TECHNICAL NOTE

Agilent™ 8753 to Anritsu Program Conversion Guide for HP-Basic and Visual Basic Users

Scorpion®

Introduction

With the obsolescence of the Agilent 8753, developers and maintainers of Automated Test Equipment code are faced with the task of converting and reworking their code to accommodate a new VNA.

This document outlines an approach to modifying existing 8753 code, in HP-BASIC or Visual Basic, to work with an Anritsu Scorpion VNA. We also list the key areas where care should be taken.

The Scorpion software architecture is a good match to the 8753. A typical 1000 line HP-Basic or Visual Basic ATE program can be modified to work on a Scorpion VNA in just a few hours.

What about 8753 Emulation Mode on Scorpion?

Scorpion has an 8753 Emulation mode available. This is descried in Appendix C of the Scorpion Programming Manual (10410-00204). In emulation mode there are two common areas where problems occur: (1) calibrations and (2) save/recall to registers. The 8753 to ENA translation task has five problem areas: (1) Triggers, (2) Limit Lines, (3) Calibration, (4) Status Monitoring, (5) Outputting Raw Data.

If a problem does occur in emulation mode, and it doesn't appear to be a simple fix, we recommend turning emulation mode off and utilizing the native Scorpion GPIB commands. If equivalent commands exist then translation is straightforward. For some calibration functions, such as loading raw standards data into the 8753 and generating the 12 term model calibration coefficients, we provide a small PC based DLL to emulate the 8753 functionality.

Command Translation

Translating commands from 8753 to Scorpion is quite straightforward. The list below shows a sampling of command equivalents.

8753 Command	Scorpion Command	Comments
FORM3	FMA;LSB	IEEE 64 bit, LSB for Intel PC
CHAN2;S11;IFBW;3000	CH2;S11;IFBW3K	
EDITLIST	(see discrete frequency list notes)	
STAR;10;MHZ;STOP;1;GHZ	SRT 10 MHZ;STP 1 GHZ	Sweep 10 MHz to 1 GHz
SWET;1	(Use Averages on Scorpion)	Manual Sweep time
POIN;51	NP51	51 Points
LOGM	MAG	Log Magnitude display
SING	TRS;WFS	Single Sweep
GATESTAR;25NS	FGT;GST 25 NS	gate start
GATESTOP;-10NS	GSP -10 NS	gate stop
S11;OUTPRAW1	OS11R	output Raw S11 data (R+I)
POIN?	ONP	output number of points
OUTPFORM	OFD	output data on display
CLASS11A;STANB	(See calibration notes)	DLL function
INPUCAL01	(See Calibration Notes)	DLL function
CALIFUL2	(See calibration notes)	DLL function
GATEON	GON	Gate ON (on active channel)
GATEOFF	GOF	Gate OFF
SMIC	SMI	Smith Chart Display
RECAREG06	(See save recall functions)	
SAV2	(See save recall functions)	
REAL	REL	Display Real Data
CONVZREF	(See notes on Impedance)	Z Reflection
REFV;75	(See notes on Impedance)	Reference Impedance

SAVE/RECALL Functions

Scorpion uses registers and calibration files to save the state of the VNA. Registers are of limited use – don’t use them if the setup uses a discrete frequency list. Calibration files can quickly fill the Scorpion internal hard disk. We recommend expanding the Scorpion hard disk with an internal or external SCSI hard drive.

HP BASIC Example

Save current VNA state to a Cal file (Turn off cal first)

```
  OUTPUT @Na;"*OPC?;COF;SAVE 'REG1.Cal'"
```

Save Current VNA State to a Register
(Cal info is not saved, do not use registers for discrete fill data)

```
  string1$ = "*OPC?;COF;SV" & VAL$(Band) & CHR$(10)
  OUTPUT @Na;string1$;
```

Recall a Cal File

```
  OUTPUT @Na;"*OPC?;RECALL 'REG1.Cal'"
```

Recall a Register

```
  string1$ = "*OPC?;RC" & VAL$(Band) & CHR$(10)
  OUTPUT @Na;string1$;
```

