Agilent Technologies 8960 Series 10 E5515B Wireless Communications Test Set
Agilent Technologies E1964A GPRS Mobile Test Application

Programming Guide

Test Application Revision A.00

Printed in U.S.A. October 2000

Agilent Part Number: E1964-90002

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

Agilent Technologies

Notice

Agilent Technologies, Inc.
Learning Products Department
24001 E. Mission
Liberty Lake, WA 99019-9599

U.S.A.

Agilent Technologies

Safety, Warranty, and Regional Sales and Service Offices Information

Manufacturer’s Declaration

This statement is provided to comply with the requirements of the German Sound Emission Directive, from 18 January 1991.

This product has a sound pressure emission (at the operator position) < 70 dB(A).

- Sound Pressure Lp < 70 dB(A).
- At Operator Position.
- Normal Operation.
- According to ISO 7779:1988/EN 27779:1991 (Type Test).

Herstellerbescheinigung

- Schalldruckpegel Lp < 70 dB(A).
- Diese Information steht im Zusammenhang mit den Anforderungen der Maschinenlärminformationsverordnung vom 18 Januar 1991.
- Am Arbeitsplatz.
- Normaler Betrieb.
- Nach ISO 7779:1988/EN 27779:1991 (Typprüfung).

Safety Considerations

GENERAL

This product and related documentation must be reviewed for familiarization with safety markings and instructions before operation.

This product has been designed and tested in accordance with *IEC Publication 1010*, "Safety Requirements for Electronic Measuring Apparatus," and has been supplied in a safe condition. This instruction documentation contains information and warnings which must be followed by the user to ensure safe operation and to maintain the product in a safe condition.

SAFETY EARTH GROUND

A uninterruptible safety earth ground must be provided from the main power source to the product input wiring terminals, power cord, or supplied power cord set.

SAFETY SYMBOLS

⚠ Indicates instrument damage can occur if indicated operating limits are exceeded.

⚡ Indicates hazardous voltages.

⏚ Indicates earth (ground) terminal

WARNING **A WARNING note denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a WARNING sign until the indicated conditions are fully understood and met.**

CAUTION A CAUTION note denotes a hazard. It calls attention to an operation procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond an CAUTION note until the indicated conditions are fully understood and met.

WARNING

This product is a Safety Class I instrument (provided with a protective earthing ground incorporated in the power cord). The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. Any interruption of the protective conductor inside or outside of the product is likely to make the product dangerous. Intentional interruption is prohibited.

Whenever it is likely that the protection has been impaired, the instrument must be made inoperative and be secured against any unintended operation.

If this instrument is to be energized via an autotransformer (for voltage reduction), make sure the common terminal is connected to the earth terminal of the power source.

If this product is not used as specified, the protection provided by the equipment could be impaired. This product must be used in a normal condition (in which all means for protection are intact) only.

No operator serviceable parts in this product. Refer servicing to qualified personnel. To prevent electrical shock, do not remove covers.

Servicing instructions are for use by qualified personnel only. To avoid electrical shock, do not perform any servicing unless you are qualified to do so.

The opening of covers or removal of parts is likely to expose dangerous voltages. Disconnect the product from all voltage sources while it is being opened.

The power cord is connected to internal capacitors that my remain live for 5 seconds after disconnecting the plug from its power supply.

For Continued protection against fire hazard, replace the line fuse(s) only with 250 V fuse(s) or the same current rating and type (for example, normal blow or time delay). Do not use repaired fuses or short circuited fuseholders.

Always use the three-prong ac power cord supplied with this product. Failure to ensure adequate earth grounding by not using this cord may cause product damage.

This product is designed for use in Installation Category II and Pollution Degree 2 per *IEC 1010* and *IEC 664* respectively. FOR INDOOR USE ONLY.

This product has autoranging line voltage input, be sure the supply voltage is within the specified range.

To prevent electrical shock, disconnect instrument from mains (line) before cleaning. Use a dry cloth or one slightly dampened with water to clean the external case parts. Do not attempt to clean internally.

Ventilation Requirements: When installing the product in a cabinet, the convection into and out of the product must not be restricted. The ambient temperature (outside the cabinet) must be less than the maximum operating temperature of the product by 4° C for every 100 watts dissipated in the cabinet. If the total power dissipated in the cabinet is greater than 800 watts, then forced convection must be used.

Product Markings

CE - the CE mark is a registered trademark of the European Community. A CE mark accompanied by a year indicated the year the design was proven.

CSA - the CSA mark is a registered trademark of the Canadian Standards Association.

CERTIFICATION

Agilent Technologies, Inc. certifies that this product met its published specifications at the time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members

WARRANTY

This Agilent Technologies instrument product is warranted against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Agilent Technologies, Inc. will at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Agilent. Buyer shall prepay shipping charges to Agilent and Agilent shall pay shipping charges, duties, and taxes for products returned to Agilent from another country.

Agilent warrants that its software and firmware designated by Agilent for use with an instrument will execute its programming instructions when properly installed on that instrument. Agilent does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. AGILENT SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. AGILENT SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASE ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

ASSISTANCE

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products. For any assistance, contact your nearest Agilent Technologies Sales and Service Office.

DECLARATION OF CONFORMITY

According to ISO/IEC Guide 22 and CEN/CENELEC EN45014

Manufacturer's Name:	**Agilent Technologies UK Ltd.**	**Agilent Technologies, Inc.**
Manufacturer's Address:	**Electronic Products & Solutions Group - Queensferry South Queensferry West Lothian, EH30 9TG Scotland, United Kingdom**	**24001 E. Mission Avenue Liberty Lake Washington 99019-9599 USA**

Declares that the product

Product Name: 8960 Series 10 Wireless Communications Test Set

Model Number: E5515B

Product Options: This declaration covers all options of the above product.

EMC:

Conforms with the following product specifications:

Standard:	Limit:
CISPR11:1990 / EN55011:1991	Group 1 Class A
IEC 801-2:1991 / EN 50082-1:1992	4kV CD, 8kV AD
IEC 801-3:1984 / EN 50082-1:1992	3V/m, 27-500 MHz
IEC 801-4:1988 / EN 50082-1:1992	0.5kV signal lines, 1kV power lines

Safety:

The product conforms to the following safety standards:

IEC 61010-1(1990) +A1(1992) +A2(1995) / EN 61010-1:1993
Canada / CSA-C22.2 No. 1010.1-93

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC, and the EMC Directive 89/336/EEC, and carries the CE mark accordingly.

South Queensferry, Scotland. 04 May 2000

R.M. Evans / Quality Manager

Spokane, Washington, USA. 04 May 2000

W.V.Roland / Reliability & Regulatory Engineering Manager

For further information, please contact your local Agilent Technologies sales office, agent, or distributor.

Table 1. Regional Sales and Service Offices

United States of America: Agilent Technologies Test and Measurement Call Center P.O. Box 4026 Englewood, CO 80155-4026 (tel) 1 800 452 4844	Canada: Agilent Technologies Canada Inc. 5159 Spectrum Way Mississaua, Ontario L4W 5G1 (tel) 1 877 894 4414	Europe: Agilent Technologies European Marketing Organization P.O. Box 999 1180 AZ Amstelveen The Netherlands (tel) (3120) 547 9999
Japan: Agilent Technologies Japan Ltd. Measurement Assistance Center 9-1 Takakura-Cho, Hachioji-Shi, Tokyo 192-8510, Japan (tel) (81) 456-56-7832 (fax) (81) 426-56-7840	Latin America: Agilent Technologies Latin America Region Headquarters 5200 Blue Lagoon Drive, Suite #950 Miami, Florida 33126 U.S. A. (tel) (305) 267 4245 (fax) (305) 267 4286	Australia/New Zealand: Agilent Technologies Australia Pty Ltd 347 Burwood Highway Forest Hill, Wictoria 3131 (tel) 1 800 629 485 (Austrailia) (fax) (61 3) 9272 0749 (tel) 0 800 738 378 (New Zealand) (fax) (64 4) 802 6881
Asia Pacific: Agilent Technologies 19/F, Cityplaza One, 111 Kings Road, Taikoo shing, Hong Kong, SAR (tel) (852) 2599 7899 (fax) (852) 2506 9233		

Contents

Programming the Agilent Technologies 8960 Series 10 for GPRS Mobile Testing in Active Cell Operating Mode

Introduction

Conventions used in this Programming Guide

Throughout this Programming Guide the term "test set" refers to an Agilent Technologies 8960 Series 10 wireless communications test set with the GPRS mobile test application installed.

Purpose of this Programming Guide

The test set represents state-of-the-art technology in one-box-testers and contains many powerful test capabilities which are accessible through easy-to-use GPIB programming commands. The purpose of this Programming Guide is to teach you how to write a basic control program, using the test set's GPIB command set. This program will perform fundamental manufacturing tests on a mobile station with the test set operating in active cell mode.

