

Agilent E5070B/E5071B ENA Series RF Network Analyzers

Duplexer Measurements

Second Edition



Agilent Technologies

No. 16000-95009

August 2002

Notices

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

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Agilent Technologies.

Agilent Technologies Japan, Ltd.

Component Test PGU-Kobe

1-3-2, Murotani, Nishi-ku, Kobe, Hyogo, 651-2241 Japan

MS-DOS®, Windows®, Windows 98, Windows NT®, Visual C++®, Visual Basic®, VBA, Excel and PowerPoint® are U.S. registered trademarks of Microsoft Corporation.

Portions ©Copyright 1996, Microsoft Corporation. All rights reserved.

© Copyright Agilent Technologies Japan, Ltd. 2002

Sample Program

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 purposes 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 any responsibility for the operation of the SAMPLE PROGRAMS to be uninterrupted or error-free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS ANY IMPLIED WARRANTY 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 right 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.

Application Programs

This chapter describes sample programs (VBA programs) based on actual measurement examples.

Measuring a multi-port device

Example 2-1 shows a sample program (VBA program) that demonstrates how to measure a (3-terminal) duplexer. You can find the source file of this program, named “apl_sys.vba”, on the sample program disk. This VBA program consists of the following modules:

NOTE

For the E5070B/E5071B with Option 213 or 214 (2-port S-parameter test set), a runtime error occurs because there are parameters that it cannot measure.

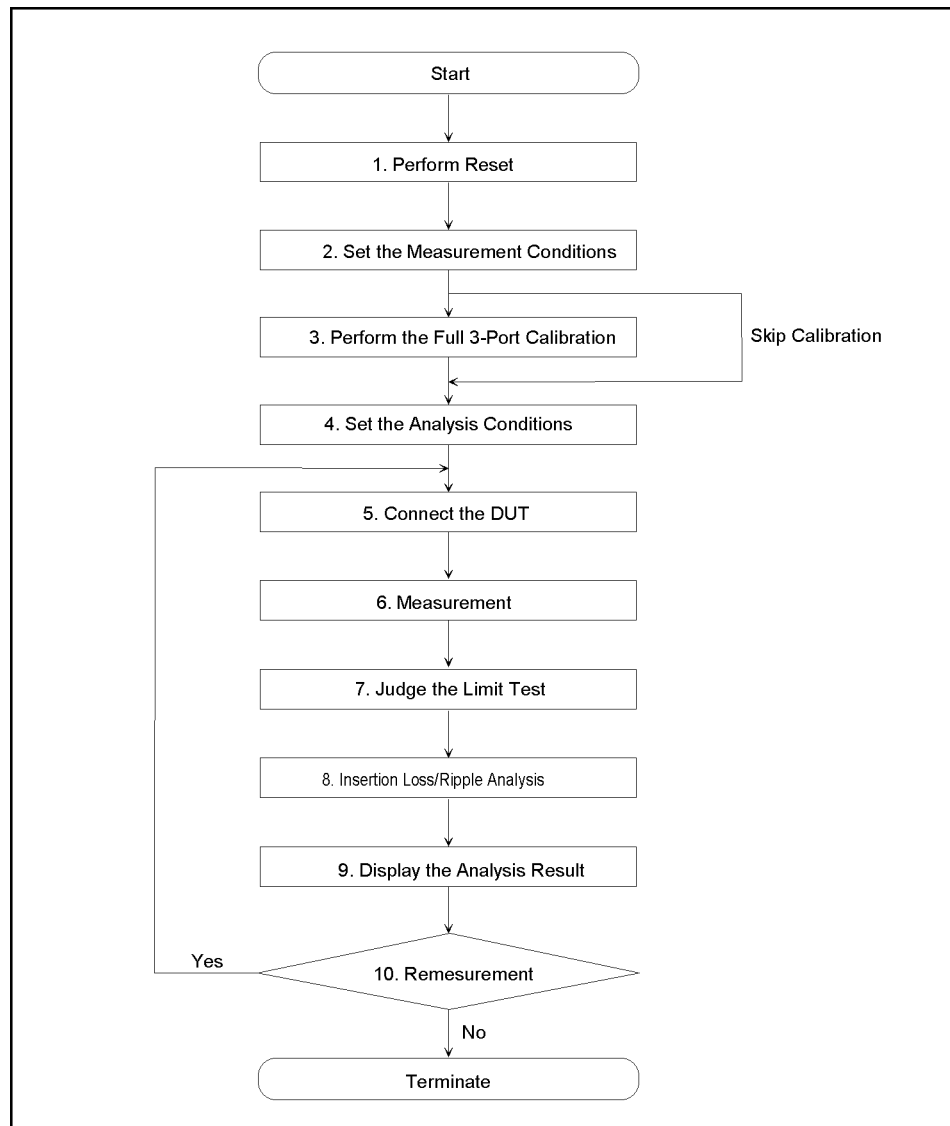
Object name	Module type	Description
frmDupRes	User form	Displays the analysis result.
mdlDupMeas	Standard module	Performs duplexer measurement.

