile_Q

Changes in Revision B.02.00 from Revision B.01.20

1. The following functions have been added.
    - hpe5022_popcornNoiseWritePatternConfig
    - hpe5022_popcornNoiseWritePatternConfig_Q
    - hpe5022_popcornNoiseCtWritePatternConfig
    - hpe5022_popcornNoiseCtWritePatternConfig_Q
    - hpe5022_headAmpInfoFile
    - hpe5022_headAmpInfoFile_Q
    - hpe5022_headAmpIcName_Q

Changes in Revision B.01.20 from Revision B.01.10.01

No changes in functions.

Service Release for Revision B.01.00

Changes at Revision B.01.00.01

1. The following functions have been added.
   - hpe5022_motorStateEx
   - hpe5022_motorStateEx_Q

Changes in Revision B.01.00 from Revision A.04.30.03

1. The following functions have been added.
   - hpe5022_headAmpRegister
   - hpe5022_headAmpRegister_Q
   - hpe5022_subModuleIdn_Q
   - hpe5022_pwReferenceRange_Q
   - hpe5022_writeDataSource
   - hpe5022_writeDataSource_Q
   - hpe5022_autoConfigEraseBand
   - hpe5022_autoConfigEraseBand_Q
   - hpe5022_autoConfigTrackProfile
   - hpe5022_autoConfigTrackProfile_Q
2. The error message "VI_ERROR_NSUP_OPER" has been changed to "hpe5022_ERROR_NSUP_OPER".
3. The following functions have been renamed. The old functions are also available.

Old	**New**
hpe5022_spinStandTtlOutput	hpe5022_spinStand_ttlOutput
hpe5022_spinStandTtlOutput_Q	hpe5022_spinStand_ttlOutput_Q
hpe5022_spinStandTtlInput_Q	hpe5022_spinStand_ttlInput_Q

Service Release for Revision A.04.30

Changes at Revision A.04.30.03

1. The following functions has been modified. The maximum number of repeat for multiple write/erase is extended from 1000 to 1000000.
    - hpe5022_writeMultiple
    - hpe5022_eraseMultiple

Changes at Revision A.04.30.02

1. The following functions has been added.
    - hpe5022_spinStandTtlOutput
    - hpe5022_spinStandTtlInput_Q

Changes at Revision A.04.30.01

1. The following functions has been added.
    - hpe5022_senseStimulusMode
    - hpe5022_senseStimulusMode_Q
    - hpe5022_senseStimulus
    - hpe5022_senseStimulus_Q
    - hpe5022_senseStimulusRange_Q
    - hpe5022_senseReferenceDcr
    - hpe5022_senseReferenceDcr_Q
    - hpe5022_measureSenseStimulusSweep
    - hpe5022_setupSenseStimulusSweep
    - hpe5022_measureTrackOffsetPositionError
    - hpe5022_trackOffsetPositionError_Q
    - hpe5022_trackOffsetUserSettlingTime
    - hpe5022_trackOffsetUserSettlingTime_Q
    - hpe5022_trackOffsetImmediateQ

Changes in Revision A.04.30 from Revision A.04.30.04

1. The following functions has been added.
    - hpe5022_trackOffsetCompPatternPosition
    - hpe5022_trackOffsetCompPatternPosition_Q
    - hpe5022_diskSizeRestriction
    - hpe5022_diskSizeRestriction_Q
    - hpe5022_headPositionSpeed
    - hpe5022_headPositionSpeed_Q
    - hpe5022_headPositionSpeedRange_Q
    - hpe5022_BER_optimizeSequenceConfig
    - hpe5022_BER_optimizeSequenceConfig_Q

Service Release for Revision A.04.20

Changes at Revision A.04.20.02

1. The following functions has been added.
    - hpe5022_trackOffsetCompInterval
    - hpe5022_trackOffsetCompInterval_Q
    - hpe5022_narrowBandTaaTawDataSize_Q
    - hpe5022_narrowBandTaaTawData_Q
    - hpe5022_narrowBandTaaTawStatistic_Q