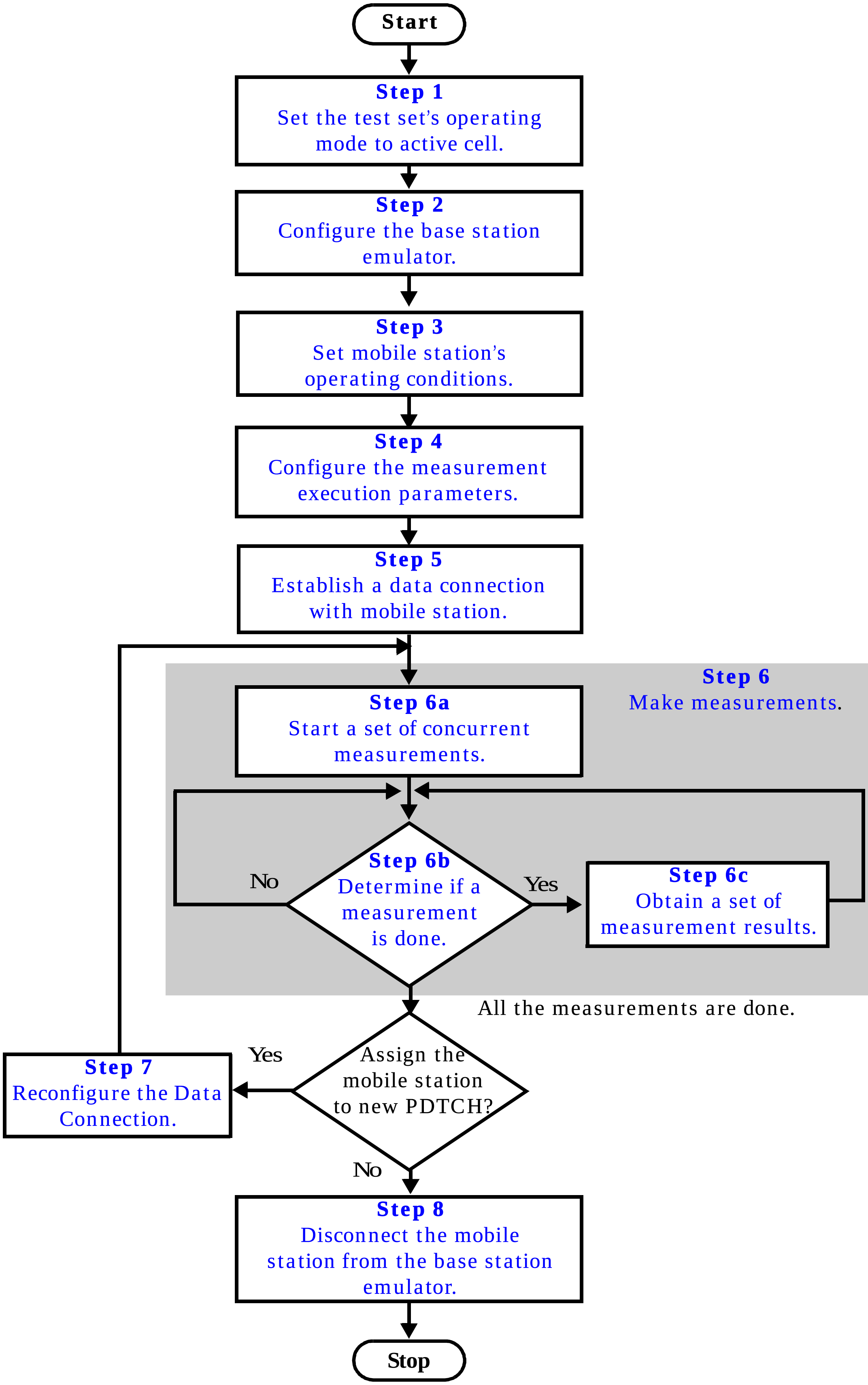
How this Programming Guide is Organized

The Programming Guide is organized around a typical set of tasks a control program would normally perform when testing a mobile station in a manufacturing environment. The set of tasks is shown in "Figure 1. Typical Flow of Tasks Performed by a Control Program" on page 15.

Typically in a manufacturing environment:

- steps 1, 2, 3, and 4 are done once each time a production run is started,
- steps 5 and 8 are done once for each mobile station tested during the production run,
- steps 6 and 7 are done iteratively for each mobile station tested during the production run. The number of iterations for these steps is dependent upon how many mobile station operating conditions are being tested (that is, number of channels, number of power levels, and so fourth).

Figure 1. Typical Flow of Tasks Performed by a Control Program



How to use this Programming Guide

This Programming Guide is divided into 9 sections. Sections 1 through 8 (Step 1 through Step 8) should be read in sequence. Each section, in order, describes one of the tasks to be performed by the control program and shows how to accomplish that task using the test set's GPIB command set. As you progress through each section your understanding of how the test set's GPIB interface operates will increase as you see the control program evolve.

The last section of the Programming Guide presents a "Comprehensive Programming Example" on page 49 which uses all of the topics discussed in sections 1 through 8 together in one program to give you a sense of how to tie everything together.

About the Programming Examples Presented in This Programming Guide

Programming Language:

The programming examples presented in this guide are written in the HP BASIC programming language, also known as Rocky Mountain BASIC or RMB.

Syntax Used in Programming Examples:

- The programming examples use the shortened form of the command syntax to reduce test time by minimizing GPIB bus transactions. The shortened form of a command is defined by use of capital letters in the command syntax. For example, for the command syntax:

  ```
  SETup:TXPower:CONTinuous <ON|1|OFF|2>
  ```

  the shortened form would be:

  ```
  SET:TXP:CONT <ON|1|OFF|2>
  ```

- 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:PDTCH[:ARFCn][:SELected] <numeric value>
  ```

  appears in the programming examples as:

  ```
  CALL:PDTCH <numeric value>
  ```

- The programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the test set's reference documentation for details on the definition and use of these command separators.

Step 1: Set the Test Set's Operating Mode to Active Cell

Background

The test set contains a base station emulator with the primary purpose of providing the call processing necessary for parametric measurements on the RF signals of a mobile station with GPRS capability.

An important characteristic of the test set's base station emulator is its operating mode. The operating mode sets the way in which the base station emulator interacts with the mobile station. Active cell mode is used when emulating a normal cell.

This Programming Guide focuses on programming the test set's base station emulator in active cell operating mode.

Overview of Active Cell Operating Mode

Active cell is the only operating mode currently available. It provides active signaling between the mobile station and the base station emulator.

Active Cell Features

The basic features provided by the base station emulator when the operating mode is set to active cell are:

- Generation of a BCH (broadcast channel).
- Support for Routing Area and Location Area updates.
- Support for the Agilent proprietary BLER data connection.
- The ATTACH procedure is supported.
- The DETACH procedure is supported.
- Packet data transfers on the uplink and downlink are supported.
- All measurements supported in the test application are available.
- The base station emulator automatically controls the test set's receiver.

Step 2: Configure the Base Station Emulator

The Base Station Emulator

In active cell operating mode the base station emulator, using the test set's GMSK modulated source, generates a downlink (base station to mobile station direction) broadcast channel (BCH) which represents a cell. The mobile station can "camp" to this signal, just as it would camp to a cell on a real network, and performs an attach procedure to register the mobile station with the network. A Packet Data Traffic Channel (PDTCH) can then be established using an Agilent proprietary data connection designed to accommodate block error rate reports from the mobile station. You can also make transmitter measurements using this data connection. Data is transferred in both the downlink and uplink directions. Measurements can be made, using the base station emulator's measurement receiver, under essentially identical conditions to that which the mobile station would experience on a real network.

The base station emulator emulates a cell in any one of the following GSM frequency bands:

- PGSM - Primary (band) GSM, also known as GSM900
- EGSM - Extension (band) GSM (includes PGSM)
- DCS - Also known as DCS1800

NOTE The term GSM is used to refer to any combination of, or all of, the supported bands. It is not used as a shortened term for PGSM.

To configure the base station emulator, you must configure the BCH and PDTCH as described in the following sections. It may not be necessary to configure all the parameters all the time. The test set's default settings allow a properly functioning mobile station to successfully camp on the cell under most circumstances.

In a manufacturing environment it may be desirable to explicitly configure 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.

Configuring the Broadcast Channel Parameters

The BCH parameters are configured using the CALL processing subsystem commands shown in the following table.

Table 1. Broadcast Channel Settable Parameters

Parameter	Command Syntax	Footnote
Cell Band	CALL[:CELL[1]]:BAND <PGSM|EGSM|DCS>	
Cell Power	CALL[:CELL[1]]:POWer:AMPLitude <numeric value>[<suffix>]	

Configuring the Packet Data Traffic Channel Parameters

The PDTCH parameters are configured using the CALL processing subsystem commands shown in the following table.

Table 2. Packet Data Traffic Channel Settable Parameters

Parameter	Command Syntax	Footnote
Packet Data Traffic Channel Band	CALL:<PDTCH|PDTChannel>:BAND <PGSM|EGSM|DCS>	1
Packet Data Traffic Channel Number (ARFCN)	CALL:<PDTCH|PDTChannel>[:ARFCn][:SELected] <numeric value> OR CALL:<PDTCH|PDTChannel>[:ARFCn]:<PGSM|EGSM|DCS> <numeric value>	2
Data Connection Type	CALL:FUNCtion:DATA:TYPE <BLER>	3
Power Reduction Downlink Level	CALL:<PDTCH|PDTChannel>:PREDuction:LEVel<1|2> <numeric value>[<suffix>]	4
Downlink Burst Power Reduction Selection	CALL:<PDTCH|PDTChannel>:PREDuction:BURSt<1|2> <PRL1|PRL2>	5

Table Footnotes

1. The PDTCH band setting becomes the selected band (see Note 2).
2. Sets the PDTCH channel for the PDTCH band selected with the PDTCH Band command (see Note 1).
3. The BLER data connection type is Agilent proprietary. It allows block error rate measurements to be made.
4. Assigns power reduction level values to PRL1 and PRL2.
5. Assigns either PRL1 or PRL2 power reduction levels to a downlink burst.

Programming Example

The following program example illustrates the use of the base station emulator BCH and PDTCH configuration commands.

