Agilent Technologies 8960 Series 10 E5515B Wireless Communications Test Set Agilent Technologies E1961A AMPS/136 Mobile Test Application

Active Cell Programming Guide

Test Application Revision A.03

Agilent Part No: E1961-90007

Printed in U.S.A.

October 2000

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

Agilent Technologies

Edition/Print Date

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

March 2000

June 2000

October 2000

Safety Summary

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Agilent Technologies Inc. assumes no liability for the customer's failure to comply with these requirements.

GENERAL

This product is a Safety Class 1 instrument (provided with a protective earth terminal). The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

All Light Emitting Diodes (LEDs) used in this product are Class 1 LEDs as per IEC 60825-1.

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.

ENVIRONMENTAL CONDITIONS

This instrument is intended for indoor use in an installation category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 95% and at altitudes of up to 2000 meters. Refer to the specifications tables for the ac mains voltage requirements and ambient operating temperature range.

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.

BEFORE APPLYING POWER

Verify that the product is set to match the available line voltage, the correct fuse is installed, and all safety precautions are taken. Note the instrument's external markings described under Safety Symbols.

GROUND THE INSTRUMENT

To minimize shock hazard, the instrument chassis and cover must be connected to an electrical protective earth ground. The instrument must be connected to the ac power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.

FUSES

Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuse holders. To do so could cause a shock or fire hazard.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE

Do not operate the instrument in the presence of flammable gases or fumes.

DO NOT REMOVE THE INSTRUMENT COVER

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified service personnel.

Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

WARNING **The WARNING sign 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 The CAUTION sign denotes a hazard. It calls attention to an operating procedure, 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 a CAUTION sign until the indicated conditions are fully understood and met.

Safety Symbols

⚠ Caution, refer to accompanying documents

Warning, risk of electric shock

Earth (ground) terminal

Alternating current

Frame or chassis terminal

Standby (supply). Units with this symbol are not completely disconnected from ac mains when this switch is off.

To completely disconnect the unit from ac mains, either disconnect the power cord, or have a qualified electrician install an external switch.

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 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

Agilent Technologies Warranty Statement for Commercial Products

Agilent Technologies 8960 Series 10 E5515B Wireless Communications Test Set, E1961A AMPS/136 Mobile Test Application

Duration of Warranty: 1 year

1. Agilent Technologies warrants Agilent Technologies hardware, accessories and supplies against defects in materials and workmanship for the period specified above. If Agilent Technologies receives notice of such defects during the warranty period, Agilent Technologies will, at its option, either repair or replace products which prove to be defective. Replacement products may be either new or like-new.
2. Agilent Technologies warrants that Agilent Technologies software will not fail to execute its programming instructions, for the period specified above, due to defects in material and workmanship when properly installed and used. If Agilent Technologies receives notice of such defects during the warranty period, Agilent Technologies will replace software media which does not execute its programming instructions due to such defects.
3. Agilent Technologies does not warrant that the operation of Agilent Technologies products will be uninterrupted or error free. If Agilent Technologies is unable, within a reasonable time, to repair or replace any product to a condition as warranted, customer will be entitled to a refund of the purchase price upon prompt return of the product.
4. Agilent Technologies products may contain remanufactured parts equivalent to new in performance or may have been subject to incidental use.
5. The warranty period begins on the date of delivery or on the date of installation if installed by Agilent Technologies. If customer schedules or delays Agilent Technologies installation more than 30 days after delivery, warranty begins on the 31st day from delivery.
6. Warranty does not apply to defects resulting from (a) improper or inadequate maintenance or calibration, (b) software, interfacing, parts or supplies not supplied by Agilent Technologies, (c) unauthorized modification or misuse, (d) operation outside of the published environmental specifications for the product, or (e) improper site preparation or maintenance.
7. TO THE EXTENT ALLOWED BY LOCAL LAW, THE ABOVE WARRANTIES ARE EXCLUSIVE AND NO OTHER WARRANTYOR CONDITION, WHETHER WRITTEN OR ORAL IS EXPRESSED OR IMPLIED AND AGILENT TECHNOLOGIES SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR CONDITIONS OR MERCHANTABILITY, SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE.
8. Agilent Technologies will be liable for damage to tangible property per incident up to the greater of $300,000 or the actual amount paid for the product that is the subject of the claim, and for damages for bodily injury or death, to the extent that all such damages are determined by a court of competent jurisdiction to have been directly caused by a defective Agilent Technologies product.

9. TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY STATEMENT ARE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES. EXCEPT AS INDICATED ABOVE, IN NO EVENT WILL AGILENT TECHNOLOGIES OR ITS SUPPLIERS BE LIABLE FOR LOSS OF DATA OR FOR DIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST PROFIT OR DATA), OR OTHER DAMAGE, WHETHER BASED IN CONTRACT, TORT, OR OTHERWISE.

   FOR CONSUMER TRANSACTIONS IN AUSTRALIA AND NEW ZEALAND: THE WARRANTY TERMS CONTAINED IN THIS STATEMENT, EXCEPT TO THE EXTENT LAWFULLY PERMITTED, DO NOT EXCLUDE RESTRICT OR MODIFY AND ARE IN ADDITION TO THE MANDATORY STATUTORY RIGHTS APPLICABLE TO THE SALE OF THIS PRODUCT TO YOU.

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.

Service and Support

Any adjustment, maintenance, or repair of this product must be performed by qualified personnel. Contact your customer engineer through your local Agilent Technologies Service Center. You can find a list of local service representatives on the Web at:

http://www.agilent-tech.com/services/English/index.html

If you do not have access to the Interned, one of these centers can direct you to your nearest representative:

United States Test and Measurement Call Center
(Toll free in US)

(800) 452-4844

Europe

(31 20) 547 9900

Canada

(905) 206-4725

Japan Measurement Assistance Center

(81) 426 56 7832
(81) 426 56 7840 (FAX)

Latin America

(305) 267 4288 (FAX)

Australia/New Zealand

1 800 629 485 (Australia)
0800 738 378 (New Zealand)

Asia-Pacific

(852) 2599 7777
(852) 2506 9285 (FAX)

Regional Sales Offices

United States of America:

Agilent Technologies (tel) 1 800 452 4844
Test and Measurement Call Center
P.O. Box 4026
Englewood, CO 80155-4026

Canada:

Agilent Technologies Canada Inc. (tel) 1 877 894 4414
2660 Matheson Blvd. E.
Mississauga, Ontario
L4W 5G1

Europe:

Agilent Technologies (tel) (3120) 547 9999
European Marketing Organization
P.O. Box 999
1180 AZ Amstelveen
The Netherlands

Japan:

Agilent Technologies Japan Ltd. (tel) (81) 456-56-7832
Measurement Assistance Center (fax) (81) 426-56-7840
9-1 Takakura-Cho, Hachioji-Shi,
Tokyo 192-8510, Japan

Latin America:

Agilent Technologies (tel) (305) 267 4245
Latin America Region Headquarters (fax) (305) 267 4286
5200 Blue Lagoon Drive,
Suite #950
Miami, Florida 33126
U.S. A.

Australia/New Zealand:

Agilent Technologies Australia Pty Ltd.	*Australia*	*New Zealand*
347 Burwood Highway	(tel) 1 800 629 485	(tel) 0 800 738 378
Forest Hill, Victoria 3131	(fax) (61 3) 9272 0749	(fax) (64 4) 802 6881

Asia Pacific:

Agilent Technologies (tel) (852) 3197 7777
24/F, Cityplaza One, (fax) (852) 2506 9233
111 Kings Road,
Taikoo Shing, Hong Kong

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.

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

Diese Information steht im Zusammenhang mit den Anforderungen der Maschinenlärminformationsverordnung vom 18 Januar 1991.

- Schalldruckpegel Lp < 70 dB(A).
- Am Arbeitsplatz.
- Normaler Betrieb.
- Nach ISO 7779:1988/EN 27779:1991 (Typprüfung).

Contents

Contents

Comprehensive Program Example

1 Programming the Agilent Technologies 8960 Series 10 for TIA/EIA-136 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 E1961A AMPS/136 mobile test application installed.

Purpose of This Programming Guide

The test capabilities of the test set 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 136 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 136 mobile station in a manufacturing environment. The set of tasks is shown in “Figure 1. Typical Flow of Tasks Performed by Control Program” on page 18.

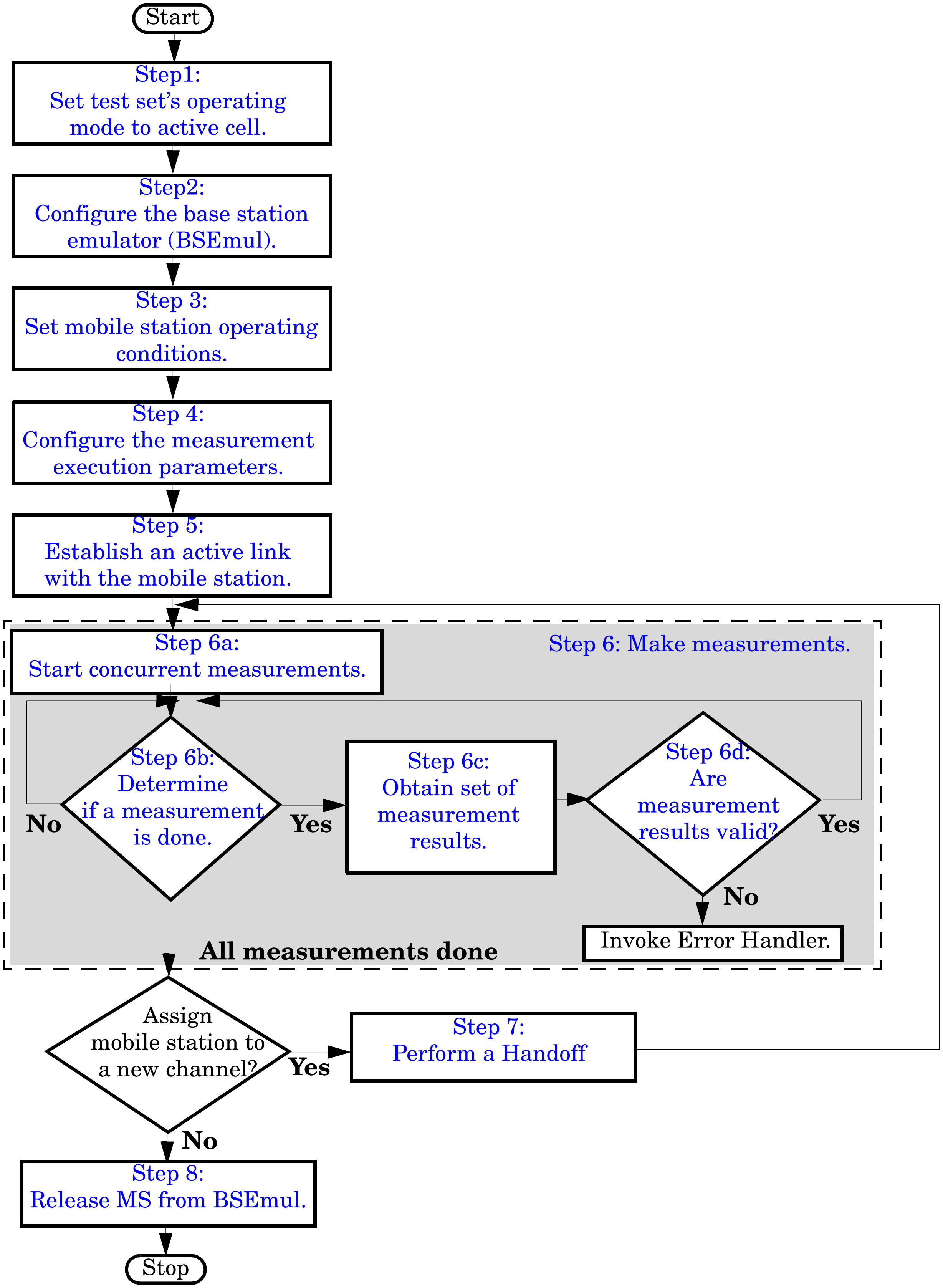
Typically in a manufacturing environment, steps 1, 2, 3, and 4 are done once each time a production run is started, steps 5 and 8 are done once for each mobile station tested during the production run, and steps 6 and 7 are done iteratively for each mobile station tested during the production run. The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

How to Use This Programming Guide

This Programming Guide is divided into 9 sections. Sections 1 through 8 (step 1 through 8) should be read in sequence. Each section discusses one of the tasks to be performed by the control program, showing 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 Program Example" on page 73 which uses all of the topics discussed in sections 1 through 8 together in one program to give the programmer a sense of how to tie everything together.

Figure 1. Typical Flow of Tasks Performed by Control Program



About the Programming Examples Presented in This Programming Guide

Programming Language:

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

Syntax Used in Programming Examples:

1. 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.

   Example 1. Command Syntax:

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

   Example 2. Shortened Form:

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

2. Programming examples do not include default nodes. Default nodes in the command syntax are defined by enclosing the node inside the [ ] brackets.

   Example 3. Command Syntax:

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

   Example 4. Command Syntax without Default Nodes:

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

3. Programming examples make extensive use of compound commands using the ; and the ;: separators. Refer to the test set's reference information for information on 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.

Example 5. Configuring a Parameter State

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

turns the state of the parameter on.

Example 6. Configuring a Parameter Value

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

assigns a value to the parameter.