Overview of the program

The program performs full 3-port calibration using the 85032F calibration kit, measure a (3-terminal) duplexer, and calculates and displays the limit test result, insertion loss, and band-pass ripple. Figure 1 shows the simple flow of the (3-terminal) duplexer measurement program.

Figure 1

Flow of duplexer measurement



e5070ave035

Description of the program

When you run this VBA program, reset is performed, the measurement conditions are set, and a message "Perform the full 3-port calibration." is displayed. To perform the full 3-port calibration, click the **Yes** button; to skip it, click the **No** button.

To perform the calibration, follow the onscreen messages to connect each standard of the Agilent 85032F calibration kit to the specified port, and click the **OK** button to measure the calibration data. Click the **Cancel** button to return to the beginning of the calibration. You cannot skip the isolation calibration. When the calibration data measurement for all standards is complete, a message "All calibration data completion." is displayed, and the calibration coefficient is calculated.

NOTE

When you cancel the calibration data measurement before completing the measurement of necessary calibration data, the settings condition may not be returned to its former state.

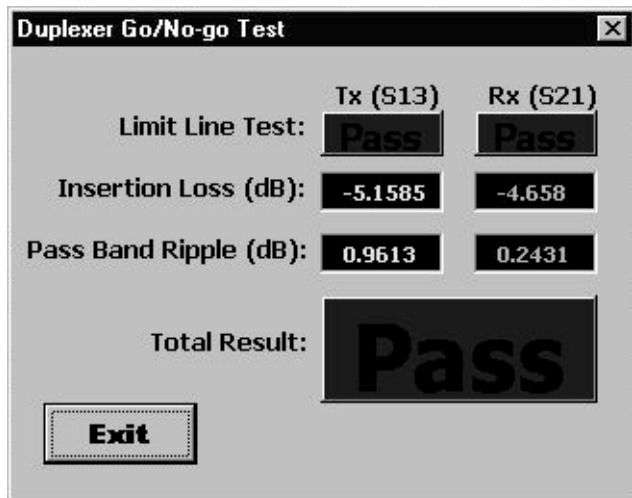
Then, the limit line is set and the setting required for the limit test judgment is performed.

The message "Set the DUT between test cables." is displayed. Connect the DUT (duplexer) between the test cables and click the **Yes** button. The limit line is set and a single sweep is executed. Then, for each of trance 1 (Tx: S13) and trace 2 (Rx: S21), the Pass/Fail judgment of the limit test and the insertion loss and ripple analysis result within the pass band (Figure 2) are displayed.

Click the **Exit** button on the user form to display the analysis result (Figure 2). The message prompting for remeasurement is displayed. To perform remeasurement, click the **Yes** button; to terminate the program, click the **No** button. Note that the detail of the program within the user form to display the analysis result (object name: frmDupRes) is not described here.

Figure 2

Display of the execution result of the program of Example 2-1



e5070avj041

The duplexer measurement program (object name: mdlDupMeas) is described in detail below. Line numbers are added for description purpose only, and do not appear in the actual program source code.