```
OUTPUT Test_set;”CALL:BAND EGSM;POW:AMPL -60”
OUTPUT Test_set;”CALL:PDTCH 1023”
OUTPUT Test_set;”CALL:FUNC:DATA:TYPE BLER”
OUTPUT Test_set;”CALL:PDTCH:PRED:LEV1 2 DB”
OUTPUT Test_set;”CALL:PDTCH:PRED:LEV2 10 DB”
OUTPUT Test_set;”CALL:PDTCH:PRED:BURS1 PRL2”
OUTPUT Test_set;”CALL:PDTCH:PRED:BURS2 PRL1”
```

Step 3: Set the Mobile Station’s Operating Conditions

Mobile Station Uplink Burst Transmit Power Level

The mobile station’s uplink burst transmit power level is specified by the test set in a command sent to the mobile station. In this way, the test set emulates a base station, which sends transmit level change commands to a mobile. The test set also uses the mobile station’s uplink burst TX level parameter to determine the power to expect from the mobile station when making measurements on the mobile station’s transmitter.

Overview

You can set the mobile station’s operating conditions using the CALL processing subsystem commands shown in the following table.

Table 3. Settable Mobile Station Operating Conditions

Parameter	Command Syntax	Footnote
Uplink Burst TX Level	CALL:<PDTCH|PDTChannel>:MS:TXLevel[:SELected]:BURSt<n> <numeric value> OR CALL:<PDTCH|PDTChannel>:MS:TXLevel:<PGSM|EGSM|DCS>:BURSt<n> <numeric value>	1,2

Table Footnotes

1. The Packet Data Traffic Channel band setting is the selected band.
2. There is only one uplink burst available for this release. Therefore, it is not necessary to specify a value in your code for <n>. It defaults to 1.

Programming Example

The following example illustrates how to set the mobile station’s operating conditions. The first line sets the mobile station’s transmit power level for the currently active band to 14. The second line sets the power level when the DCS band becomes the active band.

```
OUTPUT Test_set;”CALL:PDTCH:MS:TXL:BURS 14”
OUTPUT Test_set;”CALL:PDTCH:MS:TXL:DCS:BURS 8
```

Step 4: Configure the Measurement Execution Parameters

Measurement Execution Parameters

Measurement execution parameters allow you to control the conditions under which a measurement operates. Not all measurements use all the execution parameters shown below. Additionally, some measurements have parameters that are specific to the measurement such as offset frequency lists or filter settings. Each measurement has its own set of parameters which are unique to it and have no affect on the execution of other measurements. Refer to the GPIB syntax listing in the reference documentation for a detailed list of execution parameters for individual measurements. The general set of measurement execution parameters and their generic categories are:

- Measurement Averaging (used by most measurements)

  Multi-Measurement Count
  Multi-Measurement Count State

- Measurement Triggering (used by most measurements)

  Trigger Arm
  Trigger Source
  Trigger Delay
  Trigger Qualifier

- Measurement Synchronization (used by some measurements)

  Burst Synchronization

- Measurement Timeouts (used by all measurements)

  Measurement Timeout
  Measurement Timeout State

- Measurement Specific (execution parameters specific to an individual measurement)

Overview of the SETup subsystem

The SETup subsystem allows you to configure the measurement parameters. Each individual measurement parameter can be set and queried using the associated SETup subsystem command. The general hierarchy of the SETup subsystem command structure is as follows:

```
SETup:<measurement mnemonic>:<measurement parameter> <parameter setting/value>
```

For example, the SETup subsystem is used below to set a timeout value for the transmit power measurement:

```
SET:TXP:TIM 5
```

The following table shows the measurements available in the test application and their associated <measurement mnemonic> used in the SETup command syntax.

Table 4. Measurement Mnemonics used in the SETup Subsystem

Measurement	<measurement mnemonic>
Transmit Power	TXPower
Power vs Time	PVTime
Phase & Frequency Error	PFERror
Output RF Spectrum	ORFSpectrum

Configuring Measurement Averaging Parameters

Table 1. Measurement Averaging Parameters

Parameter	Command Syntax
Measurement Count Number and State	`SETup:<measurement mnemonic>:COUNt[:SNUMber] <numeric value>`
Measurement Count State	`SETup:<measurement mnemonic>:COUNt:STATe <ON|1|OFF|0>`
Measurement Count Number	`SETup:<measurement mnemonic>:COUNt:NUMBer <numeric value>`

Configuring Multi-Measurement Count State and Count Number Simultaneously

The multi-measurement count state can be set to ON and the multi-measurement count number can be set to some value using a single command. For example:

```
OUTPUT Test_set;”SET:TXP:COUN 10”
```

would set the multi-measurement count state to ON and set the number of averages to 10 for the transmit power measurement.

Multi-Measurement Count State Parameter

The Multi-Measurement Count State parameter is used to turn measurement averaging on and off.

Multi-Measurement Count Number Parameter

The Multi-Measurement Count Number parameter sets the number of measurement samples taken during each measurement cycle when the COUNt:STATe parameter is set to ON.

Configuring Measurement Triggering Parameters

Table 2. Measurement Triggering Parameters

Parameter	Command Syntax
Trigger Arm	`SETup:<measurement mnemonic>:CONTinuous <ON|1|OFF|0>`
Trigger Source	`SETup:<measurement mnemonic>:TRIGger:SOURce <AUTO|IMMediate|PROTocol|RISE>`
Trigger Delay	`SETup:<measurement mnemonic>:TRIGger:DELay <numeric value>[<suffix>]`
Trigger Qualifier	`SETup:<measurement mnemonic>:TRIGger:QUALifier <ON|1|OFF|0>`

Trigger Arm Parameter

The Trigger Arm parameter allows you to set whether a measurement either:

- make one measurement then stop (single), or
- automatically re-arm upon completion of one measurement and repeat the process (continuous).

For example,

```
OUTPUT Test_set;"SET:TXP:CONT OFF"
```

would set the trigger arming to single for the transmit power measurement.

NOTE If you are using the test set remotely, set the trigger arm to single (CONTinuous OFF) for each measurement or, a single command (SET:CONT:OFF) or the *RST command can be used to set the trigger arm to single for all measurements.

Trigger Source Parameter

The Trigger Source parameter selects the source of the measurement trigger signal.

```
OUTPUT Test_set;"SET:TXP:TRIG:SOUR AUTO"
```

would set the trigger source to AUTO for the transmit power measurement.

Trigger Delay Parameter

The Trigger Delay parameter controls the delay between the trigger event (the point in time at which the trigger signal is received) and the start of sampling. Negative values indicate that the sampling occurs prior to the trigger event.

```
OUTPUT Test_set;"SET:TXP:TRIG:DEL 10 US"
```

would set the trigger delay to 10 μs for the transmit power measurement.

Trigger Qualifier Parameter

The Trigger Qualifier parameter allows you to enable or disable automatic trigger re-arming following a trigger event which occurred when no valid signal (burst) was present.

```
OUTPUT Test_set;"SET:TXP:TRIG:QUAL ON"
```

would turn the trigger qualifier on for the transmit power measurement.

Configuring the Burst Synchronization Parameter

Burst Synchronization Parameter

The burst synchronization parameter allows you to specify where in the sampled data stream the test set starts analyzing the captured data. Burst synchronization occurs after the measurement data is captured. The burst synchronization parameter's setting determines how the measurement's time reference is developed from the sampled data.

Not all measurements have synchronization choices and not all synchronization choices are available in measurements that use synchronization. Measurement synchronization and measurement triggering are independent from each other and may be used in any combination.

The command syntax for setting the burst synchronization parameter is:

```
SETup:<measurement mnemonic>:BSYNc <MIDamble|AMPLitude|NONE>
```

The burst synchronization is set to midamble for the power versus time measurement in the following example:

```
OUTPUT Test_set;"SET:PVT:BSYN MID"
```

Configuring Measurement Timeout Parameters

Table 3. Measurement Timeout Parameters

Parameters	Command Syntax
Measurement Timeout Time and State	`SETup:<meas-mnemonic>:TIMeout[:STIMe] <numeric value>[<suffix>]`
Measurement Timeout State	`SETup:<meas-mnemonic>:TIMeout:STATe <ON|1|OFF|0>`
Measurement Timeout Time	`SETup:<meas-mnemonic>:TIMeout:TIME <numeric value>[<suffix>]`

Configuring Measurement Timeout State and Timeout Time Simultaneously

The measurement timeout state can be set to ON and the measurement timeout time can be set to some value using a single complex command.