Example 7. Single Complex Command to Configure a Parameter

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

would set the parameter state to ON and set 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.

Example 8. Shortened Complex Command

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

would also set the state to ON and set the value to 10 seconds. Note that in this example, the optional command mnemonic :STIMe has been left off to shorten the complex command. This is the format that will be used throughout this programming guide.

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

Background

The test set contains a TIA/EIA-136 base station emulator. The base station emulator's primary purpose is to provide the call processing necessary for parametric measurements on the RF and audio signals of a 136 mobile station.

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. The base station emulator has four operating modes; CW mode, DTC test mode, AVC test mode, and active cell mode.

Active cell mode is used when emulating a normal TIA/EIA-136 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 default operating mode of the test set's base station emulator and is used when emulating a normal TIA/EIA-136 cell (that is, 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 DCCH (Digital Control Channel).
- Generation of an ACC (Analog Control Channel).
- Generation of a DTC (Digital Traffic Channel).
- Generation of an AVC (Analog Voice Channel).
- Call setup, both mobile station and base station emulator originated.
- Changing DTC and AVC parameters during a call using over-the-air signaling.
- Base station emulator initiated and mobile station initiated call release.
- A user specified bit error ratio can be inserted into the data sent to the mobile station while on a digital traffic channel.
- All measurements supported in the test application are available.
- The base station emulator automatically controls the test set's receiver.

Setting the Test Set's Operating Mode to Active Cell

The test set's operating mode is set using the CALL:OPERating:MODE command.

Example 1. Command Syntax

```
CALL:OPERating:MODE <CALL|DTCTest|AVCTest|CW>
```

Example 2. Programming Example

```
!*****************************************************************
! Step 1: Set Test Set Operating Mode To Active Cell
!*****************************************************************
!
OUTPUT 714;"CALL:OPER:MODE CALL"
```

Step 2: Configure the Base Station Emulator

Background

In active cell operating mode the base station emulator, using the test set's modulated source, generates a forward channel (base station emulator to mobile station direction) digital control channel (DCCH) or analog control channel (ACC) which represents a cell. The mobile station can camp to the control channel signal, just as it would camp to a cell on a real network. The base station emulator can then page the mobile station on the control channel and listen to the response of the mobile station on the reverse channel (mobile station to base station emulator direction), using the test set's demodulating receiver. Calls can then be set up with the establishment of a digital traffic channel (DTC) or an analog voice channel (AVC) in both the forward and reverse directions. Measurements can be made, using the base station emulator's measuring receiver, under essentially identical conditions to those which the mobile station would experience on a real network.

The base station emulator can emulate a cell in any one of the following TIA/EIA-136 frequency bands and modes:

- Cellular Analog (800 MHz)
- Cellular Digital (800 MHz)
- PCS Digital (1900 MHz)

The task of configuring the base station emulator consists of configuring the control and traffic channels. There are numerous parameters that can be configured for both the control and traffic channels. It may not be necessary to configure all the parameters all the time. The test set's default settings should 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 all parameters to ensure that the settings have not been corrupted by someone setting a parameter's value through the test set's front panel. This will also allow the mobile to camp to the base station emulator more quickly.

Configuring the Control Channel Parameters

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

Table 1. Control Channel Settable Parameters

Parameter	Command Syntax	Footnote
Cell Power	CALL[:CELL[1]]:POWer:AMPlitude<numeric value>[<suffix>]	
Cell Power State	CALL[:CELL[1]]:POWer:STATe <ON|1|OFF|0>	
Cell Power and State	CALL[:CELL[1]]:POWer[:SAMPlitude]<numeric value>[<suffix>]	1
System Identification	CALL[:CELL]:SID <numeric value>	
Mobile Country Code	CALL[:CELL]:MCCode <numeric value>	
System Operator Code	CALL[:CELL]:SOCode <numeric value>	
Control Channel Type	CALL:CCHannel:TYPE <DCCH|ACC>	
Analog Control Channel Number	CALL:ACC[:CHANnel][:SELected] <numeric value> OR CALL:ACC[:CHANnel]:CELLular <numeric value>	2
DCCH Band	CALL:DCCH:BAND <CELLular|PCS>	3
DCCH Digital Verification Color Code	CALL:DCCH:DVCCode <numeric value>	
DCCH Burst Size	CALL:DCCH:BURSt[:SIZE] <NORMal | SHORtened>	
Digital Control Channel Number	CALL:DCCH[:CHANnel][:SELected] <numeric value> OR CALL:DCCH[:CHANnel]:<CELLular|PCS> <numeric value>	4

Table Footnotes

1 Complex command to set the cell power amplitude value and to turn the cell power state to ON in one command.

2 The selected band for analog is fixed to cellular.

3 The DCCH band setting becomes the selected band (see Note 4).

4 Sets the DCCH channel for the DCCH band selected with the DCCH Band command (see Note 3).

Example 3. Programming Example:

```
OUTPUT Test_set;"CALL:POW -50 DBM"
OUTPUT Test_set;"CALL:DCCH:BAND CELL"
OUTPUT Test_set;"CALL:DCCH 320"
```

Configuring the Traffic Channel Type

You can specify the current traffic channel type to be either a digital traffic channel or an analog traffic channel using the following command:

CALL:TCHannel:TYPE <DTC|AVC>

Configuring the Digital Traffic Channel Parameters

The digital traffic channel parameters are configured using the CALL processing subsystem commands shown in the following table.

Table 2. Digital Traffic Channel Settable Parameters

Parameter	Command Syntax	Footnotes
Digital Traffic Channel Band	CALL:SETup:DTC:BAND <CELLular|PCS>	1
Digital Traffic Channel Number	CALL:SETup:DTC[:CHANnel][:SELected] <numeric value> OR CALL:SETup:DTC[:CHANnel]:<CELLular|PCS> <numeric value>	2
Digital Traffic Channel Burst Size	CALL:DTC:BURSt[:SIZE] <NORMal|SHORtened>	3
Digital Traffic Channel Vocoder Type	CALL:SETup:DTC:VOCoder[:TYPE] <VSELp|ACELp>	
Digital Traffic Channel Timeslot	CALL:SETup:DTC:TSLot <numeric value>	
Digital Verification Color Code	CALL:SETup:DTC:DVCCode <numeric value>	
Digital Traffic Channel Induced Bit Error Rate	CALL:DTC:IBERror:RATio <numeric value>	4
Digital Traffic Channel Induced Bit Error Rate State	CALL:DTC:IBERror:STATe <ON|1|OFF|0>	4
Digital Traffic Channel Induced Bit Error Rate and State	CALL:DTC:IBERror[:SRATio] <numeric value>	4,5

Table Footnotes

1 The DTC band setting becomes the selected band (see Note 2).

2 Sets the DTC channel for the DTC band selected with the DTC Band command (see Note 1).

3 This parameter is not available during an ACC to DTC origination. Refer to the user documentation for more details.

4 This parameter is available in Active Cell operating mode when generating a DTC, or in DTC Test operating mode.

5 Complex command to set the DTC induced bit error rate and to turn the DTC induced bit error rate to ON in one command

Configuring the Analog Voice Channel Parameters

The analog voice channel parameters are configured using the CALL processing subsystem commands shown in the following table.

Table 3. Analog Voice Channel Settable Parameters

Parameter	Command Syntax	Footnotes
Analog Voice Channel Number	CALL:SETup:AVC[:CHANnel][:SELected] <numeric value> OR CALL:SETup:AVC[:CHANnel]:CELLular <numeric value>	1
SAT Frequency	CALL:SETup:AVC:SATone[:CCODE] <SAT1|SAT2|SAT3>	2
Internal FM Source State and Deviation	CALL:FM:INTernal[:SDEViation] <numeric value>[<suffix>]	2,3
Internal FM Source State	CALL:FM:INTernal:STATe <ON|1|OFF|0>	
Internal FM Source Deviation	CALL:FM:INTernal:DEViation <numeric value>[<suffix>]	2
Internal FM Source Frequency	CALL:FM:INTernal:FREQuency <numeric value>[<suffix>]	
External FM Input State	CALL:FM:EXTernal:STATe <ON|1|OFF|0>	2,4

Table Footnotes

1 The selected (:SELected) band is Cellular, the only available band for an AVC.

2 Make sure that the sum of the 3 FM sources (Internal FM, External FM, and SAT) does not exceed 20kHz deviation.

3 Complex command to set the internal FM source deviation and to turn the internal FM source state to ON in one command.

4 The FM deviation is fixed at 20kHz/volt with a 1 volt maximum input voltage.

Programming Example

The following programming example illustrates how you might setup the traffic channel parameters at the beginning of your program. First, in line 40, you would specify on which traffic channel type the call is first brought up on. In this case, it will be a digital traffic channel. This command is used again later when you want to handoff to an analog voice channel (See “Performing a Group 2 Handoff” on page 63.) Then, in lines 60 thru 90, you specify the parameters for the digital traffic channel. These include setting the band to cellular, the channel to 42, the timeslots used to 2 and 5, the vocoder to ACELP, and the induced bit error rate to 2 percent.

Finally, in lines 120 thru 140, you setup the parameters for an analog voice channel to be used later in the program.

```
10    !  Configure the traffic channel parameters
20    !
30    !  Specify the channel type
40    OUTPUT Test_set;”CALL:TCH:TYPE DTC”
50    !  Configure the DTC
60    OUTPUT Test_set;”CALL:SET:DTC:BAND CELL”
70    OUTPUT Test_set;”CALL:SET:DTC 42”
80    OUTPUT Test_set;”CALL:SET:DTC:TSL 2;VOC ACEL”
90    OUTPUT Test_set;”CALL:DTC:IBER 2”
100   !  Specify the AVC parameters to be used
110   !  when the channel type becomes AVC.
120   OUTPUT Test_set;”CALL:SET:AVC:SAT SAT1”
130   OUTPUT Test_set;”CALL:SET:AVC:CHAN 354”
140   OUTPUT Test_set;”CALL:FM:INT:STAT OFF;DEV 8KHZ;FREQ 1004HZ”
```

Step 3: Set the Mobile Station’s Operating Conditions

Background

The mobile station’s transmit level is specified by the test set in a command sent to the mobile station. In this way, the test set emulates the functionality of a 136 base station, which sends transmit level change commands to a mobile. The test set also uses the MS TX Level parameter to determine the power it should expect from the mobile station when making measurements on the mobile station’s transmitter.

The test set has two options for setting the mobile station transmit level. When a call is originated or a handoff is made to another band or channel type, the test set will send the mobile a default MS TX Level of 2. This level can be specified before any signalling occurs between the test set and the mobile station by using the CALL:SET:MS commands.

The MS TX Level can also be sent to the mobile independent of the call/handoff setup parameters. The test set immediately sends a message to the mobile station to change its power level when it receives the immediate MS TX Level command. These commands are valid only when a link is established with the mobile station.

Overview

The mobile station’s operating conditions are set up before a call origination or handoff using the CALL processing subsystem commands shown in the following table.

Table 4. Settable Mobile Station Operating Conditions

Parameter	Command Syntax	Table Footnotes
Analog Transmit Level	CALL:SETup:MS:ANALog:TXLevel[:SELected] <numeric value> OR CALL:SETup:MS:ANALog:TXLevel:CELLular <numeric value>	1
Digital Transmit Level	CALL:SETup:MS:DIGital:TXLevel[:SELected] <numeric value> OR CALL:SETup:MS:DIGital:TXLevel:<CELLular|PCS> <numeric value>	2

Table Footnotes

1 The selected band is fixed to cellular for analog.

2 The DTC band setting becomes the selected band.

Example 4. Programming Example:

```
!***********************************************************************
! Step 3: Set the Mobile Station’s Operating Conditions
!***********************************************************************
!
OUTPUT Test_set;”CALL:SET:MS:DIG:TXL:CELL 3”
```

The MS TX Level of the mobile can be changed immediately when an active link is established between the mobile station and the test set using the following commands.

Table 5. Settable Mobile Station Operating Conditions

Parameter	Command Syntax
Analog Transmit Level	CALL:MS:ANALog:TXLevel[:IMMediate]
Digital Transmit Level	CALL:MS:DIGital:TXLevel[:IMMediate]

Example 5. Programming Example:

```
OUTPUT Test_set;”CALL:MS:DIG:TXL 3”
```

Step 4: Configure the Measurement Execution Parameters

Background

Measurement execution parameters control the conditions under which a measurement operates. The general set of measurement execution parameters and their generic categories are as follows:

- Statistical Measurement Results (used by most measurements)

  Multi-Measurement Count State
  Multi-Measurement Count Number

- Measurement Timeouts (used by all measurements)

  Measurement Timeout
  Measurement Timeout State

- Measurement Triggering (used by most measurements)

  Trigger Arm (used by all measurements)
  Trigger Source (not applicable to analog measurements)
  Trigger Delay (not applicable to analog measurements)

NOTE Not all measurements use all the execution parameters shown above. Additionally, some measurements have parameters that are specific to the measurement. Each measurement has its own set of parameters that are unique to it and have no effect on the execution of other measurements. Refer to the GPIB syntax listing for a detailed list of execution parameters for individual measurements.