Line 90	Stores the calibration kit number (4: 85032F) to the CalKit variable.
Line 110	Turns off display update. Turning off display update shortens drawing time and object processing time.
Line 140	Returns the E5070B/E5071B to the preset state.
Lines 170 to 180	For channel 1, turns on the continuous trigger startup mode to on and sets the trigger source to the bus trigger.
Line 200	Calls the Setup_Parameter procedure (lines 910 to 1140). For information on the Setup_Parameter procedure, see the description later.
Line 220	Calls the Setup_Segment procedure (lines 1160 to 1530). For information on the Setup_Segment procedure, see the description later.
Line 250	Stores the calibration kit number for channel 1 to the CalKit variable.
Line 260	Stores 1, 2, and 3 to the Port variable that indicates ports used for the full 3-port calibration.
Line 280	Calls the Calib_Solt procedure (lines 1550 to 2420). For information on the Calib_Solt procedure, see the description in Example 6-1, “Measuring a band-pass filter (object name: mdlBscMeas),” on page 89 later.
Line 310	Calls the Setup_Limitline procedure (lines 2420 to 3180). For information on the Setup_Limitline procedure, see the description later.
Line 330	Calls the Setup_Register procedure (lines 3200 to 3260). For information on the Setup_Register procedure, see the description later.
Line 380	Displays the message that prompts for connecting a DUT (Device Under Test) and waits for clicking the OK button after the connection.
Line 410	Clears the questionable limit status event register and questionable limit channel 1 status event register.
Lines 420 to 430	Generates a trigger to start a single sweep and waits until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).
Lines 450 to 460	For traces 1 and 2, executes the auto scale to set the optimum scale.
Lines 490 to 500	Reads out the value of the questionable limit status event register, and stores the AND of the read-out value and 2 (the value in which only bit 1 is 1) into the Test_Ch1 variable.
Lines 510 to 530	Reads out the value of the questionable limit channel 1 status event register, and stores the AND of the read-out value and 2 (the value in which only bit 1 is 1) into the Test_Tr1 variable and the AND of the read-out value and 4 (the value in which only bit 2 is 1) into the Test_Tr2 variable.

Lines 560 to 610 Specifies trace 1 as the active trace, and sets the analysis range (start point: 1.85 GHz and stop point: 1.91 GHz). Then sets the polarity of the peak search (both the positive peak and the negative peak) and the lower limit of the peak excursion value (0).

Lines 620 to 640 Searches for the minimum value within the analysis range, and stores the analysis result into the IlossTx variables.

Line 650 Uses the ripple analysis function to store the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range into the RipTx variables.

Lines 670 to 720 Specifies trace 2 as the active trace, and sets the analysis range (start point: 1.93 GHz and stop point: 1.99 GHz). Then sets the polarity of the peak search (both the positive peak and the negative peak) and the lower limit of the peak excursion value (0).

Lines 730 to 750 Searches for the minimum value within the analysis range, and stores the analysis result into the IlossRx variables.

Line 760 Uses the ripple analysis function to store the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range into the RipRx variables.

Line 790 Calls the Display_Update procedure (lines 3280 to 3620). For information on the Display_Update procedure, see the description later.

Line 810 Displays the user form (object name: frmDupRes) on the screen to display the analysis result.

Lines 830 to 870 Displays the message asking you whether you want to perform measurement again. Click the **Yes** button to return to the DUT connection section. Click the **No** button to terminate the program.

Procedure: Setup_Parameter (lines 910 to 1140).

Lines 970 to 1020 Stores the channel layout ("D1": 1-channel display), graph layout ("D1_2": upper/lower 2 part split display), measurement parameter of trace 1 (S13), measurement parameter of trace 2 (S21), data format of trace 1 (MLOG), and data format of trace 2 (MLOG) into the ChDisp, TracDisp, Par(0), Par(1), Fmt(0), and Fmt(1) variables, respectively.

Lines 1040 to 1060 Sets the number of traces for channel 1 to 2, the channel layout to the ChDisp variable, and the graph layout for channel 1 to the TracDisp variable, respectively.

Lines 1080 to 1120 Sets the measurement parameter for trace 1 to the Par(0) variable, the data format for trace 1 to the Fmt(0) variable, the measurement parameter for trace 2 to the Par(1) variable, and the data format for trace 2 to the Fmt(1) variable, respectively.

Procedure: Setup_Segment (lines 1160 to 1530).

Lines 1200 to 1260 Stores the conditions for the channel 1's segment table setting into the SegmData(0) to SegmData(6) variables. The settings are as follows. The stimulus setting mode: start/stop value. The IF bandwidth setting for each segment: off. The power setting for each segment: off. The sweep delay time setting for each segment: off. The sweep time setting for each segment: off. The number of segments: 5.

Lines 1280 to 1470 Stores the sweep start value, the sweep stop value, and the number of measurement points for channel 1's segments 1 through 5 into the SegmData(7) to SegmData(21) variables, respectively.

Line 1490 Sets the channel 1's segment table to the SegmData variable.

