Agilent Technologies 8960 Series 10 Wireless Communications Test Set

Programming: Getting Started Guide

AMPS/136 Mobile Test Application E1961A Revision: A.05
cdma2000/IS-95/AMPS Mobile Test Application E1962B Revision: B.03
GPRS Lab Application E6701A Revision A.01
GPRS Mobile Test Application E1964A Revision: A.02
GSM Mobile Test Application E1960A Revision: A.08
GSM_AMPS/136_GPRS Mobile Test Application (fast switching) E1985A Revision: A.02
W-CDMA Mobile Test Application E1963A Revision A.01

Agilent Part Number: 5967-5188

Print Date: November 2001

Agilent Technologies

www.agilent.com/find/8960support/

Edition/Print Date

All Editions and Updates of this manual and their creation dates are listed below:

March 2001 - 5967-5125

July 2001 - 5967-5157

November 2001 - 5967-5188

Contents

E1961A AMPS/136 Mobile Test Application Revision A.05

E1962B cdma2000/IS-95/AMPS Mobile Test Application B.03

Contents

E1964A GPRS Mobile Test Application A.02
E6701A GPRS Lab Application A.01

Contents

E1985A GSM_AMPS/136_GPRS Mobile Test Application Revision A.02

Programming: Getting Started Guide

for

E1961A AMPS/136 Mobile Test Application Revision A.05

Introduction

The Getting Started Guide is organized around the basic set of tasks a control program normally performs when testing a mobile station in a manufacturing environment.

Conventions used in this Getting Started Guide

Throughout this Getting Started Guide the term “test set” refers to an Agilent Technologies 8960 Series 10 wireless communications test set with the E1961A AMPS/136 mobile test application installed.

The variable `Testset` used in the steps of the Getting Started Guide refers to the test set’s GPIB address.

How to use this Getting Started Guide

The most effective way to use this Getting Started Guide is with the Programming Flowchart included in the front inside pocket of this guide and the 8960 User Documentation. This documentation is found in two locations:

- the 8960 Family Support Site (updated frequently) on the Internet, or
- the User Documentation CD-ROM shipped with your test application.

Each step on the Programming Flowchart is illustrated with example program code in this guide. Using this Getting Started Guide, the Programming Flowchart, and the on-line information about 8960 programming, you will be able to generate a control program to perform fundamental mobile station manufacturing tests.

Useful on-line links

Go to the 8960 Family Support Site on either the CD-ROM or the Internet. The CD-ROM should self-launch to its home page. The URL for the website is:

www.agilent.com/find/8960support/

Navigate to the Programming page for this test application. The illustrations on the back of the Programming Flowchart show you how to navigate to this page.

The following links are under the heading Getting Started on the Programming page:

- Programming: Getting Started Guide
  - This online version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.
- Control Program Examples
  - These examples are for you to download. You may want to use these as templates for your own control program or to execute.
  - The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

About the Programming Examples Presented in this Guide

Programming Language:

Programming examples presented in this guide are written in the HP BASIC programming language, also known as RMB or Rocky Mountain BASIC. The use of HP BASIC is not an endorsement of the HP BASIC product.

Line Numbers

All of the programming examples in the guide with line numbers are sections from a control program example available on-line for you to download.

Code that is not part of the download control program example does not have line numbers. This code may represent an alternate method of performing the task or may illustrate a feature not used by the control program example.

Syntax used in Programming Examples:

- The programming examples use the shortened form of the command syntax to minimize GPIB bus transactions. The shortened form of a command is defined by use of capital letters in the command syntax. For the command syntax:

  ```
  RFANalyzer:CONTrol:MEASurement:FREQuency:AUTO?
  ```

  the shortened form would be:

  ```
  RFAN:CONT:MEAS:FREQ:AUTO?
  ```

- The programming examples do not include optional nodes. Optional nodes in the command syntax are defined by enclosing the node inside the [ ] brackets. For example, the command syntax:

```
CALL[:CELL]:POWer[:SAMPlitude] -80dBm
```

appears in the programming examples as:

```
CALL:POW -80dBm
```

- Programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the on-line information for the definition and use of these command separators.

Complex Commands

Complex commands are used to configure the state and assign values to parameters simultaneously. Complex commands can be used to save programming steps and minimize GPIB bus transactions.

- The syntax below turns the state of the parameter on.

  OUTPUT Test_set;"SET:DTXP:TIM:STAT ON"

- The syntax below is used to assign a value to the parameter.

  OUTPUT Test_set;"SET:DTXP:TIM:TIME 10 S"

- Both of the above actions can be accomplished with one syntax command:

  OUTPUT Test_set;"SET:DTXP:TIM:STIM 10 S"

  The command above sets the parameter state to ON and the value of the parameter to 10 seconds. Note that in this example the optional command mnemonic :STIMe has been included to clarify that this complex command was used to set both the state and the value.

- This command can be shortened further by removing the optional command mnemonic :STIMe, as shown below.

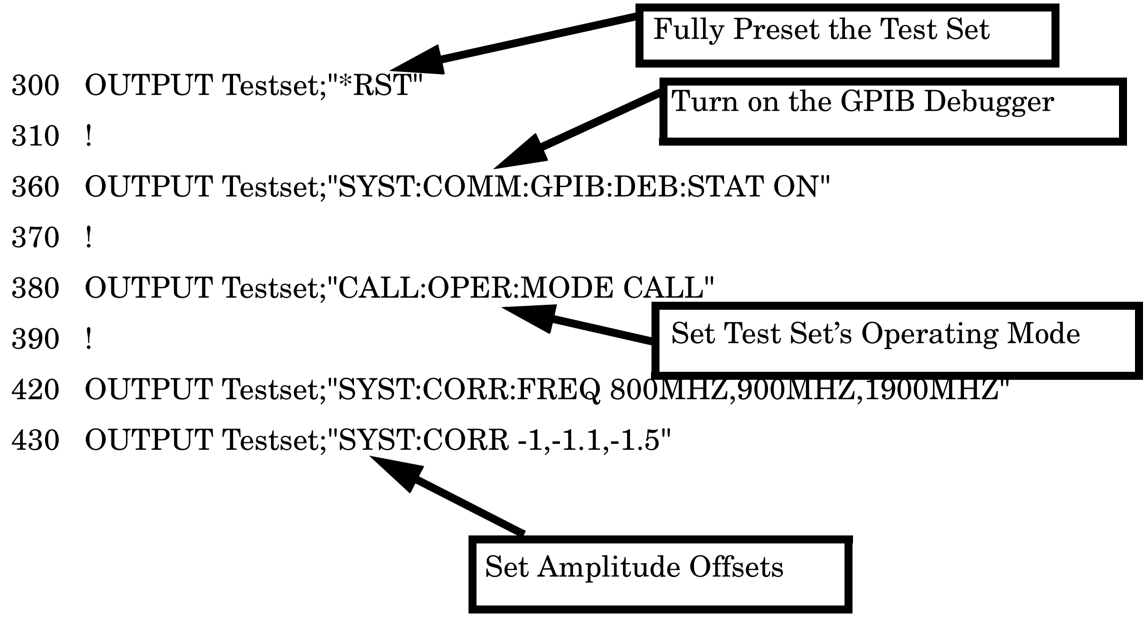
  OUTPUT Test_set;"SET:DTXP:TIM 10 S"

  This is the format that will be used throughout this guide.

Step 1: Set up the Test Set

This step explains how to:

- "Fully Preset the Test Set"
- "Turn on the GPIB Debugger"
- "Set Test Set's Operating Mode"
- "Set Amplitude Offsets"



```
300  OUTPUT Testset;"*RST"
310  !
360  OUTPUT Testset;"SYST:COMM:GPIB:DEB:STAT ON"
370  !
380  OUTPUT Testset;"CALL:OPER:MODE CALL"
390  !
420  OUTPUT Testset;"SYST:CORR:FREQ 800MHZ,900MHZ,1900MHZ"
430  OUTPUT Testset;"SYST:CORR -1,-1.1,-1.5"
```

Fully Preset the Test Set

Turn on the GPIB Debugger

Set Test Set's Operating Mode

Set Amplitude Offsets

Fully Preset the Test Set

To set up the test set, you begin by sending the *RST command. *RST is used to perform a full preset of the test set, restoring the majority of settings to their default values. *RST also sets all measurements to single trigger. More information about presets is available on the Internet.

Turn on the GPIB Debugger

Another useful tool that you may want to turn on at this time is the GPIB command debugger. While turned on, error messages appear on the test set's screen when the test set receives an unknown GPIB command. The message contains information about what type of error was caused and indicates where in the syntax the error occurred. Troubleshooting, like locating and fixing typing errors for example, becomes easier using the GPIB command debugger.

NOTE The `SYST:COMM:GPIB:DEB:STAT ON` command assists you when debugging code. This command should be taken out of your code once development is completed.

Set Test Set's Operating Mode

The `CALL:OPER:MODE` command in the diagram is used to set the test set's operating mode to active cell. This means call processing is used to maintain a link between the test set and the mobile station.

There are also three test mode operating modes available. These modes are useful when you are operating the mobile station in test mode.

Set Amplitude Offsets

You can compensate for fixture loss or cable loss by using the RF IN/OUT port's amplitude offset parameter. It is possible to set twenty offsets corresponding to twenty frequencies. Use the `SYST:CORR:FREQ` command to specify frequencies. Then, use the `SYST:CORR` complex command to set the offset values for the respective frequencies and turn the amplitude state to `ON`. The amplitude offset for frequencies not specified is determined by linear interpolation.

NOTE The amplitude offset state is not turned `OFF` when you cycle power or sent the *RST command. Therefore, when you set amplitude offset values in a test set, you must turn this state off either manually or via GPIB if you no longer want to use the offsets.

Step 2: Configure the Test Set and Mobile Station Parameters

This step explains how to:

- “Configure the Control Channel Parameters”
- “Configure the Traffic Channel Parameters”
- “Configure the Mobile Station Parameters”

NOTE Many of the parameters configured below are being configured to their default values. In a manufacturing environment it may be desirable to explicitly configure these parameters to ensure that the required settings have not been changed by someone setting a parameter’s value through the test set’s front panel. However, greater code efficiency can be achieved by not configuring them.

Configure the Control Channel Parameters

The programming example below illustrates setting the cell power to -75 dBm, specified at the beginning of the program in the variable `Rf_level`, setting the control channel type to a Digital Control Channel (DCCH), setting the DCCH band to the 800 MHz band, and specifying the DCCH channel to be 1013.

Many other control channel parameters can also be configured. Refer to the User Documentation, either on the Internet or the CD-ROM, or the GPIB syntax guide for information about these parameters.

```
500   OUTPUT Testset;”CALL:POW “;Rf_level! Set the cell power to -75 dBm
510   OUTPUT Testset;”CALL:CCH:TYPE DCCH”! Set the Control Channel Type
520   OUTPUT Testset;”CALL:DCCH:BAND CELL”! Set the DCCH band to 800 MHz
530   OUTPUT Testset;”CALL:DCCH 1013”  ! Set the DCCH channel to 1013
```

Configure the Traffic Channel Parameters

The example below illustrates how you can set some of the traffic channel parameters. In this example, the traffic channel type is being set to a Digital Traffic Channel (DTC). This command is also used when you want to perform a handoff to an Analog Voice Channel (AVC). The DTC band is set to the 800 MHz band and the channel to 542. In addition, it is possible for you to specify parameters to be used later when handoffs are performed. Here, the DTC channel for the PCS (1900 MHz) band is set to 1000. Parameters for an AVC are also established at this time. These parameters are stored in the test set and become active when the band or channel type become active. Internal FM parameters to modulate the AVC for the audio harmonic distortion measurement are set up below. Internal FM has also been turned off because you will not want to use it until the connection is on an AVC.

Refer to the User Documentation, either on the Internet or the CD-ROM, or the GPIB syntax guide for other traffic channel parameters you can configure.

```
570   OUTPUT Testset;"CALL:TCH:TYPE DTC"
580   OUTPUT Testset;"CALL:SET:DTC:BAND CELL"
590   OUTPUT Testset;"CALL:SET:DTC:CHAN:CELL 542"
600   OUTPUT Testset;"CALL:SET:DTC:CHAN:PCS 1000"
610   OUTPUT Testset;"CALL:SET:AVC 387"! Set the Analog Voice Channel
620   OUTPUT Testset;"CALL:SET:AVC:SAT SAT2"! Set the SAT tone to 6KHZ
630   ! Set up the AVC for the Audio Harmonic Distortion measurement.
640   ! Turn off the internal FM until connection is on an AVC.
650   OUTPUT Testset;"CALL:FM:INT:STAT OFF;DEV 8KHZ;FREQ 1004HZ"
```

Configure the Mobile Station Parameters

The example below illustrates how to assign mobile station transmit levels. In this example, the mobile station is being assigned to Level 2 for digital tests in both bands and for analog testing.

```
690   OUTPUT Testset;"CALL:SET:MS:DIG:TXL:CELL 2"
700   OUTPUT Testset;"CALL:SET:MS:DIG:TXL:PCS 2"
710   OUTPUT Testset;"CALL:SET:MS:ANAL:TXL 2"
```

Step 3: Set the Measurement Parameters

This step gives an example of how you can configure measurement parameters. Notice both digital and analog measurement parameters can be configured simultaneously. For more information about measurement parameters being configured, refer to the additional details about this step available on the Internet.

```
770   ! Set the trigger to single for all measurements
780   OUTPUT Testset;”SET:CONT:OFF”
790   !
800   ! Configure the Digital Measurements
810   OUTPUT Testset;”SET:DTXP:TIM 3”
820   OUTPUT Testset;”SET:MACC:TIM 3;EVM10:STAT OFF”
830   OUTPUT Testset;”SET:ACP:TIM 3”
840   !
850   ! Configure the Analog Measurements
860   OUTPUT Testset;”SET:FST:TIM 3”
870   OUTPUT Testset;”SET:ATXP:TIM 3”
880   OUTPUT Testset;”SET:FM:TIM 3”
890   OUTPUT Testset;”SET:FM:DIST:STAT ON;FREQ 6000”
900   OUTPUT Testset;”SET:FM:DET PPE”
910   OUTPUT Testset;”SET:FM:FILT TBP”
920   !OUTPUT Testset;”SET:FM:FILT:TBP 6000”
930   !
940   !  Configure the Audio Measurements
950   OUTPUT Testset;”SET:AFAN:FILT CMES;TIM 3;COUN 5;PEAK:VOLT 1”
960   OUTPUT Testset;”SET:AFAN:SDIS:STATE ON;FREQ 1004”
```

Step 3: Set the Measurement Parameters

Step 4: Make a Connection

There are several ways to establish a connection with the mobile station.

- “Originating a Call from the Mobile Station”
- “Originating a Call from the Test Set”
- “Make a Connection using Test Mode Commands”

Originating a Call from the Mobile Station

The code below illustrates how to make a connection by originating a call with the mobile station.

Synchronization between the control program and the test set is maintained by querying the test set for the state of the connection between it and the mobile station. When a Mobile Station origination occurs, the `CALL:CONN?` hanging query is used. It will return a `1` when the call is connected and a `0` otherwise. A state change detector is also armed to ensure the query does not stop hanging before the state transition from IDLE to CONNected is able to begin. Finally, to prevent the query from hanging indefinitely, which could occur if the mobile station is not turned on, badly broken, or no one pushes the “send” button on the mobile, a timeout is set for this query. In this example, 15 seconds is the value assigned to the timeout. After 15 seconds, the change detector is disarmed and the query returns either a `1` or `0`. For more information about call synchronization, refer to the additional information about this step available on the Internet.

```
1050  PRINT “Turn the phone on now.”
1060  PRINT “When the phone camps on DCCH 1013, press F2 to continue.”
1070  PAUSE
1080  CLEAR SCREEN
1090  PRINT “Originate a call on the Mobile now.”
1100  OUTPUT Testset;”CALL:CONN:TIM 15”
1110  OUTPUT Testset;”CALL:CONN:ARM”!Arm Call-State-Change Detector
1120  OUTPUT Testset;”CALL:CONN?”!Query State
1130  ENTER Testset;Callstate
1140  IF NOT Callstate THEN Orig_failed
```

Originating a Call from the Test Set

The code below illustrates how to make a connection by originating a call from the test set.

Synchronization for a test set origination is very similar to that for a mobile station originated call. The primary difference is the default timeout value associated with test set originated events and the automatic arming of the timer when a command is sent to originate an event from the test set.

```
OUTPUT Testset;”CALL:PAG:PNUM ‘0000574016’” ! Set paging number
OUTPUT Testset;”CALL:PAG:REP ON” ! Set paging repeat state
OUTPUT Testset;”CALL:ORIG” ! Start a base station originated call
OUTPUT Testset;”CALL:CONN?”  ! Hanging GPIB query
ENTER Testset;Call_connected      ! Program will hang here until
                                   ! origination passes or fails
IF NOT Call_connected THEN
   OUTPUT Test_set;”CALL:END”
   Orig_failed
END IF
```

Make a Connection using Test Mode Commands

When both the mobile station and the test set are operating in test mode, the test set provides forward channel stimulus but has no control over or communication with the mobile station. At this point, either the control program via a serial bus or other interface, or the user with the mobile’s keypad, must set up the mobile station in order for the test set to make measurements.

NOTE The method used to synchronize the mobile station with the test set is proprietary to the mobile station manufacturer. The test set has no direct control of synchronization when the mobile station is operating in test mode.

Step 5: INITiate and FETCh Measurements

- “Digital Measurements”
- “Analog Measurements”
- “Validate Measurement Results”
- “Printing Results”

Digital Measurements

The programming example below illustrates how to make a transmitter and receiver measurement simultaneously. First, set up the test set to begin a mobile-reported receiver measurement. Then initiate and fetch the transmitter measurements. After the transmitter testing is complete, the receiver measurements are queried. These are Mobile Assisted Hand Off (MAHO) values. Performing the digital measurements in this manner allows the mobile receiver measurements to be performed at the same time as the transmitter measurements. You may want to take advantage of opportunities like this to shorten test time and make test code more efficient.

For more information about using INIT:DONE? to perform concurrent measurements, refer to the additional details about the step available on the Internet.

```
1260  OUTPUT Testset;”CALL:MS:REP:MAHO ON”
1270  OUTPUT Testset;”CALL:POW “;Maho_level
1280  !
1290  OUTPUT Testset;”INIT:DTXP;MACC;ACP”
1300  REPEAT
1310    OUTPUT Testset;”INIT:DONE?”
1320    ENTER Testset;Measdone$
1330    SELECT Measdone$
1340    CASE “DTXP”
1350      OUTPUT Testset;”FETC:DTXP?”
1360      ENTER Testset;Integrity,Power
1370      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1380      Print_res(Measdone$,Power)
1390    CASE “MACC”
1400      OUTPUT Testset;”FETC:MACC?”
1410      ENTER Testset;Integrity,Evm,Ferr,Ooff,Perr,Mag
```

```
1420      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1430      Print_res(Measdone$,Evm,Ferr,Ooff,Perr,Mag)
1440    CASE “ACP”
1450      OUTPUT Testset;”FETC:ACP?”
1460      ENTER Testset;Integrity,Adjl,Adjh,Alt1l,Alt1h,Alt2l,Alt2h
1470      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1480      Print_res(Measdone$,Adjl,Adjh,Alt1l,Alt1h,Alt2l,Alt2h)
1490    END SELECT
1500  UNTIL Measdone$=”NONE”
1510  !
1520  OUTPUT Testset;”CALL:MS:REP:MAHO:BERR:NEW?”!First MAHO Report
1530  ENTER Testset;Discard$ !Discard the first mobile report.
1540  OUTPUT Testset;”CALL:MS:REP:MAHO:BERR:NEW?”!Get NEXT MAHO Report
1550  ENTER Testset;Ber$ !These values are more stable than the first
1560  OUTPUT Testset;”CALL:MS:REP:MAHO:RSSI?”
1570  ENTER Testset;Rssi
1580  Print_maho(Ber$,Rssi,Maho_level)
1590  OUTPUT Testset;”CALL:POW “;Rf_level
1600  OUTPUT Testset;”CALL:MS:REP:MAHO OFF”
```

Analog Measurements

The example below for analog measurements illustrates the same technique used above for digital measurements. Again, the test set is set up for receiver measurements, allowing the measurements to settle while initiating and fetching transmitter measurements.

