**Agilent Technologies 8960 Series 10 E5515B,C,T Wireless Communications Test Set**

# Programming: Getting Started Guide

AMPS/136 Mobile Test Application E1961A Revision: A.04
cdma2000 Mobile Test Application E1962B Revision: B.01
GPRS Mobile Test Application E1964A Revision: A.01
GSM Mobile Test Application E1960A Revision: A.07
GSM_AMPS/136 Mobile Test (fast switching) E1985A Revision A.01
IS-2000 Mobile Test Application E1962A Revision: A.01

Agilent Part No: 5967-5125

Print Date: March 2001



Agilent Technologies

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

# Edition/Print Date

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

March 2001 - 5967-5125

# 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,C,T Wireless Communications Test Set,**
**AMPS/136 Moible Test Application E1961A**
**cdma2000 Mobile Test Application E1962B**
**GPRS Mobile Test Application E1964A**
**GSM Mobile Test Application E1960A**
**IS-2000 Mobile Test Application E1962A**

**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 WARRANTY OR 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 Internet, 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
Test and Measurement Call Center
P.O. Box 4026
Englewood, CO 80155-4026
(tel) 1 800 452 4844

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

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

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

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

**Australia/New Zealand:**

Agilent Technologies Australia Pty Ltd.
347 Burwood Highway
Forest Hill, Victoria 3131
6881

*Australia*
(tel) 1 800 629 485
(fax) (61 3) 9272 0749

*New Zealand*
(tel) 0 800 738 378
(fax) (64 4) 802

**Asia Pacific:**

Agilent Technologies
24/F, Cityplaza One,
111 Kings Road,
Taikoo Shing, Hong Kong

(tel) (852) 3197 7777
(fax) (852) 2506 9233

# Declaration of Conformity

**According to ISO/IEC Guide 22 and CEN/CENELEC EN 45014**

| **Manufacturer's Name** | **Agilent Technologies UK Limited** | **Agilent Technologies, Incorporated** |
|---|---|---|
| **Manufacturer's Address** | Electronics Products & Solutions<br>Group - Queensferry<br>South Queensferry<br>West Lothian, EH30 9TG<br>Scotland, United Kingdom | RF Communications PGU<br>24001 E. Mission Avenue<br>Liberty Lake, Washington<br>99019-9599<br>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. | |

**Conforms with the following European Directives**

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (including 93/68/EEC) and carries the CD Marking accordingly.

**Conforms with the following product standards:**

| **EMC** | **Standard** | **Limit** |
|---|---|---|
| | IEC 61326-1:1997+A1:1998/EN 61326-1:1997+A1:1998 | |
| | CISPR 11:1990 / EN 55011:1991 | Group 1 Class A [1] |
| | IEC 61000-4-2:1995+A1:1998 / EN 61000-4-2:1995 | 4kV CD, 8kV AD |
| | IEC 61000-4-3:1995 / EN 61000-4-3:1995 | 3 V/m, 80-1000 MHz |
| | IEC 61000-4-4:1995 / EN 61000-4-4:1995 | 0.5V signal lines, 1kV power lines |
| | IEC 61000-4-5:1995 / EN 61000-4-5:1995 | 0.5 kV line-line, 1 kV line-ground |
| | IEC 61000-4-6:1996 / EN 61000-4-6:1996 | 3V, 0.15-80 MHz |
| | IEC 61000-4-11:1994 / EN 61000-4-11:1994 | 1 cycle, 100% |
| **Safety** | IEC 61010-1:1990+A1:1992+A2:1995 / EN 61010-1:1993+A2:1995<br>Canada CSA C22.2 No. 1010.1:1992 | |

**Supplemental Information:**

[1] The product was tested in a typical configuration with Agilent Technologies test systems

**14 December 2000**

**R.M. Evans / Quality Manager**

**14 December 2000**

**W. V. Roland / Reliability & Regulatory Engineering Manager**

For further information, please contact your local Agilent Technologies sales office, agent or distributor. Authorized EU-representative: Agilent Technologies Deutschland, GmbH, Herrenberger Strabe 130, D 71034 Boblingen, Germany

# Declaration of Conformity

**According to ISO/IEC Guide 22 and CEN/CENELEC EN 45014**

| **Manufacturer's Name** | **Agilent Technologies, Incorporated** | **Agilent Technologies, Incorporated** |
|---|---|---|
| **Manufacturer's Address** | Electronics Products & Solutions<br>Group - Queensferry<br>South Queensferry<br>West Lothian, EH30 9TG<br>Scotland, United Kingdom | |
| **Declares, that the product** | | |
| **Product Name:** | 8960 Series 10 Wireless Communications Test Set | |
| **Model Number:** | **E5515T** | |
| **Product Options:** | This declaration covers all options of the above product. | |

**Conforms with the following European Directives**

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (including 93/68/EEC) and carries the CD Marking accordingly.

**Conforms with the following product standards:**

| **EMC** | **Standard** | **Limit** |
|---|---|---|
| | IEC 61326-1:1997+A1:1998 / EN 61326-1:1997+A1:1998 | |
| | CISPR 11:1990 / EN 55011:1991 | Group 1 Class A [1] |
| | IEC 61000-4-2:1995+A1:1998 / EN 61000-4-2:1995 | 4kV CD, 8kV AD |
| | IEC 61000-4-3:1995 / EN 61000-4-3:1995 | 3 V/m, 80-1000 MHz |
| | IEC 61000-4-4:1995 / EN 61000-4-4:1995 | 0.5V signal lines, 1kV power lines |
| | IEC 61000-4-5:1995 / EN 61000-4-5:1995 | 0.5 kV line-line, 1 kV line-ground |
| | IEC 61000-4-6:1996 / EN 61000-4-6:1996 | 3V, 0.15-80 MHz |
| | IEC 61000-4-11:1994 / EN 61000-4-11:1994 | 1 cycle, 100% |
| **Safety** | IEC 61010-1:1990+A1:1992+A2:1995 / EN 61010-1:1993+A2:1995<br>Canada CSA C22.2 No. 1010.1:1992 | |

**Supplemental Information:**

[1] The product was tested in a typical configuration with Agilent Technologies test systems

**14 December 2000**

**W. V. Roland / Reliability & Regulatory Engineering Manager**

For further information, please contact your local Agilent Technologies sales office, agent or distributor. Authorized EU-representative: Agilent Technologies Deutschland, GmbH, Herrenberger Strabe 130, D 71034 Boblingen, Germany

## 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

# Contents

# Contents

# Contents

# Contents

# Programming: Getting Started Guide for AMPS/136 Mobile Test

# Introduction

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

## Conventions used in this Getting Started Guide

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

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

## How to use this Getting Started Guide

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

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

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

## Useful on-line links

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

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

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

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

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

## About the Programming Examples Presented in this Guide

### Programming Language:

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

### Line Numbers

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

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

### Syntax used in Programming Examples:

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

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

the shortened form would be:

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

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

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

appears in the programming examples as:

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

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

### Complex Commands

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

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

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

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

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

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

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

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

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

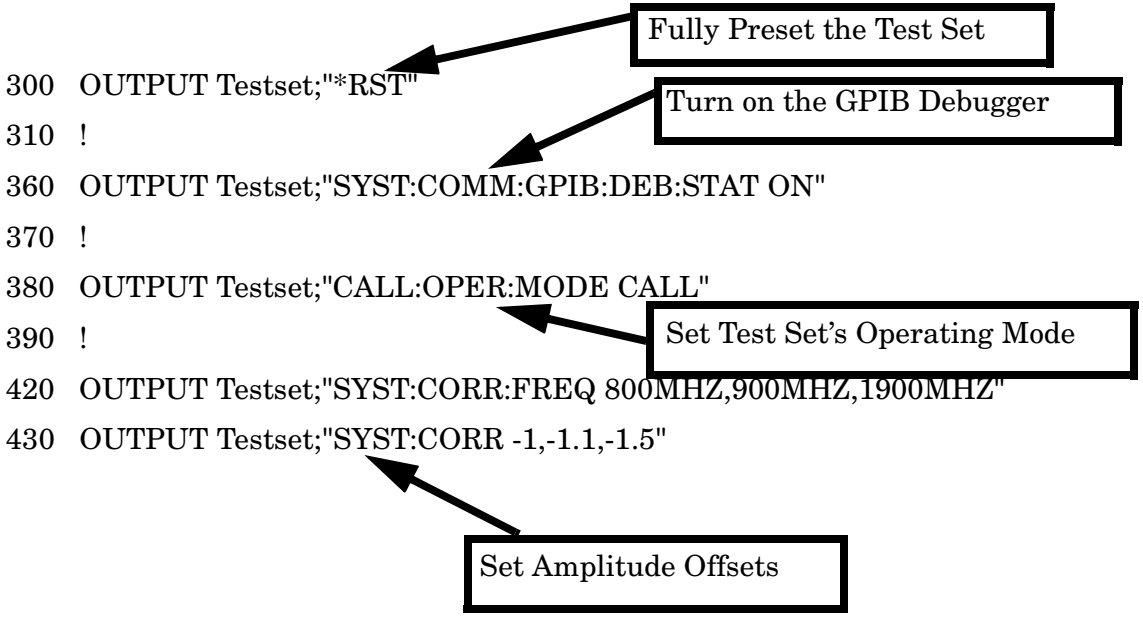
  OUTPUT Test_set;"SET:DTXP:TIM 10 S"

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

# Step 1: Set up the Test Set

This step explains how to:

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




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

## Fully Preset the Test Set

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

## Turn on the GPIB Debugger

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

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

## Set Test Set's Operating Mode

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

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

## Set Amplitude Offsets

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

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

# Step 2: Configure the Test Set and Mobile Station Parameters

This step explains how to:

- "Configure the Control Channel Parameters"
- "Configure the Traffic Channel Parameters"
- "Configure the Mobile Station Parameters"

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

## Configure the Control Channel Parameters

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

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

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

## Configure the Traffic Channel Parameters

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

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

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

## Configure the Mobile Station Parameters

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

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

# Step 3: Set the Measurement Parameters

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

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

## Step 3: Set the Measurement Parameters

# Step 4: Make a Connection

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

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

## Originating a Call from the Mobile Station

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

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

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

## Originating a Call from the Test Set

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

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

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

## Make a Connection using Test Mode Commands

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

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

# Step 5: INITiate and FETCh Measurements

- "Digital Measurements"
- "Analog Measurements"
- "Validate Measurement Results"
- "Printing Results"

## Digital Measurements

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

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

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

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

## Analog Measurements

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

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

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

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

## Validate Measurement Results

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

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

## Printing Results

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

```
2420 SUB Print_res(Meas_name$,Res1,OPTIONAL Res2,Res3,Res4,Res5,Res6)
2430    SELECT Meas_name$
2440    CASE "DTXP"
2450      PRINT USING "5X,""Ave Digital Power:"",5X,M2D.2D,"" dBm""";Res1
2460    CASE "MACC"
2470      PRINT USING "5X,""Max EVM1:"",14X,M2D.2D,"" %""";Res1
2480      PRINT USING "5X,""Worst Freq Error:"",5X,M3D.2D,"" Hz""";Res2
2490      PRINT USING "5X,""Max Mag. Error:"",8X,M2D.2D,"" %""";Res5
2500      PRINT USING "5X,""Max Origin Offset:"",5X,M2D.2D,"" dB""";Res3
2510      PRINT USING "5X,""Max Phase Error:"",7X,M2D.2D,"" Deg""";Res4
2520    CASE "ACP"
2530      PRINT USING "5X,""ACP Adj Lo:"",12X,M2D.2D,"" dBc""";Res1
2540      PRINT USING "5X,""ACP Adj Hi:"",12X,M2D.2D,"" dBc""";Res2
2550      PRINT USING "5X,""ACP Alt1 Lo:"",11X,M2D.2D,"" dBc""";Res3
2560      PRINT USING "5X,""ACP Alt1 Hi:"",11X,M2D.2D,"" dBc""";Res4
2570      PRINT USING "5X,""ACP Alt2 Lo:"",11X,M2D.2D,"" dBc""";Res5
2580      PRINT USING "5X,""ACP Alt2 Hi:"",11X,M2D.2D,"" dBc""";Res6
2590    CASE "ATXP"
2600      PRINT USING "5X,""Ave Analog Power:"",6X,M2D.2D,"" dBm""";Res1
2610    CASE "FST"
2620      PRINT USING "5X,""Worst Freq Error:"",5X,M3D.2D,"" ppm""";Res1
2630      PRINT USING "5X,""Average Freq:"",6X,M3D.2DESZ,"" Hz""";Res2
2640    CASE "FM"
2650      PRINT USING "5X,""SAT Deviation:"",9X,M5D,"" Hz""";Res1
2660      PRINT USING "5X,""Distortion:"",11X,M3D.2D,"" %""";Res2
2670    CASE "AFAN"
2680      PRINT USING "5X,""Audio Level:"",11X,M3D.2D,"" V""";Res1
2690      PRINT USING "5X,""Audio Distortion:"",6X,M3D.2D,"" %""";Res2
2700      PRINT USING "5X,""SINAD:"",16X,M3D.2D,"" dB""";Res3
2710    END SELECT
2720  SUBEND
```

# Step 6: Reconfigure Test Set and Mobile Station Parameters

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

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

## Reconfigure the Connection to a New Channel

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

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

## Reconfigure the Connection to a New Band

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

```
1630  OUTPUT Testset;"CALL:SET:DTC:BAND PCS"
1730  !
1740  OUTPUT Testset;"CALL:HAND"
1750  OUTPUT Testset;"CALL:STAT:STAT?"
1760  ENTER Testset;Connected$
1770  IF Connected$<>"CONN" THEN
1780    Dropped_call
1790  END IF
```

## Reconfigure the Connection to a New Traffic Channel Type

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

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

## Reconfigure the Connection when using Test Mode

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

# Step 7: End the Connection

You can end the connection in one of two ways:

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

## Ending the Connection from the Test Set

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

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

## Ending the Connection from the Mobile Station

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

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

## Step 7: End the Connection

# Programming: Getting Started Guide for cdma2000 Mobile Test

# Introduction

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

## Conventions used in this Getting Started Guide

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

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

## How to use this Getting Started Guide

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

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

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

## Useful on-line links

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

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

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

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

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

## About the Programming Examples Presented in this Guide

### Programming Language:

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

### Line Numbers

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

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

### Syntax used in Programming Examples:

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

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

the shortened form would be:

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

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

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

appears in the programming examples as:

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

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

### Complex Commands

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

The syntax below turns the state of the parameter on.

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

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

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

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

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

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

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

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

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

# Step 1: Set up the Test Set

This step explains how to:

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

## Fully Preset the Test Set

To set up the test set, you begin by sending the *RST command along with a global timeout. The *RST command is used to perform a full reset of the test set, returning it to a known state.

```
100   A=714 ! E5515C GPIB Address
110   ON TIMEOUT 7,20 CALL Timeout ! Calls “Timeout” routine in Step 7
120   OUTPUT A;”*RST
```

## Turn on the GPIB Debugger

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

```
130   OUTPUT A;”SYST:COMM:GPIB:DEB:STAT ON
```

**NOTE** The GPIB debugger assists you when debugging code. This command should be taken out of your code once development is completed because it slows program execution.