Line 1500 Sets channel 1's sweep type to "segment".

Line 1510 Sets the channel 1's graph horizontal axis display method to the order base (the axis on which the measurement point numbers are placed evenly in the order of measurement).

Procedure: Calib_Solt (lines 1550 to 2420).

See the Lines 1200 to 2130 of the Example 6-1 on page 89.

Procedure: Setup_Limitline (lines 2440 to 3180).

Line 2490 Stores the number of lines (5) in trace 1 limit table into the LimDataS13(0) variable.

Lines 2500 to 2790 Stores the settings in trace 1 limit table into the LimDataS13(1) to LimDataS13(25) variables.

Line 2820 Stores the number of lines (4) in trace 2 limit table into the LimDataS21(0) variable.

Lines 2830 to 3060 Stores the settings in trace 2 limit table into the LimDataS21(1) to LimDataS21(20) variables.

Lines 3080 to 3110 Specifies trace 1 as the active trace, stores the trace 1's limit line into the LimDataS13 variable, and displays it. Then, turns on the limit test function for trace 1.

Lines 3130 to 3160 Specifies trace 2 as the active trace, stores the trace 2's limit line into the LimDataS21 variable, and displays it. Then, turns on the limit test function for trace 2.

Procedure: Setup_Register (lines 3200 to 3260).

Lines 3220 to 3230 Sets the instrument so that the questionable limit channel status event register's bits 1 and 2 are set to 1 only when the questionable limit channel status register's bits 1 and 2 are changed from 0 to 1 (positive transition).

Line 3240 Enables the questionable limit channel status event register's bits 1 and 2.

Procedure: Display_Update (lines 3280 to 3620).

Line 3300 Updates the display on the LCD screen once.

Lines 3320 to 3380 When the trace 1's limit test result is Fail (Test_Tr1 = 2), displays Tx(S13) "Limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when the trace 1's limit test result is Pass (Test_Tr1 \neq 2), displays Tx(S13) "Limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3400 to 3460 When the trace 2's limit test result is Fail (Test_Tr2 = 4), displays Rx(S21) "Limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when the trace 1's limit test result is Pass (Test_Tr2 \neq 4), displays Rx(S21) "Limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3480 to 3540 When the channel 1's limit test result is Fail (Test_Ch1 = 2), displays "Overall limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when the channel 1's limit test result is Pass (Test_Ch1 \neq 2), displays "Overall limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3560 to 3600 Displays the analysis results for traces 1 and 2 (insertion loss and band-pass ripple) as Tx(S13) and Rx(S21) on the user form (object name: frmDupRes).