For more information about using INIT:DONE? to perform concurrent measurements, refer to the additional details about the step available on the Internet.

```
1890  OUTPUT Testset;”CALL:FM:INT:STAT ON”
1900  OUTPUT Testset;”CALL:POW “;Sinad_level
1910  OUTPUT Testset;”INIT:ATXP;FST;AFAN;FM”
1920  REPEAT
1930    OUTPUT Testset;”INIT:DONE?”
1940    ENTER Testset;Measdone$
1950    SELECT Measdone$
1960    CASE “ATXP”
1970      OUTPUT Testset;”FETC:ATXP?”
1980      ENTER Testset;Integrity,Power
1990      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
```

```
2000      Print_res(Measdone$,Power)
2010    CASE “FST”
2020      OUTPUT Testset;”FETC:FST?”
2030      ENTER Testset;Integrity,Ferr,Freq
2040      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
2050      Print_res(Measdone$,Ferr,Freq)
2060    CASE “AFAN”
2070      OUTPUT Testset;”FETC:AFAN?”
2080      ENTER Testset;Integrity,Level,Sinad,Dist
2090      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
2100      Print_res(Measdone$,Level,Dist,Sinad)
2110    CASE “FM”
2120      OUTPUT Testset;”FETC:FM?”
2130      ENTER Testset;Integrity,Dev,Dist
2140      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
2150      Print_res(Measdone$,Dev,Dist)
2160    END SELECT
2170  UNTIL Measdone$=”NONE”
```

Validate Measurement Results

Validating measurement results is extremely important. The test set will return a result if it is capable of making a measurement, even if this result is obtained under adverse conditions.

The test set provides an integrity indicator to inform you if any errors occurred during the measurement process. You can query the integrity indicator as a measurement result, using the `FETC?` query. A value of `0` indicates that the measurement is valid. See your GPIB syntax guide for ways to query the integrity indicator and the User Documentation on the Internet for possible returned values and the error they indicate.

Printing Results

The example below shows how you might create a subroutine to handle processes that are repeated. The variable `Meas_name$` is used to pass the name of the measurement to the subroutine. `Res1` contains measurement results. `Res2`, `Res3`, `Res4`, `Res5`, and `Res6` can contain measurement results if there are more than one for a particular measurement.

```
2420 SUB Print_res(Meas_name$,Res1,OPTIONAL Res2,Res3,Res4,Res5,Res6)
```

```
2430    SELECT Meas_name$
2440    CASE “DTXP”
2450     PRINT USING “5X,””Ave Digital Power:””,5X,M2D.2D,”” dBm”””;Res1
2460    CASE “MACC”
2470      PRINT USING “5X,””Max EVM1:””,14X,M2D.2D,”” %”””;Res1
2480      PRINT USING “5X,””Worst Freq Error:””,5X,M3D.2D,”” Hz”””;Res2
2490      PRINT USING “5X,””Max Mag. Error:””,8X,M2D.2D,”” %”””;Res5
2500      PRINT USING “5X,””Max Origin Offset:””,5X,M2D.2D,”” dB”””;Res3
2510      PRINT USING “5X,””Max Phase Error:””,7X,M2D.2D,”” Deg”””;Res4
2520    CASE “ACP”
2530      PRINT USING “5X,””ACP Adj Lo:””,12X,M2D.2D,”” dBc”””;Res1
2540      PRINT USING “5X,””ACP Adj Hi:””,12X,M2D.2D,”” dBc”””;Res2
2550      PRINT USING “5X,””ACP Alt1 Lo:””,11X,M2D.2D,”” dBc”””;Res3
2560      PRINT USING “5X,””ACP Alt1 Hi:””,11X,M2D.2D,”” dBc”””;Res4
2570      PRINT USING “5X,””ACP Alt2 Lo:””,11X,M2D.2D,”” dBc”””;Res5
2580      PRINT USING “5X,””ACP Alt2 Hi:””,11X,M2D.2D,”” dBc”””;Res6
2590    CASE “ATXP”
2600      PRINT USING “5X,””Ave Analog Power:””,6X,M2D.2D,”” dBm”””;Res1
2610    CASE “FST”
2620      PRINT USING “5X,””Worst Freq Error:””,5X,M3D.2D,”” ppm”””;Res1
2630      PRINT USING “5X,””Average Freq:””,6X,M3D.2DESZ,”” Hz”””;Res2
2640    CASE “FM”
2650      PRINT USING “5X,””SAT Deviation:””,9X,M5D,”” Hz”””;Res1
2660      PRINT USING “5X,””Distortion:””,11X,M3D.2D,”” %”””;Res2
2670    CASE “AFAN”
2680      PRINT USING “5X,””Audio Level:””,11X,M3D.2D,”” V”””;Res1
2690      PRINT USING “5X,””Audio Distortion:””,6X,M3D.2D,”” %”””;Res2
2700      PRINT USING “5X,””SINAD:””,16X,M3D.2D,”” dB”””;Res3
2710    END SELECT
2720  SUBEND
```

Step 6: Reconfigure Test Set and Mobile Station Parameters

There are multiple ways that you may want to reconfigure the connection parameters. Some examples are:

- "Reconfigure the Connection to a New Channel"
- "Reconfigure the Connection to a New Band"
- "Reconfigure the Connection to a New Traffic Channel Type"
- "Reconfigure the Connection when using Test Mode"

Reconfigure the Connection to a New Channel

The example below illustrates how to change the digital traffic channel to 556. It also illustrates changing the mobile station transmit level to 4. The process used to reconfigure the connection is to first change the parameter settings with `CALL:SET` commands. These new parameters are activated when the `CALL:HAND` command is sent. The `CALL:STAT:STAT?` query is used to ensure that the call is still connected. If the connection is compromised, the subroutine `Dropped_call` is called.

```
OUTPUT Testset;"CALL:SET:DTC 556"
OUTPUT Testset;"CALL:SET:MS:DIG:TXL 4"
OUTPUT Testset;"CALL:HAND"
!
OUTPUT Testset;"CALL:STAT:STAT?"
ENTER Testset;Call_state$
IF Call_state$<>"CONN" THEN Dropped_call
```

Reconfigure the Connection to a New Band

Again, the process for changing the connection parameters involves using the `CALL:SET` commands to set the DTC to a new band and then the `CALL:HAND` command to activate them.

```
1630  OUTPUT Testset;"CALL:SET:DTC:BAND PCS"
1730  !
```

```
1740  OUTPUT Testset;"CALL:HAND"
1750  OUTPUT Testset;"CALL:STAT:STAT?"
1760  ENTER Testset;Connected$
1770  IF Connected$<>"CONN" THEN
1780    Dropped_call
1790  END IF
```

Reconfigure the Connection to a New Traffic Channel Type

If you want to reconfigure the connection from a DTC to an AVC or from an AVC to a DTC, you must specify the new channel type. The example below illustrates how to reconfigure the connection to an AVC.

```
1670  OUTPUT Testset;"CALL:TCH:TYPE AVC"
1730  !
1740  OUTPUT Testset;"CALL:HAND"
1750  OUTPUT Testset;"CALL:STAT:STAT?"
1760  ENTER Testset;Connected$
1770  IF Connected$<>"CONN" THEN
1780    Dropped_call
1790  END IF
```

Reconfigure the Connection when using Test Mode

The test set must be put into any new configuration before the mobile station in order for the mobile station to be able to synchronize to the test set. This involves repeating steps 1 or 2 or both. It is not necessary to use the `CALL:HAND` command used when the mobile station is on an active call.

Step 7: End the Connection

You can end the connection in one of two ways:

- “Ending the Connection from the Test Set”
- “Ending the Connection from the Mobile Station”

Ending the Connection from the Test Set

Before you can end the connection, the power level must be returned to its normal level to ensure the mobile station receives the signals to end the call correctly. The `CALL:END` command is used to end the connection. Here you use the `CALL:CONN?` query for call synchronization. This query returns a `0` if the call ended successfully and a `1` if the call is not ended.

```
2280  OUTPUT Testset;”CALL:POW “;Rf_level
2290  OUTPUT Testset;”CALL:END”
2300  OUTPUT Testset;”CALL:CONN?”
2310  ENTER Testset;Callstate
2320  IF Callstate=1 THEN
2330    PRINT “Make sure the phone has released the call.”
2340    OUTPUT Testset;”SYST:PRES3”
2350  END IF
```

Ending the Connection from the Mobile Station

Because the connection is being ended from the mobile station, it is important to set a timeout value and arm the change detector. More information about using these commands to achieve call synchronization is available in the additional details about this step available on the Internet.

```
OUTPUT Testset;”CALL:CONN:TIM 5” !Set timeout time to 5 seconds.
OUTPUT Testset;”CALL:CONN:ARM”   !Arm the change detector.
DISP “Terminate the call from the mobile station.”
OUTPUT Testset;”CALL:CONN?” !Initiate call connect state query.
ENTER Testset;Call_connected     !Program will hang here until state
                                  !change or timer expires.
!Check if disconnect successful.
IF Call_connected THEN OUTPUT Testset;”SYST:PRES3”
```

Step 7: End the Connection

Programming: Getting Started Guide

for

E1962B cdma2000/IS-95/AMPS Mobile Test Application B.03

Introduction

The Getting Started Guide is organized around the basic set of tasks a control program normally performs when testing a mobile station in a manufacturing environment.

Conventions used in this Getting Started Guide

Throughout this Getting Started Guide the term “test set” refers to an Agilent Technologies 8960 Series 10 wireless communications test set with the E1962B cdma2000 mobile test application installed.

The variable `Test_set` used in the steps of the Getting Started Guide refers to the test set’s GPIB address.

How to use this Getting Started Guide

The most effective way to use this Getting Started Guide is with the Programming Flowchart included in the front inside pocket of this guide and the 8960 User Documentation. This documentation is found in two locations:

- the 8960 Family Support Site (updated frequently) on the Internet, or
- the User Documentation CD-ROM shipped with your test application.

Each step on the Programming Flowchart is illustrated with example program code in this guide. Using this Getting Started Guide, the Programming Flowchart, and the on-line information about 8960 programming, you will be able to generate a control program to perform fundamental mobile station manufacturing tests.

Useful on-line links

Go to the 8960 Family Support Site on either the CD-ROM or the Internet. The CD-ROM should self-launch to its home page. The URL for the website is:

www.agilent.com/find/8960support/

Navigate to the Programming page for this test application. The illustrations on the back of the Programming Flowchart show you how to navigate to this page.

The following links are under the heading Getting Started on the Programming page:

- Programming: Getting Started Guide
  - This online version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.
- Control Program Examples
  - These examples are for you to download. You may want to use these as templates for your own control program or to execute.
  - The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

About the Programming Examples Presented in this Guide

Programming Language:

Programming examples presented in this guide are written in the HP BASIC , also known as RMB or Rocky Mountain BASIC, and C programming languages. The use of HP BASIC is not an endorsement of the HP BASIC product.

Line Numbers

All of the programming examples in the guide with line numbers are sections from a control program example available on-line for you to download.

Code that is not part of the download control program example does not have line numbers. This code may represent an alternate method of performing the task or may illustrate a feature not used by the control program example.

Syntax used in Programming Examples:

- The programming examples use the shortened form of the command syntax to minimize GPIB bus transactions. The shortened form of a command is defined by use of capital letters in the command syntax. For the command syntax:

  ```
  RFANalyzer:CONTrol:MEASurement:FREQuency:AUTO?
  ```

  the shortened form would be:

  ```
  RFAN:CONT:MEAS:FREQ:AUTO?
  ```

- The programming examples do not include optional nodes. Optional nodes in the command syntax are defined by enclosing the node inside the [ ] brackets. For example, the command syntax:

```
CALL[:CELL]:POWer[:SAMPlitude] -80dBm
```

appears in the programming examples as:

```
CALL:POW -80dBm
```

- Programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the on-line information for the definition and use of these command separators.

Complex Commands

Complex commands are used to configure the state and assign values to parameters simultaneously. Complex commands can be used to save programming steps and minimize GPIB bus transactions.

- The syntax below turns the state of the parameter on.

  OUTPUT Test_set;"SET:CPOW:TIM:STAT ON"

- The syntax below is used to assign a value to the parameter.

  OUTPUT Test_set;"SET:CPOW:TIM:TIME 10 S"

- Both of the above actions can be accomplished with one syntax command:

  OUTPUT Test_set;"SET:CPOW:TIM:STIM 10 S"

  The command above sets the parameter state to ON and the value of the parameter to 10 seconds. Note that in this example the optional command mnemonic :STIMe has been included to clarify that this complex command was used to set both the state and the value.

- This command can be shortened further by removing the optional command mnemonic :STIMe, as shown below.

  OUTPUT Test_set;"SET:CPOW:TIM 10 S"

  This is the format that will be used throughout this guide.

Step 1: Set up the Test Set

This step explains how to:

- “Fully Preset the Test Set”
- “Declare Variables, Set Path Losses and Timeouts, Activate GPIB Debugger”

Fully Preset the Test Set

To set up the test set, you begin by sending the *RST and *OPC commands along with a global timeout. The *RST command is used to perform a full reset of the test set, returning it to a known state. The *OPC command clears all status registers to assure accurate query functionality.

```
31       CLEAR SCREEN
40       Testset=714
41 !
42  !*****************************************************************
43  ! PRESET TEST SET
44  !*****************************************************************
45 !
46       OUTPUT Testset;"*RST"
47       OUTPUT Testset;"*OPC?"
48       ENTER Testset;Opc$
49          !
```

Declare Variables, Set Path Losses and Timeouts, Activate GPIB Debugger

Another useful tool that you may want to turn on at this time is the system command GPIB debugger. When on, error messages appear on the test set’s screen when the test set receives an unknown GPIB command.

```
51  !*****************************************************************
52  ! VARIABLE DECLARATIONS FOR CALL SET UP - MOBILE SPECIFIC
53  !*****************************************************************
54   !
61       Systype$="DIGITAL2000"
70       Band$="USCELLULAR"
```

```
80       Channel=384
81       Handoffchan=500
90       Sid=1
100      Nid=1
110      Radio_config$="F3R3"
120      Service_opt$="SO2"
142 !
143  !*****************************************************************
144  ! SET PATH LOSS VALUES
145  !*****************************************************************
146 !
150       Loss_frequency$=" 851 MHZ,896 MHZ"
160       Expected_loss$=" -2,-2"
170       OUTPUT Testset;"SYST:CORR:FREQ";Loss_frequency$
180       OUTPUT Testset;"SYST:CORR";Expected_loss$
200   !
210  !*****************************************************************
211  ! SET UP TIMEOUTS
212  !*****************************************************************
213 !
214       ON TIMEOUT 7,20 CALL Timeout
220 !
233  !*****************************************************************
234  ! TURN ON 8960 GP-IB DEBUGGER - COMMENT OUT WHEN DONE DEBUGGING
PROGRAM
235  !*****************************************************************
236 !
237       OUTPUT Testset;"SYST:COMM:GPIB:DEB:STAT ON"
```

Step 2: Configure Test Set and Mobile Station Parameters

Configure the Test Set Parameters

The programming example below illustrates several cell configurations along with specifications to the setup of the generated code channels.

```
252  !*****************************************************************
253  ! SET UP CALL PARAMETERS
254  !*****************************************************************
255 !
256       DISP “Setting up Test Set Parameters”
257       OUTPUT Testset;”DISP:MODE FAST”
258       OUTPUT Testset;”CALL:OPER:MODE CALL”
270       OUTPUT Testset;”CALL:SYST “;Systype$
280       OUTPUT Testset;”CALL:BAND “;Band$
290       OUTPUT Testset;”CALL:CHAN “;Channel
300       OUTPUT Testset;”CALL:POW -50”
310       OUTPUT Testset;”CALL:SID “;Sid
320       OUTPUT Testset;”CALL:NID “;Nid
330       OUTPUT Testset;”CALL:RCON “;Radio_config$
340       OUTPUT Testset;”CALL:SOPT “;Service_opt$
350       OUTPUT Testset;”CALL:PROT PREV6”
360       OUTPUT Testset;”CALL:PAG:DRAT FULL”
370       OUTPUT Testset;”CALL:PIL -7”
380       OUTPUT Testset;”CALL:SYNC -16”
390       OUTPUT Testset;”CALL:PAG -12”
400       OUTPUT Testset;”CALL:FCH -15.6”
401       !
402       OUTPUT Testset;”SYST:SYNC?”
403       ENTER Testset;Syst_sync$
420
```

Configure the Mobile Station Parameters

There are no mobile station parameters configured in this program example.

Step 3: Set Measurement Parameters

This step gives an example of how you can configure measurement parameters. For more information about measurement parameters, refer to the *Introduction to Programming the Agilent Technologies 8960*, Step 3.

```
10 OUTPUT Testset;”SET:CONT OFF”
20 OUTPUT Testset;”SET:WQU:TIM:STIM 10”
30 OUTPUT Testset;”SET:DAP:TIM:STIM 5”
40 OUTPUT Testset;”SET:CPOW:TIM:STIM 5”
50 !
```

Step 4: Make a connection

Select an example

The mobile station (MS) must be connected with the test set in order to perform measurements. The origination of this connection can either be the mobile station or the test set. Use one of two sections of code below to fit your testing need.