Visual Basic Example (using PowerTools driver)

Open a communication session to the VNA:

```
  AwnaOpenSession 0, 6, 0, 30000, sHandle
```

Save current VNA state to a Cal file (Turn off cal first)

```
  AwnaSetString sHandle, "*OPC?;COF;SAVE 'REG1.Cal'"
```

Save Current VNA State to a Register (Cal info is not saved,
do not use registers for discrete fill data)

```
  string1$ = "*OPC?;COF;SV" & CStr(band)
  AwnaSetString sHandle, string1
  AwnaGetString sHandle, mybuf, 10
```

Recall a Cal File

```
  AwnaSetString sHandle, "*OPC?;RECALL 'REG1.Cal'"
  AwnaGetString sHandle, mybuf, 10
```

Recall a Register

```
  string1$ = "*OPC?;RC" & CStr(band)
  AwnaSetString sHandle, string1
  AwnaGetString sHandle, mybuf, 10
```

Binary Data Transfer

Transferring binary data to the VNA and from the VNA is easy to accomplish on Scorpion. HP BASIC is limited in that it only has a 64 bit data type for floating point data. On an Intel PC, make sure the transfer is set for “LSB”.

HP BASIC Example

Example: Sending the Calibration Coefficients to the VNA

1. Setup GPIB

```
ASSIGN @Fast TO GPIB_ADDRESS;FORMAT LSB FIRST
```

2. How Many Data Points?

```
Numpoints=Npts(Band)
NP2 = Numpoints*2
REDIM Tracex(1:NP2)
```

3. What Type of Data?

Scorpion can read in 32 Floats or 64 Bit Floats

```
 REAL Tracex(3202) !64 bit float
```

4. Construct the Arbitrary Block Header:

```
iSize = NP2*8
iLen = LEN(VAL$(iSize))
!Construct the Arbitrary block header
string1$=";LSB;FMB;IC" & VAL$(ii) & " #" & VAL$(iLEN) & VAL$(iSize)
```

5. Send the Arbitrary Block Header, then the Binary Data

```
OUTPUT @Na;string1$;
OUTPUT @Fast;Tracex(*);CHR$(10)
!Send an Ascii 10 at the end to signify end of data
```

Example: getting Trace Data in real and imaginary format

```
OUTPUT @Na;"ONP"
ENTER @Na;myPoints
NP2 = myPoints*2
REDIM Freqx(1:mypoints)

! Get the Frequency List
OUTPUT @Na;"LSB;FMB"; END
OUTPUT @Na;"FDH2;OFV"; END
ENTER @Fast;Freqx(*)
CLEAR @Na
!
REDIM Tracex(1:NP2)
! Get S11 Data, Real+Imaginary
OUTPUT @Na;"LSB;FMB"; END
OUTPUT @Na;"TRS;WFS";END
OUTPUT @NA;"FDH2;OS11C"; END
ENTER @Fast;TRACEX(*)
CLEAR @Na

Notes: Use FDH2 to turn off arbitrary block header.
Send Clear at the end of each transaction to flush the GPIB bus.
```

Example with Visual Basic

PowerTools has a number of easy to use functions for binary data transfer. Some of these examples are already shown in other parts of this document.

Example: sending data to the VNA (freq list is 64 bit floating point):

```
' Set a discrete frequency list of 30 frequencies
AwnaSetSweepFreqList sHandle, AtnFreq(1), 30
```

Example: sending Real/Imaginary data to the VNA:

```
' Real/Imaginary data is interleaved re1, im1, re2, im2, etc..
' here we send the cal terms from 12 term model to VNA
' specify LSB, FMB for Intel, 64 bit float
AwnaSetDataArray sHandle, "LSB;FMB;IC" & CStr(ii), tracex(1), np2
```

Example: reading Real/Imaginary data from the VNA:

```
' do a sweep and get the uncorrected data real/imag format
' getting S11 data here
AwnaSetString sHandle, "TRS;WFS"
AwnaGetMeasuredDataDouble sHandle, 11, tracex(1), np2    'get S11

' Get the Frequency List
AwnaGetSweepPointsList sHandle, 0, freqx(1), NP1
```

Dealing With Discrete Frequency Lists

Do not save Discrete Frequency Lists to Scorpion Registers – use a Cal file instead.