Example 2-1

Duplexer measurement (object name: mdlDupMeas)

```
10| Sub Main()  
20|  
30| Dim CalKit As Long, Dmy As Long, Rgst As Long, I As Long,  
Buff As Long  
40| Dim Test_Ch1 As Integer, Test_Tr1 As Integer, Test_Tr2 As  
Integer  
50| Dim IlossTx As Variant, IlossRx As Variant  
60| Dim RipTx As Double, RipRx As Double  
70| Dim Port As Variant  
80|  
90| CalKit = 4                    'Calibration kit        :85032F  
100|  
110| SCPI.DISPlay.ENABLE = False  
120|  
130| '''Presetting the E5070B/E5071B  
140| SCPI.SYSTem.PRESet  
150|  
160| '''Setting measurement conditions  
170| SCPI.INITiate(1).CONTinuous = True  
180| SCPI.TRIGger.SEQuence.Source = "BUS"  
190|  
200| Setup_Parameter  
210|  
220| Setup_Segment  
230|  
240| '''Full 3-port calibration  
250| SCPI.SENSE(1).CORRection.COLLect.CKIT.Select = CalKit
```

```

260|     Port = Array(1, 2, 3)
270|
280|     Calib_Solt 1, 3, Port
290|
300|     '''Setting analysis conditions
310|     Setup_Limitline
320|
330|     Setup_Register
340|
350| Meas_Start:
360|
370|     '''Connecting DUT
380|     MsgBox "Connect DUT between test cables."
390|
400|     '''Performing single sweep
410|     SCPI.IEEE4882.CLS
420|     SCPI.TRIGger.SEQuence.SINGLE
430|     Dmy = SCPI.IEEE4882.OPC
440|
450|     SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.AUTO
460|     SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.AUTO
470|
480|     '''Judging limit test
490|     Rgst = SCPI.STATus.QUEStionable.LIMit.EVENT
500|     Test_Ch1 = CInt(Rgst And 2)
510|     Rgst = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).EVENT
520|     Test_Tr1 = CInt(Rgst And 2)
530|     Test_Tr2 = CInt(Rgst And 4)
540|
550|     '''Analyzing insertion loss/bandpass ripple
560|     SCPI.CALCulate(1).PARAmeter(1).Select
570|     SCPI.CALCulate(1).SELeCted.FUNcTion.DOMain.START =
1850000000#
580|     SCPI.CALCulate(1).SELeCted.FUNcTion.DOMain.STOP =
1910000000#
590|     SCPI.CALCulate(1).SELeCted.FUNcTion.DOMain.STATE = True
600|     SCPI.CALCulate(1).SELeCted.FUNcTion.PPOLarity = "both"
610|     SCPI.CALCulate(1).SELeCted.FUNcTion.PEXCursion = 0
620|     SCPI.CALCulate(1).SELeCted.FUNcTion.TYPE = "MIN"
630|     SCPI.CALCulate(1).SELeCted.FUNcTion.EXECute
640|     IlossTx = SCPI.CALCulate(1).SELeCted.FUNcTion.DATA
650|     RipTx = MaxPeakToPeak(1)
660|
670|     SCPI.CALCulate(1).PARAmeter(2).Select
680|     SCPI.CALCulate(1).SELeCted.FUNcTion.DOMain.START =
1930000000#
690|     SCPI.CALCulate(1).SELeCted.FUNcTion.DOMain.STOP =
1990000000#
700|     SCPI.CALCulate(1).SELeCted.FUNcTion.DOMain.STATE = True
710|     SCPI.CALCulate(1).SELeCted.FUNcTion.PPOLarity = "both"
720|     SCPI.CALCulate(1).SELeCted.FUNcTion.PEXCursion = 0
730|     SCPI.CALCulate(1).SELeCted.FUNcTion.TYPE = "MIN"
740|     SCPI.CALCulate(1).SELeCted.FUNcTion.EXECute
750|     IlossRx = SCPI.CALCulate(1).SELeCted.FUNcTion.DATA
760|     RipRx = MaxPeakToPeak(1)
770|
780|     '''Displaying the results
790|     Display_Update Test_Tr1, Test_Tr2, Test_Ch1, IlossTx,

```

```

IlossRx, RipTx, RipRx
800|
810|     frmDupRes.Show
820|
830|     Buff = MsgBox("Do you make another measurement?", vbYesNo,
"Duplexer Measurement")
840|
850|     If Buff = vbYes Then
860|         GoTo Meas_Start
870|     End If
880|
890| End Sub
900|
910| Private Sub Setup_Parameter()
920|
930|     Dim I As Long
940|     Dim ChDisp As String, TracDisp As String
950|     Dim Par(1) As String, Fmt(1) As String
960|
970|     ChDisp = "D1"
980|     TracDisp = "D1_2"
990|     Par(0) = "S13"
1000|     Par(1) = "S21"
1010|     Fmt(0) = "MLOG"
1020|     Fmt(1) = "MLOG"
1030|
1040|     SCPI.CALCulate(1).PARAMeter.Count = 2
1050|     SCPI.DISPlay.Split = ChDisp
1060|     SCPI.DISPlay.WINDow(1).Split = TracDisp
1070|
1080|     For I = 1 To 2
1090|         SCPI.CALCulate(1).PARAMeter(I).DEFINE = Par(I - 1)
1100|         SCPI.CALCulate(1).PARAMeter(I).Select
1110|         SCPI.CALCulate(1).SElected.Format = Fmt(I - 1)
1120|     Next I
1130|
1140| End Sub
1150|
1160| Private Sub Setup_Segment()
1170|
1180|     Dim SegmData(21) As Variant
1190|
1200|     SegmData(0) = 5                                'Anytime 5 is set at
segment settings
1210|     SegmData(1) = 0                                'Allows stimulus range
to be set using Start/Stop frequency
1220|     SegmData(2) = 0                                'Not allows IF bandwidth
to be set
1230|     SegmData(3) = 0                                'Not allows power to be
set
1240|     SegmData(4) = 0                                'Not allows delay time
to be set
1250|     SegmData(5) = 0                                'Not allows sweep time
to be set
1260|     SegmData(6) = 5                                'Number of segments
1270|
1280|     ''Segment 1
1290|     SegmData(7) = 1730000000#                        'Start frequency