Changes at Revision A.04.20.01

1. The following functions has been added.
    - hpe5022_eraseWideBand
    - hpe5022_searchTrack

Changes in Revision A.04.20 from Revision A.04.10

1. The following functions has been added.
    - hpe5022_writeMultiple
    - hpe5022_eraseMultiple
    - hpe5022_senseCurrentPolarity
    - hpe5022_senseCurrentPolarity_Q
    - hpe5022_trackOffsetCompProfile
    - hpe5022_trackOffsetCompProfile_Q
    - hpe5022_trackOffsetCompPattern
    - hpe5022_trackOffsetCompPattern_Q
    - hpe5022_trackOffsetCompPatternAuto
    - hpe5022_trackOffsetCompPatternState
    - hpe5022_trackOffsetCompPatternState_Q
    - hpe5022_writeTrackOffsetCompPattern
    - hpe5022_trackOffsetCompRevolution
    - hpe5022_trackOffsetCompRevolution_Q
    - hpe5022_executeTrackOffsetComp
    - hpe5022_trackOffsetCompValue
    - hpe5022_trackOffsetCompValue_Q
    - hpe5022_releaseSeqAll

Changes in Revision A.04.10 from Revision A.04.00.01

1. The following functions has been added.
    - hpe5022_measureSideReading
    - hpe5022_setupSideReading
    - hpe5022_sideReading_Q
    - hpe5022_sideReadingDataSize_Q
    - hpe5022_sideReadingData_Q
    - hpe5022_sideReadingStatistic_Q
    - hpe5022_measureSideErase
    - hpe5022_setupSideErase
    - hpe5022_sideErase_Q
    - hpe5022_sideEraseDataSize_Q
    - hpe5022_sideEraseData_Q
    - hpe5022_sideEraseStatistic_Q
    - hpe5022_stabilitySequenceConfig
    - hpe5022_stabilitySequenceConfig_Q
2. The parameter of ‘hpe5022_MICRO_TRACK_STEP_MIN’ in the hpe5022_nucriTrackConfig function has been changed from 50 nm to 1 nm.

Service Release for Revision A.04.00

Changes in Revision A.04.00.01 from Revision A.04.00

1. In Bit Error Measurement, the following functions are added.
   - hpe5022_BER_adjacentTrackPattern
   - hpe5022_BER_adjacentTrackPattern_Q

Changes in Revision A.04.00 from Revision A.03.81.04

1. The resource names of E5037B and E5040A have been added in hpe5022_init function.
2. The video band width in spectrum measurement is set at the same value of the resolution band width in case of E5040A.
   - hpe5022_spectrumBandWidth
   - hpe5022_spectralSnrBandWidth

Service Release for Revision A.03.81

Changes in Revision A.03.81.04 from Revision A.03.81.03

The following functions are added.

* hpe5022_measureChannelBitRateSweepOptSa
* hpe5022_setupChannelBitRateSweepOptSa
* hpe5022_channelBitRateSweepOptSaData_Q
* hpe5022_measureRollOffOptSa
* hpe5022_setupRollOffOptSa
* hpe5022_rollOffOptSaData_Q
* hpe5022_calculateD50OptSa_Q
* hpe5022_spectrumOptSaFrequency
* hpe5022_spectrumOptSaFrequency_Q
* hpe5022_spectrumOptSaBandWidth
* hpe5022_spectrumOptSaBandWidth_Q
* hpe5022_measureSpectrumOptSa
* hpe5022_setupSpectrumOptSa
* hpe5022_spectrumOptSaData_Q
* hpe5022_spectrumOptSaDataAll_Q
* hpe5022_spectrumOptSaNoise_Q
* hpe5022_spectralSnrOptSaFrequency
* hpe5022_spectralSnrOptSaFrequency_Q
* hpe5022_spectralSnrOptSaBandWidth
* hpe5022_spectralSnrOptSaBandWidth_Q
* hpe5022_measureSpectralSnrOptSa
* hpe5022_setupSpectralSnrOptSa
* hpe5022_spectralSnrOptSa_Q

The following function has been modified.

* hpe5022_init

Changes in Revision A.03.81.02 from Revision A.03.81.01

When the hpe5022_setupXXXXX functions related to E5038A, such as hpe5022_setupTaa(), are performed repeatedly, the measurement speed will be improved.

Changes in Revision A.03.81 from Revision A.03.70

The following function is added.

- hpe5022_spindleAccelerationRange_Q

(This function returns the maximum and minimum allowable accelerations of the spindle which is used as a reference to specify the acceleration “accel” in the “hpe5022_spindleAcceleration” function.)

The parameter “hpe5022_BER_COUN_MAX” has been modified for the following functions. Refer to Table A-1 “Parameter Modification”.