Test Set initiated connection

The following code queries the test set for the connection status and then stops the program if a connection is not established. A hard handoff is also performed.

```
513  !*****************************************************************
514  ! CONNECT CALL
515  !*****************************************************************
516 !
526   !***Page the mobile station***
527   !
528       DISP “Paging mobile station...”
529       OUTPUT Testset;”CALL:ORIG”
530       OUTPUT Testset;”CALL:CONN:STAT?”
531       ENTER Testset;Call_connected
532       IF NOT Call_connected THEN
540           PRINT “Call attempt failed”
550           STOP
560       ELSE
570           PRINT “Call connected”
571           PRINT
580       END IF
581       Tot_timer=TIMEDATE
```

Mobile Station initiated connection

In this case, there is no need to use atimeout or arm the state change detector. These two functions occur automatically when attempting a test set originated call.

```
1 PRINT “Originate a call on the mobile now.”
10 OUTPUT Testset;”CALL:CONN:ARM” !Arm the Call-State-Change Detector
20 OUTPUT Testset;”CALL:CONN?” !Query state
```

Step 4: Make a connection

```
30 ENTER Testset;Callstate
40 IF NOT Callstae THEN
41     PRINT “Call Attempt Failed”
42     STOP
43 ELSE
44     PRINT “Call Connected”
45 END IF
50 !
```

Step 5: INITiate and FETCh Measurements

INITiate measurements and FETCh results

Below are examples of performing tests and retrieving their results. Refer to the *Introduction to 8960 Programming* for more information about the different measurement results that are available and how to fetch them.

```
785  !***********************
786  ! WAVEFORM QUALITY TEST*
787  !***********************
788 !
789   !***Specify test parameters for waveform quality test***
790 !
791       Ior=-75
792       Pilot=-7
793       Traffic=-7.4
794 !
796   !***Set up measurement parameters***
797 !
799       OUTPUT Testset;”CALL:POW “;Ior
800       OUTPUT Testset;”CALL:PIL “;Pilot
810       OUTPUT Testset;”CALL:FCH “;Traffic
820   !
830   !***Obtain and display measurement results***
831 !
840       OUTPUT Testset;”SYST:SYNC?”
841       ENTER Testset;Syst_sync$
842       !
843       Wqu_timer=TIMEDATE
850       OUTPUT Testset;”INIT:WQU”
860       LOOP
870           OUTPUT Testset;”INIT:DONE?”
880           ENTER Testset;Meas_done$
890       EXIT IF Meas_done$=”WQU”
900       END LOOP
910       OUTPUT Testset;”FETC:WQU?”
920       ENTER
Testset;Integrity,Rho,Freq_error,Time_error,Carr_feed,Phase_err,Mag_err,
Evm
```

```
921 !
930 !***Print Waveform Quality Results
931 !
932       DISP “Waveform Quality Test complete”
933       PRINT “Waveform Quality Test Results:”
934 !
935       PRINT “     Integrity indicator =    ”,Integrity
940       Print_results(“Rho = “,Rho,””)
941       Print_results(“Frequency error = “,Freq_error,”Hz”)
942       Print_results(“Time error = “,Time_error,”us”)
943       Print_results(“Carrier feedthrough = “,Carr_feed,”dBc”)
950       Print_results(“Phase error = “,Phase_err,”deg”)
951       Print_results(“Magnitude error = “,Mag_err,”%”)
952       Print_results(“Error vector magnitude = “,Evm,”%”)
953       PRINT
980       Wqu_time=PROUND(TIMEDATE-Wqu_timer,-2)
981       Print_results(“WQU Test Time= “,Wqu_time,”secs”)
982       PRINT
983       !
984       !
985  !***********************
986  ! FRAME ERROR RATE TEST*
987  !***********************
988 !
989   !***Set up measurement parameters***
990   !
991       Ior=-70
993       Pilot=-7
994       Traffic=-15.6
995 !
997       OUTPUT Testset;”CALL:POW “;Ior
998       OUTPUT Testset;”CALL:PIL “;Pilot
999       OUTPUT Testset;”CALL:FCH “;Traffic
1000   !
1001   !***Obtain measurement results***
1002       Fer_timer=TIMEDATE
1003       OUTPUT Testset;”INIT:CFER”
1004       DISP “Measuring FER”
1005       LOOP
1006           OUTPUT Testset;”INIT:DONE?”
1007           ENTER Testset;Meas_done$
1008       EXIT IF Meas_done$=”CFER”
```

```
1009       END LOOP
1010       OUTPUT Testset;”FETC:CFER?”
1012       ENTER
Testset;Integrity,Fer_test,Fer_ratio,Fer_count,Frames_tested
1013   !
1014       SELECT Fer_test
1015       CASE 0
1016           Pass_fail$=”PASSED”
1017       CASE 1
1018           Pass_fail$=”FAILED”
1019       CASE 2
1020           Pass_fail$=”MAXIMUM NUMBER OF FRAMES WERE TESTED”
1021       CASE 3
1022           Pass_fail$=”UNKNOWN”
1023       END SELECT
1024       !
1025   !***Print FER Results***
1026 !
1027       DISP “FER Test complete”
1028       !
1029       PRINT “Frame Error Rate Test Results:”
1030 !
1031       PRINT “     Integrity indicator =    “,Integrity
1032       Print_res_str(“FER Test Results”,Pass_fail$)
1033       Print_results(“FER Ratio”,Fer_ratio,”%”)
1034       Print_results(“FER Errors Count”,Fer_count,”frames”)
1035       Print_results(“Frames Tested”,Frames_tested,””)
1036       PRINT
1037       !
1039       Fer_time=PROUND(TIMEDATE-Fer_timer,-2)
1040       Print_results(“FER Test Time= “,Fer_time,”secs”)
1041       PRINT
1042       !
1043  !***************
1044  !Max Power Test*
1045  !***************
1046 !
1047   !***Set up measurement parameters***
1048   !
1049       Ior=-70
1051       Pilot=-7
1052       Traffic=-7.4
```

```
1053 !
1055       OUTPUT Testset;”CALL:POW “;Ior
1056       OUTPUT Testset;”CALL:PIL “;Pilot
1057       OUTPUT Testset;”CALL:FCH “;Traffic
1058       OUTPUT Testset;”CALL:CLPC:REV:MODE UP”
1060   !***Obtain measurement results***
1061   !
1062       WAIT 1!Settling time
1063       Max_timer=TIMEDATE
1064       OUTPUT Testset;”INIT:DAP”
1065       LOOP
1066           OUTPUT Testset;”INIT:DONE?”
1067           ENTER Testset;Meas_done$
1068       EXIT IF Meas_done$=”DAP”
1069       END LOOP
1070       OUTPUT Testset;”FETC:DAP?”
1071       ENTER Testset;Integrity,Avg_power_dbm
1072       !
1073       Avg_power_dbw=Avg_power_dbm-30
1074   !
1075   !***Display measurements if the mobile station did not drop the
call***
1076   !
1077       OUTPUT Testset;”CALL:STAT?”
1078       ENTER Testset;Call_status$
1079       IF Call_status$<>”CONN” THEN
1080           PRINT “Mobile station dropped call, invalid results”
1081       END IF
1082       !
1083   !***Print Max Power Test Results
1084   !
1085       DISP “Max Power Test complete”
1086       PRINT “Max Power Test Results:”
1087 !
1088 !
1089       PRINT “     Integrity indicator =    ”,Integrity
1090       Print_results(“Maximum Power dBm:”,Avg_power_dbm,”dBm”)
1091       Print_results(“Maximum Power dBW:”,Avg_power_dbw,”dBW”)
1095       PRINT
1097      !
1098       Max_time=PROUND(TIMEDATE-Max_timer,-2)
1099       Print_results(“Max Power Test Time= “,Max_time,”secs”)
```

```
1100       PRINT
1101       !
1102   !***Post Test clean up***
1103   !
1104       OUTPUT Testset;”CALL:CLPC:REV:MODE ACT”
1105 !
1106  !*******************
1107  !Minimum Power Test*
1108  !*******************
1109 !
1110 !
1111   !***Set up measurement parameters***
1112   !
1113       Ior=-25
1114       Pilot=-7
1115       Traffic=-7.4
1116 !
1118       OUTPUT Testset;”CALL:CONN:DROP:TIM 0”
1119       OUTPUT Testset;”CALL:POW “;Ior
1120       OUTPUT Testset;”CALL:PIL “;Pilot
1121       OUTPUT Testset;”CALL:FCH “;Traffic
1122       OUTPUT Testset;”CALL:CLPC:REV:MODE DOWN”
1124   !
1125   !***Obtain measurement results***
1126   !
1127       WAIT 1
1128       Min_timer=TIMEDATE
1129       OUTPUT Testset;”INIT:CPOW”
1130       LOOP
1131           OUTPUT Testset;”INIT:DONE?”
1132           ENTER Testset;Meas_done$
1133       EXIT IF Meas_done$=”CPOW”
1134       END LOOP
1135       !
1136       OUTPUT Testset;”FETC:CPOW?”
1137       ENTER Testset;Integrity,Channel_power
1138           !
1139                !
1140            IF Integrity=6 THEN
1142                OUTPUT Testset;”RFAN:CONT:POW:AUTO OFF”
1143                OUTPUT Testset;”RFAN:MAN:POW “;INT(Channel_power);”
DBM”
```

```
1145            END IF
1146       EXIT IF Meas_done$=”CPOW” AND Integrity=0
1147       END LOOP
1148       !
1149         !
1150    ***Print measurement results
1151       !
1152       DISP “Min Power Test complete”
1153       PRINT “Min Power Test Results:”
1154       PRINT “     Integrity indicator =    “,Integrity
1156       Print_results(“Maximum Power dBm:”,Channel_power,”dBm/1.23
MHz”)
1157       PRINT
1158       Min_time=PROUND(TIMEDATE-Min_timer,-2)
1161       Print_results(“Min Power Test Time= “,Min_time,”secs”)
1162       PRINT
1163       !
1164   !***Post Minimum Power Test Cleanup***
1165   !
1166       OUTPUT Testset;”CALL:CLPC:REV:MODE ACT”
1167       OUTPUT Testset;”CALL:CONN:DROP:TIM:STAT 1”
1168       OUTPUT Testset;”RFAN:CONT:POW:AUTO ON”
```

Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

Reconfigure the Test Set

The example below contains the code appropriate for handing the call off to a different channel.

```
600    !***Set up a hard handoff***
610       OUTPUT Testset;”CALL:SET:BAND USC”
620       OUTPUT Testset;”CALL:SET:CHAN “;Handoffchan
630 !
640    !***Perform a hard handoff***
650       OUTPUT Testset;”CALL:HAND”
660       OUTPUT Testset;”CALL:CONN:STAT?”
670       ENTER Testset;Call_connected
680       IF NOT Call_connected THEN
690           DISP “Handoff attempt failed, program stopped”
700           STOP
710       ELSE
720           PRINT “Handoff completed”
730           DISP “Call connected”
731           PRINT
740       END IF
784   !
```

Reconfigure the MS

There are no MS parameters reconfigured in this program example.

Step 7: End the Connection

End the Connection

The CALL:END command ends the mobile connection. Subroutines which are called in the program are also shown.

```
1172  !****************************
1173  ! End of All Testing
1174  !****************************
1175 !
1176 !***End Call and Return to default call processing settings***
1177 !
1178       OUTPUT Testset;”CALL:POW:DIG2000 -50”
1179       OUTPUT Testset;”CALL:END”
1180       OUTPUT Testset;”CALL:STAT:STAT?”
1181       ENTER Testset;Call_status$
1182       PRINT “Call Ended, Status:”,Call_status$
1183       Tot_time=PROUND(TIMEDATE-Tot_timer,-2)
1184       PRINT
1185       Print_results(“Test Time= “,Tot_time,”secs”)
1186       PRINT
1187       !
1188   !***End Program***
1189  End_program: ! Secondary timeout handler
1190               DISP “End of Program”
1191               END
1192  !
1193  !********************
1194  ! Subroutine Section*
1195  !********************
1196  !
1197  Timeout: SUB Timeout
1198           PRINT “Program timed out”
1199           CLEAR 7
1200           CLEAR 714
1201           STOP
1202       SUBEND
1203  Print_results: SUB Print_results(Meas_name$,Res1,Units$)
1204 !
```

Step 7: End the Connection

```
1205           PRINT USING
“5X,20A,5X,M4D.2D,1X,5A”;Meas_name$;Res1;Units$
1206 !
1404       SUBEND
1414  Print_res_str: SUB Print_res_str(String_name$,Results$)
1415 !
1424           PRINT USING “5X,20A,5X,20A”;String_name$;Results$
1425 !
1444       SUBEND
```

Programming: Getting Started Guide

for

E6701A GPRS Lab Application A.01
E1964A GPRS Mobile Test Application A.02

Introduction

The Getting Started Guide is organized around the basic set of tasks a control program normally performs when testing a mobile station in a manufacturing environment.

Conventions used in this Getting Started Guide

Throughout this Getting Started Guide the term “test set” refers to an Agilent Technologies 8960 Series 10 wireless communications test set with either the E1964A GPRS mobile test application or the E6701A GPRS lab application installed or both.

The variable `Test_set` used in the steps of the Getting Started Guide refers to the test set’s GPIB address.

How to use this Getting Started Guide

The most effective way to use this Getting Started Guide is with the Programming Flowchart included inside this guide and the 8960 User Documentation. This documentation is found in two locations:

- the 8960 Family Support Site (updated frequently) on the Internet, or
- the User Documentation CD-ROM shipped with your test application.

Each step on the Programming Flowchart is illustrated with example program code in this guide. Using this Getting Started Guide, the Programming Flowchart, and the on-line information about 8960 programming, you will be able to generate a control program to perform fundamental mobile station manufacturing tests.

Useful on-line links

Go to the 8960 Family Support Site on either the CD-ROM or the Internet. The CD-ROM should self-launch to its home page. The URL for the website is:

http://www.agilent.com/find/8960support/

Navigate to the Programming page for this test application. The illustrations on the back of the Programming Flowchart show you how to navigate to this page.

The following links are under the heading Getting Started on the Programming page:

- Programming: Getting Started Guide
  - This on-line version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.
- Control Program Examples
  - These examples are for you to download. You may want to use these as templates for your own control program or to execute.
  - The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

About the Programming Examples Presented in this Guide

Programming Language:

Programming examples presented in this guide are written in the HP BASIC programming language, also known as RMB or Rocky Mountain BASIC. The use of HP BASIC is not an endorsement of the HP BASIC product.

Line Numbers

All of the programming examples in the guide with line numbers are sections from a control program example available on-line for you to download.

Code that is not part of the download control program example does not have line numbers. This code may represent an alternate method of performing the task or may illustrate a feature not used by the control program example.

Syntax used in Programming Examples:

- The programming examples use the shortened form of the command syntax to minimize GPIB bus transactions. The shortened form of a command is defined by use of capital letters in the command syntax. For the command syntax:

  ```
  RFANalyzer:CONTrol:MEASurement:FREQuency:AUTO?
  ```

  the shortened form would be:

  ```
  RFAN:CONT:MEAS:FREQ:AUTO?
  ```

- The programming examples do not include optional nodes. Optional nodes in the command syntax are defined by enclosing the node inside the [ ] brackets. For example, the command syntax:

```
CALL[:CELL]:POWer[:SAMPlitude] -80dBm
```

appears in the programming examples as:

```
CALL:POW -80dBm
```

- Programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the on-line information for the definition and use of these command separators.

Complex Commands

Complex commands are used to configure the state and assign values to parameters simultaneously. Complex commands can be used to save programming steps and minimize GPIB bus transactions.

- The syntax below turns the state of the parameter on.

  OUTPUT Test_set;"SET:DTXP:TIM:STAT ON"

- The syntax below is used to assign a value to the parameter.

  OUTPUT Test_set;"SET:DTXP:TIM:TIME 10 S"

- Both of the above actions can be accomplished with one syntax command:

  OUTPUT Test_set;"SET:DTXP:TIM:STIM 10 S"

  The command above sets the parameter state to ON and the value of the parameter to 10 seconds. Note that in this example the optional command mnemonic :STIMe has been included to clarify that this complex command was used to set both the state and the value.

- This command can be shortened further by removing the optional command mnemonic :STIMe, as shown below.

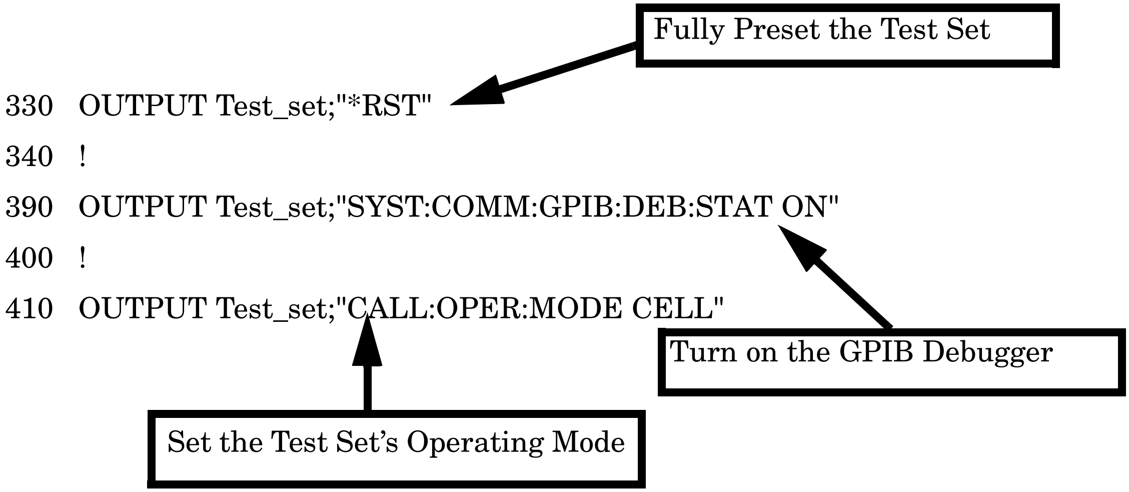
  OUTPUT Test_set;"SET:DTXP:TIM 10 S"

  This is the format that will be used throughout this guide.

Step 1: Set up the Test Set

This step explains how to:

- “Fully Preset the Test Set”
- “Turn on the GPIB Debugger”
- “Set Test Set’s Operating Mode”



```
330   OUTPUT Test_set;"*RST"
340   !
390   OUTPUT Test_set;"SYST:COMM:GPIB:DEB:STAT ON"
400   !
410   OUTPUT Test_set;"CALL:OPER:MODE CELL"
```

Fully Preset the Test Set

Turn on the GPIB Debugger

Set the Test Set’s Operating Mode

Fully Preset the Test Set

To set up the test set, you begin by sending the *RST command. The *RST is used to perform a full preset of the test set, returning it to a known state. *RST also sets all measurements to single trigger.

Turn on the GPIB Debugger

Another useful tool that you may want to turn on at this time is the GPIB command debugger. While turned on, error messages appear on the test set’s screen when the test set receives an unknown GPIB command. The message contains information about what type of error was caused and indicates where in the syntax the error occurred. Troubleshooting, like locating and fixing typing errors for example, becomes easier using the GPIB command debugger.

NOTE The SYST:COMM:GPIB:DEB:STAT ON command assists you when debugging code. This command should be taken out of your code once development is completed.

Set Test Set's Operating Mode

The `CALL:OPER:MODE` command in the diagram is used to set the test set's operating mode to active cell. This means call processing is used to maintain a link between the test set and the mobile station.

There are also three test mode operating modes available. These test modes are useful when you are operating the mobile station in test mode. See the 8960 Family Support Site on the Internet for more information on test modes.

Step 2: Configure Test Set and Mobile Station Parameters

This step explains how to:

- “Set up data connection synchronization parameters”
- “Configure the Broadcast Channel Parameters”
- “Configure the Packet Data Traffic Channel Parameters”
- “Configure the Mobile Station Operating Conditions”

Set up data connection synchronization parameters

You may choose to set the data connection synchronization timeout parameter at this point in the program. The data connection synchronization command associated with this timeout parameter is used in “Step 4: Make a connection” on page 61 and “Step 7: End the Connection” on page 69.

```
490   OUTPUT Test_set;”CALL:DCON:TIM 10”
```

NOTE Many of the parameters are being configured to their default values. In a manufacturing environment it may be desirable to explicitly configure these parameters to ensure that the required settings have not been changed by someone setting a parameter’s value through the test set’s front panel. However, greater code efficiency can be achieved by not configuring them.