Overview

The SETup subsystem is used to configure 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:<meas-mnemonic>:<measurement parameter><parameter setting/value>
```

The following table shows the measurements available in the Agilent Technologies E1961A 136/AMPS mobile test application and their associated <meas-mnemonic> used in the SETup command syntax.

Table 1. Measurement Mnemonics Used In The SETup Subsystem

Measurement	<meas-mnemonic>
Adjacent Channel Power	ACPower
Analog Transmit Power	ATXPower
Audio Analyzer	AFANalyzer
Digital Transmit Power	DTXPower
Dynamic Transmit Power	DPOWer
Frequency Modulation	FM
Frequency Stability	FSTability
IQ Tuning	IQTuning
Modulation Accuracy	MACCuracy

Configuring Statistical Measurement Results Parameters

Table 2. Statistical Measurement Results Parameters

Parameter	Command Syntax
Measurement Count Number and State	`SETup:<meas-mnemonic>:COUNt[:SNUMber] <numeric value>`
Measurement Count State	`SETup:<meas-mnemonic>:COUNt:STATe <ON|1|OFF|0>`
Measurement Count Number	`SETup:<meas-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 complex command.

Example 6. Programming Example:

```
OUTPUT Test_set;"SET:DTXP:COUN 10"
```

would set the multi-measurement count state to ON and set the number of averages to 10 for the digital 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 3. Measurement Triggering Parameters

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

Trigger Arm Parameter The Trigger Arm parameter determines whether a measurement will make one measurement then stop (single), or automatically re-arm upon completion of one measurement and repeat the process (continuous).

NOTE The recommended trigger arm setting for all measurements when programming the test set is single (CONTinuous OFF).

Example 7. Programming Example:

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

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

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

NOTE The recommended trigger source setting for all measurements when programming the test set is AUTO.

Example 8. Programming Example:

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

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

NOTE Trigger Source is always IMMediate for analog measurements and cannot be changed by the user.

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 should occur prior to the trigger event.

Example 9. Programming Example:

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

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

NOTE Trigger Delay is not applicable to analog measurements.

Configuring Measurement Timeout Parameters

Table 4. 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.

Example 10. Programming Example:

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

would set the measurement timeout state to ON and set the measurement timeout time to 10 seconds for the digital transmit power measurement.

Measurement Timeout State Parameter The Measurement Timeout State parameter is used to enable or disable measurement timeout functionality.

Example 11. Programming Example:

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

would enable measurement timeouts for the modulation accuracy measurement.

Measurement Timeout Time Parameter

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

Example 12. Programming Example:

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

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

Example 13. Programming Example

```
!*********************************************************************
! Step 4: Configure Measurement Execution Parameters
!*********************************************************************
!
!  Configure Modulation Accuracy Measurement:
!
OUTPUT Test_set;"SET:MACC:COUN 5"         ! Example of using a complex
                                          ! command to set multi-meas
                                          ! state and count at same time.
OUTPUT Test_set;"SET:MACC:TRIG:SOUR AUTO"! Set trig source to Auto.
OUTPUT Test_set;"SET:MACC:CONT OFF"       ! Set trig mode to single.
OUTPUT Test_set;"SET:MACC:TIM 15"         ! Set timeout time to 15 sec.
!  Turn the EVM10 State ON
OUTPUT Test_set;"SET:MACC:EVM10:STAT ON"
!
!  Configure Digital TX Power Measurement:
!
OUTPUT Test_set;"SET:DTXP:COUN 5"
OUTPUT Test_set;"SET:DTXP:TRIG:SOUR AUTO"
OUTPUT Test_set;"SET:DTXP:CONT OFF"
OUTPUT Test_set;"SET:DTXP:TIM 5"
!
!  Configure Frequency Stability Measurement:
!
OUTPUT Test_set;"SET:FST:COUN 3"
OUTPUT Test_set;"SET:FST:CONT OFF"
OUTPUT Test_set;"SET:FST:TIM 10"
!
!  Configure Analog TX Power Measurement:
!
OUTPUT Test_set;"SET:ATXP:COUN 5"
OUTPUT Test_set;"SET:ATXP:TRIG:SOUR AUTO"
OUTPUT Test_set;"SET:ATXP:CONT OFF"
OUTPUT Test_set;"SET:ATXP:TIM 15"
!
```

Step 5: Establish an Active Link with Mobile Station

Background

Why is Call Connect/Release Synchronization Important?

When the control program requires that an active link be established/terminated between the mobile station and the test set, it must also be able to obtain knowledge about the status of that link. The control program issues the commands necessary to initiate the call connect/release process, either to the test set (for a base station originated/terminated call) or to the mobile station (for an mobile station originated/terminated call). The control program must then be able to determine when the call has successfully connected/released so that the control program can proceed. The control program must also be able to determine if the call has not been successfully connected/released so that the control program can take appropriate action.

Call States

At any instant in time a call can be in one of the following states:

- Idle
- Set Up Request*
- Alerting*
- Releasing*
- Handoff*
- Registering*
- Paging*
- Connected

The states marked by a * are referred to as transitory states because the amount of time which the call can spend in any of these states is limited by TIA/EIA-136 protocol. Therefore, the call is not allowed to stay in a transitory state forever.

NOTE If repeat paging or repeat registration are on, it is possible for the call process to stay in one of the transitory states beyond the time specified by the TIA/EIA-136 timers.

CALL:STATus[:STATe]? Query

The CALL:STATus[:STATe]? query immediately returns the state of the call at the time the query is received.

Table 6. Responses returned from the CALL:STAT:STAT? query

Response	Meaning
IDLE	The call is in the Idle state
SREQ	The call is in the Set Up Request transitory state.
ALER	The call is in the Alerting transitory state.
REL	The call is in the Releasing transitory state.
HAND	The call is in the Handoff transitory state.
REG	The call is in the Registering transitory state.
PAG	The call is in the Paging transitory state.
CONN	The call is in the Connected state.

Using the CALL:STATus[:STATe]? query for Call Connect/Release Synchronization

The most common technique used by control programs to determine if a call connect/release process has completed (either successfully or unsuccessfully), is to repeatedly query the call state using the CALL:STATus[:STATe]? query command inside a program loop. The return value from each query is checked to determine if the connect/release 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. It may also completely miss transient states so it is far less dependable and robust.
- 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.

Call Connect/Release Synchronization Commands provided by the Test Set

The test set implements a set of commands designed specifically for call connect/release synchronization. These commands directly address many of the inherent problems discussed above. When properly used these commands eliminate the need for rapid polling of the instrument, and relieve the programmer of many of the tasks associated with error handling.

Table 7. Call Connect/Release Synchronization Commands

Synchronization Command	Command Syntax
Call Connected State Query	CALL:CONNected[:STATe]?
Call State Change Detector Arm	CALL:CONNected:ARM[:IMMediate]
Call State Change Detector Time-out	CALL:CONNected:TIMeout

Call Connected State Query Command The CALL:CONNected[:STATe]? query allows the control program to determine if a call is in the Connected state or in the Idle state. There is a built in provision to automatically wait if the call is in one of the transitory states.

Table 8. Responses Returned by the CALL:CONN:STAT? query

Response	Meaning
1	The call is in the Connected state.
0	The call is in the Idle state.

This query allows the user to quickly determine if the call is connected or not and eliminates the need for rapid polling and the problems associated with that method.

NOTE If repeat paging or repeat registration are on, a call origination process could stay in one of the transitory states until the mobile either answers the page or until the user stops the paging process. This means that if a call-connected-state query command is sent to the test set, the query could hang “forever”.

Call Connected Arm Command The test set has a call-state-change detector which can be used to temporarily hold off the response to a call-connected-state query until the call origination/release process has reached the correct state. The call-connected-state query only hangs if the call is in a transitory state. Otherwise, it immediately returns a 1 or 0. Therefore, if an origination process is started and the query is satisfied before the call state has transitioned from Idle to one of the transitory states, the query response would be zero indicating that the call had failed. The call-state-change-detector can be used to hold off the query's response until the appropriate state change has occurred.

The CALL:CONNected:ARM[:IMMediate] command is used to 'arm' this call-state-change-detector.

If the call-state-change-detector is armed when a call-connected-state query is received, the reply is held off until the call-state-change-detector is disarmed. The call-state-change-detector is disarmed upon a state change from any of the transitory states to the Idle or Connected state.

The call-state-change-detector is not disarmed by a state change from Idle to any of the transitory states, from Connected to any of the transitory states, nor is it disarmed by any transitions from Idle to Idle, or Connected to Connected.

These restrictions ensure that when the call-connected-state query returns an answer:

- the connect/release process has started since the call state must have moved from either Idle or Connected to one of the transitory states

  AND

- the connect/release process has finished since the call state has moved from a transitory state to either the Idle or Connected state.

The arm state of the change detector can be queried with the call-connected-arm-state query command. This query never hangs and immediately returns a 1 if the change detector is armed and a 0 if it is disarmed. The command is:

```
CALL:CONNected:ARM:STATe?
```

Call Connected Time-out Command If the change detector is armed and a call connection is attempted but the call state never progresses from the Idle state, the call-connected-state query would hang forever. This could easily happen if the mobile is badly broken, the mobile is not connected to the test set, no one pushes the “send” button on the mobile, etc. The CALL:CONNected:TIMeout command is used to set the time-out value for the call-state-change-detector time-out timer.

Using the Call State Change Detector Time-out for Call Connect Synchronization The call-state-change-detector time-out mechanism allows the test set to disarm the call-state-change-detector which releases the call connected state query if it is currently hanging.

The time-out timer is started whenever the call-state-change-detector is armed or gets rearmed when already armed. The duration of the time-out is set using the call-connected-time-out command and should be set to the maximum amount of time the control program should wait between arming and the connect process to begin. Once the process starts and the call state has moved into one of the transitory states, the TIA/EIA-136 defined protocol timers take over and prevent the call state from staying in a transitory state forever.

If the timer expires while the call is in the Idle or Connected state, the call-state-change-detector is disarmed, which releases the call connected state query if it is currently hanging.

If the timer expires while the call is in one of the transitory states it is ignored as, once in any transitory state, the TIA/EIA-136 defined protocol timers limit the amount of time that can be spent in any transitory state.

Overview

Establishing an active link with the mobile station when the test set is in active cell operating mode can be accomplished by making a:

- Base station originated call
- Mobile station originated call

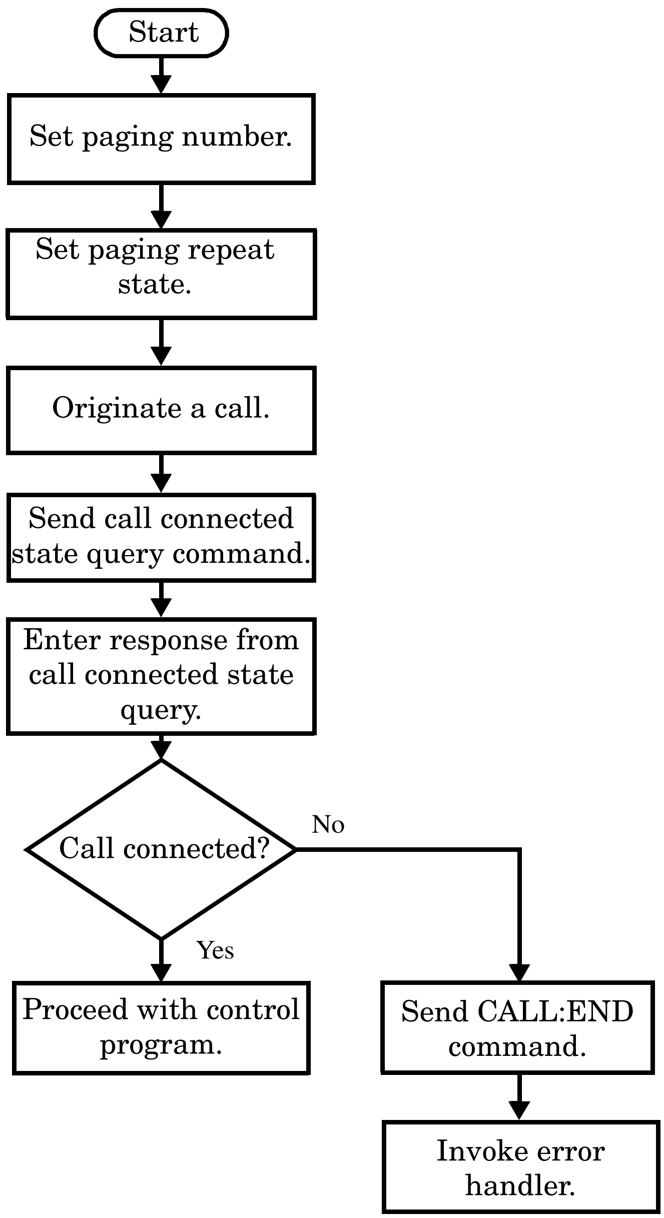
Process for Making a Base Station Originated Call

The recommended process for making a base station originated call is shown in "Figure 1. Process for Making a Base Station Originated Call" on page 44.

The CALL:ORIGinate command is used to initiate a base station originated call.

If the call origination process fails it is necessary to send the CALL:END command to the test set to force immediate termination of all processes associated with the current call origination. This ensures that if another CALL:ORIGinate command is sent to the test set before all processes associated with the failed call origination have been terminated, it will not be ignored. Note that if the test set is currently executing a call origination and it receives another call origination command it will be ignored (that is, you are telling the test set to do something it is already doing and hence it will accept the command but it will be ignored).

Figure 1. Process for Making a Base Station Originated Call



Example 9. Programming Example:

```
!*****************************************************************
! Step 5: Establish an Active Link with Mobile Station
!*****************************************************************
!
OUTPUT Test_set;"CALL:PAG:PNUM '0000574016'" ! Set paging number
OUTPUT Test_set;"CALL:PAG:REP ON" ! Set paging repeat state
OUTPUT Test_set;"CALL:ORIG" ! Start a base station originated call
OUTPUT Test_set;"CALL:CONN:STAT?"  ! Hanging GPIB query
ENTER Test_set;Call_connected      ! Program will hang here until
                                   ! origination passes or fails
IF NOT Call_connected THEN
   OUTPUT Test_set;"CALL:END"
! <put error handler here>
END IF
! Call is connected so proceed with control program
```

Call Origination Process Commands

Paging the Mobile Station Paging the mobile station is accomplished using the CALL:ORIGinate command.

Example 1. Command Syntax:

```
CALL:ORIGinate
```

Example 2. Programming Example:

```
OUTPUT Test_set;"CALL:ORIG"
```

would start the process of making a base station originated call.

Setting the Paging Number The paging number is set using the PAGing:PNUMber command.

Example 3. Command Syntax:

```
CALL:PAGing:PNUMber <string>
```

Example 4. Programming Example:

```
OUTPUT Test_set;"CALL:PAG:PNUM '0000574016'"
```

would set the paging number to 0000574016.

Setting the Paging Repeat State The paging repeat state is set using the PAGing:REPeat:STATe command.

Example 5. Command Syntax:

```
CALL:PAGing:REPeat[:STATe] <ON|1|OFF|0>
```

Example 6. Programming Example:

```
OUTPUT Test_set;"CALL:PAG:REP OFF"
```

would turn off paging repeat.

Process for Making a Mobile Station Originated Call

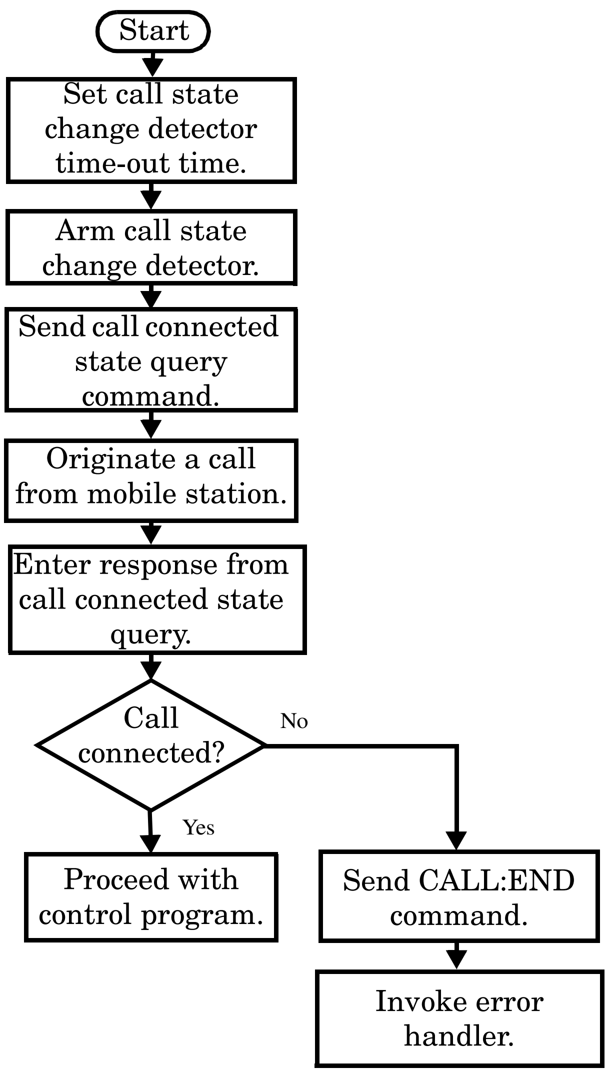
The recommended process for making a mobile station originated call is shown in "Figure 2. Process For Making A Mobile Station Originated Call" on page 46.

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

If the call origination process fails it is necessary to preset the test set to force immediate termination of all processes associated with the current call origination. This ensures that if the mobile station attempts another originate before all processes associated with the failed call origination have been terminated, it will not be ignored. Note that if the test set is currently executing a call origination and it receives another call origination command it will be ignored (that is, you are telling the test set to do something it is already doing and hence it will accept the command but it will be ignored).

For mobile station originated calls where the call is originated by physically dialing a number (as opposed to using a test bus) ensure that the call-state-change-detector time-out time is long enough to allow a human to dial the number.

Figure 2. Process For Making A Mobile Station Originated Call



Example 7. Programming Example:

```
OUTPUT Test_set;"CALL:CONN:TIM 10"         ! Set timeout time to 10 seconds
OUTPUT Test_set;"CALL:CONN:ARM"            ! Arm the change detector
DISP "Initiate a call from the mobile"
OUTPUT Test_set;"CALL:CONN:STAT?"          ! Initiate a call connect state query
ENTER Test_set;Call_connected              ! Program will hang here until
                                           ! origination passes or fails
IF NOT Call_connected THEN
   OUTPUT Test_set;"CALL:END"
! <put error handler here>
END IF
! Call is connected.  Proceed with the control program.
```

Step 6: Make Measurements

Background

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 (concurrently)
- 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

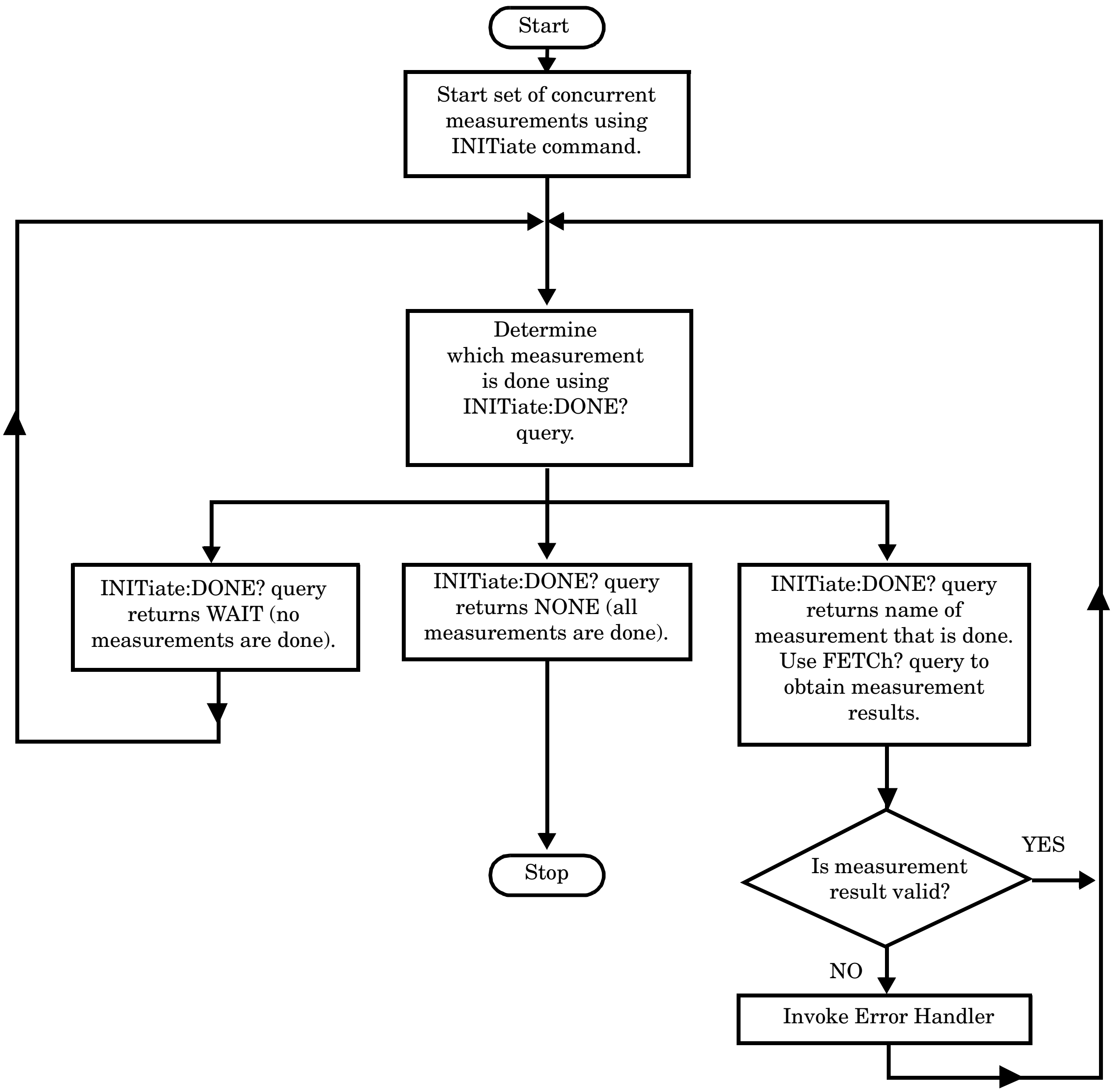
There are no special programming commands required to implement measurement concurrency.

"Step 6: Figure 1. Process for Making Concurrent Measurements" on page 50 shows the recommended process for making concurrent measurements using the test set's command set.

Making Sequential Measurements

Step 6 explains how to initiate measurements concurrently and what process to use when fetching results. These can be applied to sequential measurements. It is unnecessary to use the commands in Step 6b when performing sequential measurements.

Step 6: Figure 1. Process for Making Concurrent Measurements



Example 8. Programming Example:

The following program segment illustrates making a digital transmit power measurement and a modulation accuracy measurement concurrently using the recommended process shown in “Step 6: Figure 1. Process for Making Concurrent Measurements” on page 50.

```
!*****************************************************************
!Step 6:  Make Measurements
!*****************************************************************
!
!Step 6a:  Start Set of Concurrent Measurements
!
OUTPUT Test_set;”INIT:DTXP;MACC”
!
!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 “DTXP”
      OUTPUT Test_set;”FETC:DTXP?”
      ENTER Test_set;Integrity,Avg_dig_pow
!
!Step 6d:  Validate Measurmement Results
!
      IF Integrity=0 THEN
        PRINT “AVERAGE DIGITAL POWER = “;Avg_dig_pow
      ELSE
         GOSUB Meas_fail   !  The subroutine “Meas_fail”
                           !    is the error handler.
      END IF
!
   CASE “MACC”
      OUTPUT Test_set;”FETC:MACC:INT?;EVM10:AVER?”
      ENTER Test_set;Integrity,Avg_evm10
      IF Integrity=0 THEN
         PRINT “AVERAGE EVM 10 = “;Avg_evm10
      ELSE
         GOSUB Meas_fail
      END IF
!
   END SELECT
EXIT IF Meas_done$=”NONE”
END LOOP
```

Things That Can Go Wrong

Measurement Integrity Always Returns a Value of 6

Background A measurement integrity value of 6 indicates that some characteristic of the input signal is under range. Typically this will be the amplitude (power) of the DUT signal. This low amplitude will cause the level of the DSP sampler to be below a threshold required by the measurement algorithm to produce results of specified accuracy.

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

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

Each measurement in a test application can be started using the INITiate command. The syntax of the INITiate command is as follows:

Example 1. Command Syntax:

```
INITiate:<meas-mnemonic>[:ON]
```

The following table shows the measurements available in the Agilent Technologies E1961A AMPS/136 mobile test application and their associated <meas-mnemonic> used in the INITiate command syntax.

Table 1. Measurement Mnemonics Used In The INITiate Subsystem

Measurement	<meas-mnemonic>
Adjacent Channel Power	ACPower
Analog Transmit Power	ATXPower
Audio Analyzer	AFANalyzer
Digital Transmit Power	DTXPower
Dynamic Transmit Power	DPOWer
Frequency Modulation	FM
Frequency Stability	FSTability
IQ Tuning	IQTuning
Modulation Accuracy	MACCuracy

NOTE The Dynamic Transmit Power measurement closes all other active measurements when it is activated, therefore it should not be used concurrently with other measurements. Refer to the user documentation for more information.

Example 9. Programming Example:

```
OUTPUT Test_set;"INIT:DTXP"
```

would start the digital transmitter power measurement.

Using Compound Commands to Start Concurrent Measurements

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

```
OUTPUT Test_set;"INIT:DTXP;MACC"
```

would start the digital transmit power measurement and the modulation accuracy measurement. These measurements would then run concurrently.

Step 6b: Determine if a Measurement Is Done

Background

After a set of concurrent measurements has been started, it is desirable that the control program be able to determine when individual measurement results are available. This enables the control program to request results for a completed measurement without having to wait for other measurements to finish.

NOTE Step 6b applies only to measurements that have been initiated concurrently. When measurements have been initiated one at a time (sequential measurements) it is unnecessary to use the INITiate:DONE? query command. Step 6b can be skipped for sequential measurements.

Overview

The INITiate:DONE? query command is used to determine which measurement is finished.

As the name implies, the query returns the name of whichever active measurement is done so that the control program can request that measurement's results.

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

Once a measurement is reported as being done via the INITiate:DONE? query it is removed from the done list (measurements are only reported as being done once). The design of the INITiate:DONE? query assumes the control program immediately fetches a measurement's results once it is reported as being done.

Table 2. Responses Returned from INITiate:DONE? Query

Response	Meaning
ACP	The adjacent channel power measurement is done.
ATXP	The analog transmit power measurement is done.
AFAN	The audio analyzer measurement is done.
DPOW	The dynamic transmit power measurement is done.
DTXP	The digital transmit power measurement is done.
FM	The frequency modulation measurement is done.
FST	The frequency stability measurement is done.
IQT	The IQ Tuning measurement is done.
MACC	The modulation accuracy 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.

Example 10. Command Syntax:

```
INITiate:DONE?
```

Step 6c: Obtain a Set of Measurement Results

Background

In order to minimize bus traffic in the manufacturing environment, the test set's high-level measurements have been designed to return multiple measured values in response to a single measurement request.

For example, if a digital transmit power measurement with statistical measurement results is initiated, there will be five measurement results available as follows:

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

The test set has been designed with the capability to return the measurement results in a variety of formats to suit the needs of the measurement environment. For example, the digital 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 can be found in the test set's FETCh subsystem's GPIB command syntax reference information.

Overview

The measurement results from each measurement in a test application can be queried using the FETCh subsystem. The general hierarchy of the FETCh command structure is as follows:

```
FETCh:<meas-mnemonic>:<result format>?
```

The following table shows the measurements available in the Agilent Technologies E1961A AMPS/136 mobile test application and their associated <meas-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 3. Measurement Mnemonics Used In The FETCh Subsystem

Measurement	<meas-mnemonic>
Adjacent Channel Power	ACPower
Analog Transmit Power	ATXPower
Audio Analyzer	AFANalyzer
Digital Transmit Power	DTXPower
Dynamic Transmit Power	DPOWer
Frequency Modulation	FM
Frequency Stability	FSTability
IQ Tuning	IQTuning
Modulation Accuracy	MACCuracy

Example 1. Command Syntax:

```
FETCh:<meas-mnemonic>:<result format>?
```

Example 2. Programming Example:

```
OUTPUT Test_set;"FETC:DTXP:POW:MIN?"
```

would return the minimum value from the set of samples taken during the digital transmit power measurement (when statistical measurement results are turned on and number of samples taken >1). The returned value can be assigned to a variable using the following HP Basic command:

```
ENTER Test_set;Min_dig_pow
```

Step 6d: Validate Measurement Results

Background

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.

Overview

The integrity indicator is a measurement result and therefore is queried using the FETCh subsystem. A value of 0 indicates that the measurement is valid. A value other than 0 indicates that an error occurred during the measurement process.

Table 9. Integrity Indicators

Value Returned	Description (Message also appears on Test Set)
0	Normal
1	No Result Available
2	Measurement Timeout
3	Hardware Not Installed
4	Hardware Error
5	Over Range
6	Under Range
7	Burst Short
8	Trigger Early or Fall Early
9	Trigger Late or Rise Late
10	Signal Too Noisy
11	Sync Not Found
12	Oven Out of Range
13	Unidentified Error

Example 11. Programming Example

```
OUTPUT Test_set;"FETC:DTXP?"
ENTER Test_set;Integrity,Avg_dig_pow
IF Integrity=0 THEN
   PRINT "AVG DIG POW= ";Avg_dig_pow
ELSE
   PRINT "DTXP Measurement Error"
   PRINT "DTXP Measurement Integrity is ";Integrity
END IF
```

Step 7: Perform a Handoff

Background

A handoff is defined as assigning the mobile station to a new channel. The test set is capable of performing several types of handoffs. For simplicity, they are grouped below.

- Group 1:

  AVC to AVC
  Cellular(800MHz) DTC to Cellular(800MHz) DTC
  PCS(1900 MHz) DTC to PCS(1900 MHz) DTC

- Group 2: Mode to Mode

  AVC to Cellular(800MHz) DTC
  Cellular(800 MHz) DTC to AVC

- Group 3: Band to Band

  Cellular(800 MHz) DTC to PCS(1900 MHz) DTC
  PCS(1900 MHz) DTC to Cellular(800 MHz) DTC

- PCS(1900 MHz) DTC to AVC

NOTE A handoff from an analog voice channel in the cellular band to a digital traffic channel in the PCS(1900 MHz) band is not currently supported by the TIA/EIA-136 standard.

Overview

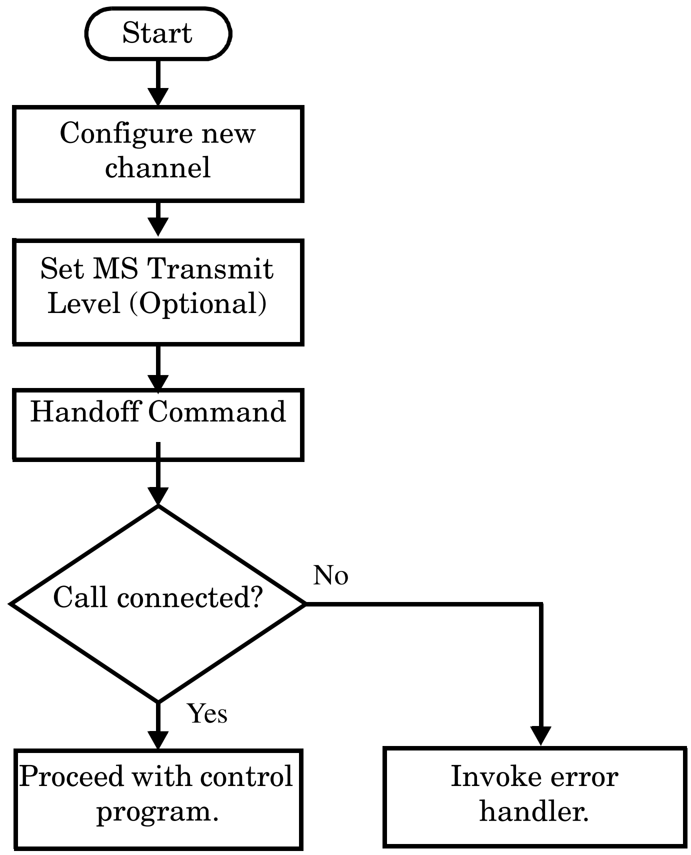
Handoffs are accomplished by using the following three steps:

1. Use base station emulator configuration commands from "Step 2: Configure the Base Station Emulator" to set up a new traffic or voice channel.
2. Use mobile station configuration commands from "Step 3: Set the Mobile Station's Operating Conditions" to set mobile station transmit levels. (Optional)
3. Issue the CALL:HANDoff[:IMMediate] command.

Performing a Group 1 Handoff

The recommended process for performing the handoffs in Group 1 is shown in the following figure.

Step 7: Figure 1. Process for Performing a Group 1 Handoff



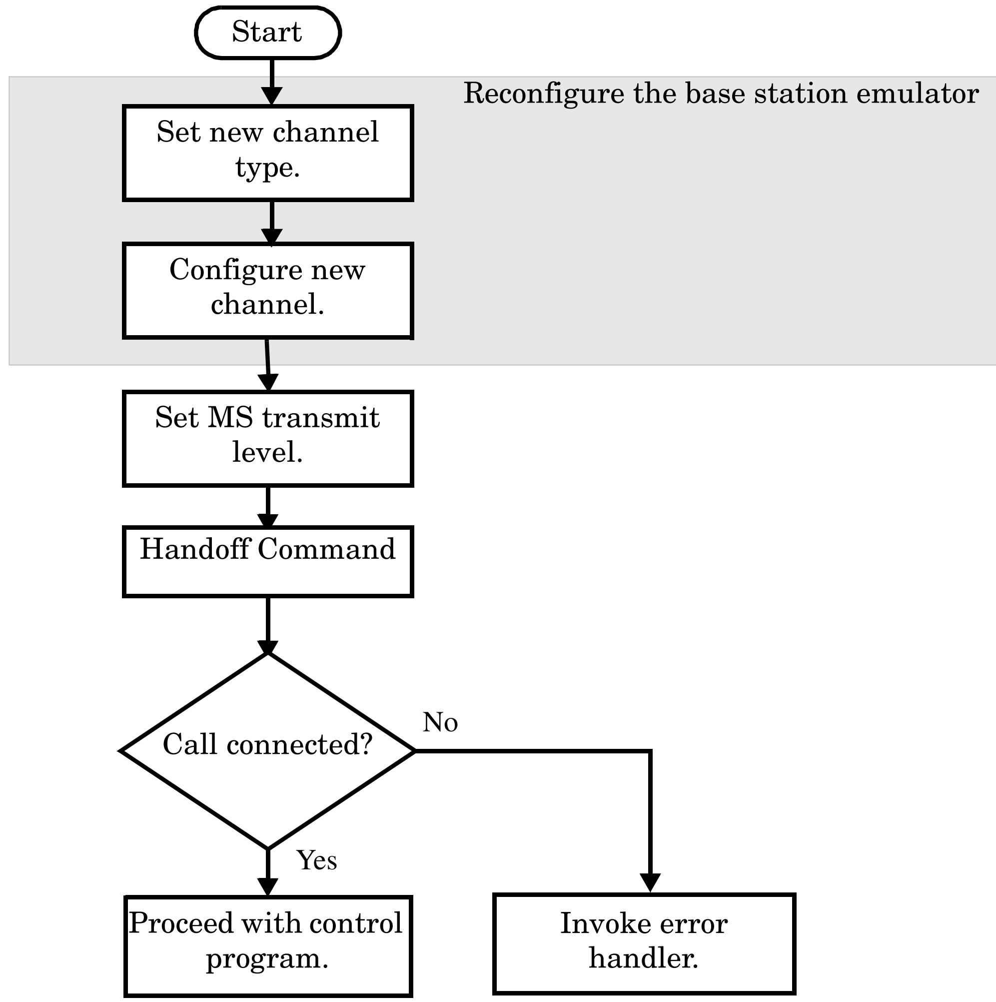
Example 12. Programming Example: Cellular DTC to Cellular DTC

```
!  Existing conditions: Mobile station is connected to the test set on
!     DTC 300 of the cellular band.
!  It is not necessary to respecify the band as cellular
!
!  Step 1:  Configure the new traffic channel
OUTPUT Test_set;"CALL:SET:DTC 556"
!  Step 2:  Configure the MS TX Level (optional)
OUTPUT Test_set;"CALL:SET:MS:DIG:TXL 4"
!  Step 3:  Issue the handoff immediate command
OUTPUT Test_set;"CALL:HAND"
!  Step 4:  Check the status of the call connection to be
!     sure the handoff completed successfully.
OUTPUT Test_set;"CALL:STAT:STAT?"
ENTER Test_set;Call_state$
IF Call_state$<>"CONN" THEN
!  Invoke the error handler
END IF
!  Handoff was successful.  Proceed with control program.
```

Performing a Group 2 Handoff

The recommended process for performing the handoffs found in Group 2 is shown in the following figure.

Step 7: Figure 2. Process for Performing a Group 2 Handoff



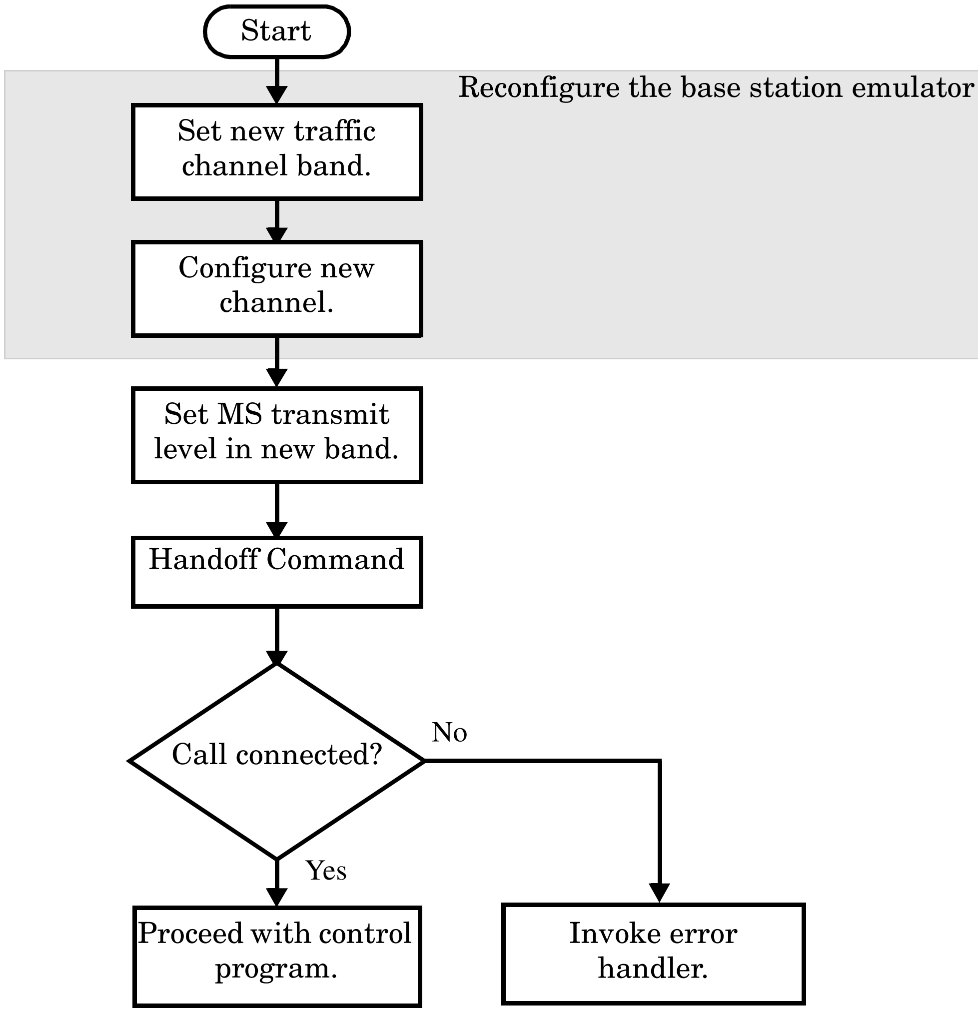
Example 13. Programming Example: Cellular DTC to AVC