- hpe5022_BER_measureBer
- hpe5022_BER_setupBer

Table A-1 Parameter Modification

Parameter	Max.(Old)	Max.(New)
hpe5022_BER_COUN_MAX	100	**64**

The parameter “segment” in the “hpe5022_measureStability” function has been modified. The user must take the specified RPM into account when setting the gate time parameters (i.e, write time, delay and read time) for the following functions.

- hpe5022_measureStability
- hpe5022_setupStability

Timing Limitations
RPM = spindle speed, when spindle speed ≥ 2400 rpm
RPM = 2400 rpm, when spindle speed < 2400 rpm

The following functions have been added for Stability Measurement. These newly incorporated functions allow the user to set up the gate timing of read and write operation by absolute time.

- hpe5022_stabilityStatistic_Q
- hpe5022_stabilityAbsWritePatternConfig
- hpe5022_stabilityAbsWritePatternConfig_Q
- hpe5022_stabilityAbsWriteModeConfig
- hpe5022_stabilityAbsWriteModeConfig_Q
- hpe5022_measureStabilityAbs
- hpe5022_setupStabilityAbs

- hpe5022_stabilityAbsDataSize_Q
- hpe5022_stabilityAbs_Q
- hpe5022_stabilityAbsData_Q
- hpe5022_stabilityAbsStatistic_Q
- hpe5022_stabilityAbsSenseModeConfig
- hpe5022_stabilityAbsSenseModeConfig_Q

The following functions have been added for BER Measurement.

- hpe5022_BER_trackProfileDataEx_Q

  (This function returns the number of lost sectors, error symbols, total sectors, etc. of the bathtub test)

Changes in Revision A.03.70 from Revision A.03.60

The following functions have been added.

- hpe5022_writeBurst
- hpe5022_setupWriteBurst
- hpe5022_burstPatternConfig
- hpe5022_burstPatternConfig_Q
- hpe5022_burstFrequencyConfig
- hpe5022_burstFrequencyConfig_Q
- hpe5022_burstGateConfg
- hpe5022_burstGateConfig_Q

(The functions mentioned above allows the user to write burst patterns as position information on the disk media.)

The following functions have been added for Head Amplifier.

- hpe5022_updateHeadAmp
- hpe5022_headAmpSerialNumber_Q
- hpe5022_headAmpBufferName_Q
- hpe5022_headAmpBufferSerialNumber_Q

(The functions mentioned above are used to report the part number and serial number of the head amplifier's buffer board and to update the head amplifier's information.)

The minimum amount of read time for popcorn noise test has been modified for the following parameters.

- hpe5022_POPCORN_READ_TIME_MIN
- hpe5022_POPCORN_CT_READ_TIME_MIN

Table A-2 Read Time of Popcorn Noise Test

Module	Parameter	Min.(Old)	Min.(New)
E5038A (Parametric)	hpe5022_POPCORN_READ_TIME_MIN	10×10^{-6}	**1×10^{-6}**
E5041A (Dual Counter)	hpe5022_POPCORN_CT_READ_TIME_MIN	10×10^{-6}	**1×10^{-6}**

Changes in Revision A.03.60 from Revision A.03.50

The following functions have been added to allow users to change the acceleration of the spindle rotation in order to avoid disk slip.

- hpe5022_spindleAcceleration
- hpe5022_spindleAcceleration_Q

The limit of the skew parameters of the functions below have been modified. (The minimum skew angle is increased from -30 to -60 degrees.)

- hpe5022_headLoadConfigEx
- hpe5022_headPosition

The maximum and minimum number of repetitive periods have been modified for the following functions.

- hpe5022_overwriteLfPattern
- hpe5022_overwriteHfPattern
- hpe5022_lfPattern
- hpe5022_hfPattern
- hpe5022_IsolatedPulsePattern

Table A-3 Number of Repetitive Periods

Data Pattern	Min.(Old)	Min.(New)	Max. (Old)	Max. (New)
Overwrite LF	4	**1**	20	**50**
Overwrite HF	1	**1**	4	**50**
LF	4	**1**	20	**50**
HF	1	**1**	4	**50**
Isolated Pulse	5	**1**	50	**50**

For BER test the following functions have been added to allow users to save and recall the setup registers in the EEPROM.

- hpe5022_BER_channelIcRegistersSave
- hpe5022_BER_channelIcRegistersRecall

For BER test the following function has been modified. When track format is set to one sector per revolution, the read back raw data information can not be acquired.