HP BASIC Example:

```
    iSize = Numpoints*8
    iLen = LEN(VAL$(iSize))
!Construct the Arbitrary block header
    string1$=";LSB;FMB;IFV #" & VAL$(iLEN) & VAL$(iSize)
    DATA   10E6, 30E6, 50E6, 55E6, 108E6, 211E6, 250E6, 270E6, 300E6, 325E6
    DATA 350E6,400E6,450E6,500E6, 550E6, 600E6, 650E6, 700E6, 750E6, 800E6
    DATA 862E6,870E6,900E6,950E6,1000E6,1200E6,1450E6,1750E6,2000E6,2200E6
    READ Atnfrq(*)
    OUTPUT @Na;string1$;
    OUTPUT @Fast;Atnfrq(*);CHR$(10)
!! On Scorpion we need to save this discrete fill to a File
!! rather than a register
    OUTPUT @Na;"*OPC?;IFBW30K;COF;SAVE 'REG1.Cal'"
    ENTER @Na;Dummy
```

Visual Basic Example:

```
Dim PV As Variant
PV = Array(10000000#, 30000000#, 50000000#, 55000000#, 108000000#, _ 211000000#,
250000000#, 270000000#, 300000000#, 325000000#, _
350000000#, 400000000#, 450000000#, 500000000#, 550000000#, _
600000000#, 650000000#, 700000000#, 750000000#, 800000000#, _
862000000#, 870000000#, 900000000#, 950000000#, 1000000000#, _
1200000000#, 1450000000#, 1750000000#, 2000000000#, 2200000000#)
For i = 1 To 30
    AtnFreq(i) = PV(i)
Next
AwnaSetSweepFreqList sHandle, AtnFreq(1), 30
AwnaSetString sHandle, "COF;SAVE 'REG1.Cal'"
```

Manual Calibration

The Scorpion S-Parameter calibration procedure is slightly different than the 8753 procedure. A VNA setup is a frequency list, IF Bandwidth setting, screen display, etc. With Scorpion, the user is prompted through a standard procedure to calibrate one setup at a time. With the 8753 and clever use of registers, it is possible to calibrate multiple setups while minimizing the connections of calibrations standards. In the case where there is no Scorpion equivalent to an 8753 procedural we have created a Windows DLL to emulate the 8753 functionality.

HP BASIC Example: Calibration

Loading the DLL into HTBasic

```
  DLL LOAD "htbstring"
  DLL GET "CHARPTR htbstring:Append"
  DLL GET "CHARPTR htbstring:setData" AS "D1"
  DLL GET "CHARPTR htbstring:initSOLT" AS "D2"
  DLL GET "CHARPTR htbstring:calFull" AS "D3"
```

Get the Frequency Data from VNA

```
  ! Get the Frequency List
    OUTPUT @Na;"LSB;FMB"; END
    OUTPUT @Na;"FDH2;OFV"; END
    ENTER @Fast;Freqx(*)
    CLEAR @Na
```

Initializing the Calibration

```
  ! initialize the Full 2 Port Cal function with
  ! VNA System Impedance set to 75 Ohms
  ! NP2 is 2*number of Frequency points
  ! Impedance of 75 Ohms
  ! Thru length of 0 meters
    String2$=FND2$((NP2),(75.0),(0.0))
```