## Set Test Set's Operating Mode

The command below can be used to set the test set's operating mode to active cell. This means call processing will be used to maintain a link between the test set and the mobile station.

```
140   OUTPUT A;"CALL:OPER:MODE CALL"
```

There is also a test mode operating mode available. It is applicable when the mobile station is being operated without over-the-air signalling.

# Step 2: Configure Test Set and Mobile Station Parameters

## Configure the Test Set Parameters

The programming example below illustrates several cell configurations.

```
150   OUTPUT A;"CALL:BAND:DIG2000 USC" ! Active band is US Cellular
160   OUTPUT A;"CALL:CHAN:DIG2000:USC 29" ! Channel is 29
170   OUTPUT A;"CALL:POW:DIG2000 -50" ! Cell power is -50 dBm
180   OUTPUT A;"CALL:SID 65535" ! System ID is 65535
190   OUTPUT A;"CALL:RCON F3R3" ! Radio config is forward link 4, reverse
link 3
200   OUTPUT A;"CALL:SOPT:RCON3 SO2" ! Service option is 2 on radio
config 3
210   OUTPUT A;"CALL:PROT:DIG2000 PREV6" ! Protocol standard is IS-2000
220   OUTPUT A;"CALL:D2KT:ESN:HEX '00000000'" ! ESN assignment (must use
single quotes
```

## Configure the Generated Code Channels

Specifications to the setup of the generated code channels may be necessary. The commands below configure many of the available settable parameters.

```
230   OUTPUT A;"CALL:FCH -10" ! F-FCH on and set to -10 dB
240   OUTPUT A;"CALL:FCH:WALS CODE14" ! F-FCH assigned to Walsh code 14
250   OUTPUT A;"CALL:OCNS:WALS CODE5" ! OCNS assigned to Walsh code 5
260   OUTPUT A;"CALL:PAG -12" ! F-Paging on and set to -12 dB
270   OUTPUT A;"CALL:PAG:DRAT HALF" ! F-Paging data rate set to half rate
280   OUTPUT A;"CALL:PIL -8" ! F-Pilot set to -8 dB
290   OUTPUT A;"CALL:QPCH:RTP -3" ! F-QPCH on and set to -3 dB
300   OUTPUT A;"CALL:SCH -15.6" ! F-SCH on and set to -15.6 dB
310   OUTPUT A;"CALL:SCH:DRAT:RCON1 BPS38400" F-SCH data rate set to
38.4k for RC3
320   OUTPUT A;"CALL:SYNC -16" ! F-Sync on and set to -16 dB
```

## Configure the Mobile Station Parameters

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

# Step 3: Set Measurement Parameters

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

```
330   ! Setup E1962B digital average power
340   OUTPUT A;"SET:DAP:CONT OFF" ! Single measurement
350   OUTPUT A;"SET:DAP:TIM 3" ! Timeout set to 3 secs
360   OUTPUT A;"SET:DAP:COUN 5" ! 5 measurement average
370   ! Setup E1962B channel power
380   OUTPUT A;"SET:CPOW:CONT ON" ! Continuous measurement
390   OUTPUT A;"SET:CPOW:TIM 3"
400   OUTPUT A;"SET:CPOW:COUN 5"
410   OUTPUT A;"SET:CPOW:MSP FAST" ! Channel power sampling set to fast
420   ! Setup E1962B waveform quality
430   OUTPUT A;"SET:WQU:CONT ON"
440   OUTPUT A;"SET:WQU:TIM 10"
450   OUTPUT A;"SET:WQU:COUN 3"
460   OUTPUT A;"SET:WQU:CDP:IQIN:LIM -30" ! Sets dB limit for inactive
code channels
470   ! Setup E1962B handoff waveform quality
480   OUTPUT A;"SET:HWQ:TIM 5"
```

# Step 4: Make a connection

## Select an example

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

## MS initiated connection

The following code queries the test set for the connection status and then stops the program if a connection is not established within fifteen seconds of arming the state change detector. This gives adequate time for the MS to connect with the test set.

```
490   PRINT "Make a call from the Mobile Station"
500   WHILE MS_State=0 ! This while loop checks for an established call
510     OUTPUT A;"CALL:CONN:TIM 15"
520     OUTPUT A;"CALL:CONN:ARM" ! State change detector armed
530     OUTPUT A;"CALL:CONN:STAT?"
540     ENTER A;MS_State
550   END WHILE
560   PRINT "Connection established, test in progress"o
```

## Test Set initiated connection

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

```
570   PRINT "Attempting connection from test set..."
580   WHILE TS_State=0 ! This while loop checks for an established call
590     OUTPUT A;"CALL:ORIG"
600     OUTPUT A;"CALL:ORIG:DONE?"
610     ENTER A;TS_State
620   END WHILE
630   PRINT "Connection established, test in progress"
```

# Step 5: INITiate and FETCh Measurements

## INITiate measurements and FETCh results

In a typical control program, many measurements are run concurrently. The code below is an example of how you might choose to organize your program to efficiently run and time concurrent measurements. Refer to the additional details about this step available on the Internet for more information about the different measurement results that are available and how to fetch them.

```
640   Simul_start=TIMEDATE
650   OUTPUT A;"INIT:DAP;CPOW;WQU"
660   LOOP
670     OUTPUT A;"INIT:DONE?"
680     ENTER A;measdone$
690       SELECT measdone$
700         CASE "DAP"
710           OUTPUT A;"FETC:DAP?"
720           ENTER A;"dap_integ,avgpow"
730           IF dap_integ=0 THEN
740             PRINT "Average power = ";avgpow;" dBm"
750           ELSE
760             PRINT "Integrity error for average power = ",dap_integ
770           END IF
780         CASE "CPOW"
790           OUTPUT A;"FETC:CPOW?"
800           ENTER A;cpow_integ,chpow
810           IF cpow_integ=0 THEN
820             PRINT  "Channel power = ";chpow;" dBm"
830           ELSE
840             PRINT "Integrity error for channel power = ",cpow_integ
850           END IF
860         CASE "WQU"
870           OUTPUT A;"FETC:WQU:INT?"
880           ENTER A;wq_int
890           OUTPUT A;"FETC:WQU:RHO?"
900           ENTER A;rho
910           OUTPUT A;"FETC:WQU:CDP:IQN;IQIN?"
920           ENTER A;noise_res,iqin_res
930           IF wq_int=O THEN
```

```
940             PRINT "Waveform quality multi-code rho = ",rho
950             IF noise_res=0 THEN
960               PRINT "Noise level OK"
970             ELSE
980               PRINT "Noise level FAILS!"
990             END IF
1000  IF iqin_res=0 THEN
1010                PRINT "Inactive code channels OK"
1020              ELSE
1030                PRINT "Inactive code channels FAIL!"
1040              END IF
1050            ELSE
1060            PRINT "Integrity error for waveform quality = ",wq_int
1070          END IF
1080       END SELECT
1090     EXIT IF measdone$="NONE"
1100  END LOOP
1110    Simul_end=TIMEDATE
1120  PRINT
1130  PRINT "Concurrent measurement time = ";Simul_end-Simul_start
1140  Hand_start=TIMEDATE
1150  OUTPUT A;"INIT:HWQ"! Handoff automatically occurs
1160  WAIT 5
1170  OUTPUT A;"FETC:HWQ:INT?"
1180  ENTER A;hwq_int
1190  OUTPUT A;"FETC:HWQ:RHO?"
1200  ENTER A;hwq_rho
1210    IF hwq_int=0 THEN
1220      PRINT "Handoff waveform qualuty multi-code rho = ",hwq_rho
1230    ELSE
1340      PRINT "Integrity error for handoff waveform quality = ",hwq_int
1350    END IF
1360  Hand_end=TIMEDATE
1370  PRINT
1380  PRINT "Handoff measurement time = ";Hand_end-Hand_start
1390  PRINT
1400  PRINT "TOTAL TEST TIME =";Hand_end-Simul_start
```

# Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

## Reconfigure the Channel, Band, and Power Level

The example below contains the code appropriate for testing at a new power level, and then handing the call off to a different band and channel.

```
1410  OUTPUT A;"CALL:POW:DIG2000 -72"
1420  OUTPUT A;"CALL:SET:BAND:DIG2000 USPC" ! Handoff active band to US
PCS
1430  OUTPUT A;"CALL:SET:CHAN 384" ! Channel is 384
1440  OUTPUT A;"CALL:HAND" ! Execute handoff
1450  WAIT 2
1460  OUTPUT A;"CALL:STAT:STAT?"
1470  ENTER A;hand_stat$
1480  IF hand_stat$<>"CONN"
1490    PRINT "Handoff unsuccessful"
1500    GOTO 1390
1510  ELSE IF hand_stat$="CONN"
1610    PRINT "Handoff successful"
1620  END IF
1630  OUTPUT A;"SET:DAP:CONT ON" ! Set average power measurement to
continuous
```

## Reconfigure the MS

There are no MS parameters reconfigured in this program example.

## Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

# Step 7: End the Connection

## End the Connection

The CALL:END command ends the mobile connection. A CALL:DCON:STAT? query is used to ensure the connection has ended.

```
1640  PRINT "End call"
1650  PRINT
1660  OUTPUT A;"CALL:END"
1670  OUTPUT A;"CALL:CONN:STAT?" ! Queries status of connection
1680  ENTER A;State
1690  IF State=1
1700    PRINT "Mobile did not disconnect"
1710  ELSE IF State=0
1720    PRINT "Mobile successfully disconnected"
1730  END IF
1740  Timeout:   SUB Timeout ! Global timeout handler (from Step 1)
1750             PRINT "Program timed out"
1760             CLEAR 7
1770             CLEAR 714
1780             SUBEND
1790  OUTPUT A;"SYST:PRES3" ! Partial reset; retains test set
configuration
1800  END
```

### Step 7: End the Connection

# Programming: Getting Started Guide for GPRS Mobile Test

# Introduction

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

## Conventions used in this Getting Started Guide

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

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

## How to use this Getting Started Guide

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

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

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

## Useful on-line links

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

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

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

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

Programming: Getting Started Guide

This on-line version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.

Control Program Examples

These examples are for you to download. You may want to use these as templates for your own control program or to execute.

The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

## About the Programming Examples Presented in this Guide

### Programming Language:

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

### Line Numbers

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

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

### Syntax used in Programming Examples:

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

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

the shortened form would be:

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

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

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

appears in the programming examples as:

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

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

### Complex Commands

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

The syntax below turns the state of the parameter on.

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

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

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

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

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

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

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

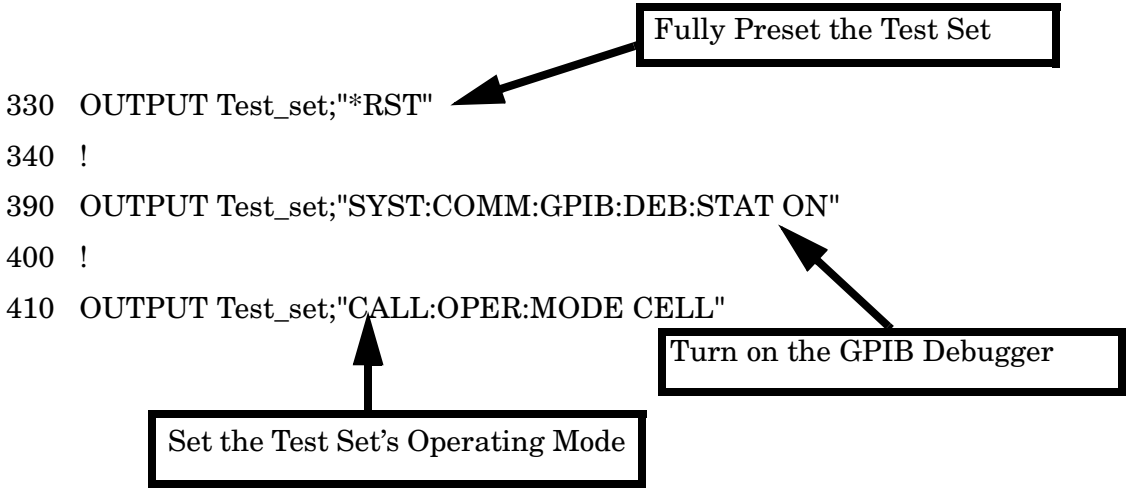
```
OUTPUT Test_set;"SET:DTXP:TIM 10 S"
```

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

# Step 1: Set up the Test Set

This step explains how to:

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



## Fully Preset the Test Set

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

## Turn on the GPIB Debugger

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

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

## Set Test Set's Operating Mode

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

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

# Step 2: Configure Test Set and Mobile Station Parameters

This step explains how to:

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

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

## Configure the Broadcast Channel Parameters

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

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

## Configure the Packet Data Traffic Channel Parameters

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

```
520   OUTPUT Test_set;"CALL:PDTCH 45" ! Set packet data traffic channel
530   !  Set Data Connection Type to BLER
540   OUTPUT Test_set;"CALL:FUNC:DATA:TYPE BLER"
550   ! Set Multi-slot Configuration to two downlinks and one uplinks.
560   OUTPUT Test_set;"CALL:PDTCH:MSL:CONF D2U1"
570   OUTPUT Test_set;"CALL:PDTCH:CSCH CS4"! Set Coding Scheme to CS4
580   !
590   ! Assign values to the power reduction levels
600   OUTPUT Test_set;"CALL:PDTCH:PZER:LEV 30"   ! Assign a value to P0
610   OUTPUT Test_set;"CALL:PDTCH:PRED:LEV1 11"! Set PRL1 to 11 dB
620   OUTPUT Test_set;"CALL:PDTCH:PRED:LEV2 0"   ! Set PRL2 to 0 dB
630   !
640   ! Assign power levels to the downlink bursts
650   ! Assign Burst 1 a PRL of 0 dB
660   OUTPUT Test_set;"CALL:PDTCH:PRED:BURS1 PRL2"
670   ! Assign Burst 2 a PRL of 11 dB
680   OUTPUT Test_set;"CALL:PDTCH:PRED:BURS2 PRL1"
```

## Configure the Mobile Station Operating Conditions

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

```
710   OUTPUT Test_set;"CALL:PDTCH:MS:TXL:BURS1 5"
720   OUTPUT Test_set;"CALL:PDTCH:MS:TXL:BURS2 3"
```