Configure the Broadcast Channel Parameters

The programming example below illustrates configuring the active band to PGSM and setting the cell power to -80 dBm.

```
520   OUTPUT Test_set;”CALL:BAND PGSM”! Set active broadcast band
530   OUTPUT Test_set;”CALL:POW -80”  ! Set cell power to -80 dBm.
```

Configure the Packet Data Traffic Channel Parameters

In the next example, several packet data traffic channel parameters are specified. Note that the PDTCH ARFCN is set to 45. This is on a different ARFCN than the broadcast channel (20 is the default broadcast channel ARFCN for PGSM), allowing the use of the power reduction levels specified and assigned to the downlink bursts.

```
560   OUTPUT Test_set;”CALL:PDTCH 45” ! Set packet data traffic channel
570   !  Set Data Connection Type to BLER
580   OUTPUT Test_set;”CALL:FUNC:DATA:TYPE BLER”
590   !
600   !  The following commands are helpful in enabling the test set to
610   !  establish a data connection with mobiles which cannot establish
620   !  a BLER data connection under normal conditions.
630   !
640   !  Set the test set to send an invalid FCS to the mobile
650   OUTPUT Test_set;”CALL:FUNC:DATA:BLER:LLC:FCS CORR”!
660   !  Set the block polling interval used during BLER connection
670   OUTPUT Test_set;”CALL:FUNC:DATA:BLER:POLL:INT 4”! Set polling rate
680   !  Set the type of frame numbering scheme used during attach.
690   OUTPUT Test_set;”CALL:FUNC:DATA:FRAM:STAR ABS”! Absolute attach
700   !
710   ! Set Multi-slot Configuration to two downlinks and one uplinks.
720   OUTPUT Test_set;”CALL:PDTCH:MSL:CONF D2U1”
730   OUTPUT Test_set;”CALL:PDTCH:CSCH CS4”! Set Coding Scheme to CS4
740   !
750   ! Assign values to the power reduction levels
760   OUTPUT Test_set;”CALL:PDTCH:PZER:LEV 30”   ! Assign a value to P0
770   OUTPUT Test_set;”CALL:PDTCH:PRED:LEV1 11”! Set PRL1 to 11 dB
780   OUTPUT Test_set;”CALL:PDTCH:PRED:LEV2 0”   ! Set PRL2 to 0 dB
790   !
800   ! Assign power levels to the downlink bursts
810   ! Assign Burst 1 a PRL of 0 dB
820   OUTPUT Test_set;”CALL:PDTCH:PRED:BURS1 PRL2”
830   ! Assign Burst 2 a PRL of 11 dB
840   OUTPUT Test_set;”CALL:PDTCH:PRED:BURS2 PRL1”
```

Configure the Mobile Station Operating Conditions

The example below illustrates how to assign mobile station transmit levels to two uplink bursts.

```
870   OUTPUT Test_set;”CALL:PDTCH:MS:TXL:BURS 5”
880   OUTPUT Test_set;”CALL:PDTCH:MS:TXL:BURS2 3”
```

Step 3: Set Measurement Parameters

This step gives an example of how you can configure measurement parameters. For more information about measurement parameters, refer to the additional details about this step available on the Internet.

```
940   ! Configure ORFS Measurement:
950   !
960   ! The lines below are examples of using complex commands to set
970   ! multi-meas state and count at the same time.
980   OUTPUT Test_set;”SET:ORFS:SWIT:COUN 5”
990   OUTPUT Test_set;”SET:ORFS:MOD:COUN 10”
1000  !
1010  OUTPUT Test_set;”SET:ORFS:TRIG:SOUR AUTO”! Set trig source to
AUTO.
1020  OUTPUT Test_set;”SET:ORFS:CONT OFF”      ! Set trig mode to single.
1030  OUTPUT Test_set;”SET:ORFS:TIM 60”        ! Set timeout time.
1040  ! Put switching and modulation offsets to be tested into string
1050  ! variables.  Swit$ contains switching offsets.  Mod$ contains
1060  ! modulation offsets.
1070  Swit$=”.4MHZ,-.4MHZ,.6MHZ,-.6MHZ,1.2MHZ,-1.2MHZ,1.8MHZ,-1.8MHZ”
1080  Mod$=”.4MHZ,-.4MHZ,.6MHZ,-.6MHZ,1.2MHZ,-1.2MHZ,1.8MHZ,-1.8MHZ”
1090  OUTPUT Test_set;”SET:ORFS:SWIT:FREQ “&Swit$
1100  OUTPUT Test_set;”SET:ORFS:MOD:FREQ “&Mod$
1110  !
1120  ! Configure TX Power Measurement:
1130  !
1140  OUTPUT Test_set;”SET:TXP:COUN 3;CONT OFF;TIM 20”
1150  OUTPUT Test_set;”SET:TXP:TRIG:SOUR AUTO;QUAL ON”
1160  !
1170  ! Configure Phase & Frequency Error Measurement:
1180  !
1190  OUTPUT Test_set;”SET:PFER:COUN 8;CONT OFF;TIM 30;SYNC MID”
1200  OUTPUT Test_set;”SET:PFER:TRIG:SOUR AUTO”
1210  !
1220  ! Configure multislot measurements to be made on burst 1
1230  OUTPUT Test_set;”RFAN:MSL:MEAS:BURS 1”
```

Step 3: Set Measurement Parameters

Step 4: Make a connection

This step explains how to make a connection with the mobile station. The two actions you must perform to accomplish this are:

- "Mobile Station performs a GPRS Attach"
- "Start a Data Connection"

Mobile Station performs a GPRS Attach

The mobile station must initiate a GPRS attach. It is assumed the phone is turned on and is attempting a GPRS attach. The following code queries the test set for the connection status and then stops the program if an attached state is not reached within one minute. This gives adequate time for the mobile station to attach.

Synchronization between the control program and the test set is maintained by querying the test set for the state of the connection between it and the mobile station. When a GPRS Attach is being performed, the `CALL:ATT?` hanging query is the appropriate query to use. It will return a "1" when the mobile station is attached and a "0" otherwise. A state change detector is also armed to ensure the query does not stop hanging before the state transition from "IDLE" to "ATTached" is able to begin. Finally, to prevent the query from hanging indefinitely, which could occur if the mobile station is not turned on or not sent the correct commands, a timeout is set for this query. In this example, 10 seconds is the value assigned to the timeout. After 10 seconds, the change detector is disarmed and the query returns either a "1" or "0".

```
1350  DISP "Turn the phone on now.  Make the phone perform a GPRS
attach."
1360  Start_time=TIMEDATE
1370  LOOP
1380    OUTPUT Test_set;"CALL:DCON:ARM"
1390    OUTPUT Test_set;"CALL:ATT?"
1400    ENTER Test_set;Att_state
1410   EXIT IF Att_state
1420    Current_time=TIMEDATE-Start_time
1430    IF Current_time>=Timer THEN       ! Timer value is 1 minute
1440      BEEP
1450      DISP ""
1460      PRINT "GPRS attach did not complete. Program terminated."
```

```
1470      STOP
1480    END IF
1490  END LOOP
```

Start a Data Connection

Once the mobile station has successfully attached, you can start the data connection using the CALL:FUNC:DATA:STAR command. The CALL:FUNC:DATA:STAR command is a overlapped command, meaning the test set accepts other commands before completely processing this command. Because this is an overlapped command, synchronization is maintained by using the CALL:TRAN? hanging query. It allows the test set to initiate the data connection and then returns a “1” if the data connection starts successfully and a “0” if the state of the connection returns to either “IDLE” or “ATTached”. Because the test set originated the data connection, it is not necessary to assign a timeout value or arm the change detector for this query. Instead, there is a default timer associated with this query and the change detector is armed automatically.

```
1520  OUTPUT Test_set;”CALL:FUNC:DATA:STAR”
1530  OUTPUT Test_set;”CALL:TRAN?”
1540  ENTER Test_set;Tran_state
1550  IF NOT Tran_state THEN
1560    BEEP
1570    DISP ““
1580    PRINT “Data connection failed.  Program terminated.”
1590    STOP
1600  END IF
```

Step 5: INITiate and FETCh Measurements

This step explains how to:

- “INITiate a set of measurements”
- “FETCh measurement results”
- “Making the BLER Measurement”

INITiate a set of measurements

The example below demonstrates how to start three measurements running concurrently. The SYST:MEAS:RES command resets the Block Error Rate measurement results. This is done to ensure BLER reports are collected during a known good connection between the test set and the mobile station.

```
1680  OUTPUT Test_set;”SYST:MEAS:RES” ! Reset the BLER results
1690  OUTPUT Test_set;”INIT:ORFS;TXP;PFER”
```

FETCh measurement results

In a typical control program, measurements are repeated on various frequencies and power levels. Therefore, it is desirable to have a subroutine capable of fetching multiple measurement results. The example code below demonstrates how you might create a subroutine for fetching the measurement results. The variable Pdtch contains the ARFCN the measurement is being made on. The variables Ms_pwr_bs1 and Ms_pwr_bs2 refer to the current power level assigned to the uplink bursts. Refer to the additional details about this step available on the Internet for more information about the different measurement results that are available and how to fetch them.

```
1730  CALL Global_fetch

3160  SUB Global_fetch
3170    OPTION BASE 1
3180    COM /Address/Test_set
3190    OUTPUT Test_set;”CALL:PDTCH?”
3200    ENTER Test_set;Pdtch
3210    OUTPUT Test_set;”CALL:PDTCH:MS:TXL:BURS?;BURS2?”
3220    ENTER Test_set;Ms_pwr_bs1;Ms_pwr_bs2
```

```
3230  !
3240  !  Determine if a measurement is done:
3250  !
3260   LOOP
3270     OUTPUT Test_set;”INIT:DONE?”
3280     ENTER Test_set;Meas_done$
3290  !
3300  ! Obtain measurement results: Each measurement illustrates a
3310  ! different way of reading in results. There is no one right way.
3320  ! The method used is application dependent. Note that the examples
3330  ! do not show all possible ways.
3340  !
3350     SELECT Meas_done$
3360  !
3370     CASE “TXP”  ! TX Power measurement done.
3380       ALLOCATE Txpower(4)
3390       OUTPUT Test_set;”FETC:TXP:INT?;POW:ALL?”
3400       ENTER Test_set;Integrity,Txpower(*)
3410       IF (Integrity=0) THEN  ! Always check integrity value.
3420         PRINT “TX Power results: PDTCH=”;Pdtch
3430         PRINT “                  Burst1 TXL=”;Ms_pwr_bs1
3440         PRINT “                  Burst2 TXL=”;Ms_pwr_bs2
3450         PRINT USING “5X,””Minimum:””,M2D.2D,”” dBm”””;Txpower(1)
3460         PRINT USING “5X,””Maximum:””,M2D.2D,”” dBm”””;Txpower(2)
3470         PRINT USING “5X,””Average:””,M2D.2D,”” dBm”””;Txpower(3)
3480         PRINT USING “5X,””Std Dev:””,M2D.2D,”” dB”””;Txpower(4)
3490         DEALLOCATE Txpower(*)
3500       ELSE
3510         GOSUB Bad_measurement
3520       END IF
3530  !
3540     CASE “PFER”  ! Phase & Frequency Error measurement done.
3550       OUTPUT Test_set;”FETC:PFER:ALL?”
3560       ENTER Test_set;Integrity,Rms_ph_er,Peak_ph_er,Worst_frq_er
3570       IF (Integrity=0) THEN
3580         PRINT “PFERror results: PDTCH=”;Pdtch
3590         PRINT “                 Burst1 TXL=”;Ms_pwr_bs1
3600         PRINT “                 Burst2 TXL=”;Ms_pwr_bs2
3610         PRINT “RMS Phase Error: “;Rms_ph_er;” deg”
3620         PRINT “Peak Phase Error: “;Peak_ph_er;” deg”
3630         PRINT “Worst Freq Error: “;Worst_frq_er;” Hz”
3640       ELSE
```

```
3650          GOSUB Bad_measurement
3660        END IF
3670  !
3680      CASE “ORFS”  ! ORFS measurement done.
3690  !
3700  ! This code illustrates a more ‘generic’ approach to reading
3710  ! measurement results. By using the capabilities designed into
3720  ! high-level measurements, routines that access measurement
3730  ! results do not have to explicitly know what the measurement
3740  ! execution conditions were.  That information can be determined
3750  ! at the time the measurement results are queried.
3760  !
3770        OUTPUT Test_set;”FETC:ORFS:INT?”        ! Check integrity.
3780        ENTER Test_set;Integrity
3790        IF (Integrity=0) THEN
3800          ! Get the number of offsets tested.
3810          OUTPUT Test_set;”SET:ORFS:SWIT:FREQ:POIN?”
3820          ENTER Test_set;Points
3830          IF Points THEN  ! Only query if one or more offsets tested.
3840            ALLOCATE Swit_res(Points),Swit_offs(Points)
3850            ! Get measurement offsets.
3860            OUTPUT Test_set;”SET:ORFS:SWIT:FREQ?”
3870            ENTER Test_set;Swit_offs(*)
3880            ! Get results
3890            OUTPUT Test_set;”FETC:ORFS:POW?;:FETC:ORFS:SWIT?”
3900            ENTER Test_set;Tx_power,Swit_res(*)
3910            PRINT “ORFS Swit Results: PDTCH=”;Pdtch
3920            PRINT “                   Burst1 TXL=”;Ms_pwr_bs1
3930            PRINT “                   Burst2 TXL=”;Ms_pwr_bs2
3940           PRINT USING “19X,””TX Power =””,M2D.2D,”” dBm”””;Tx_power
3950            PRINT “     Offset(kHz)          Level(dBm)”
3960            PRINT “     -----------          ----------”
3970 Orfs_image:  IMAGE 6X,M4D.2D,12X,M4D.2D
3980            FOR J=1 TO Points
3990             PRINT USING Orfs_image;(Swit_offs(J)/1.E+3),Swit_res(J)
4000            NEXT J
4010            DEALLOCATE Swit_res(*),Swit_offs(*)
4020          END IF
4030          ! Get the number of offsets tested.
4040          OUTPUT Test_set;”SET:ORFS:MOD:FREQ:POIN?”
4050          ENTER Test_set;Points
4060          IF Points THEN ! Only query if one or more offsets tested.
```

```
4070            ALLOCATE Mod_res(Points),Mod_offs(Points)
4080            ! Get measurement offsets
4090            OUTPUT Test_set;”SET:ORFS:MOD:FREQ?”
4100            ENTER Test_set;Mod_offs(*)
4110            ! Get results
4120            OUTPUT Test_set;”FETC:ORFS:POW?;:FETC:ORFS:MOD?”
4130            ENTER Test_set;Tx_power,Pwr_30khz,Mod_res(*)
4140            PRINT “ORFS Mod Results: TCH=”;Tch;”and TXL=”;Ms_pwr_lvl
4150            PRINT “30 KHz BW Power =”;Pwr_30khz;” dBm”
4160            PRINT “     Offset(kHz)          Level(dB)”
4170            PRINT “     -----------          ---------”
4180            FOR J=1 TO Points
4190              PRINT USING Orfs_image;(Mod_offs(J)/1.E+3),Mod_res(J)
4200            NEXT J
4210            DEALLOCATE Mod_res(*),Mod_offs(*)
4220          END IF
4230        ELSE
4240          GOSUB Bad_measurement
4250        END IF
4260      END SELECT
4270    EXIT IF Meas_done$=”NONE”
4280    END LOOP  ! If ‘WAIT’ is returned from ‘INIT:DONE?’ query, it
4290              ! just falls through the loop.
4300    SUBEXIT
```

Making the BLER Measurement

After initiating and fetching measurement results, the phone has been able to make several BLER measurements. In the example below, the range for the number of blocks tested is between 2000 and 2300. Test time has been saved by resetting the BLER reports before making any other measurements. This allows the BLER results for a number of blocks to be obtained concurrent to the transmitter measurements being performed. The example below illustrates how to finally retrieve the BLER measurement results and the number of blocks tested.

```
1740  REPEAT
1750    OUTPUT Test_set;”CALL:STAT:PDTCH:BLER?”
1760    ENTER Test_set;Bler,Blocks
1770  UNTIL (Blocks>=2000 AND Blocks<=2300)
1780  PRINT
1790  PRINT “Blocks tested at -101 dBm = “;Blocks
1800  PRINT “BLER Result at downlink PDTCH power of -101 dBm = “;Bler
```

Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

There are several ways you may want to reconfigure the connection parameters when you reach this step. Some examples are:

- "Reconfigure the Mobile Station Parameters"
- "Reconfigure the connection to a new PDTCH ARFCN"
- "Reconfigure the connection to a new PDTCH ARFCN in a different band"

Reconfigure the Mobile Station Parameters

The example below illustrates how to change the mobile station's transmit level for burst 1 to level 10. The default burst for this command is burst 1.

```
2030  OUTPUT Test_set;"CALL:PDTCH:MS:TXL:BURS 10"
```

Reconfigure the connection to a new PDTCH ARFCN

When reconfiguring the connection to a new packet data traffic channel ARFCN, you may also want to change the mobile station TX Level as well. In this case, using deferred parameter commands would create the most efficient code. The code below shows how to set a new mobile station TX level and PDTCH ARFCN with deferred parameter commands, and then use the `CALL:HAND` command to apply the new parameters. The code also shows you how to use the `CALL:STAT:DATA?` synchronization command to make sure the connection was maintained. The `CALL:STAT:DATA?` query returns the current state of the connection. In this case, "TRAN" should be returned, indicating the connection is still in the transferring state and the handover was successful. This query can be used because the `CALL:HAND` command is sequential, meaning its operation completes before the test set accepts a new command.

```
2250  OUTPUT Test_set;"CALL:SET:PDTCH:MS:TXL:BURS 5"
2260  OUTPUT Test_set;"CALL:SET:PDTCH 120"
2270  OUTPUT Test_set;"CALL:HAND"
2280  ! Use a call synchronization command to ensure the
2290  ! reconfiguration succeeded.
2300  OUTPUT Test_set;"CALL:STAT:DATA?"
```

```
2310  ENTER Conn_status$
2320  IF Conn_status$<>"TRAN" THEN
2330     PRINT "Data connection failed to reconfigure properly."
2340     PRINT "Program terminated."
2350     STOP
2360  END IF
```

Reconfigure the connection to a new PDTCH ARFCN in a different band

When configuring the connection to a new band, remember to specify the band for the new mobile station TX level and PDTCH ARFCN. In the code below, DCS has been added to the commands for reconfiguring both these parameters. The result is that those parameters are stored until the DCS band is made active by the CALL:HAND command. If they are not specified as DCS band parameters, they become active immediately.

Note that reconfiguring the connection to a new band uses the same synchronization method as changing to a new PDTCH.

```
2520  OUTPUT Test_set;"CALL:PDTCH:MS:TXL:DCS:BURS 15"
2530  OUTPUT Test_set;"CALL:PDTCH:DCS 665"
2540  OUTPUT Test_set;"CALL:PDTCH:BAND DCS"
2550  ! Use a call syncrhonization command to ensure the
2560  ! reconfiguration succeeded.
2570  OUTPUT Test_set;"CALL:STAT:DATA?"
2580  ENTER Conn_status$
2590  IF Conn_status$<>"TRAN" THEN
2600     PRINT "Data connection failed to reconfigure properly."
2610     PRINT "Program terminated."
2620     STOP
2630  END IF
```

Step 7: End the Connection

This step explains how to end the connection with the mobile station. The two actions you perform to accomplish this are:

- “End the Data Connection”
- “GPRS Detach”

End the Data Connection

The CALL:FUNC:DATA:STOP command ends the data connection. As in “Step 4: Make a connection” on page 61, synchronization with the control program is important. The CALL:FUNC:DATA:STOP command is a sequential command, meaning its operation is completed before the test set accepts another command. Therefore, it is only necessary to use the CALL:STAT:DATA? query to ensure the data connection has ended and the connection is in the “ATTached” state.

```
2790  OUTPUT Test_set;”CALL:FUNC:DATA:STOP”
2800  OUTPUT Test_set;”CALL:STAT:DATA?”
2810  ENTER Test_set;Conn_status$
2820  IF Conn_status$<>”ATT” THEN
2830    PRINT “Unable to terminate data connection correctly.”
2840    PRINT “PROGRAM TERMINATED.”
2850    STOP
2860  END IF
```

GPRS Detach

The test set does not require you to perform a GPRS detach. No errors are generated if a GPRS detach is not performed. Therefore, you may choose to remove the tested phone after the data connection has ended.

There are two ways to initiate a GPRS Detach.