Loading Cal Terms into DLL

```
  ! Pass the Raw Standards data into the DLL
    MAT Tracex=Refl(1,1:NP2,Band)
    String2$=FND1$((NP2),"O1",Tracex(1))
    MAT Tracex=Refl(2,1:NP2,Band)
    String2$=FND1$((NP2),"S1",Tracex(1))
    MAT Tracex=Refl(3,1:NP2,Band)
    String2$=FND1$((NP2),"L1",Tracex(1))
  !
    MAT Tracex=Refl(7,1:NP2,Band)
    String2$=FND1$((NP2),"O2",Tracex(1))
    MAT Tracex=Refl(8,1:NP2,Band)
    String2$=FND1$((NP2),"S2",Tracex(1))
    MAT Tracex=Refl(9,1:NP2,Band)
    String2$=FND1$((NP2),"L2",Tracex(1))
  !
    MAT Tracex=Refl(5,1:NP2,Band)
    String2$=FND1$((NP2),"T11",Tracex(1))
    MAT Tracex=Refl(6,1:NP2,Band)
    String2$=FND1$((NP2),"T21",Tracex(1))
    MAT Tracex=Refl(11,1:NP2,Band)
    String2$=FND1$((NP2),"T22",Tracex(1))
    MAT Tracex=Refl(12,1:NP2,Band)
    String2$=FND1$((NP2),"T12",Tracex(1))
```

Setting up cal kit parameters (Optional)

```
! ENTER the PORTMODEL FOR 75 OHM Cal Kit
! short(meters), open(meters), c0,c1,c2,c3,l0,l1,l2,l3
!Anritsu Male N 75 Standard = 0 mm
!Anritsu Female N 75 Standard = 5.2 mm
!
DATA 0,0,55E-15,0,0,0,0,0,0,0
DATA .0052,.0052,23E-15,0,0,0,0,0,0,0
READ P1M(*)
READ P2M(*)
! Send in new parameters for the Cal Kit Data
String2$=FND1$((10),"PORTMODEL1",P1M(1))
String2$=FND1$((10),"PORTMODEL2",P2M(1))
```

Generate the 12 Term Calibration

```
! Generate the Full 2 Port Model
  String2$=FND3$((NP1),"FREQ",Freqx(1))
```

Setting up VNA for Receiving Calibration Data

```
! Set up Scorpion for Cal First
  string1$ = "*OPC?;HLD;A120;CON;CTN;" & CHR$(10)
  OUTPUT @Na;string1$;
  ENTER @Na;Dummy
!
  string1$ = "*OPC?;HLD;COF" & CHR$(10)
  OUTPUT @Na;string1$;
  ENTER @Na;Dummy
!Load the Cal Terms into VNA
  FOR ii = 1 to 12
    String1$ = "OC" & VAL$(ii)
    ! Get the Cal Coefficients from DLL
    String2$=FNAppend$((NP2),String1$,Tracex(1))
    iSize = NP2*8
    iLen = LEN(VAL$(iSize))
    !Construct the Arbitrary block header
    string1$=";LSB;FMB;IC" & VAL$(ii) & " #" & VAL$(iLEN) & VAL$(iSize)
    OUTPUT @Na;string1$;
    OUTPUT @Fast;Tracex(*);CHR$(10)
  NEXT ii
 ! Turn on the Cal
  string1$ = "*OPC?;CON;CTN" & CHR$(10)
  OUTPUT @Na;string1$;
  ENTER @Na;Dummy
```

Visual Basic Example: Calibration

Declaring the DLL in Visual Basic

```
Declare Function setData Lib "djCalLib.DLL" (ByVal keyname As
  String, xdata As Double, ByVal points As Integer) As String
Declare Function getData Lib "djCalLib.DLL" (ByVal keyname As String,
  xdata As Double, ByVal points As Integer) As String
Declare Function initSOLT Lib "djCalLib.DLL" (ByVal points As
  Integer, ByVal ohms As Double, ByVal thrulength As Double) As String
Declare Function calFull Lib "djCalLib.DLL" (ByVal keyname
  As String, xdata As Double, ByVal points As Integer) As String
Declare Function djAdd Lib "djCalLib.DLL" (ByVal num1 As Double,
  ByVal num2 As Double) As Double
Declare Sub djLog Lib "djCalLib.DLL" (ByVal mesg As String)
```