# Step 3: Set Measurement Parameters

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

```
780   ! Configure ORFS Measurement:
790   !
800   ! The lines below are examples of using complex commands to set
810   ! multi-meas state and count at the same time.
820   OUTPUT Test_set;"SET:ORFS:SWIT:COUN 5"
830   OUTPUT Test_set;"SET:ORFS:MOD:COUN 10"
840   !
850   OUTPUT Test_set;"SET:ORFS:TRIG:SOUR AUTO"! Set trig source to AUTO.
860   OUTPUT Test_set;"SET:ORFS:CONT OFF"      ! Set trig mode to single.
870   OUTPUT Test_set;"SET:ORFS:TIM 60"        ! Set timeout time.
880   ! Put switching and modulation offsets to be tested into string
890   ! variables.  S$ contains switching offsets.  M$ contains
900   ! modulation offsets.
910   Swit$=".4MHZ,-.4MHZ,.6MHZ,-.6MHZ,1.2MHZ,-1.2MHZ,1.8MHZ,-1.8MHZ"
920   Mod$=".4MHZ,-.4MHZ,.6MHZ,-.6MHZ,1.2MHZ,-1.2MHZ,1.8MHZ,-1.8MHZ"
930   OUTPUT Test_set;"SET:ORFS:SWIT:FREQ "&Swit$
940   OUTPUT Test_set;"SET:ORFS:MOD:FREQ "&Mod$
950   !
960   ! Configure TX Power Measurement:
970   !
980   OUTPUT Test_set;"SET:TXP:COUN 3;CONT OFF;TIM 20"
990   OUTPUT Test_set;"SET:TXP:TRIG:SOUR AUTO;QUAL ON"
1000  !
1010  ! Configure Phase & Frequency Error Measurement:
1020  !
1030  OUTPUT Test_set;"SET:PFER:COUN 8;CONT OFF;TIM 30;SYNC MID"
1040  OUTPUT Test_set;"SET:PFER:TRIG:SOUR AUTO"
1050  !
1060  ! Configure multislot measurements to be made on burst 1
1070  OUTPUT Test_set;"RFAN:MSL:MEAS:BURS 1"
```

### Step 3: Set Measurement Parameters

# Step 4: Make a connection

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

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

## Mobile Station performs a GPRS Attach

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

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

```
1190  DISP "Turn the phone on now.  Make the phone perform a GPRS
attach."
1200  Start_time=TIMEDATE
1210  LOOP
1220    OUTPUT Test_set;"CALL:DCON:TIM 10"
1230    OUTPUT Test_set;"CALL:DCON:ARM"
1240    OUTPUT Test_set;"CALL:ATT?"
1250    ENTER Test_set;Att_state
1260   EXIT IF Att_state
1270    Current_time=TIMEDATE-Start_time
1280    IF Current_time>=Timer THEN       ! Timer value is 1 minute
1290      BEEP
1300      DISP ""
```

```
1310      PRINT "GPRS attach did not complete. Program terminated."
1320      STOP
1330    END IF
1340  END LOOP
```

## Start a Data Connection

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

```
1370  OUTPUT Test_set;"CALL:FUNC:DATA:STAR"
1380  OUTPUT Test_set;"CALL:TRAN?"
1390  ENTER Test_set;Tran_state
1400  IF NOT Tran_state THEN
1410    BEEP
1420    DISP ""
1430    PRINT "Data connection failed.  Program terminated."
1440    STOP
1450  END IF
```

# Step 5: INITiate and FETCh Measurements

This step explains how to:

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

## INITiate a set of measurements

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

```
1510  !  Start a set of concurrent measurements:
1520  !
1530  OUTPUT Test_set;"SYST:MEAS:RES" ! Reset the BLER results
1540  OUTPUT Test_set;"INIT:ORFS;TXP;PFER"
```

## FETCh measurement results

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

```
1580  CALL Global_fetch

2980  SUB Global_fetch
2990    OPTION BASE 1
3000    COM /Address/Test_set
3010    OUTPUT Test_set;"CALL:PDTCH?"
3020    ENTER Test_set;Pdtch
3030    OUTPUT Test_set;"CALL:PDTCH:MS:TXL:BURS?;BURS2?"
3040    ENTER Test_set;Ms_pwr_bs1;Ms_pwr_bs2
```

```
3050  !
3060  !  Determine if a measurement is done:
3070  !
3080    LOOP
3090      OUTPUT Test_set;"INIT:DONE?"
3100      ENTER Test_set;Meas_done$
3110  !
3120  ! Obtain measurement results: Each measurement illustrates a
3130  ! different way of reading in results. There is no one right way.
3140  ! The method used is application dependent. Note that the examples
3150  ! do not show all possible ways.
3160  !
3170      SELECT Meas_done$
3180  !
3190      CASE "TXP"  ! TX Power measurement done.
3200        ALLOCATE Txpower(4)
3210        OUTPUT Test_set;"FETC:TXP:INT?;POW:ALL?"
3220        ENTER Test_set;Integrity,Txpower(*)
3230        IF (Integrity=0) THEN  ! Always check integrity value.
3240          PRINT "TX Power results: PDTCH=";Pdtch
3250          PRINT "                  Burst1 TXL=";Ms_pwr_bs1
3260          PRINT "                  Burst2 TXL=";Ms_pwr_bs2
3270          PRINT USING "5X,""Minimum:"",M2D.2D,"" dBm""";Txpower(1)
3280          PRINT USING "5X,""Maximum:"",M2D.2D,"" dBm""";Txpower(2)
3290          PRINT USING "5X,""Average:"",M2D.2D,"" dBm""";Txpower(3)
3300          PRINT USING "5X,""Std Dev:"",M2D.2D,"" dB""";Txpower(4)
3310          DEALLOCATE Txpower(*)
3320        ELSE
3330          GOSUB Bad_measurement
3340        END IF
3350  !
3360      CASE "PFER"  ! Phase & Frequency Error measurement done.
3370        OUTPUT Test_set;"FETC:PFER:ALL?"
3380        ENTER Test_set;Integrity,Rms_ph_er,Peak_ph_er,Worst_frq_er
3390        IF (Integrity=0) THEN
3400          PRINT "PFERror results: PDTCH=";Pdtch
3410          PRINT "                 Burst1 TXL=";Ms_pwr_bs1
3420          PRINT "                 Burst2 TXL=";Ms_pwr_bs2
3430          PRINT "RMS Phase Error: ";Rms_ph_er;" deg"
3440          PRINT "Peak Phase Error: ";Peak_ph_er;" deg"
3450          PRINT "Worst Freq Error: ";Worst_frq_er;" Hz"
3460        ELSE
```

```
3470          GOSUB Bad_measurement
3480        END IF
3490  !
3500      CASE "ORFS"  ! ORFS measurement done.
3510  !
3520  ! This code illustrates a more 'generic' approach to reading
3530  ! measurement results. By using the capabilities designed into
3540  ! high-level measurements, routines that access measurement
3550  ! results do not have to explicitly know what the measurement
3560  ! execution conditions were.  That information can be determined
3570  ! at the time the measurement results are queried.
3580  !
3590        OUTPUT Test_set;"FETC:ORFS:INT?"        ! Check integrity.
3600        ENTER Test_set;Integrity
3610        IF (Integrity=0) THEN
3620          ! Get the number of offsets tested.
3630          OUTPUT Test_set;"SET:ORFS:SWIT:FREQ:POIN?"
3640          ENTER Test_set;Points
3650          IF Points THEN  ! Only query if one or more offsets tested.
3660            ALLOCATE Swit_res(Points),Swit_offs(Points)
3670            ! Get measurement offsets.
3680            OUTPUT Test_set;"SET:ORFS:SWIT:FREQ?"
3690            ENTER Test_set;Swit_offs(*)
3700            ! Get results
3710            OUTPUT Test_set;"FETC:ORFS:POW?;:FETC:ORFS:SWIT?"
3720            ENTER Test_set;Tx_power,Swit_res(*)
3730            PRINT "ORFS Swit Results: PDTCH=";Pdtch
3740            PRINT "                   Burst1 TXL=";Ms_pwr_bs1
3750            PRINT "                   Burst2 TXL=";Ms_pwr_bs2
3760           PRINT USING "19X,""TX Power ="",M2D.2D,"" dBm""";Tx_power
3770            PRINT "     Offset(kHz)          Level(dBm)"
3780            PRINT "     -----------          ----------"
3790 Orfs_image:  IMAGE 6X,M4D.2D,12X,M4D.2D
3800            FOR J=1 TO Points
3810             PRINT USING Orfs_image;(Swit_offs(J)/1.E+3),Swit_res(J)
3820            NEXT J
3830            DEALLOCATE Swit_res(*),Swit_offs(*)
3840          END IF
3850          ! Get the number of offsets tested.
3860          OUTPUT Test_set;"SET:ORFS:MOD:FREQ:POIN?"
3870          ENTER Test_set;Points
3880          IF Points THEN ! Only query if one or more offsets tested.
```

```
3890            ALLOCATE Mod_res(Points),Mod_offs(Points)
3900            ! Get measurement offsets
3910            OUTPUT Test_set;"SET:ORFS:MOD:FREQ?"
3920            ENTER Test_set;Mod_offs(*)
3930            ! Get results
3940            OUTPUT Test_set;"FETC:ORFS:POW?;:FETC:ORFS:MOD?"
3950            ENTER Test_set;Tx_power,Pwr_30khz,Mod_res(*)
3960            PRINT "ORFS Mod Results: TCH=";Tch;"and TXL=";Ms_pwr_lvl
3970            PRINT "30 KHz BW Power =";Pwr_30khz;" dBm"
3980            PRINT "     Offset(kHz)          Level(dB)"
3990            PRINT "     -----------          ---------"
4000            FOR J=1 TO Points
4010              PRINT USING Orfs_image;(Mod_offs(J)/1.E+3),Mod_res(J)
4020            NEXT J
4030            DEALLOCATE Mod_res(*),Mod_offs(*)
4040          END IF
4050        ELSE
4060          GOSUB Bad_measurement
4070        END IF
4080      END SELECT
4090    EXIT IF Meas_done$="NONE"
4100    END LOOP  ! If 'WAIT' is returned from 'INIT:DONE?' query, it
4110              ! just falls through the loop.
4120    SUBEXIT
4130 Bad_measurement: !
4140  PRINT "Measurement error: "&Meas_done$
4150  PRINT "Measurement Integrity value =";Integrity
4160  RETURN
4170  !
4180  SUBEND
```