```
OUTPUT Test_set;"SET:TXP:TIM:STIM 10"
```

would set the measurement timeout state to ON and set the measurement timeout time to 10 seconds for the transmit power measurement. Note that in this example the optional command mnemonic :STIMe has been included for purposes of clarity.

Measurement Timeout State Parameter

The Measurement Timeout State parameter allows you to enable or disable measurement timeouts.

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

would enable measurement timeouts for the power versus time measurement.

Measurement Timeout Time Parameter

The Measurement Timeout Time parameter allows you to set the maximum time that a measurement executes before failing with a timeout error (when the TIMEout:STATe parameter is set to ON).

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

would set the measurement timeout time to 10 seconds for the transmit power measurement.

Configuring Measurement Specific Parameters

Some measurements have parameters that are specific to the measurement. Refer to the GPIB syntax listings for a detailed list of execution parameters for individual measurements. Some of the possible programming techniques that you can use to configure measurement specific execution parameters are described below.

Sending Comma-Separated Parameter Configuration Lists to the Test Set

Numerous parameters may be necessary to configure a measurement. For example, the output RF spectrum measurement can require up to 22 frequency offsets for the modulation part of the measurement and up to 8 frequency offsets for the switching part of the measurement. You send these offsets as comma separated lists. There are a variety of techniques that you can use to send these lists. Some of these techniques are shown below.

1. Include each individual parameter in the command itself. For example:

```
OUTPUT Test_set;"SET:ORFS:SWIT:FREQ .4MHZ,.6MHZ,-.4MHZ,-.6MHZ"
```

2. Store the parameter values in a data structure and send the command with the data structure appended to it. For example:

   - Using a string variable:

```
DIM Swit_offs$[255]
Swit_offs$=”.4MHZ,.6MHZ,-.4MHZ,-.6MHZ,1.2MHZ,-1.2MHZ”
OUTPUT Test_set;”SET:ORFS:SWIT:FREQ “&Swit_offs$
```

   - Using numeric arrays:

```
OPTION BASE 1
REAL Swit_offs(8),Mod_offs(22)
!
DATA 400,-400,600,-600,1200,-1200,1800,-1800
DATA .1,-.1,.2,-.2,.25,-.25,.4,-.4,.6,-.6,.8,-.8
DATA 1,-1,1.2,-1.2,1.4,-1.4,1.6,-1.6,1.8,-1.8
!
READ Swit_offs(*)
READ Mod_offs(*)
!
Swit_img:IMAGE K,7(K,”KHZ,”),K,”KHZ”
Mod_img:IMAGE K,21(K,”MHZ,”),K,”MHZ”
OUTPUT Test_set USING Swit_img;”SET:ORFS:SWIT:FREQ”,Swit_offs(*)
OUTPUT Test_set USING Mod_img;”SET:ORFS:MOD:FREQ”,Mod_offs(*)
```

Programming Example

The following example illustrates configuring the measurement execution parameters for the output RF spectrum, transmit power, and phase and frequency error measurements.

```
! Configure ORFS Measurement:
!
OUTPUT Test_set;”SET:ORFS:SWIT:COUN 5”     ! Examples of using complex
OUTPUT Test_set;”SET:ORFS:MOD:COUN 10”     ! commands to set multi-meas
                                           ! state and count at same time.
OUTPUT Test_set;”SET:ORFS:TRIG:SOUR AUTO”  ! Set trig source to AUTO.
OUTPUT Test_set;”SET:ORFS:CONT OFF”        ! Set trig mode to single.
OUTPUT Test_set;”SET:ORFS:TIM 60”          ! Set timeout time to 60 sec.
! Put switching and modulation offsets to be tested into string variables.
Swit_offs$=”400KHZ,-400KHZ,600KHZ,-600KHZ,1200KHZ,-1200KHZ,1800KHZ,-1800KHZ”
Mod_offs$=”.2MHZ,-.2MHZ,.4MHZ,-.4MHZ,.6MHZ,-.6MHZ,.8MHZ,-.8MHZ,1MHZ,-1MHZ”
OUTPUT Test_set;”SET:ORFS:SWIT:FREQ “&Swit_offs$
OUTPUT Test_set;”SET:ORFS:MOD:FREQ “&Mod_offs$
!
! Configure TX Power Measurement:
!
OUTPUT Test_set;”SET:TXP:COUN 3;CONT OFF;TIM 20”
OUTPUT Test_set;”SET:TXP:TRIG:SOUR RISE;QUAL ON”
!
! Configure Phase & Frequency Error Measurement:
!
OUTPUT Test_set;”SET:PFER:COUN 8;CONT OFF;TIM 30;BSYN MID”
OUTPUT Test_set;”SET:PFER:TRIG:SOUR PROT;QUAL ON”
```

Step 5: Establish a Data Connection with the Mobile Station

- "Data Connection Status Synchronization" on page 31
- "Process for Establishing a Data Connection" on page 33

Data Connection Status Synchronization

When the control program requires that a data connection be established between the mobile station and the test set, it must also be able to obtain knowledge about the status of the connection. The control program issues the commands necessary to initiate the connection process to the test set. The control program must then be able to determine when the connection has transitioned through various states successfully so that it can proceed. The control program must also be able to determine that the transition has not been successful so appropriate action can be taken.

Data Connection States

At any instant in time a data connection is in one of the following states:

- Idle
- Attaching*
- Detaching*
- Attached
- Starting*
- Ending*
- Transferring

The states marked by a * are referred to as transitory states because the amount of time which the connection spends in any of these states is limited. Therefore, the connection is not allowed to stay in a transitory state forever.

CALL:STATus:DATA? Query

The CALL:STATus:DATA? query immediately returns the state of the data connection at the time the query is received.

Table 5. Responses Returned from the CALL:STAT:DATA? Query

Response	Meaning
IDLE	The connection is in the Idle state.
ATTG	The connection is in the Attaching transitory state.
DET	The connection is in the Detaching transitory state.
ATT	The connection is in the Attached state.
STAR	The connection is in the Starting transitory state.
END	The connection is in the Ending transitory state.
TRAN	The connection is in the Transferring state.

Using the CALL:STATus[:DATA]? query for Connection State Synchronization

The most common technique used to determine if a process has completed (either successfully or unsuccessfully), is to repeatedly query the connection state using the CALL:STATus:DATA? query command inside a program loop. The return value from each query is checked to determine if the process is proceeding or has reached the desired state.

There are, however, some inherent problems associated with this technique:

- The rapid polling of the instrument increases bus traffic and places increased demand on the instrument's processors to respond to the constant stream of queries.
- The control program must handle failure conditions. For example: if a call origination process is started but the call never leaves the Idle state, the control program must incorporate some technique to prevent the program from staying in the loop forever waiting for a transition out of the Idle state.

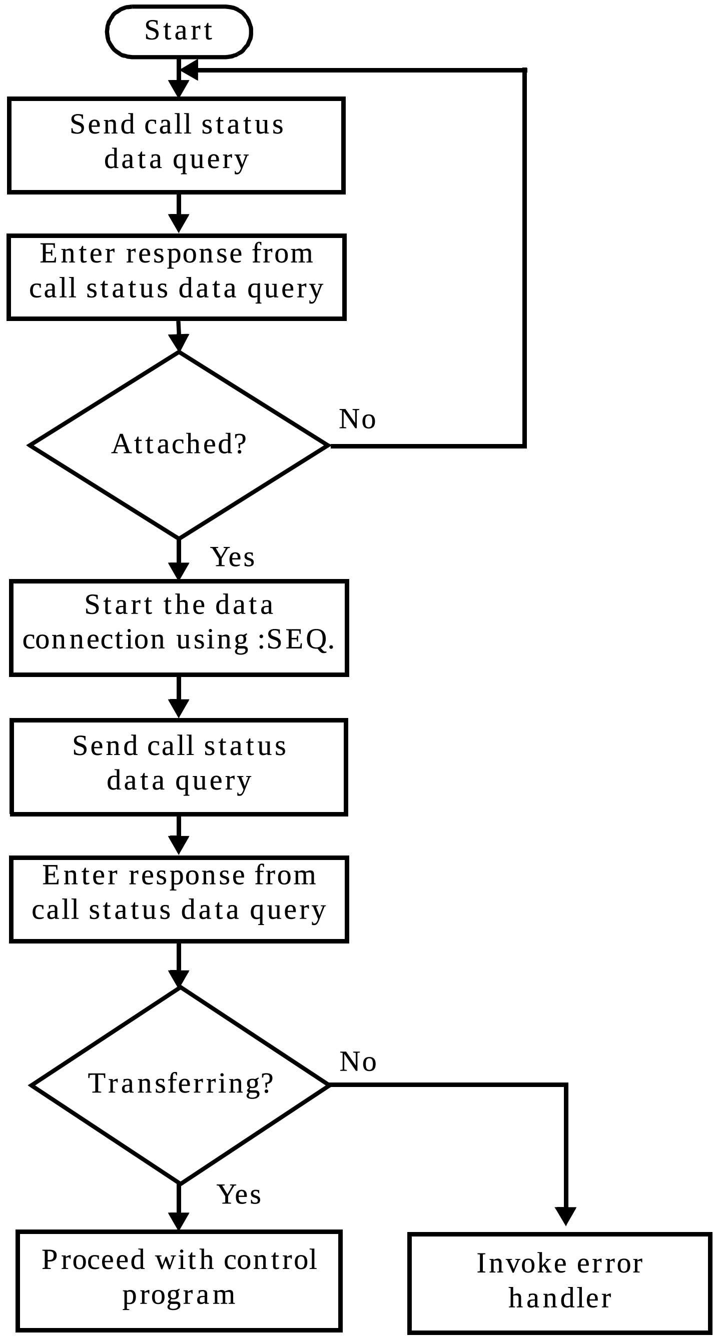
You can avoid rapidly polling the test set by attaching the :SEQuential synchronization command to the command used to initiate the data connection. This will force the test set to complete this command before executing any other commands. Therefore, the status of the data connection is stable and only needs to be queried once to determine if the connection succeeded or failed.

Process for Establishing a Data Connection

Use the CALL:FUNCtion:DATA:STARt command in conjunction with the :SEQuential synchronization command to establish a data connection.

The recommended process for establishing a data connection is shown in the following figure.

Figure 1. Process for Establishing a Data Connection



Programming Example

The following programming example illustrates the process for establishing a data connection.