- “Initiating the GPRS Detach from the Test Set”
- “Mobile Station initiated GPRS Detach”

Initiating the GPRS Detach from the Test Set

The example below illustrates initiating a GPRS Detach from the Test Set.

```
2920  OUTPUT Test_set;”CALL:FUNC:DATA:DET”
2930  !
2940  Start_time=TIMEDATE
2950  LOOP
2960    OUTPUT Test_set;”CALL:DCON:ARM”
2970    OUTPUT Test_set;”CALL:ATT?”
2980    ENTER Test_set;Att_state
2990   EXIT IF NOT Att_state
3000    Current_time=TIMEDATE-Start_time
3010    IF Current_time>=Timer THEN
3020      DISP “”
3030      PRINT “GPRS detach did not occur.  Program terminated”
3040      STOP
3050    END IF
3060    IF Conn_state$=”DET” THEN
3070      DISP “GPRS detach is in process.”
3080    END IF
3090  END LOOP
```

Mobile Station initiated GPRS Detach

The example below illustrates initiating a GPRS Detach from the mobile station.

```
DISP “Initiate a GPRS Detach”
Start_time=TIMEDATE
LOOP
  OUTPUT Test_set;”CALL:STAT:DATA?”
  ENTER Test_set;Conn_state$
 EXIT IF Conn_state$=”IDLE”
  Current_time=TIMEDATE-Start_time
  IF Current_time>=Timer THEN
    DISP “”
    PRINT “GPRS detach did not occur.  Program terminated”
    STOP
  END IF
  IF Conn_state$=”DET” THEN
    DISP “GPRS detach is in process.”
  END IF
END LOOP
```

Programming: Getting Started Guide

for

E1960A GSM Mobile Test Application Revision A.08

Introduction

The Getting Started Guide is organized around the basic set of tasks a control program normally performs when testing a mobile station in a manufacturing environment.

Conventions used in this Getting Started Guide

Throughout this Getting Started Guide the term “test set” refers to an Agilent Technologies 8960 Series 10 wireless communications test set with the E1960A GSM mobile test application installed.

The variable `Test_set` used in the steps of the Getting Started Guide refers to the test set’s GPIB address.

How to use this Getting Started Guide

The most effective way to use this Getting Started Guide is with the Programming Flowchart included inside this guide and the 8960 User Documentation. This documentation is found in two locations:

- the 8960 Family Support Site (updated frequently) on the Internet, or
- the User Documentation CD-ROM shipped with your test application.

Each step on the Programming Flowchart is illustrated with example program code in this guide. Using this Getting Started Guide, the Programming Flowchart, and the on-line information about 8960 programming, you will be able to generate a control program to perform fundamental mobile station manufacturing tests.

Useful on-line links

Go to the 8960 Family Support Site on either the CD-ROM or the Internet. The CD-ROM should self-launch to its home page. The URL for the website is:

www.agilent.com/find/8960support/

Navigate to the Programming page for this test application. The illustrations on the back of the Programming Flowchart show you how to navigate to this page.

The following links are under the heading Getting Started on the Programming page:

- Programming: Getting Started Guide
  - — This online version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.
- Control Program Examples
  - — These examples are for you to download. You may want to use these as templates for your own control program or to execute.
  - — The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

About the Programming Examples Presented in this Guide

Programming Language:

Programming examples presented in this guide are written in the HP BASIC programming language, also known as RMB or Rocky Mountain BASIC. The use of HP BASIC is not an endorsement of the HP BASIC product.

Line Numbers

All of the programming examples in the guide with line numbers are sections from a control program example available on-line for you to download.

Code that is not part of the download control program example does not have line numbers. This code may represent an alternate method of performing the task or may illustrate a feature not used by the control program example.

Syntax used in Programming Examples:

- The programming examples use the shortened form of the command syntax to minimize GPIB bus transactions. The shortened form of a command is defined by use of capital letters in the command syntax. For the command syntax:

  ```
  RFANalyzer:CONTrol:MEASurement:FREQuency:AUTO?
  ```

  the shortened form would be:

  ```
  RFAN:CONT:MEAS:FREQ:AUTO?
  ```

- The programming examples do not include optional nodes. Optional nodes in the command syntax are defined by enclosing the node inside the [ ] brackets. For example, the command syntax:

```
CALL[:CELL]:POWer[:SAMPlitude] -80dBm
```

appears in the programming examples as:

```
CALL:POW -80dBm
```

- Programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the on-line information for the definition and use of these command separators.

Complex Commands

Complex commands are used to configure the state and assign values to parameters simultaneously. Complex commands can be used to save programming steps and minimize GPIB bus transactions.

- The syntax below turns the state of the parameter on.

  OUTPUT Test_set;"SET:DTXP:TIM:STAT ON"

- The syntax below is used to assign a value to the parameter.

  OUTPUT Test_set;"SET:DTXP:TIM:TIME 10 S"

- Both of the above actions can be accomplished with one syntax command:

  OUTPUT Test_set;"SET:DTXP:TIM:STIM 10 S"

  The command above sets the parameter state to ON and the value of the parameter to 10 seconds. Note that in this example the optional command mnemonic :STIMe has been included to clarify that this complex command was used to set both the state and the value.

- This command can be shortened further by removing the optional command mnemonic :STIMe, as shown below.

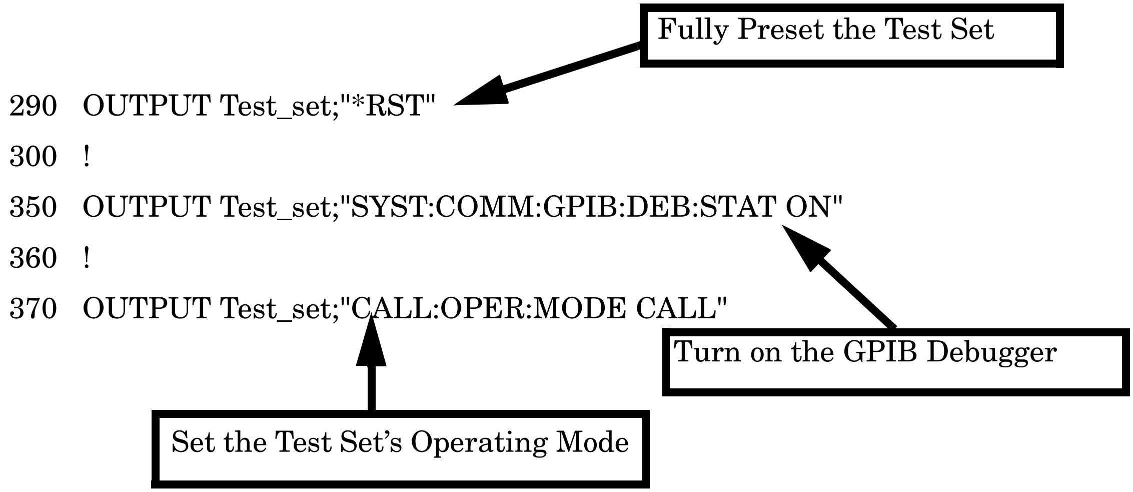
  OUTPUT Test_set;"SET:DTXP:TIM 10 S"

  This is the format that will be used throughout this guide.

Step 1: Set up the Test Set

This step explains how to:

- “Fully Preset the Test Set”
- “Turn on the GPIB Debugger”
- “Set the Test Set’s Operating Mode”



```
290  OUTPUT Test_set;"*RST"
300  !
350  OUTPUT Test_set;"SYST:COMM:GPIB:DEB:STAT ON"
360  !
370  OUTPUT Test_set;"CALL:OPER:MODE CALL"
```

Fully Preset the Test Set
Turn on the GPIB Debugger
Set the Test Set’s Operating Mode

Fully Preset the Test Set

To set up the test set, you begin by sending the *RST command. The *RST is used to perform a full preset of the test set, returning it to a known state. *RST also sets all measurements to single trigger.

Turn on the GPIB Debugger

Another useful tool that you may want to turn on at this time is the GPIB command debugger. While turned on, error messages appear on the test set’s screen when the test set receives an unknown GPIB command. The message contains information about what type of error was caused and indicates where in the syntax the error occurred. Troubleshooting, like locating and fixing typing errors for example, becomes easier using the GPIB command debugger.

NOTE The `SYST:COMM:GPIB:DEB:STAT ON` command assists you when debugging code. This command should be taken out of your code once development is completed.

Set the Test Set's Operating Mode

The `CALL:OPER:MODE` command in the diagram is used to set the test set's operating mode to active cell. This means call processing is used to maintain a link between the test set and the mobile station.

There are also three test mode operating modes available. These test modes are useful when you are operating the mobile station in test mode.

Step 2: Configure Test Set and Mobile Station Parameters

This step explains how to:

- “Configure the Broadcast Channel Parameters”
- “Configure the Traffic Channel Parameters”
- “Configure the Mobile Station Operating Parameters”

NOTE Many of the parameters configured below are being configured to their default values. In a manufacturing environment it may be desirable to explicitly configure these parameters to ensure that the required settings have not been changed by someone setting a parameter’s value through the test set’s front panel. However, greater code efficiency can be achieved by not configuring them.

Configure the Broadcast Channel Parameters

The example below illustrates how to set up the broadcast channel parameters.

You must deactivate the cell, as shown in line 530 below, before setting the network parameters in line 550. If you do not deactivate the cell, the test set generates the following error:

```
GSM operation rejected; Attempting to set <MCC|MNC|LAC|NCC|BCC> while
generating a BCH.
```

```
500   ! Set the broadcast channel parameters
510   OUTPUT Test_set;”CALL:BAND PGSM” ! Set active broadcast band.
520   ! Deactivate cell to set network parameters.
530   OUTPUT Test_set;”CALL:ACT OFF”
540   ! Set network parameters
550   OUTPUT Test_set;”CALL:MCC 1;LAC 1;MNC 1;NCC 1;BCC 5”
560   !
570   OUTPUT Test_set;”CALL:ACT ON”    ! Reactivate the cell.
580   OUTPUT Test_set;”CALL:BCH 20”    ! Set broadcast channel to 20.
590   OUTPUT Test_set;”CALL:POW -85”   ! Set cell power to -85 dBm and
600                                    ! cell power state to ON with
610                                    ! a complex command.
```

Configure the Traffic Channel Parameters

The following example illustrates setting the traffic channel ARFCN and timeslot. Refer to the User Documentation, either on the Internet or the CD-ROM, or the GPIB syntax guide for other traffic channel parameters you can configure.

```
640   OUTPUT Test_set;”CALL:TCH 45”          ! Set traffic channel to 45.
650   OUTPUT Test_set;”CALL:TCH:TSL 4”       ! Set timeslot to 4
```

Configure the Mobile Station Operating Parameters

The example below illustrates how to configure the mobile station’s parameters. In this example, you can see how to set the discontinuous transmission state and how to set the mobile station’s transmit level.

```
680   OUTPUT Test_set;”CALL:MS:DTX OFF”! Turn DTX off for all MS tests.
690   OUTPUT Test_set;”CALL:MS:TXL 5”  ! Set the MS Transmit Level
```

Step 3: Set the Measurement Parameters

This step gives an example of how you can configure measurement parameters. For more information about measurement parameters, refer to the additional details about this step available on the Internet.

```
750   ! Configure ORFS Measurement:
760   !
770   ! The lines below are examples of using complex commands to set
780   ! multi-meas state and count at the same time.
790   OUTPUT Test_set;”SET:ORFS:SWIT:COUN 5”
800   OUTPUT Test_set;”SET:ORFS:MOD:COUN 10”
810   !
820  OUTPUT Test_set;”SET:ORFS:TRIG:SOUR AUTO”! Set trig source to AUTO.
830  OUTPUT Test_set;”SET:ORFS:CONT OFF”      ! Set trig mode to single.
840   OUTPUT Test_set;”SET:ORFS:TIM 60”        ! Set timeout time.
850   ! Put switching and modulation offsets to be tested into string
860   ! variables.  Swit$ contains switching offsets.  Mod$ contains
870   ! modulation offsets.
880   Swit$=”.4MHZ,-.4MHZ,.6MHZ,-.6MHZ,1.2MHZ,-1.2MHZ,1.8MHZ,-1.8MHZ”
890   Mod$=”.4MHZ,-.4MHZ,.6MHZ,-.6MHZ,1.2MHZ,-1.2MHZ,1.8MHZ,-1.8MHZ”
900   OUTPUT Test_set;”SET:ORFS:SWIT:FREQ “&Swit$
910   OUTPUT Test_set;”SET:ORFS:MOD:FREQ “&Mod$
920   !
930   ! Configure TX Power Measurement:
940   !
950   OUTPUT Test_set;”SET:TXP:COUN 3”
960   OUTPUT Test_set;”SET:TXP:TRIG:SOUR RISE;QUAL ON”
970   OUTPUT Test_set;”SET:TXP:CONT OFF”
980   OUTPUT Test_set;”SET:TXP:TIM 20”
990   !
1000  ! Configure Phase & Frequency Error Measurement:
1010  !
1020  OUTPUT Test_set;”SET:PFER:COUN 8”
1030  OUTPUT Test_set;”SET:PFER:TRIG:SOUR PROT;QUAL ON”
1040  OUTPUT Test_set;”SET:PFER:CONT OFF”
1050  OUTPUT Test_set;”SET:PFER:TIM 30”
1060  OUTPUT Test_set;”SET:PFER:BSYN MID”
```

Step 4: Make a Connection

There are two possible ways to make a connection with the mobile station.

- “Originating a Call from the Test Set”
- “Originating a Call from the Mobile Station”

Originating a Call from the Test Set

The code below illustrates how to make a connection by originating a call from the test set.

Synchronization for a test set origination is very similar to that for a mobile station originated call. However, as a programming convenience the test set automatically arms the state change detector with a fixed timeout value of 60 seconds for test set initiated events. Therefore, there is no need for you to specify a timeout value or arm the change detector when originating a call from the test set.

```
1120  ! Set the paging IMSI
1130  OUTPUT Test_set;”CALL:PAG:IMSI ‘001012345678901’”
1140  OUTPUT Test_set;”CALL:PAG:REP OFF” ! Set the paging repeat state.
1150  !
1280  Tries=1
1290  LOOP
1300    OUTPUT Test_set;”CALL:ORIG”       ! Originate a call.
1310    OUTPUT Test_set;”CALL:CONN:STAT?” ! CALL:CONN hanging query.
1320    ENTER Test_set;Call_connected
1330  ! Program will hang here until origination process completes.  If
1340  ! successful and the call is connected the query will return a 1.
1350  ! If unsuccessful and the call is not connected, the query
1360  ! returns 0.
1370  !
1380  EXIT IF Call_connected
1390    OUTPUT Test_set;”CALL:END”
1400    IF Tries=50 THEN
1410      BEEP
1420      DISP ““
1430      PRINT “Call did not connect after”;Tries;”.”
1440      PRINT “Program terminated.”
1450      STOP
```

```
1460    END IF
1470    DISP "Call has not connected after";Tries;"attempts."
1480    Tries=Tries+1
1490  END LOOP
```

Originating a Call from the Mobile Station

The code below illustrates how to make a connection by originating a call with the mobile station. This code is not included in the control program available on-line for you to download. That example originates the connection from the test set.

Synchronization between the control program and the test set is maintained by querying the test set for the state of the connection between it and the mobile station. When a Mobile Station origination occurs, the CALL:CONN? hanging query is used. It will return a “1” when the call is connected and a “0” otherwise. A state change detector is also armed to ensure the query does not stop hanging before the state transition from “IDLE” to “CONNected” is able to begin. Finally, to prevent the query from hanging indefinitely, which could occur if the mobile station is not turned on, badly broken, or no one pushes the “send” button on the mobile, a timeout is set for this query. In this example, 15 seconds is the value assigned to the timeout. After 15 seconds, the change detector is disarmed and the query returns either a “1” or “0”. For more information about call synchronization, refer to the Internet.

```
OUTPUT Test_set;"CALL:CONN:TIM 15"    ! Set timeout time to 15 seconds
OUTPUT Test_set;"CALL:CONN:ARM"       ! Arm the change detector
OUTPUT Test_set;"CALL:CONN:STAT?"     ! Initiate call connect state query
DISP "Originate call from mobile station."
ENTER Test_set;Call_connected         ! Program will hang here until
                                      ! origination passes or fails
IF NOT Call_connected THEN            ! Check if connection successful
  OUTPUT Test_set;"CALL:END"
  PRINT "Origination failed.  Program terminated."
  STOP
END IF
```

Step 5: INITiate and FETCh Measurements

This step explains how to:

- "INITiate a set of measurements"
- "FETCh measurement results using a subroutine"

INITiate a set of measurements

The example below illustrates how to start three measurements running concurrently.

```
1560  ! Start a set of concurrent measurements:
1570  !
1580  OUTPUT Test_set;"INIT:TXP;PFER;ORFS"
```

FETCh measurement results using a subroutine

In a typical control program, measurements are repeated on various frequencies and power levels. Therefore, it is desirable to have a subroutine capable of fetching multiple measurement results. The example code below demonstrates how you might create a subroutine for fetching the measurement results. The variable `Tch` contains the ARFCN the measurement is being made on. The variable `Ms_pwr_lvl` refers to the current power level assigned to the phone. Refer to the additional details on the Internet for more information about the different measurement results that are available and how to fetch them.

```
2470  SUB Global_fetch
2480    OPTION BASE 1
2490    COM /Address/Test_set
2500    OUTPUT Test_set;"CALL:TCH?;MS:TXL?"
2510    ENTER Test_set;Tch,Ms_pwr_lvl
2520  !
2530  !  Determine if a measurement is done:
2540  !
2550    LOOP
2560      OUTPUT Test_set;"INIT:DONE?"
2570      ENTER Test_set;Meas_done$
2580  !
```

```
2590  ! Obtain measurement results: Each measurement illustrates a
2600  ! different way of reading in results. There is no one right way.
2610  ! The method used is application dependent. Note that the examples
2620  ! do not show all possible ways.
2630  !
2640      SELECT Meas_done$
2650  !
2660      CASE “TXP”  ! TX Power measurement done.
2670        ALLOCATE Txpower(4)
2680        OUTPUT Test_set;”FETC:TXP:INT?;POW:ALL?”
2690        ENTER Test_set;Integrity,Txpower(*)
2700        IF (Integrity=0) THEN  ! Always check integrity value.
2710          PRINT “TX Power results: TCH=”;Tch;”and TXL=”;Ms_pwr_lvl
2720          PRINT USING “5X,””Minimum:””,M2D.2D,”” dBm”””;Txpower(1)
2730          PRINT USING “5X,””Maximum:””,M2D.2D,”” dBm”””;Txpower(2)
2740          PRINT USING “5X,””Average:””,M2D.2D,”” dBm”””;Txpower(3)
2750          PRINT USING “5X,””Std Dev:””,M2D.2D,”” dB”””;Txpower(4)
2760          DEALLOCATE Txpower(*)
2770        ELSE
2780          GOSUB Bad_measurement
2790        END IF
2800  !
2810      CASE “PFER”  ! Phase & Frequency Error measurement done.
2820        OUTPUT Test_set;”FETC:PFER:ALL?”
2830        ENTER Test_set;Integrity,Rms_ph_er,Peak_ph_er,Worst_frq_er
2840        IF (Integrity=0) THEN
2850          PRINT “PFERror results: TCH=”;Tch;”and TXL=”;Ms_pwr_lvl
2860          PRINT “RMS Phase Error: “;Rms_ph_er;” deg”
2870          PRINT “Peak Phase Error: “;Peak_ph_er;” deg”
2880          PRINT “Worst Freq Error: “;Worst_frq_er;” Hz”
2890        ELSE
2900          GOSUB Bad_measurement
2910        END IF
2920  !
2930      CASE “ORFS”  ! ORFS measurement done.
2940  !
2950  ! This code illustrates a more ‘generic’ approach to reading
2960  ! measurement results. By using the capabilities designed into
2970  ! high-level measurements, routines that access measurement
2980  ! results do not have to explicitly know what the measurement
2990  ! execution conditions were.  That information can be determined
3000  ! at the time the measurement results are queried.
```

```
3010  !
3020        OUTPUT Test_set;”FETC:ORFS:INT?”           ! Check integrity.
3030        ENTER Test_set;Integrity
3040        IF (Integrity=0) THEN
3050          ! Get the number of offsets tested.
3060          OUTPUT Test_set;”SET:ORFS:SWIT:FREQ:POIN?”
3070          ENTER Test_set;Points
3080          IF Points THEN  ! Only query if one or more offsets tested.
3090            ALLOCATE Swit_res(Points),Swit_offs(Points)
3100            ! Get measurement offsets.
3110            OUTPUT Test_set;”SET:ORFS:SWIT:FREQ?”
3120            ENTER Test_set;Swit_offs(*)
3130            ! Get results
3140            OUTPUT Test_set;”FETC:ORFS:POW?;:FETC:ORFS:SWIT?”
3150            ENTER Test_set;Tx_power,Swit_res(*)
3160           PRINT “ORFS Swit Results: TCH=”;Tch;”and TXL=”;Ms_pwr_lvl
3170           PRINT USING “19X,””TX Power =””,M2D.2D,”” dBm”””;Tx_power
3180            PRINT “     Offset(kHz)          Level(dBm)”
3190            PRINT “     -----------          ----------”
3200 Orfs_image:  IMAGE 6X,M4D.2D,12X,M4D.2D
3210            FOR J=1 TO Points
3220             PRINT USING Orfs_image;(Swit_offs(J)/1.E+3),Swit_res(J)
3230            NEXT J
3240            DEALLOCATE Swit_res(*),Swit_offs(*)
3250          END IF
3260          ! Get the number of offsets tested.
3270          OUTPUT Test_set;”SET:ORFS:MOD:FREQ:POIN?”
3280          ENTER Test_set;Points
3290          IF Points THEN ! Only query if one or more offsets tested.
3300            ALLOCATE Mod_res(Points),Mod_offs(Points)
3310            ! Get measurement offsets
3320            OUTPUT Test_set;”SET:ORFS:MOD:FREQ?”
3330            ENTER Test_set;Mod_offs(*)
3340            ! Get results
3350            OUTPUT Test_set;”FETC:ORFS:POW?;:FETC:ORFS:MOD?”
3360            ENTER Test_set;Tx_power,Pwr_30khz,Mod_res(*)
3370            PRINT “ORFS Mod Results: TCH=”;Tch;”and TXL=”;Ms_pwr_lvl
3380            PRINT “30 KHz BW Power =”;Pwr_30khz;” dBm”
3390            PRINT “     Offset(kHz)          Level(dB)”
3400            PRINT “     -----------          ---------”
3410            FOR J=1 TO Points
3420              PRINT USING Orfs_image;(Mod_offs(J)/1.E+3),Mod_res(J)
```

```
3430            NEXT J
3440            DEALLOCATE Mod_res(*),Mod_offs(*)
3450          END IF
3460        ELSE
3470          GOSUB Bad_measurement
3480        END IF
3490      END SELECT
3500    EXIT IF Meas_done$=”NONE”
3510    END LOOP  ! If ‘WAIT’ is returned from ‘INIT:DONE?’ query, it
3520              ! just falls through the loop.
3530    SUBEXIT
3540 Bad_measurement: !
3550  PRINT “Measurement error: “&Meas_done$
3560  PRINT “Measurement Integrity value =”;Integrity
3570  RETURN
3580  !
3590  SUBEND
```

Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

There are several ways you may want to reconfigure the connection parameters. Some examples are:

- “Reconfigure the Mobile Station Parameters”
- “Reconfigure the Connection to a New ARFCN”
- “Reconfigure the Connection to a New ARFCN in a Different Band”

Reconfigure the Mobile Station Parameters

The example below illustrates how to change the mobile station’s transmit level. The `:SEQuential` synchronization command appended to the end of the command ensures that the command has executed before the test set accepts any other commands. This is important because the mobile station needs to have received the command to be on the new power level before transmitter measurements can be made accurately.

```
1650  ! Assign a new power level to the Mobile Station
1660  OUTPUT Test_set;”CALL:MS:TXL:SEQ 10”
```

Reconfigure the Connection to a New ARFCN

The example below illustrates how to reconfigure the connection to a new ARFCN. You may also want to change the mobile station transmit level at this time as well. The recommended process for reconfiguring the mobile station transmit level and the ARFCN at the same is shown in the example below.

The example below also shows you how to use the `CALL:STAT:STAT?` query to determine if the connection was maintained. The `CALL:STAT:STAT?` query returns the current state of the connection. In this case, “CONN” is returned if the connection is still in the connected state, indicating the handover succeeded. This query can be used in this instance because the `:SEQuential` command forced the `CALL:TCH` command to operate sequentially.

```
1810  OUTPUT Test_set;”CALL:MS:TXL 5”
1820  OUTPUT Test_set;”CALL:TCH:SEQ 120” ! Use :SEQ to force sequential
```

```
1830                                    ! execution of the TCH command.
1840  OUTPUT Test_set;"CALL:STAT:STAT?"  ! Verify that the call is still
1850  ENTER Test_set;Call_status$        ! in the connected state after
1860                                    ! handover.
1870  IF Call_status$<>"CONN" THEN
1880    PRINT "Call handover failed. New channel assignment =";Tch
1890    PRINT "Program terminated."
1900    STOP
1910  END IF
```

Reconfigure the Connection to a New ARFCN in a Different Band

The recommended process for reconfiguring the connection to a new band is illustrated in the example below. DCS is added to the commands for reconfiguring the mobile station transmit level parameter and the TCH ARFCN. This results in the new parameter values being stored until the DCS band is made active by the `CALL:TCH:BAND` command. If they are not specified as DCS band parameters, they become active immediately

Note that reconfiguring the connection to a new band uses the same synchronization method as changing to a new TCH ARFCN.

```
2060  OUTPUT Test_set;"CALL:MS:TXL:DCS 5"
2070  OUTPUT Test_set;"CALL:TCH:DCS 600"
2080  OUTPUT Test_set;"CALL:TCH:BAND DCS"
2090  !
2100  OUTPUT Test_set;"CALL:STAT:STAT?" ! Verify that the call is still
2110  ENTER Test_set;Call_status$       ! in the connected state after
2120                                   ! handover.
2130  IF Call_status$<>"CONN" THEN
2140    PRINT "Call handover failed. New channel assignment =";Tch
2150    PRINT "Program terminated."
2160    STOP
2170  END IF
```

Step 7: End the Connection

You can end the connection in one of two ways:

- "Ending the Connection from the Test Set"
- "Ending the Connection from the Mobile Station"

Ending the Connection from the Test Set

When you are ending the connection from the test set use the CALL:END command. The example below illustrates how you use the CALL:CONN:STAT? query for call synchronization. This query returns a "0" if the call ended successfully and a "1" if the call is not ended. It is not necessary for you to arm the change detector or set a change detector timeout when using the test set to terminate a call. The test set automatically arms the change detector and uses a default timeout in this situation.

```
2320  OUTPUT Test_set;"CALL:END"
2330  OUTPUT Test_set;"CALL:CONN:STAT?"
2340  ENTER Test_set;Call_connected
2350  IF Call_connected THEN
2360    BEEP
2370    PRINT "Unable to complete BS termination. Program terminated."
2380    STOP
2390  END IF
```

Ending the Connection from the Mobile Station

When the connection is being ended from the mobile station, it is important to set a timeout value and arm the change detector. More information about using these commands to achieve call synchronization is available in the additional details about this step found on the Internet.

This code is not included in the control program available on-line for you to download. That example ends the connection from the test set.

```
OUTPUT Test_set;"CALL:CONN:TIM 5" !Set timeout time to 5 seconds.
OUTPUT Test_set;"CALL:CONN:ARM"   !Arm the change detector.
OUTPUT Test_set;"CALL:CONN:STAT?" !Initiate call connect state query.
DISP "Terminate the call from the mobile station."
ENTER Test_set;Call_connected     !Program will hang here until state
```

```
                                  !change or timer expires.
IF Call_connected THEN            !Check if disconnect successful.
  OUTPUT Test_set;”CALL:END”
  PRINT “Call failed to end correctly.  Program terminated.”
  STOP
END IF
```

Programming: Getting Started Guide

for

E1985A GSM_AMPS/136_GPRS Mobile Test Application (fast switching)
Revision A.02

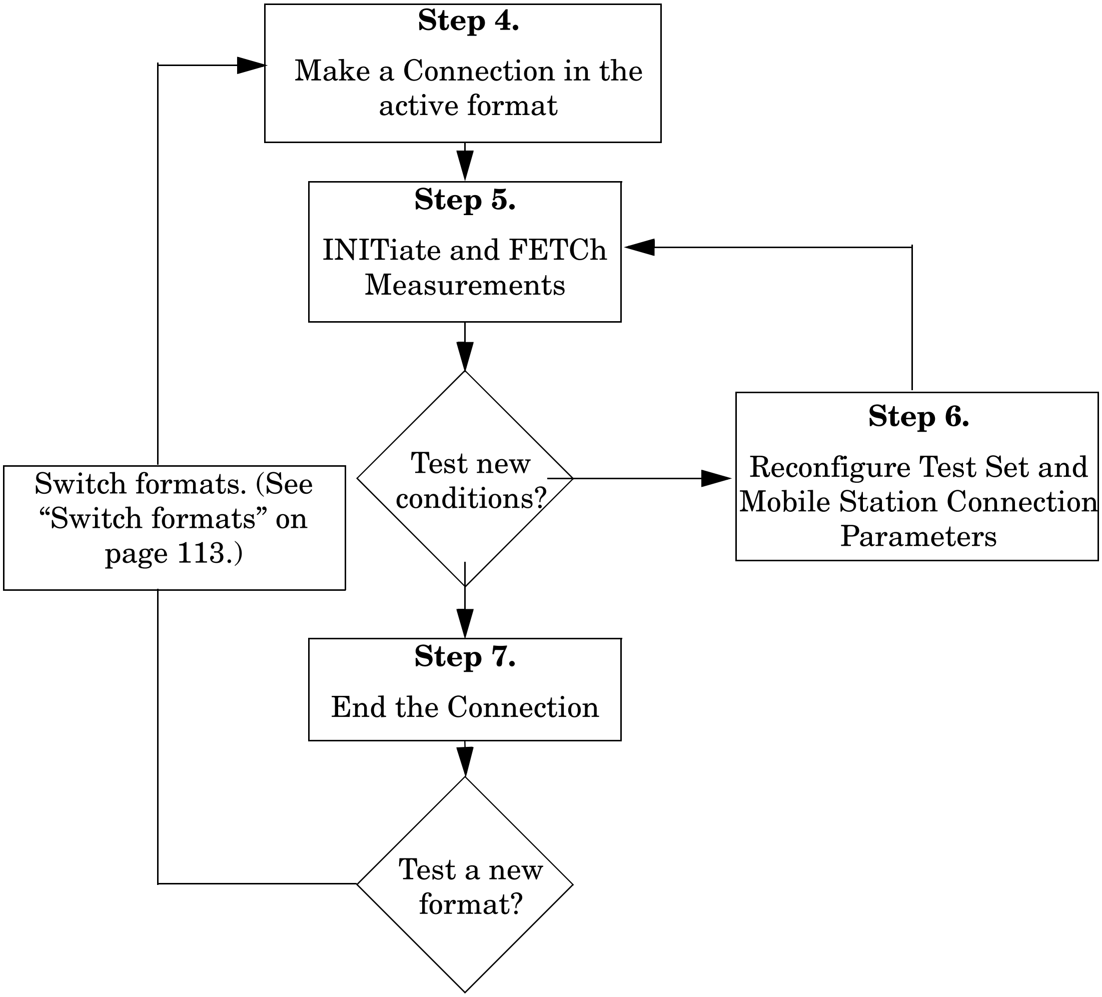
Introduction

The Getting Started Guide is organized around the basic set of tasks a control program normally performs when testing a mobile station in a manufacturing environment.

How to use the programming flowchart for switching between formats

It is important for you to understand how to utilize the programming flowchart when designing a control program to switch between multiple formats. In steps 1,2, and 3 of the programming flowchart, you set up the test set and configure parameters for each applicable format all at once. In contrast, you implement steps 4-7 for only one format at a time. Figure 1. on page 93 illustrates in greater detail how to perform these steps for each format. Example code for multiple formats is contained in steps 4-7 of this guide.

Figure 1. Flowchart for the fast switching test application


Step 4.
Make a Connection in the active format
Step 5.
INITiate and FETCh Measurements
Test new conditions?
Step 6.
Reconfigure Test Set and Mobile Station Connection Parameters
Switch formats. (See "Switch formats" on page 113.)
Step 7.
End the Connection
Test a new format?

Conventions used in this Getting Started Guide

Throughout this Getting Started Guide the term “test set” refers to an Agilent Technologies 8960 Series 10 wireless communications test set with the E1985A GSM_AMPS/136_GPRS mobile test application installed.

The variable `Testset` used in the steps of the Getting Started Guide refers to the test set’s GPIB address.

How to use this Getting Started Guide

The most effective way to use this Getting Started Guide is with the Programming Flowchart included inside this guide and the 8960 User Documentation. This documentation is found in two locations:

- the 8960 Series 10 Support Site (updated frequently) on the Internet, or
- the User Documentation CD-ROM shipped with your test application.

Each step on the Programming Flowchart is illustrated with example program code in this guide. Using this Getting Started Guide, the Programming Flowchart, and the on-line information about 8960 programming, you will be able to generate a control program to perform fundamental mobile station manufacturing tests.

Useful on-line links

Go to the 8960 Series 10 Support Site on either the CD-ROM or the Internet. The CD-ROM should self-launch to its home page. The URL for the website is:

http://www.agilent.com/find/8960support/

Navigate to the Programming page for this test application. The illustrations on the back of the Programming Flowchart show you how to navigate to this page.

The following links are under the heading Getting Started on the Programming page:

- Programming: Getting Started Guide
  - — This on-line version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.
- Control Program Examples
  - — These examples are for you to download. You may want to use these as templates for your own control program or to execute.
  - — The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

About the Programming Examples Presented in this Guide

Programming Language:

Programming examples presented in this guide are written in the HP BASIC programming language, also known as RMB or Rocky Mountain BASIC. The use of HP BASIC is not an endorsement of the HP BASIC product.

Line Numbers

All of the programming examples in the guide with line numbers are sections from a control program example available on-line for you to download.

Code that is not part of the download control program example does not have line numbers. This code may represent an alternate method of performing the task or may illustrate a feature not used by the control program example.

Syntax used in Programming Examples:

- The programming examples use the shortened form of the command syntax to minimize GPIB bus transactions. The shortened form of a command is defined by use of capital letters in the command syntax. For the command syntax:

  ```
  RFANalyzer:CONTrol:MEASurement:FREQuency:AUTO?
  ```

  the shortened form would be:

  ```
  RFAN:CONT:MEAS:FREQ:AUTO?
  ```

- The programming examples do not include optional nodes. Optional nodes in the command syntax are defined by enclosing the node inside the [ ] brackets. For example, the command syntax:

  ```
  CALL[:CELL]:POWer[:SAMPlitude] -80dBm
  ```

  appears in the programming examples as:

  ```
  CALL:POW -80dBm
  ```

- Programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the on-line information for the definition and use of these command separators.

Complex Commands

Complex commands are used to configure the state and assign values to parameters simultaneously. Complex commands can be used to save programming steps and minimize GPIB bus transactions.

- The syntax below turns the state of the parameter on.

  ```
  OUTPUT Test_set;"SET:DTXP:TIM:STAT ON"
  ```

- The syntax below is used to assign a value to the parameter.

  ```
  OUTPUT Test_set;"SET:DTXP:TIM:TIME 10 S"
  ```

- Both of the above actions can be accomplished with one syntax command:

  ```
  OUTPUT Test_set;"SET:DTXP:TIM:STIM 10 S"
  ```

  The command above sets the parameter state to ON and the value of the parameter to 10 seconds. Note that in this example the optional command mnemonic :STIMe has been included to clarify that this complex command was used to set both the state and the value.

- This command can be shortened further by removing the optional command mnemonic :STIMe, as shown below.

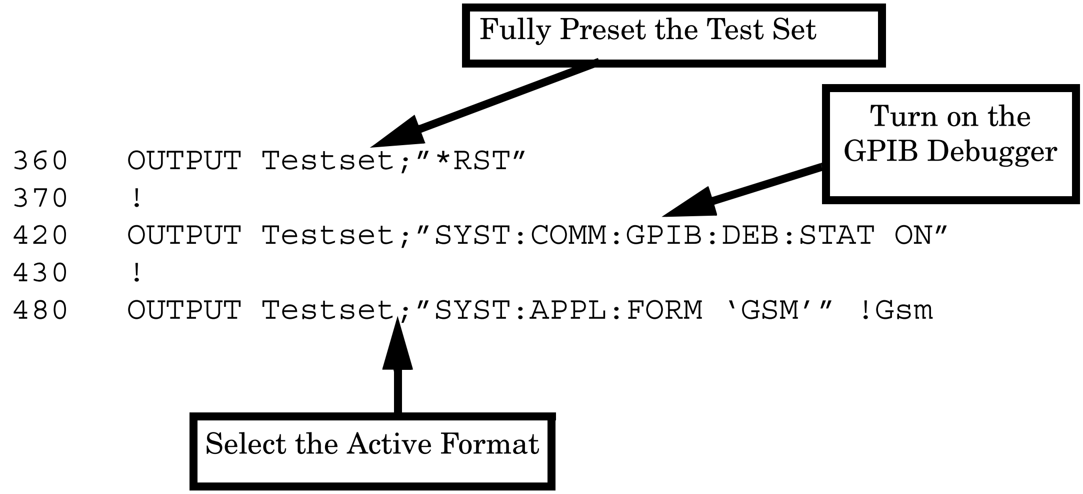
  ```
  OUTPUT Test_set;"SET:DTXP:TIM 10 S"
  ```

  This is the format that will be used throughout this guide.

Step 1: Set up the Test Set

This step explains how to:

- “Fully Preset the Test Set”
- “Turn on the GPIB Debugger”
- “Select the Active Format”



```
360   OUTPUT Testset;”*RST”
370   !
420   OUTPUT Testset;”SYST:COMM:GPIB:DEB:STAT ON”
430   !
480   OUTPUT Testset;”SYST:APPL:FORM ‘GSM’” !Gsm
```

Fully Preset the Test Set

To set up the test set, you begin by sending the `*RST` command. `*RST` is used to perform a full preset of the test set, returning it to a known state. `*RST` also sets all measurements to single trigger.

NOTE All measurement triggers are set to single in all the test formats supported by the test application when the `*RST` command is sent.

Turn on the GPIB Debugger

A useful tool that you may want to turn on at this time is the GPIB command debugger. While turned on, error messages containing debugging information appear on the test set’s screen when the test set receives an unknown GPIB command. The information indicates what type of error was caused and where in the syntax the error occurred. Troubleshooting, like locating and fixing typing errors for example, becomes easier using the GPIB command debugger.

NOTE *This command should be taken out of your code once development is completed.* It does increase your test time. The `SYST:COMM:GPIB:DEB:STAT ON` command only assists you when debugging code.

Select the Active Format

The `SYST:APPL:FORM` command used in the diagram switches the test set to the correct format. This is a very fast switch because it occurs within the test application. Switches to other test applications use a different command and take much longer. Refer to user documentation on your CD-ROM or the Internet for more information about fast switching.

Step 2: Configure Test Set and Mobile Station Parameters

NOTE Many of the parameters configured in this step are being set to their default values. In a manufacturing environment it may be desirable to explicitly configure these parameters to ensure that the required settings have not been changed by someone setting a parameter’s value through the test set’s front panel. However, greater code efficiency can be achieved by not configuring them.

Configure test set and mobile station parameters

It is possible for you to configure the test set and mobile station parameters for all test formats supported by the test application at the beginning of your program. The following examples illustrate how to set up parameters for all available formats after having selected the GSM format as the active format in “Step 1: Set up the Test Set” on page 97.