## Making the BLER Measurement

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

```
1590  REPEAT
1600    OUTPUT Test_set;"CALL:STAT:PDTCH:BLER?"
1610    ENTER Test_set;Bler,Blocks
1620  UNTIL (Blocks>=2000 AND Blocks<=2300)
1630  PRINT
1640  PRINT "Blocks tested at -101 dBm = ";Blocks
1650  PRINT "BLER Result at downlink PDTCH power of -101 dBm = ";Bler
```

# Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

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

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

## Reconfigure the Mobile Station Parameters

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

```
1880  OUTPUT Test_set;"CALL:PDTCH:MS:TXL:BURS 10"
```

## Reconfigure the connection to a new PDTCH ARFCN

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

```
2100  OUTPUT Test_set;"CALL:SET:PDTCH:MS:TXL:BURS 5"
2110  OUTPUT Test_set;"CALL:SET:PDTCH 120"
2120  OUTPUT Test_set;"CALL:HAND"
2130  ! Use a call synchronization command to ensure the
2140  ! reconfiguration succeeded.
2150  OUTPUT Test_set;"CALL:STAT:DATA?"
```

```
2160  ENTER Conn_status$
2170  IF Conn_status$<>"TRAN" THEN
2180     PRINT "Data connection failed to reconfigure properly."
2190     PRINT "Program terminated."
2200     STOP
2210  END IF
```

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

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

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

```
2370  OUTPUT Test_set;"CALL:PDTCH:MS:TXL:DCS:BURS 15"
2380  OUTPUT Test_set;"CALL:PDTCH:DCS 665"
2390  OUTPUT Test_set;"CALL:PDTCH:BAND DCS"
2400  ! Use a call syncrhonization command to ensure the
2410  ! reconfiguration succeeded.
2420  OUTPUT Test_set;"CALL:STAT:DATA?"
2430  ENTER Conn_status$
2440  IF Conn_status$<>"TRAN" THEN
2450     PRINT "Data connection failed to reconfigure properly."
2460     PRINT "Program terminated."
2470     STOP
2480  END IF
```

# Step 7: End the Connection

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

- "End the Data Connection"
- "GPRS Detach"

## End the Data Connection

The CALL:FUNC:DATA:STOP command ends the data connection. As in "Step 4: Make a connection" on page 73, synchronization with the control program is important. The CALL:FUNC:DATA:STOP command is a sequential command, meaning its operation is completed before the test set accepts another command. Therefore, it is only necessary to use the CALL:STAT:DATA? query to ensure the data connection has ended and the connection is in the "ATTached" state.

```
2640  OUTPUT Test_set;"CALL:FUNC:DATA:STOP"
2650  OUTPUT Test_set;"CALL:STAT:DATA?"
2660  ENTER Test_set;Conn_status$
2670  IF Conn_status$<>"ATT" THEN
2680    PRINT "Unable to terminate data connection correctly."
2690    PRINT "PROGRAM TERMINATED."
2700    STOP
2710  END IF
```

## GPRS Detach

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

The example below illustrates how you could maintain synchronization if you choose to perform the GPRS detach.

```
2760  DISP "Initiate a GPRS Detach"
2770  Start_time=TIMEDATE
2780  LOOP
2790    OUTPUT Test_set;"CALL:STAT:DATA?"
2800    ENTER Test_set;Conn_state$
2810   EXIT IF Conn_state$="IDLE"
2820    Current_time=TIMEDATE-Start_time
2830    IF Current_time>=Timer THEN
2840      DISP ""
2850      PRINT "GPRS detach did not occur.  Program terminated"
2860      STOP
2870    END IF
2880    IF Conn_state$="DET" THEN
2890      DISP "GPRS detach is in process."
2900    END IF
2910  END LOOP
```

# Programming: Getting Started Guide for GSM Mobile Test

# Introduction

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

## Conventions used in this Getting Started Guide

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

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

## How to use this Getting Started Guide

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

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

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

## Useful on-line links

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

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

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

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

Programming: Getting Started Guide

This online version of this guide contains links to more detailed information about each step which may be useful as you develop your control program.

Control Program Examples

These examples are for you to download. You may want to use these as templates for your own control program or to execute.

The control program explained in the Getting Started Guide is also available here for you to download. It is a fully functional control program.

## About the Programming Examples Presented in this Guide

### Programming Language:

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

### Line Numbers

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

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

### Syntax used in Programming Examples:

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

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

the shortened form would be:

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

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

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

appears in the programming examples as:

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

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

### Complex Commands

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

The syntax below turns the state of the parameter on.