```
REPEAT                              ! Loop to determine when connection state
                                    !  is Attached
OUTPUT Test_set;"CALL:STAT:DATA?" ! Query state of data connection
ENTER Test_set;Conn_stat$
! <code to prevent infinite looping>
UNTIL Conn_stat$="ATT"
!
OUTPUT Test_set;"CALL:FUNC:DATA:STAR:SEQ"  ! Start the data connection
!
OUTPUT Test_set;"CALL:STAT:DATA?"
ENTER Test_set;Conn_stat$
IF Conn_stat$<>"TRAN" THEN
! <put error handler here>
END IF
! Data Connection is established so proceed with control program
```

Step 6: Make Measurements

- "Measurement Concurrency"
- "Programming Example"
- "Things That Can Go Wrong"

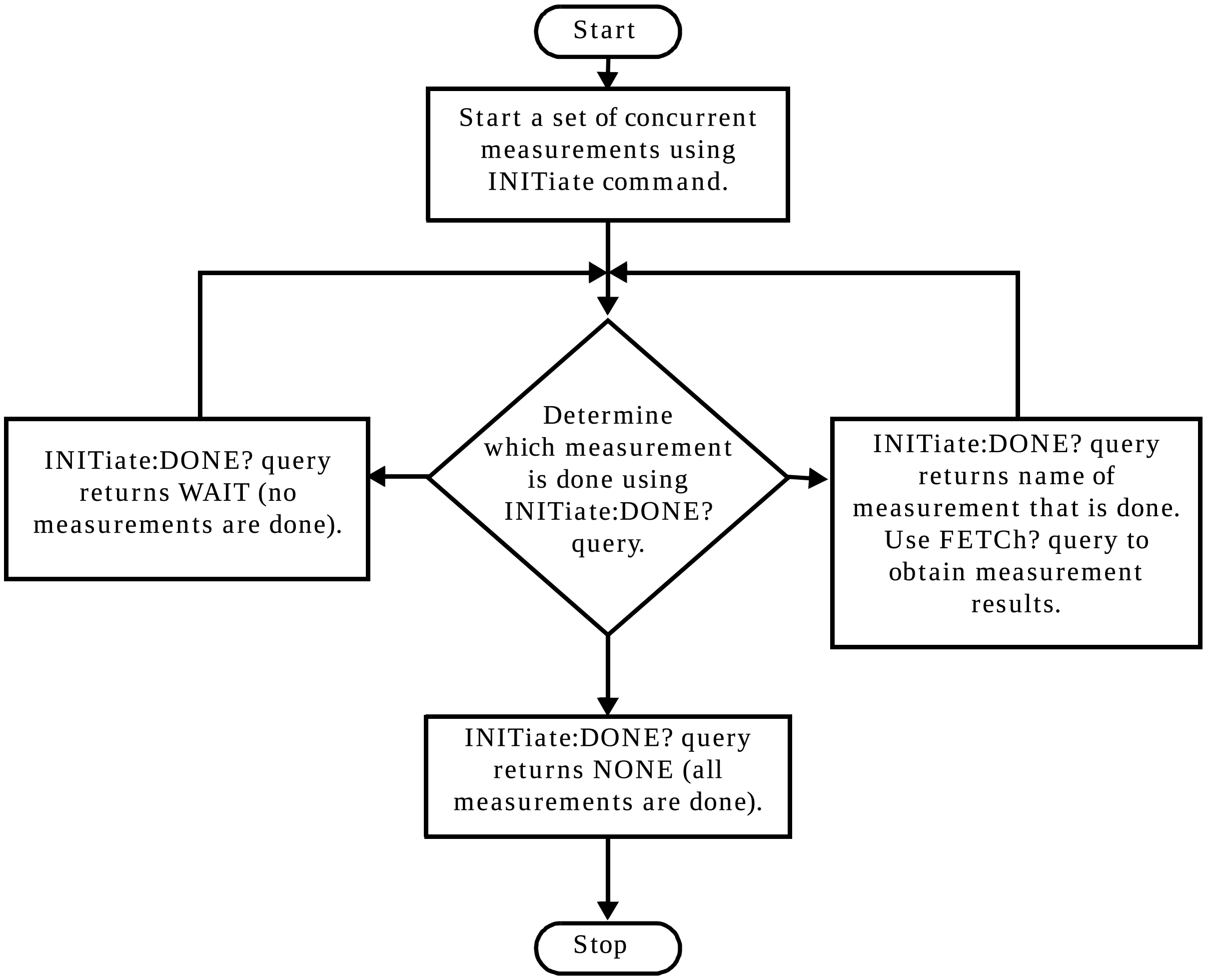
Measurement Concurrency

The multiple signal path, DSP based, multiple processor architecture of the test set allows the test set to make concurrent measurements. This means that:

- multiple measurements can execute and finish at the same time.
- individual measurement completion is not influenced by other measurement processes.
- availability of measurement results is not dependent upon the sequence that the measurements were requested in.
- results from measurements that take few processor cycles are available without having to wait for measurements that take many processor cycles.

"Figure 1. Process for Making Measurements" on page 36 shows the recommended process for making concurrent measurements using the test set's command set.

Figure 1. Process for Making Measurements



Programming Example

The following programming example shows you how to make a transmit power measurement and a phase and frequency error measurement concurrently using the process shown in "Figure 1. Process for Making Measurements" on page 36.

```
! Step 6a: Start Set of Concurrent Measurements:
!
OUTPUT Test_set;"INIT:TXP;PFER"
!
! Step 6b: Determine If A Measurement Is Done:
!
LOOP
  OUTPUT Test_set;"INIT:DONE?"
  ENTER Test_set;Meas_done$
!
! Step 6c: Obtain Measurement Results
!
  SELECT Meas_done$
    CASE "TXP"
      OUTPUT Test_set;"FETC:TXP:POW?"
      ENTER Test_set;Avg_tx_power
    CASE "PFER"
      OUTPUT Test_set;"FETC:PFER:RMS?"
      ENTER Test_set;Max_rms_phas_er
  END SELECT
EXIT IF Meas_done$="NONE"
END LOOP
```

Things That Can Go Wrong

Measurement Integrity Always Returns a Value of 6

A measurement integrity value of 6 indicates that some characteristic of the input signal is under range. Typically this is the amplitude (power) of the signal. This low amplitude causes the level of the DSP sampler to be below a threshold required by the test set to produce results with the accuracy specified for the test set.

Possible Cause One of the most likely causes of a measurement underrange condition is signal loss caused by either the test fixture or cabling.

Suggested Workaround You can compensate for fixture or cable loss by using the RF IN/OUT port's amplitude offset parameters. Twenty different offsets can be set up for twenty different frequencies. To set the frequencies use the command:

```
SYSTem:CORRection:SFRequency (comma separated numeric values)<suffix>
```

Similarly, to then set the offsets for these frequencies, use the following command syntax:

```
SYSTem:CORRection:SGAin (comma separated numeric values)<suffix>
```

Refer to the reference documentation for details about how to use these commands properly.

Step 6a: Start Set Of Concurrent Measurements

Starting Measurements

The INITiate command is used to start measurements. Each individual measurement can be started using the INITiate command. For starting measurements, the syntax of the INITiate command is as follows:

```
INITiate:<measurement mnemonic>[:ON]
```

The following table shows the measurements available in the test application and their associated <measurement mnemonic> used in the INITiate command syntax.

Table 6. Measurement Mnemonics Used In The INITiate Subsystem

Measurement	<measurement mnemonic>
Transmit Power	TXPower
Power vs Time	PVTime
Phase & Frequency Error	PFERror
Output RF Spectrum	ORFSpectrum

Programming Example

The following command starts the transmitter power measurement.