- “Set up parameters for GSM”
- “Set up parameters for AMPS/136”
- “Set up parameters for GPRS”

Set up parameters for GSM

Notice the use of the `:GSM` identifier appended to the `CALL:POW:AMPL` command. Identifiers are used with commands or queries that are shared by test formats. They enable you to send commands to an inactive test format. Therefore, it is not necessary to use the `:GSM` format identifier below because GSM is the active test format. It is included here to demonstrate the use of format identifiers.

```
670   OUTPUT Testset;”CALL:POW:AMPL:GSM “;Gsm_level
680   OUTPUT Testset;”CALL:BAND EGSM”
690   OUTPUT Testset;”CALL:TCH 5;BCH 32;:CALL:MS:TXL 0;TADV 0”
```

Set up parameters for AMPS/136

Notice the use of the format identifier appended to the `CALL:POW:AMPL` command.

The format identifier for the AMPS/136 Mobile Test format is :TA136. The format identifier for the digital system in AMPS/136 format is :DIG136. None of the other commands sent below require this format identifier because they are unique to the AMPS/136 test format.

```
720   OUTPUT Testset;"CALL:OPER:MODE CALL"
730   OUTPUT Testset;"CALL:POW:AMPL:TA136 ";Tdma_level
740   OUTPUT Testset;"CALL:DCCH 1013"
750   OUTPUT Testset;"CALL:SET:AVC 387"
760   OUTPUT Testset;"CALL:SET:DTC:BAND CELL"
770   OUTPUT Testset;"CALL:SET:DTC:CHAN:CELL 542"
780   OUTPUT Testset;"CALL:SET:DTC:CHAN:PCS 1000"
790   OUTPUT Testset;"CALL:TCH:TYPE DTC"
800   OUTPUT Testset;"CALL:SET:MS:ANAL:TXL 2"
810   OUTPUT Testset;"CALL:SET:MS:DIG:TXL:CELL 2"
820   OUTPUT Testset;"CALL:SET:MS:DIG:TXL:PCS 2"
830   ! Set up the AVC for the SINAD measurement.
840   ! Turn off the internal FM until connection is on an AVC.
850   OUTPUT Testset;"CALL:FM:INT:STAT OFF;DEV 8KHZ;FREQ 1004HZ"
```

Set up parameters for GPRS

You can see the use of the format identifier for the GPRS Mobile Test format, :GPRS, in the following example code.

```
!  Configure GPRS parameters
OUTPUT Test_set;"CALL:BAND:GPRS PGSM"! Set broadcast band
OUTPUT Test_set;"CALL:POW:GPRS -60"! Set cell power to -80 dBm.
OUTPUT Test_set;"CALL:PDTCH 45"! Set packet data traffic channel
OUTPUT Test_set;"CALL:FUNC:DATA:TYPE BLER"! Data connection type
!
! Set Multi-slot Configuration to two downlinks and one uplink.
OUTPUT Test_set;"CALL:PDTCH:MSL:CONF D2U1"
OUTPUT Test_set;"CALL:PDTCH:CSCH CS4"! Set Coding Scheme to CS4
! Assign values to the power reduction levels
OUTPUT Test_set;"CALL:PDTCH:PZER:LEV 25" ! Assign a value to P0
OUTPUT Test_set;"CALL:PDTCH:PRED:LEV1 5"! Set PRL1 to 5 dB
OUTPUT Test_set;"CALL:PDTCH:PRED:LEV2 11" ! Set PRL2 to 11 dB
! Assign power levels to the downlink bursts
OUTPUT Test_set;"CALL:PDTCH:PRED:BURS1 PRL1"
OUTPUT Test_set;"CALL:PDTCH:PRED:BURS2 PRL2"
!
! Assign a power level to the uplink burst
OUTPUT Test_set;"CALL:PDTCH:MS:TXL:BURS 5"
```

Step 3: Set Measurement Parameters

- “Set trigger to single for all measurements”
- “Set measurement parameters”

Set trigger to single for all measurements

You can send either the *RST command, as discussed in “Fully Preset the Test Set” on page 97, or the following command to set the trigger to single for all measurements in all the test formats supported by the test application. In the example below, the command is commented out (by preceding it with an !) because in the control program example utilized throughout this guide, the *RST command is used, making the SET:CONT:OFF command unnecessary. If you choose to not send the *RST command, make sure to uncomment this line.

```
950   ! OUTPUT Testset;”SET:CONT:OFF”
```

Set measurement parameters

The most efficient use of a test application supporting multiple test formats is to complete measurement setups at times when the test system is doing something else. This example illustrates setting up measurement parameters for the test formats at the beginning of the program, presumably while a phone is being prepared for RF test.

For more information about the measurement parameters, refer to the additional details about this step available on your CD-ROM or the Internet.

- “Set GSM measurement parameters”
- “Set AMPS/136 measurement parameters”
- “Set GPRS measurement parameters”

Set GSM measurement parameters

For more details about specific GSM measurement parameters, refer to the 8960 User Documentation for the GSM Mobile Test Application available on your CD-ROM or the Internet.

```
980   OUTPUT Testset;”SET:PFER:TIM 5;COUN 1”
990   OUTPUT Testset;”SET:PVT:TIM 5;COUN 10”
1000  OUTPUT Testset;”SET:PVT:TIME -28us,-10us,321.2us,552.8us,570.8us”
1010  OUTPUT Testset;”SET:ORFS:TIM 7;COUN:STAT OFF”
1020  !
1030  ! This example puts the switching and modulation offsets to be
1040  ! tested into string variables.  Mod$ contains the modulation
1050  ! offsets while Swit$ contains the switching offsets.
1060  Mod$=”200KHZ,-200KHZ,400KHZ,-400KHZ”
1070  Swit$=”400KHZ,-400KHZ,600KHZ,-600KHZ”
1080  OUTPUT Testset;”SET:ORFS:MOD:COUN 10;FREQ “&Mod$
1090  OUTPUT Testset;”SET:ORFS:SWIT:COUN 20;FREQ “&Swit$
1100  OUTPUT Testset;”SET:FBER:COUN 10000”
```

Set AMPS/136 measurement parameters

For more details about specific AMPS/136 measurement parameters, refer to the 8960 User Documentation for the AMPS/136 Mobile Test Application available on your CD-ROM or the Internet.

```
1140  ! Digital Measurement Parameters
1150  OUTPUT Testset;”SET:DTXP:TIM 3;COUN 10”
1160  OUTPUT Testset;”SET:MACC:TIM 3;COUN 1”
1170  OUTPUT Testset;”SET:MACC:EVM10:STAT ON”
1180  OUTPUT Testset;”SET:ACP:TIM 3;COUN 1”
1190  !
1200  ! Analog Measurement Parameters
1210  OUTPUT Testset;”SET:FST:TIM 3;COUN 1”
1220  OUTPUT Testset;”SET:ATXP:TIM 3;COUN 10”
1230  OUTPUT Testset;”SET:FM:TIM 3;COUN 1”
1240  OUTPUT Testset;”SET:FM:FREQ:STAT ON”
1250  OUTPUT Testset;”SET:FM:DIST:STAT ON;FREQ 6000”
1260  OUTPUT Testset;”SET:FM:DET PPE”
1270  OUTPUT Testset;”SET:FM:FILT:TBP 6000”
1280  OUTPUT Testset;”SET:FM:FILT TBP”
1290  !
1300  ! Audio Measurement Parameters
1310  OUTPUT Testset;”SET:AFAN:FREQ:STAT ON”
1320  OUTPUT Testset;”SET:AFAN:FILT CMES;TIM 3;COUN 5;PEAK:VOLT 1”
1330  OUTPUT Testset;”SET:AFAN:SDIS:STAT ON;FREQ 1004”
```

Set GPRS measurement parameters

For more details about specific GPRS measurement parameters, refer to the 8960 User Documentation for the GPRS Mobile Test Application available on your CD-ROM or the Internet.

```
! Configure ORFS Measurement
!
OUTPUT Test_set;”SET:ORFS:TIM:GPRS 20”! Set timeout time.
Swit$=”.4MHZ,-.4MHZ,.6MHZ,-.6MHZ,1.2MHZ,-1.2MHZ,1.8MHZ,-1.8MHZ”
OUTPUT Test_set;”SET:ORFS:SWIT:COUN:GPRS 20”
OUTPUT Test_set;”SET:ORFS:SWIT:FREQ:GPRS “&Swit$
!
! Configure Power vs. Time Measurement for GSM and GPRS:
!
Pvt$=”-28us,-10us,321.2us,552.8us,570.8us”
OUTPUT Test_set;”SET:PVT:TIM:GPRS 10”
OUTPUT Test_set;”SET:PVT:COUN:GPRS 5”
OUTPUT Test_set;”SET:PVT:TIME:GPRS “&Pvt$
!
! Configure Phase & Frequency Error Measurement for GSM and GPRS:
!
OUTPUT Test_set;”SET:PFER:COUN 10;TIM 5”
OUTPUT Test_set;”SET:PFER:TIM:GPRS 5”
OUTPUT Test_set;”SET:PFER:COUN:GPRS 10”
OUTPUT Test_set;”SET:PFER:SYNC MID;SYNC:GPRS MID”
```

Step 4: Make a Connection

NOTE For more information about various ways to make a connection with the mobile station, refer to the Programming: Getting Started Guide for the specific mobile test technologies.

In this control program example, a GSM call is established first. After you switch formats at the end of “Step 7: End the Connection” on page 113, it will again be necessary to establish a call. (See the fast switching test application flowchart, Figure 1. on page 93 to see the process for using multiple formats.) Following are code examples for making a connection in different formats.

- “Subroutine for making GSM and AMPS/136 connections”
- “Make a GSM connection”
- “Make an AMPS/136 connection”

Subroutine for making GSM and AMPS/136 connections

A subroutine for making connections is used because this same code is used more than once. This provides greater code efficiency.

The code for the subroutine is below:

```
3870  SUB Orig_call(Format$)
3880    COM INTEGER Testset
3890    PRINT “Turn the “;Format$;” phone on now.”
3900    PRINT “When the phone camps, press ‘F2’ to continue.”
3910    PAUSE
3920    CLEAR SCREEN
3930    PRINT “Originate a call on the “;Format$;” Mobile now.”
3940    OUTPUT Testset;”CALL:CONN:TIM 10”
3950    OUTPUT Testset;”CALL:CONN:ARM”!Arm the Call-State-Change
3960                                  ! Detector
3970    OUTPUT Testset;”CALL:CONN?”!Query State
3980    ENTER Testset;Callstate
3990    IF NOT Callstate THEN
4000      Orig_failed
4010    END IF
```

```
4020  SUBEND
```

Make a GSM connection

The subroutine to establish the GSM connection is called using the code below.

```
1390  Orig_call("GSM") ! Subroutine for originating a call from the MS
```

Make an AMPS/136 connection

After performing all GSM measurements, ending the GSM connection, and switching the test set to the AMPS/136 format, you can establish an AMPS/136 connection by calling the subroutine again. See Figure 1. on page 93 illustrates this process for making use of the multiformat capability.

```
2020  Orig_call("AMPS/136")
```

Step 5: INITiate and FETCh Measurements

This step explains how to:

- “INITiate a set of measurements”
- “FETCh measurement results using a subroutine”
- “Measurement Integrity Indicators”

INITiate a set of measurements

The current active format is GSM. Therefore, GSM measurements are INITiated and FETChed first. Figure 1. on page 93 diagrams the process for using multiple formats.

- “INITiating GSM measurements”
- “INITiating AMPS/136 measurements”

INITiating GSM measurements

The example below illustrates how to start four GSM measurements running concurrently. The FBER (Fast Bit Error Rate) measurement is a receiver measurement. This makes use of the test set’s ability to make transmitter and receiver measurements at the same time.

```
1510    OUTPUT Testset;”INIT:FBER;PFER;PVT;ORFS”
```

INITiating AMPS/136 measurements

The sections below contain examples illustrating how to start AMPS/136 digital and analog measurements.

Digital

```
2110    OUTPUT Testset;”CALL:MS:REP:MAHO ON”
2120    OUTPUT Testset;”CALL:POW:TA136 “;Maho_level
2130    OUTPUT Testset;”INIT:DTXP;MACC;ACP”
```

Analog

```
2410  OUTPUT Testset;”CALL:FM:INT:STATE ON”
```

```
2420  OUTPUT Testset;”CALL:POW “;Sinad_level
2430  OUTPUT Testset;”INIT:ATXP;FST;AFAN;FM”
```

FETCh measurement results using a subroutine

In a typical control program, measurements are repeated on various frequencies and power levels. Therefore, it is desirable to have a subroutine capable of fetching multiple measurement results. The example code below demonstrates how you might create a subroutine for fetching the measurement results. Refer to the additional details available on your CD-ROM or the Internet for more information about the different measurement results that are available and how to fetch them.

```
4100  SUB Fetch_results
4110    COM INTEGER Testset
4120    REPEAT
4130      OUTPUT Testset;”INIT:DONE?”
4140      ENTER Testset;Measdone$
4150      SELECT Measdone$
4160      CASE “DTXP”
4170        OUTPUT Testset;”FETC:DTXP?”
4180        ENTER Testset;Integrity,Power
4190        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4200        Print_results(Measdone$,Power)
4210      CASE “MACC”
4220        OUTPUT Testset;”FETC:MACC?”
4230        ENTER Testset;Integrity,Evm,Ferr,Ooff,Perr,Mag,Evm10
4240        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4250        Print_results(Measdone$,Evm,Ferr,Ooff,Perr,Mag,Evm10)
4260      CASE “ACP”
4270        OUTPUT Testset;”FETC:ACP?”
4280        ENTER Testset;Integrity,Adjl,Adjh,Alt1l,Alt1h,Alt2l,Alt2h
4290        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4300        Print_results(Measdone$,Adjl,Adjh,Alt1l,Alt1h,Alt2l,Alt2h)
4310      CASE “ATXP”
4320        OUTPUT Testset;”FETC:ATXP?”
4330        ENTER Testset;Integrity,Power
4340        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4350        Print_results(Measdone$,Power)
4360      CASE “FST”
4370        OUTPUT Testset;”FETC:FST?”
4380        ENTER Testset;Integrity,Ferr,Freq
4390        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
```

```
4400        Print_results(Measdone$,Ferr,Freq)
4410      CASE “AFAN”
4420        OUTPUT Testset;”FETC:AFAN?”
4430        ENTER Testset;Integrity,Level,Sinad,Dist
4440        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4450        OUTPUT Testset;”FETC:AFAN:FREQ?”
4460        ENTER Testset;Freq
4470        Print_results(Measdone$,Level,Dist,Sinad,Freq)
4480      CASE “FM”
4490        OUTPUT Testset;”FETC:FM?”
4500        ENTER Testset;Integrity,Dev,Dist,Freq
4510        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4520        Print_results(Measdone$,Dev,Dist,Freq)
4530      CASE “PVT”
4540        OUTPUT Testset;”FETC:PVT?”
4550        ENTER Testset;Integrity,Mask,Power,Pvt1,Pvt2,Pvt3,Pvt4,Pvt5
4560        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4570        Print_results(Measdone$,Mask,Power,Pvt1,Pvt2,Pvt3,Pvt4,Pvt5)
4580      CASE “PFER”
4590        OUTPUT Testset;”FETC:PFER?”
4600        ENTER Testset;Integrity,Rmsperr,Peakperr,Ferr
4610        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4620        Print_results(Measdone$,Rmsperr,Peakperr,Ferr)
4630      CASE “ORFS”
4640        OUTPUT Testset;”FETC:ORFS:INT?”
4650        ENTER Testset;Integrity
4660        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4670        OUTPUT Testset;”FETC:ORFS:SWIT?”
4680        ENTER Testset;S1,S2,S3,S4
4690        OUTPUT Testset;”FETC:ORFS:MOD?”
4700        ENTER Testset;Pow_30khz,M1,M2,M3,M4
4710        Print_results(Measdone$,S1,S2,S3,S4,M1,M2,M3,M4,Pow_30khz)
4720      CASE “FBER”
4730        OUTPUT Testset;”FETC:FBER:INT?;RAT?”
4740        ENTER Testset;Integrity,Fber
4750        IF Integrity<>0 THEN CALL Bad_integrity(Integrity,Measdone$)
4760        Print_results(Measdone$,Fber)
4770      END SELECT
4780    UNTIL Measdone$=”NONE”
4790  SUBEND
```

Measurement Integrity Indicators

When the test set performs a measurement, it always returns a measurement result even when the measurement was made under adverse conditions. Therefore, you should check the measurement integrity indicator value to ensure it is zero. If the value is something other than zero, the returned measurement result is not valid. Refer to the user documentation on your CD-ROM or the Internet for more information about integrity indicators.

```
3380  SUB Bad_integrity(Integrity,Meas_name$)
3390    IF Integrity<>0 THEN
3400       PRINT “Warning: “;Meas_name$;” Integrity =”;Integrity
3410    END IF
3420  SUBEND
```

Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

NOTE There are several ways you may want to reconfigure the connection parameters. For more information refer to the Programming: Getting Started Guide for the specific mobile test technologies.

The first active format in the control program example is GSM. Figure 1. on page 93 diagrams the process for using multiple formats.

Below are examples of reconfiguring parameters from the control program example for:

- “GSM”
- “AMPS/136”

GSM

Here, the GSM call is reconfigured to a new channel, a new mobile station transmit level, and a new band. The state of the call is checked after the reconfiguration to ensure the call is maintained.

```
1640      OUTPUT Testset;”CALL:TCH 25;:CALL:MS:TXL 0”
1650      OUTPUT Testset;”CALL:TCH:BAND DCS”
1660      OUTPUT Testset;”CALL:STAT?”
1670      ENTER Testset;Callstat$
1680      IF Callstat$<>”CONN” THEN Dropped_call
```

AMPS/136

The code below illustrates using a loop to reconfigure the connection first to a different band and then to an analog voice channel.

```
2220    IF I=1 THEN ! Just did TDMA 800, get ready for TDMA 1900
2230      OUTPUT Testset;”CALL:SET:DTC:BAND PCS”
2240      PRINT ““
2250      PRINT “TDMA PCS Results:”
2260    ELSE   ! Done doing TDMA 1900, get ready for Analog
2270      OUTPUT Testset;”CALL:TCH:TYPE AVC”
2280      PRINT ““
```

```
2290      PRINT “Analog Results:”
2300    END IF
2310    OUTPUT Testset;”CALL:HAND”
2320    OUTPUT Testset;”CALL:STAT?”
2330    ENTER Testset;Callstat$
2340    IF Callstat$<>”CONN” THEN Dropped_call
2350  NEXT I
```

Step 7: End the Connection

- "End the connection for GSM and AMPS/136"
- "Switch formats"
- "Make a new connection"

End the connection for GSM and AMPS/136

NOTE For more information about ways to end the connection with the mobile station, refer to the Programming: Getting Started Guide for the specific mobile technologies.

The example below illustrates how to end the connection from the test set. The commands needed to perform this function are the same for both GSM and AMPS/136 formats. They are different for other formats supported by the test application.

```
1780  OUTPUT Testset;"CALL:END"
1790  OUTPUT Testset;"CALL:CONN?"
1800  ENTER Testset;Callstate
1810  IF Callstate=1 THEN
1820    PRINT "Make sure the GSM phone has released the call."
1830    OUTPUT Testset;"SYST:PRES3"
1840  END IF
```

Switch formats

At this point in your control program, you can switch to a new format. The process for using multiple formats of the test application is diagramed in Figure 1. on page 93. The example code below illustrates activating the AMPS/136 format.

```
1960  OUTPUT Testset;"SYST:APPL:FORM 'AMPS/136'"
```

Similarly, to activate the GPRS format, use the following example code.

```
OUTPUT Test_set;"SYST:APPL:FORM 'GPRS'"
```

Make a new connection

It is possible for you to begin testing in a new format immediately. All the set up has already been done. Testing can begin by repeating steps 4, 5, and 6 as illustrated in Figure 1. on page 93. To establish the new connection, see “Step 4: Make a Connection” on page 105.