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

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

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

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

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

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

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

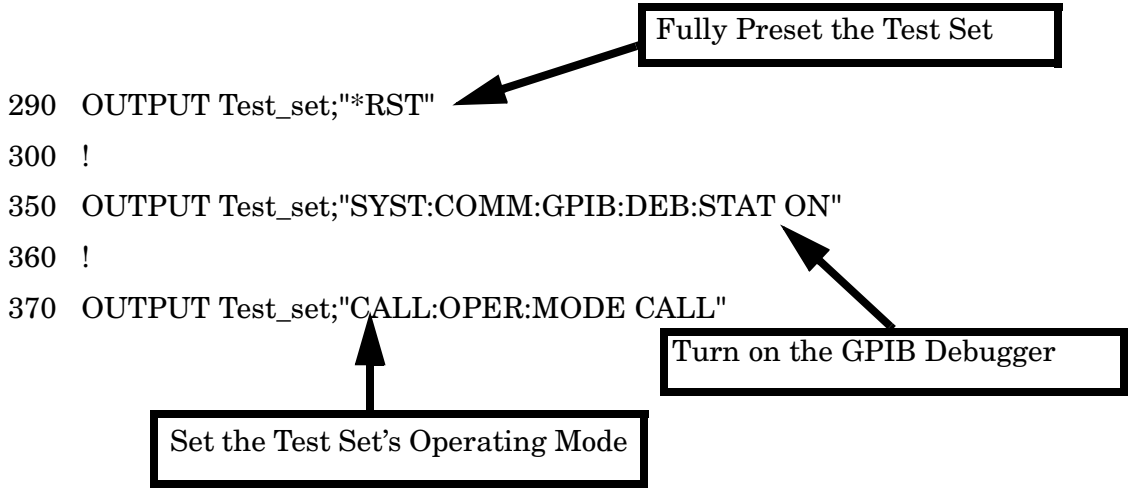
OUTPUT Test_set;"SET:DTXP:TIM 10 S"

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

# Step 1: Set up the Test Set

This step explains how to:

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




## Fully Preset the Test Set

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

## Turn on the GPIB Debugger

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

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

## Set the Test Set's Operating Mode

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

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

# Step 2: Configure Test Set and Mobile Station Parameters

This step explains how to:

- "Configure the Broadcast Channel Parameters"
- "Configure the Traffic Channel Parameters"
- "Configure the Mobile Station Operating Parameters"

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

## Configure the Broadcast Channel Parameters

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

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

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

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

## Configure the Traffic Channel Parameters

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

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

## Configure the Mobile Station Operating Parameters

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

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

# Step 3: Set the Measurement Parameters

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

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

# Step 4: Make a Connection

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

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

## Originating a Call from the Test Set

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

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

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

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

## Originating a Call from the Mobile Station

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

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

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

# Step 5: INITiate and FETCh Measurements

This step explains how to:

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

## INITiate a set of measurements

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

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

## FETCh measurement results using a subroutine

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

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

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

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

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

# Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

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

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

## Reconfigure the Mobile Station Parameters

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

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

## Reconfigure the Connection to a New ARFCN

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

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

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

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

## Reconfigure the Connection to a New ARFCN in a Different Band

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

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

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

# Step 7: End the Connection

You can end the connection in one of two ways:

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

## Ending the Connection from the Test Set

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

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

## Ending the Connection from the Mobile Station

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

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

```
OUTPUT Test_set;"CALL:CONN:TIM 5" !Set timeout time to 5 seconds.
OUTPUT Test_set;"CALL:CONN:ARM"   !Arm the change detector.
OUTPUT Test_set;"CALL:CONN:STAT?" !Initiate call connect state query.
DISP "Terminate the call from the mobile station."
ENTER Test_set;Call_connected     !Program will hang here until state
                                  !change or timer expires.
IF Call_connected THEN            !Check if disconnect successful.
  OUTPUT Test_set;"CALL:END"
  PRINT "Call failed to end correctly.  Program terminated."
  STOP
END IF
```

# Programming: Getting Started Guide for GSM_AMPS/136 Mobile Test

# GSM_AMPS/136 Programming

The Agilent Technologies E1985A, GSM_AMPS/136 Mobile Test fast switching test application allows you to switch between GSM and TIA/EIA 136 formats very quickly. The GSM_AMPS/136 test application combines the measurements and features from the E1960A GSM, and the E1961A AMPS/136 Mobile Test test applications into a single fast switching test application. Switching between test applications takes more than one minute, whereas switching formats in a fast switching test application takes less than 2 seconds. When a format is selected in a fast switching test application, that format becomes the active format.

Refer to the reference information for GSM when you need details about GSM operation, and refer to the reference information for AMPS/136 when you need details about AMPS/136 operation.

## Format Switching GPIB Commands

- To perform a format switch to the AMPS/136 radio format use this GPIB command:

```
OUTPUT 714;"SYSTEM:APPLICATION:FORMAT:NAME 'AMPS/136'"
```

- To perform a format switch to the GSM radio format use this GPIB command:

```
OUTPUT 714;"SYSTEM:APPLICATION:FORMAT:NAME 'GSM'"
```

- To query the license status of the GSM format use this GPIB query:

```
OUTPUT 714;"SYSTEM:APPLICATION:FORMAT:LICENSE? 'GSM'"
```

- To query the license status of the AMPS/136 format use this GPIB query:

```
OUTPUT 714;"SYSTEM:APPLICATION:FORMAT:LICENSE? 'AMPS/136'"
```

## Identifiers For CALL Commands

Some CALL commands/queries are shared for both radio formats in the GSM_AMPS/136 fast switching test application, identifiers are used to specify the format for the GPIB commands/queries. The command/query is sent to the active format if you don't use an identifier. An identifier must be used when sending commands to the inactive format.

**NOTE** Select the tab labeled AMPS/136 for programming information in that radio format.

Select the tab labeled GSM/GPRS for programming information in the GSM radio format.

There are four identifiers available, all identifiers are not used for each CALL command. Refer to the following list of CALL commands to determine which identifiers are available for each command.

Using an identifier with the command is recommended for these CALL commands.

### CALL[:CELL]:MCCode Identifiers

CALL[:CELL]:MCCode|[:SELected] | :GSM | :DIGital136

### CALL[:CELL]:POWer Identifiers

CALL[:CELL]:POWer:AMPLitude |[:SELected] | :GSM | :TA136

CALL[:CELL]:POWer:AMPLitude:CW |[:SELected] | :TA136

CALL[:CELL]:POWer[:SAMPlitude] |[:SELected] | :GSM | :TA136

CALL[:CELL]:POWer[:SAMPlitude]:CW |[:SELected] | :TA136

CALL[:CELL]:POWer:STATe |[:SELected] | :GSM | :TA136

CALL[:CELL]:POWer:STATe:CW |[:SELected] | :TA136

### CALL[:CELL]:RFGenerator Identifiers

CALL[:CELL]:RFGenerator:FREQuency |[:SELected] | :GSM | :TA136

### CALL:MS:REPorted Identifiers

CALL:MS:REPorted:ONUMber |[:SELected] | :GSM | :TA136

CALL:MS:REPorted:PCLass |[:SELected] | :GSM | :TA136

CALL:MS:REPorted:REVision:CHARacter:GSM

CALL:MS:REPorted:REVision[:DIGital] |[:SELected] | :GSM | :DIGital136

### CALL:PAGing:REPeat Identifiers

CALL:PAGing:REPeat[:STATe] |[:SELected] | :GSM | :TA136

## Identifiers For DPOWer and IQTuning Measurements

The GSM_AMPS/136 test application uses the DPOWer and IQTuning measurement mnemonics for both formats. The measurements for each format are different even though the mnemonics are the same.

The GSM_AMPS/136 test application allows you to send measurement SETup commands to the active or the inactive format, you should not send ABORt, INITiate, FETch or READ commands to an inactive format.

When sending SETup commands to the inactive format you must use identifiers as part of the command. Using an identifier insures that the test set applies the SETup command to the measurement in the intended format. If you don't use an identifier in the command or query, the test set assumes that you want the active format.

**NOTE** Select the tab labeled AMPS/136 for programming information in that radio format.

Select the tab labeled GSM/GPRS for programming information in the GSM radio format.