```
OUTPUT Test_set;"INIT:TXP"
```

Using Compound Commands to Start Multiple Measurements

More than one measurement can be started using a single INITiate command. For example:

```
OUTPUT Test_set;"INIT:TXP;PFER"
```

starts the transmit power measurement and the phase and frequency error measurement. These measurements then run concurrently.

Step 6b: Determine if a Measurement Is Done

Background

After a set of concurrent measurements have been started, you can set up your control program to determine when individual measurement results are available. This enables you to request results for a completed measurement while other measurements are still running.

Overview

Use the INITiate:DONE? query command to determine which measurement has completed.

This command is a query only and returns only one response per query. The responses returned and their meaning are shown in the following table.

Once a measurement is reported as being complete via the INITiate:DONE? query it is removed from the done list (it is not reported again). To use the INITiate:DONE? query properly, your control program should immediately fetch a measurement's results once it is reported as being complete.

Table 7. Responses Returned from INITiate:DONE? Query

Response	Meaning
TXP	The transmit power measurement is done.
PVT	The power versus time measurement is done.
PFER	The phase and frequency error measurement is done.
ORFS	The output RF spectrum measurement is done.
WAIT	There are one or more measurements that are in progress, but none of those measurements are done yet.
NONE	No measurements are in progress.

For an example of how the INIT:DONE? query is used, see "Programming Example" on page 37.

Step 6c: Obtain a Set of Measurement Results

Background

In order to minimize bus traffic and reduce test time, the test set's measurements are designed to return multiple measured values in response to a single measurement request.

For example, if a transmit power measurement with averaging is initiated there are five measurement results available. These are:

1. Measurement integrity value
2. Average value
3. Minimum value
4. Maximum value
5. Standard deviation value

The test set can return the measurement results in a variety of formats to suit your needs. For example, the transmitter power measurement results can be returned as:

- Measurement integrity and average value,

  or

- Average value and minimum value and maximum value and standard deviation value,

  or

- Average value only,

  or

- Minimum value only,

  or

- Maximum value only,

  or

- Standard deviation value only,

  or

- Measurement integrity value only.

The formats available for individual measurements are specifically defined in the reference documentation and can be found in the FETCh? subsystem's GPIB command syntax.

Overview

After a measurement is initiated, the measurement results can be queried using the FETCh? subsystem. The general structure of the FETCh? command is as follows:

```
FETCh:<measurement mnemonic>:<result format>?
```

The following table shows the measurements available in the test application and their associated <measurement mnemonic> used in the FETCh? command syntax.

The command syntax used to obtain the various measurement result formats (<result format>) for each measurement can be found in the test set's FETCh? subsystem's GPIB command syntax reference information.

Table 8. Measurement Mnemonics Used In The FETCh? Subsystem

Measurement	**<measurement mnemonic>**
Transmit Power	TXPower
Power vs Time	PVTime
Phase & Frequency Error	PFERror
Output RF Spectrum	ORFSpectrum

Programming Example

The following command returns the minimum value from the set of samples taken during the transmit power measurement (when averaging is turned on and number of samples taken >1).

```
OUTPUT Test_set;"FETCh:TXP:POW:MIN?"
```

Step 7: Reconfigure the Data Connection

Process for Reconfiguring a Data Connection

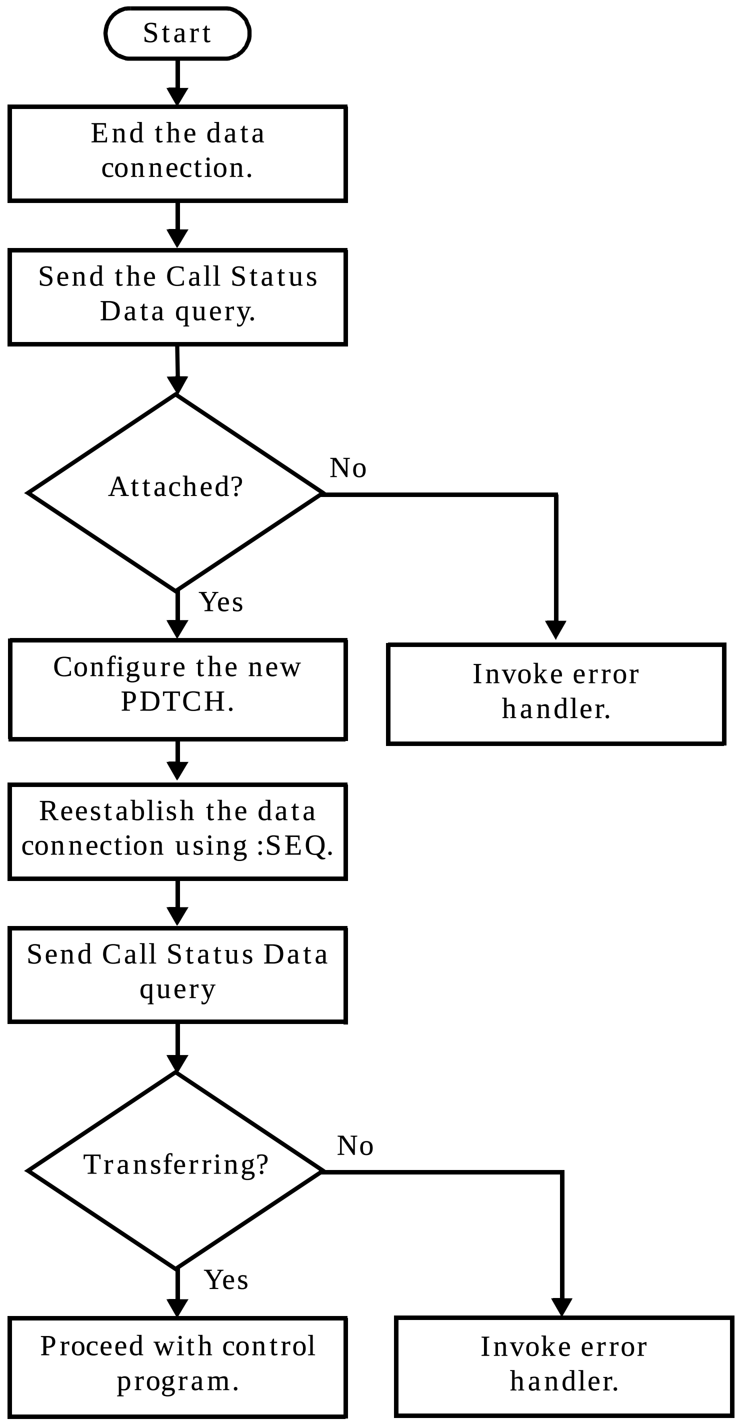
If you want to test the mobile station using different parameters, such as channel number, channel band, or mobile station transmit level, you must reconfigure the data connection.

To reconfigure a data connection you must:

- first end the data connection,
- use the PDTCH commands to setup the new channel, and
- reestablish the data connection using the CALL:FUNCtion:DATA:STARt:SEQuential command.

This process is shown in more detail "Figure 1. Process for Reconfiguring the Data Connection" on page 44 and the "Programming Example" on page 45.

Figure 1. Process for Reconfiguring the Data Connection



Programming Example

The following example illustrates how to reconfigure the data connection. Notice the packet data traffic channel commands are used to change several parameters.

```
! Existing conditions: a mobile station is connected to the test
! set, operating mode is set to active cell and a data connection
! is in the transferring state.
! Step 1: End the data connection
OUTPUT Test_set;"CALL:FUNC:DATA:STOP"
OUTPUT Test_set;"CALL:STAT:DATA?"
ENTER Test_set;Conn_stat$
IF Conn_stat$<>"ATT" THEN
!<put error handler here>
END IF
! Step 2: Change the packet data traffic channel band, number, and
!         MS TX Level
OUTPUT Test_set;"CALL:PDTCH:BAND DCS"
OUTPUT Test_set;"CALL:PDTCH 556"
OUTPUT Test_set;"CALL:PDTCH:MS:TXL:BURS 4"
OUTPUT Test_set;"CALL:FUNC:DATA:STAR:SEQ"!Reestablish the data connection
                                 !No other commands will be processed
                                 !until this operation completes
                                 !because the :SEQ has been attached.
! Step #2: Check connection status
OUTPUT Test_set;"CALL:STAT:DATA?"
ENTER Test_set;Conn_stat$
IF Conn_stat$<>"TRAN" THEN
! <put error handler here>
END IF
! Data Connection is established
```

Step 8: Disconnect the Mobile Station from the Base Station Emulator

Data Connection Status Synchronization

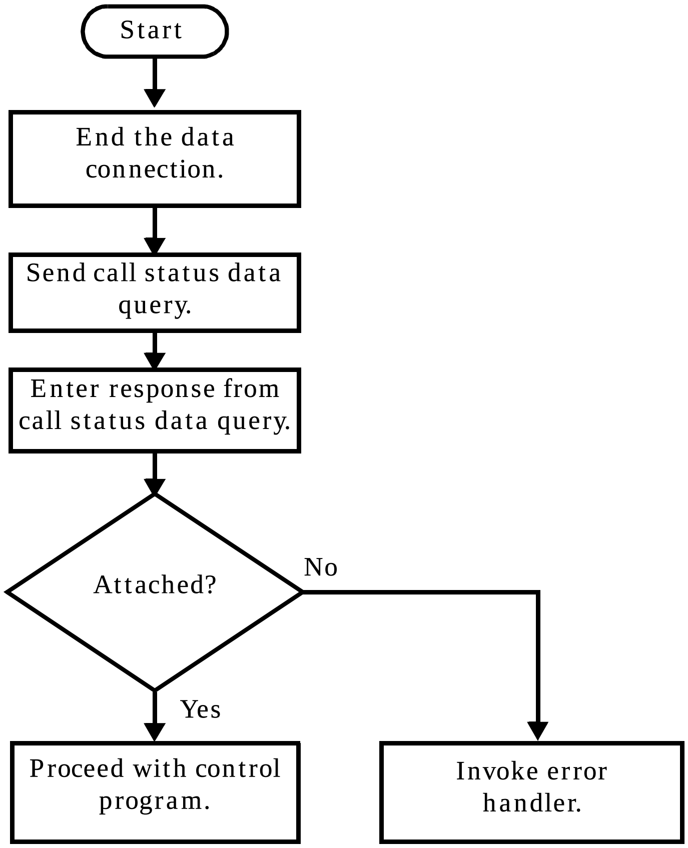
After you have completed the testing of the phone, the last step will be to disconnect the data connection established between the mobile station and the test set. As detailed in "Step 5: Establish a Data Connection with the Mobile Station" on page 31, the control program must also be able to obtain knowledge about the status of the connection. The control program issues the commands to the test set necessary to begin the disconnect process. It must able to determine what state the connection is in so that it can proceed or take appropriate action if the disconnect is not progressing correctly.

End the Data Connection

The recommended process for ending the data connection is shown in the following figure.

Use the CALL:FUNCtion:DATA:STOP command.

Figure 1. Process for Stopping ETSI Test Mode



Programming Example

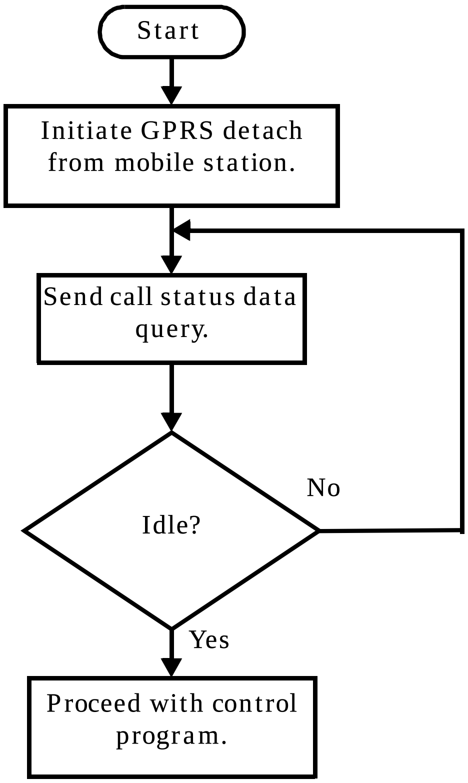
The programming example below illustrates how to return the data connection status to “Attached” by ending the data connection and how to maintain synchronization with the control program.