```
!  Existing conditions: Mobile station is connected to the test set on
!     DTC 300 of the cellular band.
!
!  Step 1a:  Configure Channel Type to AVC
OUTPUT Test_set;"CALL:TCH:TYPE AVC"
!
!  Step 1b:  Configure the new channel
!  It is not necessary to respecify the band as cellular
!
OUTPUT Test_set;"CALL:SET:AVC 42"
!
!  Step 2:  Set new MS TX Level
OUTPUT Test_set;"CALL:SET:MS:ANAL:TXL 3"
!
!  Step 3:  Issue the handoff immediate command
OUTPUT Test_set;"CALL:HAND"
!  Step 4:  Check the status of the call connection to be
!     sure the handoff completed successfully.
OUTPUT Test_set;"CALL:STAT:STAT?"
ENTER Test_set;Call_state$
IF Call_state$<>"CONN" THEN
!  Invoke the error handler
END IF
!  Handoff was successful.  Proceed with control program.
```

Performing a Group 3 Handoff

The recommended process for performing the handoffs found in Group 3 is shown in the following figure.

Step 7: Figure 3. Process for Performing a Group 3 Handoff

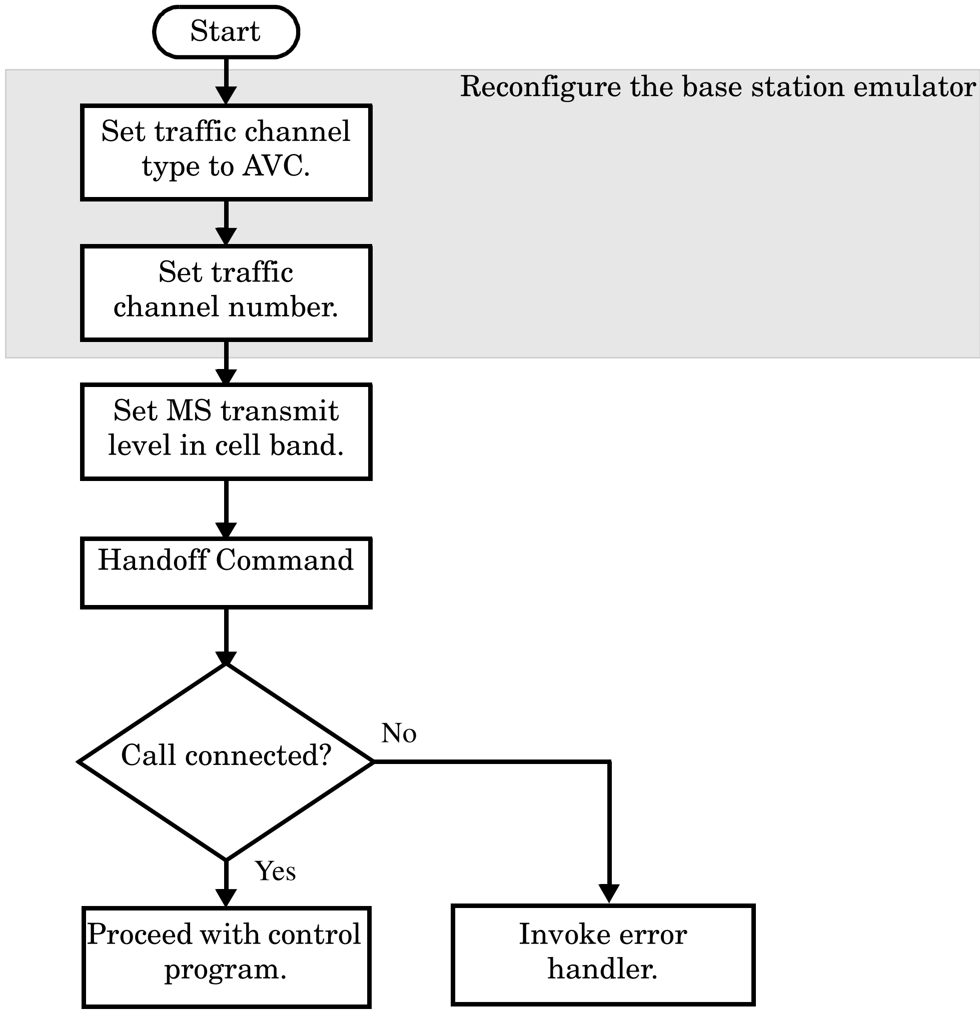


Example 14. Programming Example: PCS DTC to Cellular DTC

```
!  Existing conditions: Mobile station is connected to the test set
!     on DTC 1026 of the PCS band.
!
!  Step 1a:  Set the DTC band to cellular
OUTPUT Test_set;"CALL:SET:DTC:BAND CELL"
!  Step 1b:  Configure the new channel
OUTPUT Test_set;"CALL:SET:DTC 556"
!
!  Step 2:  Set the new MS TX level
OUTPUT Test_set;"CALL:SET:MS:DIG:TXL 3"
!
!  Step 3:  Issue the handoff immediate command
OUTPUT Test_set;"CALL:HAND"
!
!  Step 4:  Check the status of the call connection to be
!     sure the handoff completed successfully.
OUTPUT Test_set;"CALL:STAT:STAT?"
ENTER Test_set;Call_state$
IF Call_state$<>"CONN" THEN
!  Invoke the error handler
END IF
!  Handoff was successful.  Proceed with control program.
```

Performing a PCS DTC to AVC Handoff

Step 7: Figure 4. Process for Performing a PCS DTC to AVC Handoff



Example 15. Programming Example: PCS DTC to AVC

```
!  Existing conditions: Mobile station is connected to the test set on
!     DTC 1026 of the PCS band.
!
!  Step 1a:  Change the traffic channel type to AVC
OUTPUT Test_set;"CALL:TCH:TYPE AVC"
!  Step 1b:  Configure the new channel
OUTPUT Test_set;"CALL:SET:AVC:CELL 42"
!
!  Step 2:  Set a new MS TX Level
OUTPUT Test_set;"CALL:SET:MS:ANAL:TXL 3"
!
!  Step 3:  Issue the handoff immediate command
OUTPUT Test_set;"CALL:HAND"
!
!  Step 4:  Check the status of the call connection to be
!     sure the handoff completed successfully.
OUTPUT Test_set;"CALL:STAT:STAT?"
ENTER Test_set;Call_state$
IF Call_state$<>"CONN" THEN
PRINT "HANDOFF FAILED"
!  Invoke the error handler
END IF
!  Handoff was successful.  Proceed with the control program.
```

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

Background

See "Step 5: Establish an Active Link with Mobile Station" on page 39 for a discussion of call connect/disconnect synchronization.

Overview

Terminating an active call with the mobile station when the test set is in active cell operating mode can be accomplished in one of two ways:

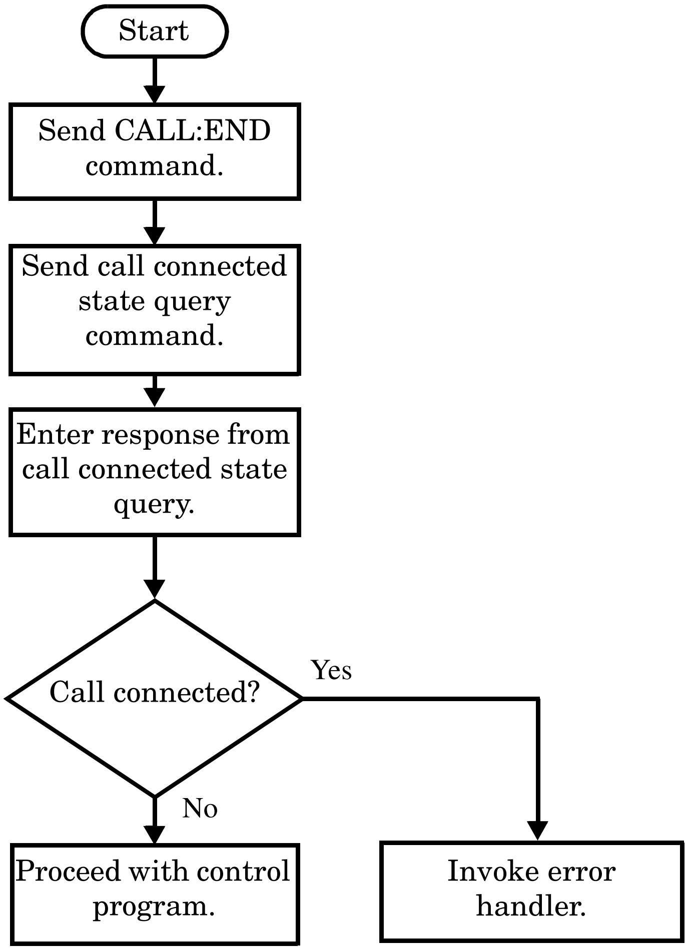
- Terminate the active call from the base station emulator
- Terminate the active call from the mobile station

Terminating an Active Call from the Base Station Emulator

The recommended process for terminating an active call from the base station emulator is shown in the following figure.

The CALL:END command is used to initiate a base station disconnect.

Step 8: Figure 1. Process for Terminating an Active Call from the Base Station Emulator



Example 10. Programming Example:

```
!*********************************************************************
! Step 8: Disconnect Mobile Station from Base Station Emulator
!*********************************************************************
!
OUTPUT Test_set;"CALL:END"        ! Initiate a base station disconnect.
OUTPUT Test_set;"CALL:CONN:STAT?" ! Initiate call connect state query.
ENTER Test_set;Call_connected     ! Program will hang here until state
                                  ! change or timer expires.
IF Call_connected THEN            ! Check if disconnect successful
! <put error handler here>
END IF
! Call is disconnected so proceed with control program
```

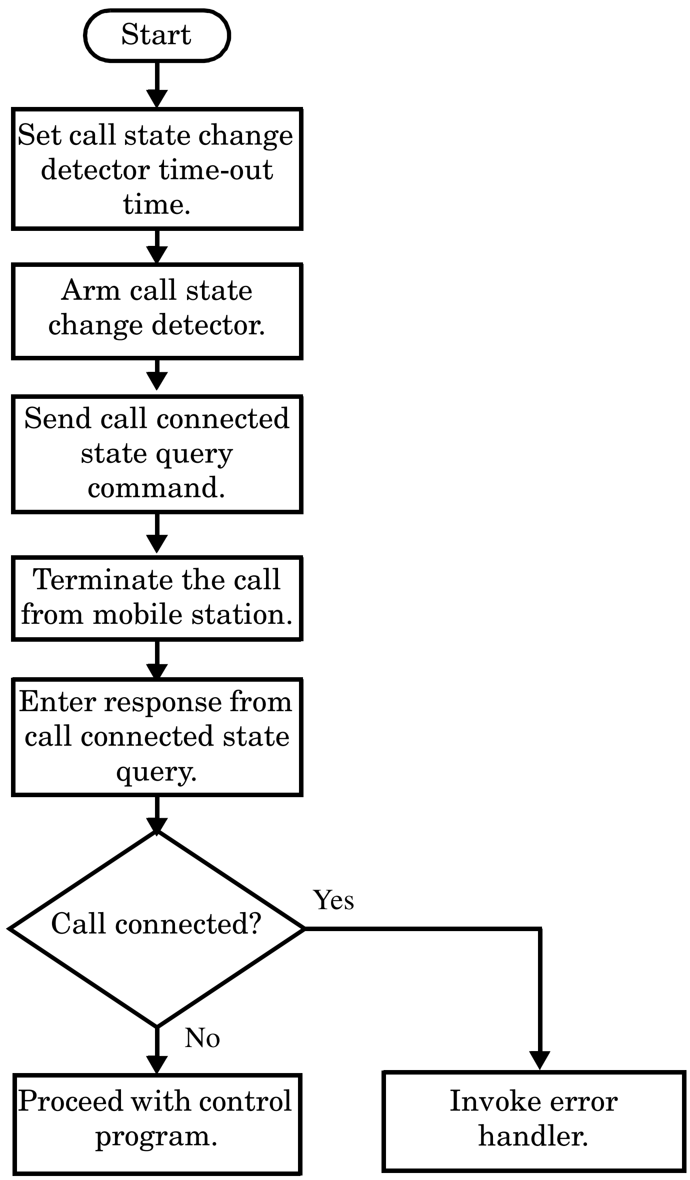
Terminating an Active Call from the Mobile Station

The process for terminating an active call from the mobile station is shown in the following figure.

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

For mobile station terminated calls where the call is terminated by physically pushing a button on the phone (as opposed to using a test bus) ensure that the call-state-change-detector time-out time is long enough to allow a human to push the button.

Step 8: Figure 2. Process for Terminating an Active Call from the Mobile Station



Example 11. Programming Example:

```
OUTPUT Test_set;"CALL:CONN:TIM 5" !Set timeout time to 5 seconds.
OUTPUT Test_set;"CALL:CONN:ARM"   !Arm the change detector.