```

```

1300|     SegmData(8) = 1830000000#           'Stop frequency
1310|     SegmData(9) = 50                   'Number of points
1320|     '''Segment 2
1330|     SegmData(10) = 1830000000#          'Start frequency
1340|     SegmData(11) = 2030000000#          'Stop frequency
1350|     SegmData(12) = 400                  'Number of points
1360|     '''Segment 3
1370|     SegmData(13) = 2030000000#          'Start frequency
1380|     SegmData(14) = 2130000000#          'Stop frequency
1390|     SegmData(15) = 50                   'Number of points
1400|     '''Segment 4
1410|     SegmData(16) = 3650000000#          'Start frequency
1420|     SegmData(17) = 4030000000#          'Stop frequency
1430|     SegmData(18) = 38                   'Number of points
1440|     '''Segment 5
1450|     SegmData(19) = 5500000000#          'Start frequency
1460|     SegmData(20) = 6020000000#          'Stop frequency
1470|     SegmData(21) = 52                   'Number of points
1480|
1490|     SCPI.SENSE(1).SEGMENT.DATA = SegmData
1500|     SCPI.SENSE(1).SWEep.TYPE = "SEGM"
1510|     SCPI.DISplay.WINDow(1).X.SPACing = "OBAS"
1520|
1530| End Sub
1540|
1550| Private Sub Calib_Solt(Chan As Long, SoltType As Long, Port
As Variant)
1560|
1570|     Dim Dmy As Long, I As Long, J As Long, Buff As Long
1580|
1590|     Cal_Start:
1600|
1610|     Buff = MsgBox("Perform the full " & SoltType & "-port
calibration.", vbOKCancel, "Full" & SoltType & "-port calibration")
1620|     If Buff = vbCancel Then
1630|         GoTo Cal_Skip
1640|     End If
1650|
1660|     Select Case SoltType
1670|         Case 1
1680|             SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT1 =
Port(0)
1690|         Case 2
1700|             SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT2 =
Port
1710|         Case 3
1720|             SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT3 =
Port
1730|         Case 4
1740|             SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT4 =
Port
1750|     End Select
1760|
1770|     For I = 1 To SoltType
1780|
1790|         Buff = MsgBox("Connect the Open standard to Port " &
CStr(Port(I - 1)) & ".", vbOKCancel, "Full" & SoltType & "-port
calibration")

```

```

1800|         If Buff = vbOK Then
1810|             SCPI.SENSE(Chan).CORRection.COLLECT.ACQUIRE.OPEN =
Port(I - 1)
1820|             Dmy = SCPI.IEEE4882.OPC
1830|         Else
1840|             GoTo Cal_Start
1850|         End If
1860|
1870|         Buff = MsgBox("Connect the Short standard to Port " &
CStr(Port(I - 1)) & ".", vbOKCancel, "Full" & SoltType & "-port
calibration")
1880|         If Buff = vbOK Then
1890|             SCPI.SENSE(Chan).CORRection.COLLECT.ACQUIRE.Short =
Port(I - 1)
1900|             Dmy = SCPI.IEEE4882.OPC
1910|         Else
1920|             GoTo Cal_Start
1930|         End If
1940|
1950|         Buff = MsgBox("Connect the Load standard to Port " &
CStr(Port(I - 1)) & ".", vbOKCancel, "Full" & SoltType & "-port
calibration")
1960|         If Buff = vbOK Then
1970|             SCPI.SENSE(Chan).CORRection.COLLECT.ACQUIRE.Load =
Port(I - 1)
1980|             Dmy = SCPI.IEEE4882.OPC
1990|         Else
2000|             GoTo Cal_Start
2010|         End If
2020|     Next I
2030|
2040|     For I = 1 To SoltType - 1
2050|         For J = I + 1 To SoltType
2060|             Buff = MsgBox("Connect the Thru standard between
Port " & CStr(Port(I - 1)) & " and Port " & CStr(Port(J - 1)) & ".",
vbOKCancel, "Full" & SoltType & "-port calibration")
2070|             If Buff = vbOK Then
2080|
SCPI.SENSE(Chan).CORRection.COLLECT.ACQUIRE.THROUGH = Array(Port(I -
1), Port(J - 1))
2090|                 Dmy = SCPI.IEEE4882.OPC
2100|
SCPI.SENSE(Chan).CORRection.COLLECT.ACQUIRE.THROUGH = Array(Port(J -
1), Port(I - 1))
2110|                 Dmy = SCPI.IEEE4882.OPC
2120|             Else
2130|                 GoTo Cal_Start
2140|             End If
2150|         Next J
2160|     Next I
2170|
2180|     If SoltType <> 1 Then
2190|         Buff = MsgBox("Do you measure the Isolation
(Optional)?", vbYesNo, "Full" & SoltType & "-port calibration")
2200|         If Buff = vbYes Then
2210|             For I = 1 To SoltType - 1
2220|                 For J = I + 1 To SoltType
2230|                     Buff = MsgBox("Connect the Load standard to