- hpe5022_BER_errorRawData_Q

Changes in Revision A.03.60 from Revision A.03.50

The following functions have been added for SHNR (sub-harmonic noise ratio) measurement.

- hpe5022_shnrConfig
- hpe5022_shnrConfig_Q
- hpe5022_shnrWritePatternConfig
- hpe5022_shnrWritePatternConfig_Q
- hpe5022_shnrWriteCurrentConfig
- hpe5022_shnrWriteCurrentConfig_Q
- hpe5022_shnrWriteModeConfig
- hpe5022_shnrWriteModeConfig_Q
- hpe5022_measureShnr
- hpe5022_setupShnr
- hpe5022_shnr_Q
- hpe5022_shnrData_Q

Changes in Revision A.03.50 from Revision A.03.40

The following functions have been added for Popcorn Noise Test using E5041A.

- hpe5022_popcornNoiseCtConfig
- hpe5022_popcornNoiseCtConfig_Q
- hpe5022_popcornNoiseCtWriteModeConfig
- hpe5022_popcornNoiseCtWriteModeConfig_Q
- hpe5022_popcornNoiseCtHysteresisConfig
- hpe5022_popcornNoiseCtHysteresisConfig_Q
- hpe5022_measurePopcornNoiseCt
- hpe5022_setupPopcornNoiseCt
- hpe5022_popcornNoiseCt_Q

(The following functions have been added for Absolute Polarity Measurement) These newly incorporated functions allows the user to specify his own data pattern.

- hpe5022_absolutePolarityPatternMode
- hpe5022_absolutePolarityPatternMode_Q
- hpe5022_absolutePolarityPattern
- hpe5022_absolutePolarityPattern_Q

Changes in Revision A.03.40 from Revision A.03.30

The following functions have been added.

- “hpe5022_measureAbsolutePolarity”
- “hpe5022_setupAbsolutePolarity”
- “hpe5022_absolutePolarity_Q”
- “hpe5022_absolutePolarityIsolatedPulseTaa_Q”
- “hpe5022_absolutePolarityTribitTaaDataSize_Q”
- “hpe5022_absolutePolarityTribitTaaData_Q”
- “hpe5022_absolutePolarityTribitTaaStatistic_Q”

(The above functions allows you to measure Absolute Polarity test.)

- “hpe5022_seqSaGateConfig”
- “hpe5022_seqSaGateConfig_Q”
- “hpe5022_seqBaseSegGateConfig”
- “hpe5022_seqBaseSegGateConfig_Q”
- “hpe5022_seqBaseSegWriteReadGateConfig”
- “hpe5022_seqBaseSegWriteReadGateConfig_Q”

(The above functions allows you to read or write data during the specified period on a track.)

- “hpe5022_repetitivePatternEx”
- “hpe5022_repetitivePatternEx_Q”
- “hpe5022_userPatternEx”
- “hpe5022_userPatternEx_Q”

(This function allows you to define more repetitive and user patterns.)

- “hpe5022_stabilityWriteModeConfig”
- “hpe5022_stabilityWriteModeConfig_Q”

(This function allows you to change the first write polarity for each segment on a write-read sequence.)

- “hpe5022_stabilityWritePatternConfig”
- “hpe5022_stabilityWritePatternConfig_Q”

(This function allows you to change the data pattern on a write sequence from one on a write-read sequence.)

- “hpe5022_BER_triggerDelayTime”
- “hpe5022_BER_triggerDelayTime_Q”

(This function allows you to set a trigger delay on the bit error measurement.)

The following function has a new values for the parameter to allows you to change the first write polarity for each segment on a write-read sequence.

- “hpe5022_seqConfiguration”

The following functions has some new values for the parameter to select repetitive and user patterns.

- “hpe5022_selectPattern”
- “hpe5022_createSeqBaseSegErase”
- “hpe5022_createSeqBaseSegWrite”
- “hpe5022_createSeqBaseSegRead”
- “hpe5022_createSeqBaseSegWriteRead”

Changes in Revision A.03.30 from Revision A.03.2x

The following functions have been added.

- hpe5022_executionMode
- hpe5022_executionMode_Q
- hpe5022_lastMeasureState_Q

(This function specifies the type of execution of the setup measurement sequence)

- hpe5022_wai

(This function waits for all modules to complete a measurement sequence.)

The following functions have been modified.

- hpe5022_filterInformation_Q
- hpe5022_hgaCassette
- hpe5022_hgaCassette_Q

(The parameter “hppn” was changed to “part_num”. This change is in line with the transition from Hewlett-Packard to Agilent Technologies name brand.