```
OUTPUT Test_set;”CALL:FUNC:DATA:STOP”  ! End the data connection.
OUTPUT Test_set;”CALL:STAT:DATA?”      ! Send call status data query.
ENTER Test_set;Conn_stat$
IF Conn_stat$<>”ATT” THEN              ! Check if exit was successful
! <put error handler here>
END IF
! The data connection has been ended. Proceed with control program.
```

Initiate the GPRS detach procedure from the Mobile Station

There is no facility in the test set to initiate a call disconnect from the mobile station. This must be done manually or through a test bus built into the mobile station.

Figure 2. Process for Initiating GPRS detach from the Mobile Station



Programming Example

This example is one way to do data connection synchronization while ending the data connection.

```
! Manually initiate a GPRS Detach or send commands via a bus
DISP “Initiate a GPRS Detach.”
REPEAT                            ! Loop to determine when connection state
                                  !  is Idle
OUTPUT Test_set;”CALL:STAT:DATA?” ! Query state of data connection
ENTER Test_set;Conn_stat$
! <code to prevent infinite looping>
UNTIL Conn_stat$=”IDLE”
!
! Data Connection is established so proceed with control program
```

Comprehensive Programming Example

This section contains an example program for making measurements using the test set. The program follows the task flow shown in “Figure 1. Typical Flow of Tasks Performed by a Control Program” on page 15.

Example Program

```
10    ! Prog Name: P1_ex.txt     Rev: A.0.1      Date Code: 07 September 2000
20    !
30    ! Configure the BASIC environment, dimension and initialize variables.
40    ! These actions are unrelated to programming the Agilent Technologies 8960.
50    !
60    OPTION BASE 1
70    COM /Address/Test_set
80    ! Allocate arrays to hold ORFS switching & modulation frequency offsets.
90    DIM Swit_offs$[255],Mod_offs$[255]
100   ! Allocate arrays to hold measurement results.
110   REAL Txpower(4)
120   Test_set=714  ! Test set’s GPIB address.
130   PRINTER IS CRT
140   CLEAR SCREEN
150   !
160   ! Reset test set to start from a known state. Not always necessary to do full
170   ! preset in a manufacturing environment but desireable in programming example.
180   !
190   OUTPUT Test_set;”*RST”
200   !
210   ! Turn on the GPIB debugger.  This is optional but very helpful for debugging
220   ! GPIB commands when developing new code.
230   !
240   OUTPUT Test_set;”SYST:COMM:GPIB:DEB:STAT ON”
250   !
260   ! Check error message queue and STOP if any errors present. This ensures that
270   ! the example program starts with no error conditions present in the test set.
280   !
290   CALL Chk_err_msg_que
300   !
310   !****************************************************************************
320   ! Step 1: Set Test Set’s Operating Mode to Active Cell
330   !****************************************************************************
340   !
350   ! Active Cell operating mode is the only mode available in this release.
360   ! There is no GPIB command to specify operating mode in this release.
370   !
380   !****************************************************************************
390   ! Step 2: Configure the Base Station Emulator
400   !****************************************************************************
410   !
420   OUTPUT Test_set;”CALL:BAND PGSM;POW:AMPL -60”! Set active broadcast band to
430                                                ! PGSM and cell power to -60 dBm.
440   OUTPUT Test_set;”CALL:PDTCH 45”         ! Set packet data traffic channel to 45
450   OUTPUT Test_set;”CALL:FUNC:DATA:TYPE BLER” ! Set ETSI Test Mode Type to BLER
460   !
470   ! Assign values to the power reduction levels
480   OUTPUT Test_set;”CALL:PDTCH:PRED:LEV1 12 DB”
490   OUTPUT Test_set;”CALL:PDTCH:PRED:LEV2 3 DB”
500   !
510   ! Assign power levels to the downlink bursts
```

```
520   OUTPUT Test_set;”CALL:PDTCH:PRED:BURS1 PRL2”
530   OUTPUT Test_set;”CALL:PDTCH:PRED:BURS2 PRL1”
540   !
550   !*****************************************************************************
560   ! Step 3: Set the Mobile Station’s Operating Conditions
570   !*****************************************************************************
580   !
590   ! Assign a power level to the uplink burst
600   OUTPUT Test_set;”CALL:PDTCH:MS:TXL:BURS 14”
610   !
620   !*****************************************************************************
630   ! Step 4: Configure the Measurement Execution Parameters
640   !*****************************************************************************
650   !
660   ! Configure ORFS Measurement:
670   !
680   OUTPUT Test_set;”SET:ORFS:SWIT:COUN 5” ! Examples of using complex commands to
690   OUTPUT Test_set;”SET:ORFS:MOD:COUN 10” ! set multi-meas state and count at
700                                          ! same time.
710   OUTPUT Test_set;”SET:ORFS:TRIG:SOUR AUTO”  ! Set trig source to AUTO.
720   OUTPUT Test_set;”SET:ORFS:CONT OFF”        ! Set trig mode to single.
730   OUTPUT Test_set;”SET:ORFS:TIM 20”          ! Set timeout time to 20 seconds.
740   ! Put switching and modulation offsets to be tested into string variables.
750   Swit_offs$=”400KHZ,-400KHZ,600KHZ,-600KHZ,1200KHZ,-1200KHZ,1800KHZ,-1800KHZ”
760   Mod_offs$=”.2MHZ,-.2MHZ,.4MHZ,-.4MHZ,.6MHZ,-.6MHZ,.8MHZ,-.8MHZ,1MHZ,-1MHZ”
770   OUTPUT Test_set;”SET:ORFS:SWIT:FREQ “&Swit_offs$
780   OUTPUT Test_set;”SET:ORFS:MOD:FREQ “&Mod_offs$
790   !
800   ! Configure TX Power Measurement:
810   !
820   OUTPUT Test_set;”SET:TXP:COUN 3;CONT OFF;TIM 20”
830   OUTPUT Test_set;”SET:TXP:TRIG:SOUR AUTO;QUAL ON”
840   !
850   ! Configure Phase & Frequency Error Measurement:
860   !
870   OUTPUT Test_set;”SET:PFER:COUN 8;CONT OFF;TIM 30;BSYN MID”
880   OUTPUT Test_set;”SET:PFER:TRIG:SOUR AUTO;QUAL ON”
890   !
900   !*****************************************************************************
910   ! Step 5: Establish a Data Connection with the Mobile Station
920   !*****************************************************************************
930   !
940   ! This example assumes the MS has initiated a GPRS attach.
950   ! The following code will query the test set for the connection status 50
960   ! times and then stop the program if an attached state is not reached. This
970   ! should give adequate time for the MS to attach.
980   !
990   Tries=1
1000  LOOP
1010    OUTPUT Test_set;”CALL:STAT:DATA?”  ! Query the connection’s state
1020    ENTER Test_set;Conn_state$
1030  EXIT IF Conn_state$=”ATT”
1040    IF Tries=50 THEN
```