```

```

Port " & Port(I - 1) & " and Port " & Port(J - 1) & ".", vbOKCancel,
"Full" & SoltType & "-port calibration")
2240|         If Buff = vbOK Then
2250|
SCPI.SENSE(Chan).CORRection.COLLect.ACQuire.ISOLation =
Array(Port(I - 1), Port(J - 1))
2260|         Dmy = SCPI.IEEE4882.OPC
2270|
SCPI.SENSE(Chan).CORRection.COLLect.ACQuire.ISOLation =
Array(Port(J - 1), Port(I - 1))
2280|         Dmy = SCPI.IEEE4882.OPC
2290|         Else
2300|             GoTo Cal_Start
2310|         End If
2320|     Next J
2330| Next I
2340| End If
2350| End If
2360|
2370| SCPI.SENSE(1).CORRection.COLLect.SAVE
2380| MsgBox "All calibration data completion."
2390|
2400| Cal_Skip:
2410|
2420| End Sub
2430|
2440| Private Sub Setup_Limitline()
2450|
2460|     Dim LimDataS13(25) As Variant, LimDataS21(20) As Variant
2470|
2480|     '''Limit line for S13
2490|     LimDataS13(0) = 5                                'Number of segment
2500|     '''Limit_line 1
2510|     LimDataS13(1) = 1                                'Maximum
2520|     LimDataS13(2) = 1730000000#                      'Beginning of stimulus
2530|     LimDataS13(3) = 1930000000#                      'End of stimulus
2540|     LimDataS13(4) = 0                                'Beginning of response
2550|     LimDataS13(5) = 0                                'End of response
2560|     '''Limit_line 2
2570|     LimDataS13(6) = 2                                'Minimum
2580|     LimDataS13(7) = 1850000000#                      'Beginning of stimulus
2590|     LimDataS13(8) = 1910000000#                      'End of stimulus
2600|     LimDataS13(9) = -8                               'Beginning of response
2610|     LimDataS13(10) = -8                              'End of response
2620|     '''Limit_line 3
2630|     LimDataS13(11) = 1                               'Maximum
2640|     LimDataS13(12) = 1930000000#                    'Beginning of stimulus
2650|     LimDataS13(13) = 1990000000#                    'End of stimulus
2660|     LimDataS13(14) = -35                             'Beginning of response
2670|     LimDataS13(15) = -35                             'End of response
2680|     '''Limit_line 4
2690|     LimDataS13(16) = 1                               'Maximum
2700|     LimDataS13(17) = 1990000000#                    'Beginning of stimulus
2710|     LimDataS13(18) = 2130000000#                    'End of stimulus
2720|     LimDataS13(19) = -40                             'Beginning of response
2730|     LimDataS13(20) = -40                             'End of response
2740|     '''Limit_line 5
2750|     LimDataS13(21) = 1                               'Maximum