The table below shows several examples of using identifiers as part of a DPOWer or IQTuning command.

**Table 2. Measurement Format Identifiers Examples**

| GPIB Examples | Description |
|---|---|
| SETup:DPOWer:TIMeout[:STIMe] 10s[:SELected] | Sets the timeout value in the DPOWer measurement for the active format. |
| SETup:IQTuning:TIMeout[:STIMe] 10s:GSM | Sets the timeout value for the IQTuning measurement in the GSM format whether or not it is the active format. |
| SETup:IQTuning:TIMeout[:STIMe] 10s:DIGital136 | Sets the timeout value for the IQTuning measurement in the AMPS/136 format whether or not it is the active format |
| INITiate:DPOWer[:ON][:SELected] | Initiates a DOWer measurement for the active format. |
| FETCh:IQTuning[:ALL][:SELected] | Fetches measurement results for the IQTuning measurement from the active format. |
| ABORt:DPOWer[:SELected] | Aborts the DPOWer measurement for the active format. |
| READ:IQTuning:[:ALL][:SELected] | Reads an IQTuning measurement for the active format. |

# Programming: Getting Started Guide for cdma2000 Test Mode Mobile Test

# Introduction

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

## Conventions used in this Getting Started Guide

Throughout this Getting Started Guide the term “test set” refers to an Agilent Technologies 8960 Series 10 wireless communications test set with the E1962A cdma2000 mobile test application installed.

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

## How to use this Getting Started Guide

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

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

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

## Useful on-line links

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

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

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

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

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

## About the Programming Examples Presented in this Guide

### Programming Language:

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

### Line Numbers

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

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

### Syntax used in Programming Examples:

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

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

the shortened form would be:

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

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

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

appears in the programming examples as:

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

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

### Complex Commands

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

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

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

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

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

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

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

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

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

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

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

# Step 1: Set up the Test Set

This step explains how to:

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

## Fully Preset the Test Set

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

```
120   COM Test_set
130   Test_set=720 ! GPIB address of test set
140   OUTPUT A;"*RST,*OPC
```

## Turn on the GPIB Debugger

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

```
150   OUTPUT Test_set;"SYST:COMM:GPIB:DEB:STAT ON"
```

**NOTE** The GPIB debugger assists you when debugging code. This command should be taken out of your code once development is completed.

## Set Test Set's Operating Mode

The command below can be used to set the test set's operating mode to test mode.

```
240   OUTPUT Test_set;"CALL:OPER:MODE D2KT"
```

The mobile station will be operated without over-the-air signalling.

# Step 2: Configure Test Set and Mobile Station Parameters

## Configure the Test Set Parameters

The programming example below illustrates several cell configurations.

```
150   OUTPUT Test_set;"SYST:CORR -2.5"! MS has a -2.5 dB fixture loss
160   OUTPUT Test_set;"CALL:POW -55"! Sets cell power to -55 dBm and turn
power state ON with complex command
170   OUTPUT Test_set;"CALL:BAND:DIG2000:USPC"! Sets active band to PCS
180   OUTPUT Test_set;"CALL:CHAN:USPC 384"! Sets traffic channel to 38
```

## Configure the Mobile Station Parameters

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

# Step 3: Set Measurement Parameters

This step gives an example of how you can configure measurement parameters.

```
190   !  Configure Digital Average Power Measurement:
200   OUTPUT Test_set;"SET:DAP:CONT ON"
210   OUTPUT Test_set;"SET:DAP:COUN 10"
220   OUTPUT Test_set;"SET:DAP:TIM:STIM 5"
230   !  Configure Channel Power Measurement:
240   OUTPUT Test_set;"SET:CPOW:CONT OFF"
250   OUTPUT Test_set;"SET:CPOW:COUN 5"
260   OUTPUT Test_set;"SET:CPOW:TIM:STIM 15
```

# Step 4: Make a connection

## Establish test mode connection

The mobile station (MS) must be connected with the test set in order to perform measurements. In test mode, the origination of this connection may be initiated through a terminal program by configuring the mobile's parameters to match those of the test set.

```
270   PRINT "SETUP THE MS FOR TEST MODE (USE A TERMINAL PROGRAM IF
NECESSARY)"
280   PRINT "PRESS CONTINUE"
290   PAUSE
```

# Step 5: INITiate and FETCh Measurements

## INITiate measurements and FETCh results

In a typical control program, many measurements are run concurrently. The code below is an example of how you might choose to organize your program to efficiently run concurrent measurements.

```
300   !  Start Set of Concurrent Measurements:
310   OUTPUT Test_set;"INIT:DAP;CPOW"
320   !  Determine if A Measurement Is Done
330   LOOP
340     OUTPUT Test_set;"INIT:DONE?"
350     ENTER Test_set;Meas_done$
360   !  Obtain Measurement Results:
370     SELECT Meas_done$
380     CASE "CPOW"
390        GOSUB "CPOW"
400     CASE "DAP"
410        OUTPUT Test_set;"FETC:DAP?"
420        ENTER Test_set;Dap_integ,Dig_avg_pow
430        IF Dap_integ=0 THEN
440          PRINT "DIG AVG POW= ",Dig_avg_pow
450        ELSE
460          GOSUB Bad_int_ind
470    !  Are Measurement Results Valid?:
480        END IF
490     END SELECT
500    EXIT IF Meas_done$="NONE"
510   END LOOP  !  If 'WAIT' is returned from the 'INIT:DONE?' query, the
loop will repeat.
```

# Step 6: Reconfigure Test Set and Mobile Station Connection Parameters

## Reconfigure the Test Set

The example below contains the code appropriate for testing at a new band and channel. Handoffs are difficult to perform in test mode so the connection and measurements must be once again initiated.

```
530  !  Reconfigure the Base Station Emulator
540  OUTPUT Test_set;"CALL:BAND:DIG2000:KPCS" ! Set active band to
Korean PCS
550  OUTPUT Test_set;"CALL:CHAN:USPC 325" ! Set traffic channel to 325
560  !  Setup The MS under new configuration using MS Test Mode Commands
570  PRINT " "
580  PRINT "SETUP THE MS FOR TEST MODE (USE A TERMINAL PROGRAM IF
NECESSARY)"
590  PRINT "PRESS CONTINUE"
600  PAUSE
610  !  Start Set of Concurrent Measurements:
630  OUTPUT Test_set;"INIT:CPOW"
640  !  Determine if A Measurement Is Done:
650  LOOP
660    OUTPUT Test_set;"INIT:DONE?"
670    ENTER Test_set;Cpow_done$
680  !  Obtain Measurement Results:
690    SELECT Cpow_done$
700    CASE "CPOW"
710       OUTPUT Test_set;"FETC:CPOW?"
720       ENTER Test_set;Cpow_integ,Chan_pow_meas
730       IF Integrity=0 THEN
740          PRINT "Channel Power Integrity = ",Cpow_integ
750       ELSE
760          GOSUB Bad_int_ind
770       END IF
780    END SELECT
790  EXIT IF Meas_done$="NONE"
800  END LOOP
810          Bad_int_ind !
```

```
820  PRINT Meas_done$&" Measurement Error"
830  PRINT "Measurement Integrity Indicator: ",Integrity
840  RETURN
850  END
```

## Reconfigure the MS

There are no MS parameters reconfigured within this program example.

# Step 7: End the Connection

## End the Connection

In test mode, the connection between the MS and the test set cannot be ended remotely, the code below simply instructs the use to manually end the call.

```
860  PRINT "End call either through terminal program or MS keypad"
870  OUTPUT A;"SYST:PRES3" ! Partial reset; retains test set
configuration
880  END
```