Get the Frequency Data from VNA

```
ReDim freqx(1 To numPoints)
AwnaGetSweepPointsList sHandle, 0, freqx(1), NP1
```

Initializing the Calibration

```
' initialize the Full 2 Port Cal function
string1 = initSOLT(np2, 50#, 0#)
```

Loading Cal Terms into DLL

```
' Pass the Raw Standards data into the DLL

  'MAT Tracex=Refl(1,1:NP2,Band)
  assignLittleArray Refl, tracex, 1, band, np2
  string2 = setData((np2),"O1", tracex(1))

  'MAT Tracex=Refl(2,1:NP2,Band)
  assignLittleArray Refl, tracex, 2, band, np2
  string2 = setData((np2),"S1", tracex(1))

  'MAT Tracex=Refl(3,1:NP2,Band)
  assignLittleArray Refl, tracex, 3, band, np2
  string2 = setData((np2),"L1", tracex(1))

  'MAT Tracex=Refl(7,1:NP2,Band)
  assignLittleArray Refl, tracex, 7, band, np2
  string2 = setData((np2),"O2", tracex(1))

  'MAT Tracex=Refl(8,1:NP2,Band)
  assignLittleArray Refl, tracex, 8, band, np2
  string2 = setData((np2),"S2", tracex(1))

  'MAT Tracex=Refl(9,1:NP2,Band)
  assignLittleArray Refl, tracex, 9, band, np2
  string2 = setData((np2), "L2", tracex(1))
```

```
    'MAT Tracex=Refl(5,1:NP2,Band)
    assignLittleArray Refl, tracex, 5, band, np2
    string2 = setData((np2),"T11", tracex(1))
    'MAT Tracex=Refl(6,1:NP2,Band)
    assignLittleArray Refl, tracex, 6, band, np2
    string2 = setData((np2),"T21", tracex(1))
    'MAT Tracex=Refl(11,1:NP2,Band)
    assignLittleArray Refl, tracex, 11, band, np2
    string2 = setData((np2),"T22", tracex(1))
    'MAT Tracex=Refl(12,1:NP2,Band)
    assignLittleArray Refl, tracex, 12, band, np2
    string2 = setData((np2),"T12", tracex(1))
```

Setting up cal kit parameters (Optional)

```
    string2 = setData((10),"PORTMODEL1", P1M(1))
    string2 = setData((10),"PORTMODEL2", P2M(1))
```

Generate the 12 Term Calibration

```
  ' Generate the Full 2 Port Model
  string2$ = calFull((NP1),"FREQ", freqx(1))
```

Setting up VNA for Receiving Calibration Data

```
  string1$ = "*OPC?;HLD;A120;CON;CTN;"
  AwnaSetString sHandle, string1$
  AwnaGetString sHandle, mybuf, 10
    string1$ = "*OPC?;HLD;COF" & Chr$(10)
    AwnaSetString sHandle, string1$
    AwnaGetString sHandle, mybuf, 10

  'Load the Cal Terms into VNA
  Dim ii As Long
  For ii = 1 To 12
    string1$ = "OC" & CStr(ii)
    string2$ = getData((np2),string1, tracex(1))
    AwnaSetDataArray sHandle, "LSB;FMB;IC" & CStr(ii), tracex(1), np2
  Next ii

  string1$ = "*OPC?;CON;CTN"
  AwnaSetString sHandle, string1$
  AwnaGetString sHandle, mybuf, 10

  ' Save the Calibration to FILE
  string1$ = "*OPC?;SAVE 'CAL" & CStr(band) & ".Cal'"
  AwnaSetString sHandle, string1$
  AwnaGetString sHandle, mybuf, 10
```

SALES CENTERS:
United States (800) ANRITSU
Canada (800) ANRITSU
South America 55 (21) 2527-6922

Europe 44 (0) 1582-433433
Japan 81 (46) 223-1111
Asia-Pacific (852) 2301-4980

Microwave Measurement Division
490 Jarvis Drive, Morgan Hill, CA 95037-2809
http://www.us.anritsu.com

Discover What's Possible®

11410-00371, Rev. A