Programming: Getting Started Guide

for

E1963A W-CDMA Mobile Test Application Revision A.01

Introduction

The Getting Started Guide is organized around the basic set of tasks a control program normally performs when testing a mobile station in a manufacturing environment.

Conventions used in this Getting Started Guide

Throughout this Getting Started Guide the term “test set” refers to an Agilent Technologies 8960 Series 10 wireless communications test set with the E1963A W-CDMA mobile test application installed.

The variable `Testset` used in the steps of the Getting Started Guide refers to the test set’s GPIB address.

How to use this Getting Started Guide

The most effective way to use this Getting Started Guide is with the Programming Flowchart included in the front inside pocket of this guide and the 8960 User Documentation. This documentation is found in two locations:

- the 8960 Family Support Site (updated frequently) on the Internet, or
- the User Documentation CD-ROM shipped with your test application.

Each step on the Programming Flowchart is illustrated with example program code in this guide. Using this Getting Started Guide, the Programming Flowchart, and the on-line information about 8960 programming, you will be able to generate a control program to perform fundamental mobile station manufacturing tests.

Useful on-line links

Go to the 8960 Family Support Site on either the CD-ROM or the Internet. The CD-ROM should self-launch to its home page. The URL for the website is:

http://www.agilent.com/find/8960support/

Navigate to the Programming page for this test application. The illustrations on the back of the Programming Flowchart show you how to navigate to this page.

The following links are under the heading Getting Started on the Programming page:

- Programming: Getting Started Guide
  - This online version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.
- Control Program Examples
  - These examples are for you to download. You may want to use these as templates for your own control program or to execute.
  - The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

About the Programming Examples Presented in this Guide

Programming Language:

Programming examples presented in this guide are written in the HP BASIC programming language, also known as RMB or Rocky Mountain BASIC. The use of HP BASIC is not an endorsement of the HP BASIC product.

Line Numbers

All of the programming examples in the guide with line numbers are sections from a control program example available on-line for you to download.

Code that is not part of the download control program example does not have line numbers. This code may represent an alternate method of performing the task or may illustrate a feature not used by the control program example.

Syntax used in Programming Examples:

- The programming examples use the shortened form of the command syntax to minimize GPIB bus transactions. The shortened form of a command is defined by use of capital letters in the command syntax. For the command syntax:

  ```
  RFANalyzer:CONTrol:MEASurement:FREQuency:AUTO?
  ```

  the shortened form would be:

  ```
  RFAN:CONT:MEAS:FREQ:AUTO?
  ```

- The programming examples do not include optional nodes. Optional nodes in the command syntax are defined by enclosing the node inside the [ ] brackets. For example, the command syntax:

  ```
  CALL[:CELL]:POWer[:SAMPlitude] -80dBm
  ```

  appears in the programming examples as:

  ```
  CALL:POW -80dBm
  ```

- Programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the on-line information for the definition and use of these command separators.

Complex Commands

Complex commands are used to configure the state and assign values to parameters simultaneously. Complex commands can be used to save programming steps and minimize GPIB bus transactions.

- The syntax below turns the state of the parameter on.

  OUTPUT Test_set;"SET:DTXP:TIM:STAT ON"

- The syntax below is used to assign a value to the parameter.

  OUTPUT Test_set;"SET:DTXP:TIM:TIME 10 S"

- Both of the above actions can be accomplished with one syntax command:

  OUTPUT Test_set;"SET:DTXP:TIM:STIM 10 S"

  The command above sets the parameter state to ON and the value of the parameter to 10 seconds. Note that in this example the optional command mnemonic :STIMe has been included to clarify that this complex command was used to set both the state and the value.

- This command can be shortened further by removing the optional command mnemonic :STIMe, as shown below.

  OUTPUT Test_set;"SET:DTXP:TIM 10 S"

  This is the format that will be used throughout this guide.

Step 1: Set Up the Test Set

In this step you initialize the test set and set up general operating conditions.

- "Initialize the Test Set"
- "Set Up General Operating Conditions"

Initialize the Test Set

Fully Preset the Test Set

It is important to start each production session with the test set in a known state. Sending the *RST command resets all parameters to their default values, ends all measurement processes, and sets all measurement triggers to single.

Clear the Error Queue

At the start of each production session it is useful to clear the error queue so that you know any messages logged are relevant to the current production session.

Programming Example

```
250   ! Fully preset the test set
260   OUTPUT Testset;"*RST"
270   ! Clear the error queue
280   OUTPUT Testset;"*CLS"
```

Set Up General Operating Conditions

Turn Debugger On

The debugger is useful while you are developing code. When it is on, the test set alerts you when you send an incorrect command. You should turn it off once your code is complete.

Set Operating Mode

You must set the operating mode to FDD Test Mode.

Set Amplitude Offsets

You can account for path loss in your system by setting amplitude offsets. You can specify up to 20 frequency/amplitude pairs.

Programming Example

```
320   ! Turn debugger on
330   OUTPUT Testset;"SYST:COMM:GPIB:DEB ON"
340   ! Set operating mode to FDD Test Mode
350   OUTPUT Testset;"CALL:OPER:MODE FDDT"
360   ! Set amplitude offsets
370   OUTPUT Testset;"SYST:CORR:FREQ 800MHZ,900MHZ,1900MHZ,2000MHZ"
380   OUTPUT Testset;"SYST:CORR:SGA -0.7,-0.8,-1.0,-1.1"
```

Step 2: Configure Test Set and Mobile Station Parameters

In this step you configure the parameters that allow a connection to be made between the test set and user equipment.

- "Set Up the Downlink"
- "Set Up the Uplink"

Set Up the Downlink

Set Downlink Power

You must set the downlink power (cell power).

Set Downlink Frequency

You must set the frequency at which the downlink will transmit, by specifying either the channel or the frequency.

To specify the downlink channel, send the following command:

```
CALL:CHAN <>
```

If you choose to specify the downlink frequency rather than channel, you must first set control of the downlink output frequency to manual (automatic off), and then set the downlink frequency, as shown in the programming example below.

Set Downlink Primary Scrambling Code

You can set the downlink primary scrambling code.

Set Up Downlink Physical Channels

There are several downlink physical channels and an AWGN (Additive White Gaussian Noise) source which you can configure. You can set the power level of each physical channel (relative to cell power) and the absolute power level of the AWGN source. For some of the physical channels you can also specify the channelization code. You can specify the downlink DPCH type (12.2k RMC is the only option at this time) and data type (such as PRBS15).

Programming Example

```
470   ! Set downlink power
480   OUTPUT Testset;”CALL:POW “;Rf_level
490   ! Set downlink frequency
500   OUTPUT Testset;”CALL:CONT:DOWN:FREQ:AUTO OFF”
510   OUTPUT Testset;”CALL:RFG:FREQ 2.0 GHZ”
520   ! Set downlink primary scrambling code
530   OUTPUT Testset;”CALL:SCOD:PRIM 1”
540   ! Set up DPCH
550   OUTPUT Testset;”CALL:DPCH:TYP RMC12”
560   OUTPUT Testset;”CALL:DPCH:RMC12:CCOD CODE9”
570   OUTPUT Testset;”CALL:FDDT:DPCH -10.3”
580   OUTPUT Testset;”CALL:DTCH:DATA PRBS15”
590   ! Set up CPICH
600   OUTPUT Testset;”CALL:FDDT:CPIC -3.3”
610   ! Set up P-CCPCH
620   OUTPUT Testset;”CALL:FDDT:CCPC:PRIM -5.35”
630   ! Set up PICH
640   OUTPUT Testset;”CALL:PICH:CCOD CODE16”
650   OUTPUT Testset;”CALL:FDDT:PICH -8.35”
660   ! Set up AWGN
670   OUTPUT Testset;”CALL:AWGN:POW:STAT OFF”
```

Set Up the Uplink

Set Power Control Bits

You can specify which closed loop power control bit sequence is sent on the downlink DPCCH.

Set Expected Uplink Power

You must specify what uplink power level the test set should expect. You can do this one of two ways.

You can set the MS Target Power (once Active Cell mode is implemented in the test set, this command will force the user equipment to the specified output power. In FDD Test Mode, since the 8960 is not performing active closed loop power control of the uplink, this command does not change the user equipment’s output power). Setting the MS Target Power automatically sets the receiver’s expected power accordingly.

CALL:MS:POW:TARG <>

You can also set the receiver’s expected power by setting control of the expected power to manual (automatic off) and then setting the expected power value, as shown in the programming example below.

Set Expected Uplink Frequency

You must indicate to the 8960 at what frequency the user equipment will transmit. (Unlike other technologies, in W-CDMA the uplink channel is not automatically determined based on the downlink channel. The offset between the uplink and downlink may vary based upon which band class you are operating in. So, you must specify the uplink frequency.) You can specify the expected uplink frequency by channel or by frequency.

To specify the expected uplink channel, send the following command:

CALL:UPL:CHAN <>

If you choose to specify the frequency rather than the channel, you must first set control of the expected frequency to manual (automatic off), and then set the uplink frequency, as shown in the programming example below.

For some testing scenarios you may choose to perform measurements on a frequency other than that which the user equipment is using for its signaling link. To accommodate this, in addition to the commands to set expected uplink channel or frequency, there is a command to set the measurement frequency. To set the measurement frequency you must set measurement frequency control to manual (automatic off) and then set the measurement frequency, as shown in the programming example below. This is only necessary if you want to perform measurements on a frequency other than the uplink channel or frequency.

Set Expected Uplink Primary Scrambling Code

You must indicate to the 8960 what primary scrambling code the user equipment is using.

Programming Example

```
710   ! Set power control bits to all up to force UE to max power
720   OUTPUT Testset;”CALL:FDDT:CLPC:UPL:MODE UP”
730   ! Set expected uplink power
740   OUTPUT Testset;”RFAN:CONT:POW:AUTO OFF”
750   OUTPUT Testset;”RFAN:MAN:POW:FDD 0”
```

```
760   ! Set expected uplink frequency and measurement frequency
770   OUTPUT Testset;”RFAN:CONT:UPL:FREQ:AUTO OFF”
780   OUTPUT Testset;”RFAN:MAN:UPL:FREQ 1.9 GHZ”
790   OUTPUT Testset;”RFAN:CONT:MEAS:FREQ:AUTO OFF”
800   OUTPUT Testset;”RFAN:MAN:MEAS:FREQ 1.9 GHZ”
810   ! Set expected uplink primary scrambling code
820   OUTPUT Testset;”CALL:UPL:DPCH:SCOD 0”
```

Step 3: Set Measurement Parameters

In this step you set up measurement parameters. This configures the measurements so that they are ready to execute in step 5.

Many of the measurements have only generic measurement parameters available, such as measurement count, timeout, trigger arm and trigger source.

There is a command available to set all measurement triggers to single (SET:CONT:OFF), which is the recommended trigger arm configuration for remote use of the test set. However, if you sent the *RST command in step 1, all measurement triggers will already be set to single.

Some measurements have measurement-specific parameters available. Consult the programming reference material available on the Internet to find out more about measurement-specific parameters.

Programming Example

```
880   ! Set all measurement triggers to single
890   OUTPUT Testset;”SET:CONT:OFF”
900   ! Set up thermal power
910   OUTPUT Testset;”SET:WTP:TIM 3;COUN 1”
920   ! Set up channel power
930   OUTPUT Testset;”SET:WCP:TIM 3;COUN 1;TRIG:SOUR IMM”
940   OUTPUT Testset;”SET:WCP:INT:TIME 10 MS”
950   ! Set up waveform quality
960   OUTPUT Testset;”SET:WWQ:TIM 3;COUN 1;TSL 1”
970   ! Set up ACLR
980   OUTPUT Testset;”SET:WACL:TIM 3;COUN 1;TRIG:SOUR IMM”
990   ! Set up Loopback BER
1000  OUTPUT Testset;”SET:WBER:TIM 10;COUN 15000”
```

Step 3: Set Measurement Parameters

Step 4: Make a Connection

In this step you establish communication between the test set and user equipment such that measurements can be made.

When the test set is operating in FDD Test Mode, you do not send any commands to the test set for this step. From the previous steps, the test set should already be properly configured and transmitting its downlink signal so that the user equipment can detect and synchronize to it. In this step you either implement a pause in your program and wait for the user equipment to be manually configured and begin transmitting, or send appropriate test mode commands to configure the user equipment and begin its transmission.

During this step, the user equipment must begin transmission of a 12.2k RMC DPCH at the proper power level, frequency and primary scrambling code. It must be synchronized to the test set for waveform quality and transmit in loopback mode 1 for loopback BER.

Programming Example

```
1060  PRINT “Configure UE for 12.2k RMC Now.”
1070  PRINT
1080  PRINT “UE Synchronization required for Waveform Quality.”
1090  PRINT “UE loopback Mode 1 required for loopback BER.”
1100  PRINT
1110  PRINT “8960 DL is 2.0 GHz”
1120  PRINT “8960 UL Expected Frequency: 1.9 GHz”
1130  PRINT “8960 UL Expected Power: 0 dBm”
1140  PRINT “8960 UL Expected Primary Scrambling Code: 0”
1150  PRINT
1160  PRINT “When UE is configured and transmitting, press ‘F2’.”
1170  PAUSE
```

Step 5: INITiate and FETCh Measurements

In this step you INITiate measurements, FETCh the results, and verify the results are valid.

- “INITiate a Set of Concurrent Measurements”
- “FETCh Measurement Results”

INITiate a Set of Concurrent Measurements

The test set is capable of performing concurrent measurements. To start the measurement process you INITiate a set of concurrent measurements.

FETCh Measurement Results

To retrieve measurement results as they complete, you must set up a loop using the INIT:DONE? query. Depending upon the output of the query, the program will either wait for a measurement to complete, FETCh the result of a completed measurement, or continue the program once all measurements are complete. Once a measurement result is FETChed, you should check its integrity indicator to verify that the result is valid.

Programming Example

```
1280  ! INITiate a set of concurrent measurements
1290    OUTPUT Testset;”INIT:WTP;WWQ;WACL;WBER”
1300  ! FETCh the measurement results (using a subroutine)
1310    Fetch_results

1780  SUB Fetch_results
1790    REPEAT
1800  ! Determine if any measurements are done
1810      OUTPUT 714;”INIT:DONE?”
1820      ENTER 714;Measdone$
1830      SELECT Measdone$
1840      CASE “WTP”
1850  ! FETCh measurement result
1860        OUTPUT 714;”FETC:WTP:INT?;POW?”
1870        ENTER 714;Integrity,Power
```

```
1880  ! Verify measurement result is valid
1890        IF Integrity=0 THEN
1900  ! For valid result, print result to screen (using a subroutine)
1910          Print_results(Measdone$,Power)
1920        ELSE
1930  ! For invalid result, invoke error handler (using a subroutine)
1940          Meas_error(Measdone$,Integrity)
1950        END IF
1960      CASE “WCP”
1970        OUTPUT 714;”FETC:WCP:INT?;POW?”
1980        ENTER 714;Integrity,Power
1990        IF Integrity=0 THEN
2000          Print_results(Measdone$,Power)
2010        ELSE
2020          Meas_error(Measdone$,Integrity)
2030        END IF
2040      CASE “WWQ”
2050        OUTPUT 714;”FETC:WWQ?”
2060        ENTER 714;Integrity,Evm,Ferr,Ooff,Perr,Merr
2070        IF Integrity=0 THEN
2080          Print_results(Measdone$,Evm,Ferr,Ooff,Perr,Merr)
2090        ELSE
2100          Meas_error(Measdone$,Integrity)
2110        END IF
2120      CASE “WACL”
2130        OUTPUT 714;”FETC:WACL:INT?;AVER?”
2140        ENTER 714;Integrity,Negfive,Posfive,Negten,Posten
2150        IF Integrity=0 THEN
2160          Print_results(Measdone$,Negfive,Posfive,Negten,Posten)
2170        ELSE
2180          Meas_error(Measdone$,Integrity)
2190        END IF
2200      CASE “WBER”
2210        OUTPUT 714;”FETC:WBER?;WBER:INT?”
2220        ENTER 714;Ber,Integrity
2230        IF Integrity=0 THEN
2240          Print_results(Measdone$,Ber)
2250        ELSE
2260          Meas_error(Measdone$,Integrity)
2270        END IF
2280      END SELECT
2290  ! Exit loop when all measurements are complete
```

```
2300    UNTIL Measdone$=”NONE”
2310  SUBEND
```

Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

In this step you change characteristics of the link between the test set and user equipment.

For example, to change the uplink frequency, you must pause the program to wait for the user equipment to be configured to transmit at the new frequency. You must also change the expected uplink frequency and measurement frequency accordingly, as shown in the programming example below.

If you change the test set configuration, such as downlink frequency or primary scrambling code, you must configure the test set and then allow the user equipment to find the new configuration and synchronize to it before proceeding with measurements.

Programming Example

```
1410      PRINT “Change UE UL Frequency to 1.850 GHz.”
1420      PRINT
1430      PRINT “UE Synchronization required for Waveform Quality.”
1440      PRINT “UE loopback Mode 1 required for loopback BER.”
1450      PRINT
1460      PRINT “When UE is configured and transmitting, press ‘F2’.”
1470      PAUSE
1480  ! Change expected uplink frequency
1490      OUTPUT Testset;”RFAN:MAN:MEAS:FREQ 1.85 GHZ”
1500      OUTPUT Testset;”RFAN:MAN:UPL:FREQ 1.85 GHZ”
```

Step 7: End the Connection

In this step you end the communication between the test set and user equipment, and prepare for the next program action.

Once you have ended the user equipment transmission, to prepare for testing the next user equipment, you should partially preset the test set. This stops all measurement processes but does not reset all parameters to their default values. That way you do not need to reset the parameters which are constant for all user equipment in the test batch.

Programming Example

```
1630  PRINT “Testing complete, end UE transmission.”
1640  ! Partially preset the test set
1650  OUTPUT Testset;”SYST:PRES3”
```

CST 学习培训课程套装

该培训套装由易迪拓培训联合微波 EDA 网共同推出，是最全面、系统、专业的 CST 微波工作室培训课程套装，所有课程都由经验丰富的专家授课，视频教学，可以帮助您从零开始，全面系统地学习 CST 微波工作的各项功能及其在微波射频、天线设计等领域的设计应用。且购买该套装，还可超值赠送 3 个月免费学习答疑…

课程网址：http://www.edatop.com/peixun/cst/24.html

HFSS 天线设计培训课程套装

套装包含 6 门视频课程和 1 本图书，课程从基础讲起，内容由浅入深，理论介绍和实际操作讲解相结合，全面系统的讲解了 HFSS 天线设计的全过程。是国内最全面、最专业的 HFSS 天线设计课程，可以帮助您快速学习掌握如何使用 HFSS 设计天线，让天线设计不再难…

课程网址：http://www.edatop.com/peixun/hfss/122.html

13.56MHz NFC/RFID 线圈天线设计培训课程套装

套装包含 4 门视频培训课程，培训将 13.56MHz 线圈天线设计原理和仿真设计实践相结合，全面系统地讲解了 13.56MHz 线圈天线的工作原理、设计方法、设计考量以及使用 HFSS 和 CST 仿真分析线圈天线的具体操作，同时还介绍了 13.56MHz 线圈天线匹配电路的设计和调试。通过该套课程的学习，可以帮助您快速学习掌握 13.56MHz 线圈天线及其匹配电路的原理、设计和调试…

详情浏览：http://www.edatop.com/peixun/antenna/116.html

我们的课程优势：

※ 成立于 2004 年，10 多年丰富的行业经验，

※ 一直致力并专注于微波射频和天线设计工程师的培养，更了解该行业对人才的要求

※ 经验丰富的一线资深工程师讲授，结合实际工程案例，直观、实用、易学

联系我们：

※ 易迪拓培训官网：http://www.edatop.com

※ 微波 EDA 网：http://www.mweda.com

※ 官方淘宝店：http://shop36920890.taobao.com