DISP "Terminate the call from the mobile station."
OUTPUT Test_set;"CALL:CONN:STAT?" !Initiate call connect state query.
ENTER Test_set;Call_connected     !Program will hang here until state
                                  !change or timer expires.
IF Call_connected THEN            !Check if disconnect successful.
! <put error handler here>
END IF
! Call is disconnected so proceed with control program
```

Comprehensive Program Example

This section presents two example programs for making measurements using the test set in active cell mode. The first program follows the task flow presented at the beginning of this programming guide and illustrates how each step presented throughout the guide fits together into a single program. The second program is an example of more efficient programming techniques.

Example Program 1

```
10    !***************************************************************
20    !   Program:  COMP_EX.TXT                           October 9,2000
30    !   Purpose:  Comprehensive Example of programming measurements on
40    !             the Agilent Technologies 8960 using Active Cell
50    !             operating mode.
60    !             This example is written in HP BASIC.
70    !***************************************************************
80    !
90    !   Configure the HP BASIC environment.
100   !   These actions are unrelated to configuring or performing tests
110   !      with the Agilent Technologies 8960.
120   !
130   OPTION BASE 1
140   Test_set=714!  Test set’s GPIB address
150   PRINTER IS CRT
160   CLEAR SCREEN
170   !
180   !   Reset the test set to start from a known state.
190   !   It is not always necessary to do a full preset in a
200   !   manufacturing environment, but is useful in a programming
210   !   example.
220   !
230   OUTPUT Test_set;”*RST”
240   !
250   !  Set a GPIB timeout.  This is for program development purposes
260   !  only.  Programming mistakes may cause the GPIB bus to hang and
270   !  not allow HP BASIC to be reset.
280   !
290   ON TIMEOUT 7,30 GOTO Timeout_routine
300   !
310   !   Turn on the GPIB debugger.  This is for program development
320   !   purposes only and should be removed once the program is
330   !   completed.
340   OUTPUT Test_set;”SYST:COMM:GPIB:DEB ON”
350   !***************************************************************
360   !Step 1:  Set the test set’s Operating Mode to Active Cell
370   !***************************************************************
380   !
390   OUTPUT Test_set;”CALL:OPER:MODE CALL”
400   !
410   !***************************************************************
420   !Step 2:  Configure the base station emulator (BSEmul)
430   !***************************************************************
440   !
450   !  The default traffic channel type is DTC,
460   !  so we will not specify it here.
470   !
480   ! Configure the Digital Traffic Channel parameters
490   OUTPUT Test_set;”CALL:SET:DTC:BAND CELL”! Set DTC Band to Cell.
500   OUTPUT Test_set;”CALL:SET:DTC 1000”     ! Set DTC to 1000.
510   !
```

```
520   !  Configure the Analog Voice Channel parameters
530   OUTPUT Test_set;”CALL:SET:AVC 1000”
540   OUTPUT Test_set;”CALL:SET:AVC:SAT SAT1”
550   !*****************************************************************
560   !Step 3:  Set the Mobile Station’s Operating Conditions
570   !*****************************************************************
580   OUTPUT Test_set;”CALL:SET:MS:DIG:TXL 4”
590   !
600   !*****************************************************************
610   !Step 4:  Configure measurement execution parameters
620   !*****************************************************************
630   !
640   !  Set trigger to single for all measurements
650   OUTPUT Test_set;”SET:CONT:OFF”
660   !
670   !  Configure Digital Measurements
680   OUTPUT Test_set;”SET:DTXP:COUN 2;TIM 5”
690   OUTPUT Test_set;”SET:MACC:EVM10 ON;COUN 1;TIM 10”
700   OUTPUT Test_set;”SET:ACP:COUN 3;TIM 5”
710   !
720   !  Configure Analog Measurements
730   OUTPUT Test_set;”SET:ATXP:COUN 2;TIM 5”
740   !
750   !*****************************************************************
760   !Step 5:  Establish an active link with the MS
770   !*****************************************************************
780   !
790   !   This example will use an BS originated call.  The phone must
800   !   be camped to the BSEmul before a call can be originated.  The
810   !   following code will use repeat paging to originate the call.
820   !   The test set will repeatedly page the mobile until it has
830   !   successfully camped and accepts the page.
840   !
850   OUTPUT Test_set;”CALL:PAG:PNUM ‘0002007344’” ! The paging number
860   OUTPUT Test_set;”CALL:PAG:REP ON” ! Repeat paging is turned on
870   OUTPUT Test_set;”CALL:ORIG”       ! The test set begins paging
880                                     ! the mobile station.
890   !
900   OUTPUT Test_set;”CALL:CONN:STAT?” ! Hanging query
910   ENTER Test_set;Call_connected     ! Program will hang here until
920                                     ! origination process completes.
930                                     ! If successful, returns 1.
940                                     ! If call does not connect,
950                                     ! returns 0.
960   IF NOT Call_connected THEN
970     OUTPUT Test_set;”CALL:END”
980     PRINT “Call was not able to connect.  Program terminated.”
990     STOP
1000  END IF
1010  !
1020  !*****************************************************************
1030  !Step 6:  Make Measurements
1040  !*****************************************************************
```

```
1050  CLEAR SCREEN
1060  PRINT “800 MHz DTC Measurement Results: “
1070  PRINT “********CH 1000**********”
1080  !
1090  !Step 6a:  Start Set of Concurrent Measurements
1100  !
1110  OUTPUT Test_set;”INIT:DTXP;MACC”
1120  !
1130  !Step 6b:  Determine if a Measurement is Done
1140  !
1150  LOOP
1160  OUTPUT Test_set;”INIT:DONE?”
1170  ENTER Test_set;Meas_done$
1180  !
1190  !Step 6c:  Obtain Measurement Results
1200  !
1210  SELECT Meas_done$
1220  !
1230  CASE “DTXP”
1240     OUTPUT Test_set;”FETC:DTXP?”
1250     ENTER Test_set;Integrity,Avg_dig_pow
1260  !
1270  !Step 6d:  Validate Measurmement Results
1280  !
1290     IF Integrity=0 THEN
1300        PRINT
1310        PRINT “AVERAGE DIGITAL POWER = “;Avg_dig_pow
1320     ELSE
1330        GOSUB Meas_fail  !  The subroutine “Meas_fail”
1340                         !    is the error handler.
1350     END IF
1360  !
1370  CASE “MACC”
1380     OUTPUT Test_set;”FETC:MACC:INT?;EVM10:AVER?”
1390     ENTER Test_set;Integrity,Avg_evm10
1400     OUTPUT Test_set;”FETC:MACC:EVM?”
1410     ENTER Test_set;Max_evm
1420     IF Integrity=0 THEN
1430        PRINT
1440        PRINT “AVERAGE EVM 10 = “;Avg_evm10
1450        PRINT “MAXIMUM EVM    = “;Max_evm
1460     ELSE
1470        GOSUB Meas_fail
1480     END IF
1490  !
1500  END SELECT
1510  EXIT IF Meas_done$=”NONE”
1520  END LOOP !  If ‘WAIT’ is returned from the ‘INIT:DONE?’ query, it
1530           !  will fall through the loop.
1540  !
1550  !*****************************************************************
1560  !Step 7: Perform a Group 1 Handoff
1570  !*****************************************************************
```

```
1580  !
1590  !  Handoff to another DTC in the 800 MHz band
1600  !
1610  !  Set up the handoff.
1620  OUTPUT Test_set;”CALL:SET:DTC 300”
1630  OUTPUT Test_set;”CALL:SET:MS:DIG:TXL 2”
1640  !
1650  !  Initiate the handoff.
1660  OUTPUT Test_set;”CALL:HAND”
1670  !
1680  !  Verify that the handoff completed successfully
1690  OUTPUT Test_set;”CALL:STAT:STAT?”
1700  ENTER Test_set;Call_state$
1710  IF Call_state$<>”CONN” THEN
1720     PRINT “HANDOFF FAILED”
1730     STOP
1740  END IF
1750  !
1760  !*****************************************************************
1770  !Step 6(REPEATED):  Make Measurements on the New Channel
1780  !*****************************************************************
1790  PRINT
1800  PRINT “********CH 300********”
1810  OUTPUT Test_set;”INIT:DTXP;MACC;ACP”
1820  LOOP
1830  OUTPUT Test_set;”INIT:DONE?”
1840  ENTER Test_set;Meas_done$
1850  SELECT Meas_done$
1860  !
1870  CASE “DTXP”
1880     OUTPUT Test_set;”FETC:DTXP?”
1890     ENTER Test_set;Integrity,Avg_dig_pow
1900     IF Integrity=0 THEN
1910        PRINT
1920        PRINT “AVERAGE DIGITAL POWER = “;Avg_dig_pow
1930     ELSE
1940        GOSUB Meas_fail  !  The subroutine “Meas_fail”
1950                         !    is the error handler.
1960     END IF
1970  !
1980  CASE “MACC”
1990     OUTPUT Test_set;”FETC:MACC:INT?;EVM10:AVER?”
2000     ENTER Test_set;Integrity,Avg_evm10
2010     OUTPUT Test_set;”FETC:MACC:EVM?”
2020     ENTER Test_set;Max_evm
2030     IF Integrity=0 THEN
2040        PRINT
2050        PRINT “AVERAGE EVM 10 = “;Avg_evm10
2060        PRINT “MAXIMUM EVM    = “;Max_evm
2070     ELSE
2080        GOSUB Meas_fail
2090     END IF
2100  !
```

```
2110  CASE “ACP”
2120     OUTPUT Test_set;”FETC:ACP?”
2130     ENTER Test_set;Integrity,L_adj,U_adj,L1_alt,U1_alt,L2_alt,U2_alt
2140     IF Integrity=0 THEN
2150        PRINT
2160        PRINT “AVERAGE LOWER ADJACENT POWER =”;L_adj
2170        PRINT “AVERAGE UPPER ADJACENT POWER =”;U_adj
2180        PRINT “AVERAGE LOWER 1ST ALTERNATE POWER =”;L1_alt
2190        PRINT “AVERAGE UPPER 1ST ALTERNATE POWER =”;U1_alt
2200        PRINT “AVERAGE LOWER 2ND ALTERNATE POWER =”;L2_alt
2210        PRINT “AVERAGE UPPER 2ND ALTERNATE POWER =”;U2_alt
2220        PRINT
2230     ELSE
2240        GOSUB Meas_fail
2250     END IF
2260  !
2270  END SELECT
2280  EXIT IF Meas_done$=”NONE”
2290  END LOOP
2300  !
2310  !****************************************************************
2320  !Step 7: Perform a Group 2 Handoff
2330  !****************************************************************
2340  !
2350  OUTPUT Test_set;”CALL:TCH:TYPE AVC”
2360  OUTPUT Test_set;”CALL:HAND”
2370  OUTPUT Test_set;”CALL:STAT:STAT?”
2380  ENTER Test_set;Call_state$
2390  IF Call_state$<>”CONN” THEN
2400     PRINT “HANDOFF FAILED”
2410     STOP
2420  END IF
2430  !
2440  !****************************************************************
2450  !Step 6(REPEATED):  Make Analog Measurements
2460  !****************************************************************
2470  !
2480  PRINT “800 MHz AVC Measurement Results: “
2490  PRINT “********CH 1000**********”
2500  !
2510  OUTPUT Test_set;”INIT:ATXP”
2520  OUTPUT Test_set;”FETC:ATXP?”
2530  ENTER Test_set;Integrity,Avg_anal_pow
2540  IF Integrity=0 THEN
2550     PRINT
2560     PRINT “AVERAGE ANALOG TRANSMIT POWER = “;Avg_anal_pow
2570  ELSE
2580     GOSUB Meas_fail
2590  END IF
2600  !
2610  !****************************************************************
2620  !Step 8:  Disconnect the mobile from the BSEmul
2630  !****************************************************************
```

```
2640  !
2650  OUTPUT Test_set;"CALL:END"
2660  !  Verify that the call released correctly
2670  OUTPUT Test_set;"CALL:CONN:STAT?"
2680  ENTER Test_set;Call_conn
2690  IF Call_conn THEN
2700     PRINT "UNABLE TO DISCONNECT PROPERLY"
2710     STOP
2720  END IF
2730  PRINT
2740  PRINT "TESTING COMPLETED SUCCESSFULLY"
2750  STOP
2760 Meas_fail:   !  Subroutine to handle measurement errors
2770   PRINT
2780   PRINT "Measurement error:  "&Meas_done$
2790   PRINT "Integrity Indicator = ";Integrity
2800   RETURN
2810 Timeout_routine: !
2820  PRINT "GPIB TIMEOUT HAS OCCURRED."
2830  END ! End of program
```

Example Program 2