```
1050      BEEP
1060      DISP ““
1070      PRINT “GPRS attach did not complete. Program terminated.”
1080      STOP
1090    END IF
1100    IF Conn_state$=”ATTG” THEN
1110      DISP “GPRS attach has started.”
1120    END IF
1130    Tries=Tries+1
1140  END LOOP
1150  DISP ““
1160  FOR Traf_chan=15 TO 115 STEP 50        ! Test channels 15, 65, and 115
1170     OUTPUT Test_set;”CALL:PDTCH “;Traf_chan
1180     FOR Ms_pwr_lvl=5 TO 15 STEP 5       ! Test power levels 5, 10, and 15
1190       OUTPUT Test_set;”CALL:PDTCH:MS:TXL:BURS1 “;Ms_pwr_lvl
1200       OUTPUT Test_set;”CALL:FUNC:DATA:STAR:SEQ” ! Starts Data Connection
1210                                           ! No other commands will be processed
1220                                           ! until this operation completes
1230                                           ! because the :SEQ has been attached
1240       OUTPUT Test_set;”CALL:STAT:DATA?”
1250       ENTER Test_set;Conn_state$
1260       IF Conn_state$<>”TRAN” THEN
1270         PRINT “Data connection failed.  Terminated program.”
1280         STOP
1290       END IF
1300  !
1310  !*****************************************************************************
1320  ! Step 6: Make Measurements
1330  !*****************************************************************************
1340  !
1350  ! Step 6a: Start a set of concurrent measurements:
1360  !
1370      OUTPUT Test_set;”INIT:TXP;PFER;ORFS”
1380  !
1390  ! Step 6b: Determine if a measurement is done:
1400  !
1410      LOOP
1420        OUTPUT Test_set;”INIT:DONE?”
1430        ENTER Test_set;Meas_done$
1440  !
1450  ! Step 6c: Obtain measurement results: Each measurement illustrates a
1460  !          different way of reading in results. There is no one right way. The
1470  !          method used is application dependent. Note that the examples do not
1480  !          show all possible ways.
1490  !
1500        SELECT Meas_done$
1510  !
1520        CASE “TXP”  ! TX Power measurement done.
1530          OUTPUT Test_set;”FETC:TXP:INT?;POW:ALL?”
1540          ENTER Test_set;Integrity,Txpower(*)
1550          IF (Integrity=0) THEN  ! Always check integrity value.
1560            PRINT “TX Power results: TCH =”;Traf_chan;”and TXL =”;Ms_pwr_lvl
1570            PRINT USING “5X,””Minimum:””,M2D.2D,”” dBm”””;Txpower(1)
```

```
1580            PRINT USING “5X,””Maximum:””,M2D.2D,”” dBm”””;Txpower(2)
1590            PRINT USING “5X,””Average:””,M2D.2D,”” dBm”””;Txpower(3)
1600            PRINT USING “5X,””Std Dev:””,M2D.2D,”” dB”””;Txpower(4)
1610          ELSE
1620            GOSUB Bad_measurement
1630          END IF
1640  !
1650        CASE “PFER”  ! Phase & Frequency Error measurement done.
1660          OUTPUT Test_set;”FETC:PFER:ALL?”
1670          ENTER Test_set;Integrity,Rms_phas_err,Peak_phas_err,Worst_freq_err
1680          IF (Integrity=0) THEN
1690            PRINT “PFERror results: TCH =”;Traf_chan;”and TXL =”;Ms_pwr_lvl
1700            PRINT USING “5X,””RMS Phase Error:””,M2D.2D,”” deg”””;Rms_phas_err
1710            PRINT USING “5X,””Peak Phase Error:””,M2D.2D,”” deg”””;Peak_phas_err
1720            PRINT USING “5X,””Worst Freq Error:””,M3D.2D,”” Hz”””;Worst_freq_err
1730          ELSE
1740            GOSUB Bad_measurement
1750          END IF
1760  !
1770        CASE “ORFS”  ! ORFS measurement done.
1780        !
1790        ! This code illustrates a more ‘generic’ approach to reading measurement
1800        ! results. By using the capabilities designed into high-level
1810        ! measurements, routines that access measurement results do not have to
1820        ! explicitly know what the measurement execution conditions were.  That
1830        ! information can be determined at the time the measurement results are
1840        ! queried.
1850        !
1860          OUTPUT Test_set;”FETC:ORFS:INT?”        ! Check measurement integrity.
1870          ENTER Test_set;Integrity
1880          IF (Integrity=0) THEN
1890            OUTPUT Test_set;”SET:ORFS:SWIT:FREQ:POIN?” ! Get number of offsets
1900                                                       ! tested.
1910            ENTER Test_set;Points
1920            IF Points THEN  ! Only query if one or more offsets tested.
1930              ALLOCATE Orfs_swit_res(Points),Orfs_swit_offs(Points)
1940              OUTPUT Test_set;”SET:ORFS:SWIT:FREQ?”  ! Get measurement offsets.
1950              ENTER Test_set;Orfs_swit_offs(*)
1960              OUTPUT Test_set;”FETC:ORFS:POW?;:FETC:ORFS:SWIT?” ! Get results.
1970              ENTER Test_set;Tx_power,Orfs_swit_res(*)
1980              PRINT “ORFS Swit Results: TCH =”;Traf_chan;”and TXL =”;Ms_pwr_lvl
1990              PRINT USING “19X,””TX Power =””,M2D.2D,”” dBm”””;Tx_power
2000              PRINT “     Offset(kHz)          Level(dBm)”
2010              PRINT “     -----------          ----------”
2020 Orfs_image:  IMAGE 6X,M4D.2D,12X,M4D.2D
2030              FOR J=1 TO Points
2040                PRINT USING Orfs_image;(Orfs_swit_offs(J)/1.E+3),Orfs_swit_res(J)
2050              NEXT J
2060              DEALLOCATE Orfs_swit_res(*),Orfs_swit_offs(*)
2070            END IF
2080            OUTPUT Test_set;”SET:ORFS:MOD:FREQ:POIN?” ! Get number of offsets
2090                                                      ! tested.
2100            ENTER Test_set;Points
```

```
2110           IF Points THEN ! Only query if one or more offsets tested.
2120             ALLOCATE Orfs_mod_res(Points),Orfs_mod_offs(Points)
2130             OUTPUT Test_set;"SET:ORFS:MOD:FREQ?" ! Get measurement offsets.
2140             ENTER Test_set;Orfs_mod_offs(*)
2150             OUTPUT Test_set;"FETC:ORFS:POW?;:FETC:ORFS:MOD?" ! Get results.
2160             ENTER Test_set;Tx_power,Pwr_30khz,Orfs_mod_res(*)
2170             PRINT "ORFS Mod Results: TCH =";Traf_chan;"and TXL =";Ms_pwr_lvl
2180             PRINT USING "18X,""30 KHz BW Power ="",M2D.2D,"" dBm""";Pwr_30khz
2190             PRINT "     Offset(kHz)          Level(dB)"
2200             PRINT "     -----------          ---------"
2210             FOR J=1 TO Points
2220               PRINT USING Orfs_image;(Orfs_mod_offs(J)/1.E+3),Orfs_mod_res(J)
2230             NEXT J
2240             DEALLOCATE Orfs_mod_res(*),Orfs_mod_offs(*)
2250           END IF
2260         ELSE
2270           GOSUB Bad_measurement
2280         END IF
2290       END SELECT
2300     EXIT IF Meas_done$="NONE"
2310     END LOOP  ! If 'WAIT' is returned from 'INIT:DONE?' query, it just falls
2320               ! through the loop.
2330  !
2340 !*****************************************************************************
2350 ! Step 7: Reconfigure the Data Connection
2360 !*****************************************************************************
2370 !
2380 OUTPUT Test_set;"CALL:FUNC:DATA:STOP" ! Ends the Data Connection
2390 OUTPUT Test_set;"CALL:STAT:DATA?"     ! Check connection status
2400 ENTER Test_set;Conn_state$
2410 IF Conn_state$<>"ATT" THEN
2420   PRINT "Data Connection was not stopped"
2430   STOP
2440 END IF
2450 !
2460    NEXT Ms_pwr_lvl
2470 NEXT Traf_chan
2480 !
2490 !*****************************************************************************
2500 ! Step 8: Disconnect the Mobile Station From the BSE
2510 !*****************************************************************************
2520 !
2530 !  Data Connction was stopped at the end of the FOR loop above.
2540 !
2550 !  MS must initiate GPRS detach
2560 DISP "Initiate a GPRS Detach"
2570 Tries=1
2580 LOOP
2590   OUTPUT Test_set;"CALL:STAT:DATA?"
2600   ENTER Test_set;Conn_state$
2610 EXIT IF Conn_state$="IDLE"
2620   IF Tries=50 THEN
2630     DISP ""
```

```
2640      PRINT “GPRS detach did not occur.  Program terminated”
2650      STOP
2660    END IF
2670    IF Conn_state$=”DET” THEN
2680      DISP “GPRS detach is in process.”
2690    END IF
2700    Tries=Tries+1
2710  END LOOP
2720  PRINT “Program completed.”
2730  STOP
2740  !
2750 Bad_measurement: !
2760  PRINT “Measurement error: “&Meas_done$
2770  PRINT “Measurement Integrity value =”;Integrity
2780  RETURN
2790  !
2800  END  ! End of program
2810  !
2820  SUB Chk_err_msg_que
2830    COM /Address/Test_set
2840    DIM Error_message$[255]
2850    Error_flag=0
2860    LOOP
2870      OUTPUT Test_set;”SYST:ERR?”
2880      ENTER Test_set;Error_number,Error_message$
2890    EXIT IF Error_number=0
2900      IF Error_number=-350 THEN
2910        Error_flag=1
2920        PRINT “Error Message Queue overflow. Error messages have been lost.”
2930      ELSE
2940        Error_flag=1
2950        PRINT Error_number,Error_message$
2960      END IF
2970    END LOOP
2980    IF NOT Error_flag THEN
2990      PRINT “No errors in Error Message Queue.”
3000      SUBEXIT
3010    END IF
3020    STOP
3030  SUBEND
```

Index