Importing the library and including declaration file for software Rev3.30 and later revisions.

When you update the system software the new destination directory for Rev 3.30 will be C:\Program Files\Agilent\E5022. In line with this the set PATH of the “autoexec.bat” will be changed. In addition the path name of the import library of E5022A VEE will also be changed. Before you install software Rev3.30 it is recommended that you uninstall Rev3.21. However, when you uninstall Rev3.21and earlier revisions, use the installer process to uninstall. “. Do not use the “Control Panel>>Add/Remove Program” icon or any other manual means to uninstall.

- You must import the library file and include the declaration file at the beginning of every program that contains Agilent Technologies E5022A function calls.

  When you create your own application programs either by VEE, Visual Basic or Visual C++, you need to do the following.

  — For VEE, the file “hpe5022_32.dll” under the directory “c:\Program Files\Agilent\E5022\bin” should be imported. The file named “hpe5022_vee.h” under the directory “c:\Program Files\Agilent\E5022\include” should be included as a definition file. For Rev3.30 you need to change the path name of the “hpe5022_32.dll” and recompile the .vxe files

  — For Visual Basic, the file “hpe5022.bas” under the directory “c:\Program Files\Agilent\ E5022\include” should be added in the development environment as an object file.

  — For C language, the file “hpe5022.h” under the directory “c:\Program Files\Agilent\E5022\include” should be included. The file named

“hpe5022_32.lib” under the directory “c:\Program Files\Agilent\E5022\lib” should be linked. You need to change the path setting of the “include/LIB” files in the development environment for software Rev3.30.

Changes in Revision A.03.2x from Revision A.03.10

The following functions have been added.

- hpe5022_measureMicroTrackProfile
- hpe5022_setupMicroTrackProfile
- hpe5022_microTrackProfileData_Q
- hpe5022_microTrackProfileTrackWidth_Q

(The above functions allows you to measure a micro track profile faster than the method using hpe5022_createMicroTrack.)

- hpe5022_hgaCassetteDimension
- hpe5022_hgaCassetteDimension_Q

(This function allows you to use a customer designed HGA cassette.)

- hpe5022_taaModulation_Q

(This function allows you to measure Modulation in the TAA measurement.)

The following functions have been modified. Modification of your program is required when you use these functions in revision A.03.10.

- hpe5022_spectrumBandWidth
- hpe5022_spectrumBandWidth_Q
- hpe5022_spectralSnrBandWidth
- hpe5022_spectralSnrBandWidth_Q
- hpe5022_seqConfiguration

  (The video band width parameter is added.)

Changes in Revision A.03.1x from Revision A.03.0x

The following functions have been added.

- hpe5022_spectrumDisplayState
- hpe5022_spectrumDisplayState_Q

(This function allows you to monitor the data on the screen of the Spectrum Analyzer.)

- hpe5022_tripleTrackAdjacentTrackConfig
- hpe5022_tripleTrackAdjacentTrackConfig_Q

(This function allows you to select a data pattern for the adjacent track. Erase can be also selected as a data pattern.)

- hpe5022_subIdn_Q

(This function allows you to query the name, the firmware revision and the serial number of system components)

For bit error rate, the parameter ‘ave’ (average) has been added to the following functions.

- hpe5022_BER_measureBer
- hpe5022_BER_setupBer
- hpe5022_BER_measureTrackProfile
- hpe5022_BER_setupTrackProfile
- hpe5022_BER_747Config
- hpe5022_BER_747Config_Q

The parameter ‘hpe5022_BER_OPT_INIT’ has been added to the following functions.

- hpe5022_BER_optimizeState
- hpe5022_BER_optimizeState_Q

The parameter ‘error count’ has been added to this function.

- hpe5022_BER_channelQualityRawData_Q

The parameter ‘total sector’ has been deleted from the following functions.

- hpe5022_BER_sectorErrorCount_Q (total sector deleted)
- hpe5022_BER_channelIcRegisterSweepData_Q (total sector deleted)

The parameters of the following functions have been modified.

- hpe5022_popcornNoiseConfig

- hpe5022_popcornNoiseConfig_Q
- hpe5022_setupPopcornNoise
- hpe5022_measurePopcornNoise

  (The read time parameter has been added, this sets the time to read the data)

The following function has been modified.

- hpe5022_spindleSpeed

  (The spindle speed can be changed during drive on.)