```

```

2760|     LimDataS13(22) = 2130000000#    'Beginning of stimulus
2770|     LimDataS13(23) = 6020000000#    'End of stimulus
2780|     LimDataS13(24) = -20             'Beginning of response
2790|     LimDataS13(25) = -20             'End of response
2800|
2810|     '''Limit line for S21
2820|     LimDataS21(0) = 4                 'Number of segment
2830|     '''Limit_line 1
2840|     LimDataS21(1) = 1                 'Maximum
2850|     LimDataS21(2) = 1730000000#      'Beginning of stimulus
2860|     LimDataS21(3) = 1850000000#      'End of stimulus
2870|     LimDataS21(4) = -40               'Beginning of response
2880|     LimDataS21(5) = -40               'End of response
2890|     '''Limit_line 2
2900|     LimDataS21(6) = 1                 'Maximum
2910|     LimDataS21(7) = 1850000000#      'Beginning of stimulus
2920|     LimDataS21(8) = 1910000000#      'End of stimulus
2930|     LimDataS21(9) = -40               'Beginning of response
2940|     LimDataS21(10) = -40              'End of response
2950|     '''Limit_line 3
2960|     LimDataS21(11) = 1                 'Maximum
2970|     LimDataS21(12) = 1910000000#      'Beginning of stimulus
2980|     LimDataS21(13) = 6020000000#      'End of stimulus
2990|     LimDataS21(14) = 0                 'Beginning of response
3000|     LimDataS21(15) = 0                 'End of response
3010|     '''Limit_line 4
3020|     LimDataS21(16) = 2                 'Minimum
3030|     LimDataS21(17) = 1930000000#      'Beginning of stimulus
3040|     LimDataS21(18) = 1990000000#      'End of stimulus
3050|     LimDataS21(19) = -10               'Beginning of response
3060|     LimDataS21(20) = -10              'End of response
3070|
3080|     SCPI.CALCulate(1).PARAMeter(1).Select
3090|     SCPI.CALCulate(1).SElected.LIMit.DATA = LimDataS13
3100|     SCPI.CALCulate(1).SElected.LIMit.DISplay.STATE = True
3110|     SCPI.CALCulate(1).SElected.LIMit.STATE = True
3120|
3130|     SCPI.CALCulate(1).PARAMeter(2).Select
3140|     SCPI.CALCulate(1).SElected.LIMit.DATA = LimDataS21
3150|     SCPI.CALCulate(1).SElected.LIMit.DISplay.STATE = True
3160|     SCPI.CALCulate(1).SElected.LIMit.STATE = True
3170|
3180| End Sub
3190|
3200| Private Sub Setup_Register()
3210|
3220|     SCPI.STATUS.QUEStionable.LIMit.CHANnel(1).PTRansition = 6
3230|     SCPI.STATUS.QUEStionable.LIMit.CHANnel(1).NTRansition = 0
3240|     SCPI.STATUS.QUEStionable.LIMit.CHANnel(1).ENABle = 6
3250|
3260| End Sub
3270|
3280| Sub Display_Update(Test_Tr1 As Integer, Test_Tr2 As Integer,
Test_Ch1 As Integer, IlossTx As Variant, IlossRx As Variant, RipTx
As Variant, RipRx As Variant)
3290|
3300|     SCPI.DISplay.UPDate.IMMediate
3310|

```

```

3320|     If Test_Tr1 = 2 Then
3330|         frmDupRes.lblJudgeS13.BackColor = RGB(255, 0, 0)
3340|         frmDupRes.lblJudgeS13.Caption = "Fail"
3350|     Else
3360|         frmDupRes.lblJudgeS13.BackColor = RGB(0, 0, 255)
3370|         frmDupRes.lblJudgeS13.Caption = "Pass"
3380|     End If
3390|
3400|     If Test_Tr2 = 4 Then
3410|         frmDupRes.lblJudgeS21.BackColor = RGB(255, 0, 0)
3420|         frmDupRes.lblJudgeS21.Caption = "Fail"
3430|     Else
3440|         frmDupRes.lblJudgeS21.BackColor = RGB(0, 0, 255)
3450|         frmDupRes.lblJudgeS21.Caption = "Pass"
3460|     End If
3470|
3480|     If Test_Ch1 = 2 Then
3490|         frmDupRes.lblResult.BackColor = RGB(255, 0, 0)
3500|         frmDupRes.lblResult.Caption = "Fail"
3510|     Else
3520|         frmDupRes.lblResult.BackColor = RGB(0, 0, 255)
3530|         frmDupRes.lblResult.Caption = "Pass"
3540|     End If
3550|
3560|         frmDupRes.txtIlossS13.Text = Format(IlossTx(0),
"0.####")
3570|         frmDupRes.txtIlossS21.Text = Format(IlossRx(0),
"0.####")
3580|
3590|         frmDupRes.txtRipS13.Text = Format(RipTx, "0.####")
3600|         frmDupRes.txtRipS21.Text = Format(RipRx, "0.####")
3610|
3620|     End Sub

```