```
10     !***************************************************************
20     ! Comprehensive Example Program
30     ! 8960 AMPS/136 Call Mode
40     !
50     ! This is a program to demonstrate the use of Momentum
60     ! with E1961A AMPS/136 TA testing a dual-mode TDMA phone.
70     ! The program makes Digital Cellular, Digital PCS, and Analog
80     ! measurements while on a call.
90     !***************************************************************
100    !
110    COM /Address/ INTEGER Testset
120    Testset=714
130    ON TIMEOUT 7,20 GOSUB Timeout
140    Rf_level=-75
150    Maho_level=-110
160    Sinad_level=-116
170    CLEAR SCREEN
180    !
190    !***************************************************************
200    ! Configure Testset Parameters
210    !***************************************************************
220    !
230    OUTPUT Testset;”*RST”
240    OUTPUT Testset;”SYST:COMM:GPIB:DEB:STAT ON”
250    ! Remove afer development is finished.
260    !
270    !***************************************************************
280    ! Configure Base Station Emulator Parameters
290    !***************************************************************
300    !
310    OUTPUT Testset;”CALL:CELL:OPER:MODE CALL”
320    OUTPUT Testset;”CALL:POW “;Rf_level
330    OUTPUT Testset;”CALL:CCH:TYPE DCCH”
340    OUTPUT Testset;”CALL:DCCH:BAND CELL”
350    OUTPUT Testset;”CALL:DCCH 1013”
360    OUTPUT Testset;”CALL:SET:AVC 387”
370    OUTPUT Testset;”CALL:SET:DTC:BAND CELL”
380    OUTPUT Testset;”CALL:SET:DTC:CHAN:CELL 542”
390    OUTPUT Testset;”CALL:SET:DTC:CHAN:PCS 1000”
400    OUTPUT Testset;”CALL:TCH:TYPE DTC”
410    OUTPUT Testset;”CALL:CONN:TIM 15”
420    OUTPUT Testset;”CALL:SET:MS:ANAL:TXL 2”
430    OUTPUT Testset;”CALL:SET:MS:DIG:TXL:CELL 2”
440    OUTPUT Testset;”CALL:SET:MS:DIG:TXL:PCS 2”
450    OUTPUT Testset;”CALL:FM:INT:STATE OFF;DEV 8KHZ;FREQ 1004HZ”
460    !
470    !***************************************************************
480    ! Configure Measurement Parameters
490    !***************************************************************
500    !
510    OUTPUT Testset;”SET:CONT:OFF”
```

```
520   OUTPUT Testset;”SET:DTXP:TIM 3”
530   OUTPUT Testset;”SET:MACC:TIM 3”
540   OUTPUT Testset;”SET:ACP:TIM 3”
550   OUTPUT Testset;”SET:FST:TIM 3”
560   OUTPUT Testset;”SET:ATXP:TIM 3”
570   OUTPUT Testset;”SET:AFAN:TIM 3”
580   OUTPUT Testset;”SET:FM:TIM 3”
590   OUTPUT Testset;”SET:MACC:EVM10:STAT OFF”
600   OUTPUT Testset;”SET:AFAN:PEAK:VOLT 1”
610   OUTPUT Testset;”SET:AFAN:SDIS:STAT ON”
620   OUTPUT Testset;”SET:AFAN:FILT CMES”
630   OUTPUT Testset;”SET:AFAN:SDIS:STATE ON;FREQ 1004”
640   OUTPUT Testset;”SET:FM:DIST:STATE ON;FREQ 6000”
650   OUTPUT Testset;”SET:FM:DETECTOR:TYPE PPEAK”
660   OUTPUT Testset;”SET:FM:FILT:TBP 6000”
670   OUTPUT Testset;”SET:FM:FILT TBP”
680   !
690   !****************************************************************
700   ! Establish Call with Mobile
710   !****************************************************************
720   !
730   PRINT “Turn the phone on now.”
740   PRINT “When the phone camps on DCCH 1013, press F2 to continue.”
750   PAUSE
760   CLEAR SCREEN
770   PRINT “Originate a call on the Mobile now.”
780   OUTPUT Testset;”CALL:CONN:ARM” !Arm Call-State-Change Detector
790   OUTPUT Testset;”CALL:CONN?” !Query State
800   ENTER Testset;Callstate
810   IF NOT Callstate THEN
820     Orig_failed
830   END IF
840   !
850   CLEAR SCREEN
860   PRINT “Cellular Results: “
870   Time=TIMEDATE
880   !
890   !****************************************************************
900   ! Make Measurements (Digital Cellular and PCS)
910   !****************************************************************
920   !
930   FOR I=1 TO 2
940     OUTPUT Testset;”CALL:POW “;Maho_level
950     OUTPUT Testset;”CALL:MS:REP:MAHO ON”
960     !
970     OUTPUT Testset;”INIT:DTXP;MACC;ACP”
980     REPEAT
990       OUTPUT Testset;”INIT:DONE?”
1000      ENTER Testset;Measdone$
1010      SELECT Measdone$
1020      CASE “DTXP”
1030        OUTPUT Testset;”FETC:DTXP?”
1040        ENTER Testset;Integrity,Power
```

```
1050        IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1060        Print_results(Measdone$,Power)
1070      CASE “MACC”
1080        OUTPUT Testset;”FETC:MACC?”
1090        ENTER Testset;Integrity,Evm,Ferr,Ooff,Perr,Mag
1100        IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1110        Print_results(Measdone$,Evm,Ferr,Ooff,Perr,Mag)
1120      CASE “ACP”
1130        OUTPUT Testset;”FETC:ACP?”
1140        ENTER Testset;Integrity,Adjl,Adjh,Alt1l,Alt1h,Alt2l,Alt2h
1150        IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1160        Print_results(Measdone$,Adjl,Adjh,Alt1l,Alt1h,Alt2l,Alt2h)
1170      END SELECT
1180    UNTIL Measdone$=”NONE”
1190    !
1200    OUTPUT Testset;”CALL:MS:REP:MAHO:BERR:NEW?” !Get NEXT MAHO Report
1210    ENTER Testset;Ber$
1220    OUTPUT Testset;”CALL:MS:REP:MAHO:RSSI?”
1230    ENTER Testset;Rssi
1240    Print_maho(Ber$,Rssi,Maho_level)
1250    OUTPUT Testset;”CALL:MS:REP:MAHO OFF”
1260    OUTPUT Testset;”CALL:POW “;Rf_level
1270  !
1280  !****************************************************************
1290  ! Make Test Set Changes (Change to PCS, then Analog)
1300  !****************************************************************
1310  !
1320    IF I=1 THEN ! Just did Digital 800, get ready for Digital 1900
1330      OUTPUT Testset;”CALL:SET:DTC:BAND PCS”
1340      PRINT “”
1350      PRINT “PCS Results:”
1360    ELSE   ! Done doing Digital 1900, get ready for Analog
1370      OUTPUT Testset;”CALL:TCH:TYPE AVC”
1380      PRINT “”
1390      PRINT “Analog Results:”
1400    END IF
1410  !
1420  !****************************************************************
1430  ! Perform handoff to new channel
1440  !****************************************************************
1450  !
1460    OUTPUT Testset;”CALL:HAND”
1470    OUTPUT Testset;”CALL:STAT:STAT?”
1480    ENTER Testset;Connected$
1490    IF Connected$<>”CONN” THEN
1500      Dropped_call
1510    END IF
1520  NEXT I
1530  !
1540  !****************************************************************
1550  ! Make Measurements (Analog)
1560  !****************************************************************
1570  !
```

```
1580  OUTPUT Testset;”CALL:FM:INT:STATE ON”
1590  OUTPUT Testset;”CALL:POW “;Sinad_level
1600  OUTPUT Testset;”INIT:ATXP;FST;AFAN;FM”
1610  REPEAT
1620    OUTPUT Testset;”INIT:DONE?”
1630    ENTER Testset;Measdone$
1640    SELECT Measdone$
1650    CASE “ATXP”
1660      OUTPUT Testset;”FETC:ATXP?”
1670      ENTER Testset;Integrity,Power
1680      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1690      Print_results(Measdone$,Power)
1700    CASE “FST”
1710      OUTPUT Testset;”FETC:FST?”
1720      ENTER Testset;Integrity,Ferr,Freq
1730      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1740      Print_results(Measdone$,Ferr,Freq)
1750    CASE “AFAN”
1760      OUTPUT Testset;”FETC:AFAN?”
1770      ENTER Testset;Integrity,Level,Sinad,Dist
1780      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1790      Print_results(Measdone$,Level,Dist,Sinad)
1800    CASE “FM”
1810      OUTPUT Testset;”FETC:FM?”
1820      ENTER Testset;Integrity,Dev,Dist
1830      IF Integrity<>0 THEN CALL Bad_measurement(Integrity,Measdone$)
1840      Print_results(Measdone$,Dev,Dist)
1850    END SELECT
1860  UNTIL Measdone$=”NONE”
1870  !
1880  !****************************************************************
1890  ! Finish Testing, Release Call
1900  !****************************************************************
1910  !
1920  PRINT ““
1930  PRINT “Testing Complete.”
1940  Testtime=PROUND(TIMEDATE-Time,-2)
1950  PRINT ““
1960  PRINT “Test time was “;Testtime;” seconds.”
1970  OUTPUT Testset;”CALL:POW “;Rf_level
1980  OUTPUT Testset;”CALL:END”
1990  OUTPUT Testset;”CALL:CONN:STAT?”
2000  ENTER Testset;Callstate
2010  IF Callstate=1 THEN
2020    PRINT “Make sure the phone has released the call.”
2030    OUTPUT Testset;”SYST:PRES3”
2040  END IF
2050  STOP
2060 Timeout:  !Comes here only when program times out
2070  PRINT “Program time out.”
2080  END
2090  !
2100  !*****************************************************************
```

```
2110  ! Subroutine Section
2120  !*****************************************************************
2130  !
2140  SUB Print_results(Meas_name$,Res1,OPTIONAL Res2,Res3,Res4,Res5,Res6)
2150    SELECT Meas_name$
2160    CASE “DTXP”
2170      PRINT USING “5X,””Ave Digital Power:””,5X,M2D.2D,”” dBm”””;Res1
2180    CASE “MACC”
2190      PRINT USING “5X,””Max EVM1:””,14X,M2D.2D,”” %”””;Res1
2200      PRINT USING “5X,””Worst Freq Error:””,5X,M3D.2D,”” Hz”””;Res2
2210      PRINT USING “5X,””Max Mag. Error:””,8X,M2D.2D,”” %”””;Res5
2220      PRINT USING “5X,””Max Origin Offset:””,5X,M2D.2D,”” dB”””;Res3
2230      PRINT USING “5X,””Max Phase Error:””,7X,M2D.2D,”” Deg”””;Res4
2240    CASE “ACP”
2250      PRINT USING “5X,””ACP Adj Lo:””,12X,M2D.2D,”” dBc”””;Res1
2260      PRINT USING “5X,””ACP Adj Hi:””,12X,M2D.2D,”” dBc”””;Res2
2270      PRINT USING “5X,””ACP Alt1 Lo:””,11X,M2D.2D,”” dBc”””;Res3
2280      PRINT USING “5X,””ACP Alt1 Hi:””,11X,M2D.2D,”” dBc”””;Res4
2290      PRINT USING “5X,””ACP Alt2 Lo:””,11X,M2D.2D,”” dBc”””;Res5
2300      PRINT USING “5X,””ACP Alt2 Hi:””,11X,M2D.2D,”” dBc”””;Res6
2310    CASE “ATXP”
2320      PRINT USING “5X,””Ave Analog Power:””,6X,M2D.2D,”” dBm”””;Res1
2330    CASE “FST”
2340      PRINT USING “5X,””Worst Freq Error:””,5X,M3D.2D,”” ppm”””;Res1
2350      PRINT USING “5X,””Average Freq:””,6X,M3D.2DESZ,”” Hz”””;Res2
2360    CASE “FM”
2370      PRINT USING “5X,””SAT Deviation:””,9X,M5D,”” Hz”””;Res1
2380      PRINT USING “5X,””Distortion:””,11X,M3D.2D,”” %”””;Res2
2390    CASE “AFAN”
2400      PRINT USING “5X,””Audio Level:””,11X,M3D.2D,”” V”””;Res1
2410      PRINT USING “5X,””Audio Distortion:””,6X,M3D.2D,”” %”””;Res2
2420      PRINT USING “5X,””SINAD:””,16X,M3D.2D,”” dB”””;Res3
2430    END SELECT
2440  SUBEND
2450  SUB Bad_measurement(Integrity,Meas_name$)
2460    PRINT “     Measurement warning: “;Meas_name$
2470    PRINT “     Integrity =”;Integrity
2480  SUBEND
2490  SUB Dropped_call
2500    PRINT “ “
2510    PRINT “Call was dropped. Program aborted.”
2520    STOP
2530  SUBEND
2540  SUB Print_maho(Ber$,Rssi,Maho_level)
2550    SELECT Ber$
2560    CASE “BERI0”
2570      PRINT “     MAHO BER at “;Maho_level;” dBm is:  <0.01”
2580    CASE “BERI1”
2590      PRINT “     MAHO BER at “;Maho_level;” dBm is:  0.01 < 0.1”
2600    CASE “BERI2”
2610      PRINT “     MAHO BER at “;Maho_level;” dBm is:  0.1 < 0.5”
2620    CASE “BERI3”
2630      PRINT “     MAHO BER at “;Maho_level;” dBm is:  0.5 < 1.0”
```

```
2640    CASE “BERI4”
2650      PRINT “     MAHO BER at “;Maho_level;” dBm is:  1.0 < 2.0”
2660    CASE “BERI5”
2670      PRINT “     MAHO BER at “;Maho_level;” dBm is:  2.0 < 4.0”
2680    CASE “BERI6”
2690      PRINT “     MAHO BER at “;Maho_level;” dBm is:  4.0 < 8.0”
2700    CASE “BERI7”
2710      PRINT “     MAHO BER at “;Maho_level;” dBm is:  > 8.0”
2720    CASE ELSE
2730      PRINT “     MAHO Results not available.”
2740      SUBEXIT
2750    END SELECT
2760    PRINT “     MAHO RSSI at “;Maho_level;” dBm is: “;Rssi
2770  SUBEND
2780  SUB Orig_failed
2790    PRINT “Origination failed. Check conditions and run again.”
2800    STOP
2810  SUBEND
```