The following function of the user defined sequence have been modified.

- hpe5022_createSeqBaseSegRead

  (The erase pattern is added to the data pattern parameter. The ‘hpe5022_MEAS_SPEC’ is added to the read option.)
- hpe5022_createSeqBaseSegWrite

  (The erase pattern is added to the data pattern parameter.)
- hpe5022_createSeqBaseSegWriteRead

  (The erase pattern is added to the data pattern parameter and the write and read data patterns can be set independently.)
- hpe5022_seqConfiguration

  (The parameter for spectrum measurement is added.)

Bug Fix, an error ‘invalid parameter’ occurs when the defauult setting for 2.5 inch clamp is executed. This bug is fixed in Rev 3.10 from its previous revision.

Changes at Revision A.3.01 from Revision A.3.00

The following functions have been added and /or modified.

- For Triple Track Test, the function “hpe5022_tripleTrackAdjacentTrackConfig” is added. This function allows you to set different data patterns for the adjacent track.
- For Popcorn Noise Measurement, the read time parameter is added in the “hpe5022_popcornNoiseConfig” Also the parameter “hpe5022_POPCORN_ WRIT_TIME_MIN” has been modified to 10×10^{-6} from 100×10^{-6}
- For Popcorn Noise Measurement, the ‘segment’ and ‘rev’ parameters have been removed from the “hpe5022_measurePopcornNoise” and “hpe5022_setupPopcornNoise” functions. In replacement the parameter ‘counts’ is added.
- For User Defined Sequence, the parameter ‘datPat’ in the “hpe5022_createSeqbaseSegWriteRead” function has been changed and divided into two parameters, namely, ‘writPat’ and ‘readPat’. These two new parameters allows you to specify different write and read data patterns.
- For UDS, four erase data patterns have been added in the data pattern ‘datpat’ setting of the base segment, namely, “hpe5022_PAT_ERASE”, “hpe_5022_PAT_ERASE_DC_NEG”, “hpe_5022_PAT_ERASE_DC_POS” and “hpe5022_PAT_ERASE_AC”.

Changes in Revision A.03.00 from Revision A.02.1x

The following functions are added.

- BER Measurement Functions
- Spectrum Measurement Functions
- Spectral SNR Measurement Functions
- hpe5022_spindleSpeedRange_Q
- hpe5022_headLoadImmediate
- hpe5022_idn_Q
- hpe5022_tripleTrackAdjacentTrackConfig
- hpe5022_tripleTrackAdjacentTrackConfig_Q

The following parameter values have been changed.

- hpe5022_TRACK_MAX is changed from 20,000 to 1,000,000.
- hpe5022_POPCORN_SEG_PER_TRACK_MAX is changed from 5 to 10.

The type of parameter of the following function have been changed from “ViString” to “ViChar”.

- hpe5022_userPattern
- hpe5022_userPattern_Q

Changes in Revision A.02.1x from Revision A.02.0x

The following functions are added.

- hpe5022_headLoadConfigEx
- hpe5022_headLoadConfigEx_Q
- hpe5022_trackToRadiusSkew
- hpe5022_isolatedPulseReferenceTaaAuto
- hpe5022_isolatedPulseReferenceTaaAuto_Q
- hpe5022_setupEraseBand
- hpe5022_popcornNoiseConfig
- hpe5022_measurePopcornNoise
- hpe5022_setupPopcornNoise
- hpe5022_popcornNoise_Q
- hpe5022_popcornNoiseHistogram_Q

The function to specify the noise reference level is added in order to improve the measurement accuracy. This change will cause a difference in measurement results between revision 2.0x and 2.1x.

- hpe5022_noiseReference_Q

The parameters of the following function has been changed.

- hpe5022_snr_Q

The baseline parameter is deleted from the parametric measurement function. The following functions have been deleted.

- hpe5022_parametricBaseline_Q
- hpe5022_parametricBaselineDataSize_Q
- hpe5022_parametricBaselineData_Q
- hpe5022_parametricBaselineStatistic_Q

The baseline parameter of the function below has been changed.

- hpe5022_parametric_Q

The following functions have been changed. The value of OTRC Write Pos. is changed from the absolute value to the ratio value of write track width.

- hpe5022_tripleTrackPositionConfig
- hpe5022_tripleTrackPositionConfig_Q

The functions have been changed. Performing the three Track Erase process is selectable.

- hpe5022_measureTripleTrack
- hpe5022_setupTripleTrack

Index

Index