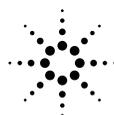


Agilent Technologies 8960 Series 10 Wireless Communications Test Set

Product Note



Designed with the
Programmer in Mind



Agilent Technologies
Innovating the HP Way

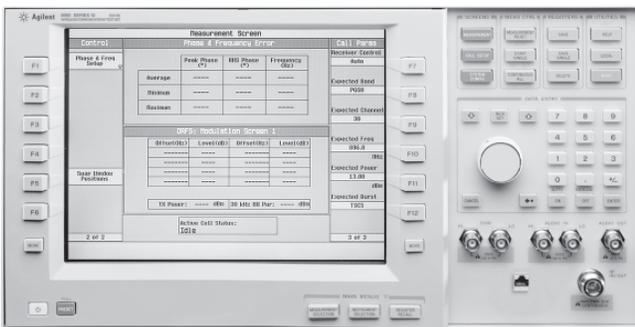


Agilent Technologies 8960 Series 10 Designed with the programmer in mind

The Agilent Technologies 8960 Series 10 wireless communications test set is Agilent Technologies' most flexible RF communications test platform. This platform contains many features that are specifically designed for automated operation

in the high-volume cellular manufacturing environment. In addition to fast measurements and production-quality accuracy and repeatability, the 8960 Series 10 is significantly easier to program than previous RF test equipment. This product note describes some of the programming features of the 8960 Series 10. These features were specifically designed to allow test engineers to write more efficient test application software that is:

- Faster
- More robust
- Requires less development, documentation, and maintenance.



High-Level Measurements

To reduce code size and increase the speed of measurements, the 8960 Series 10 has a feature called High-Level Measurements. In previous RF test equipment, test engineers were required to write complex algorithms to ensure accurate and repeatable measurements. These algorithms would often contain hundreds of lines of code just to produce a single set of measurement results. Optimizing this code for speed and robustness would often take factory expertise, or simply could not be done because key information about synchronization between the test equipment and the device-under-test was not available to the external control program.

High-Level Measurements – Easy as 1-2-3

High-Level Measurements (HLM) eliminate the task of developing optimized measurement algorithms. Each HLM contains a factory-designed measurement algorithm which has been highly optimized for that measurement. Access to these HLMs is provided through high-level, easy-to-use GPIB commands.

Using these high-level measurements, making a measurement has been reduced to a simple three step process.

The **first step** in making an HLM is to **SETUP** the measurement. In this step the control program sends commands to the test set to configure the HLM execution conditions. This is typically a few commands which set up parameters such as triggering conditions, number of bursts to measure, frequency offsets to measure, and so forth.

The **second step** in making an HLM is to **INITIATE** the measurement. This is a single command, issued from the control program, that starts the measurement.

The **final step** in making an HLM is to **FETCH** the measurement results. HLMs have been designed to return the entire set of measurement results in response

Measurement Screen			
Control	Transmit Power		Call Par
Operating Mode	Minimum	Maximum	Traffic Chan
Active Cell	12.35 dBm	12.36 dBm	30
	Average	Std Dev	Timeslot
	12.36 dBm	0.00 dB	4
End Call	Phase & Frequency Error		Timing Advanc
	Peak Phase	RMS Phase	0
	7.05°	1.96°	MS TX Level
	Frequency		15
	-12.06 Hz		Mobile Loopba
			Off
	Active Cell Status:		
	Connected		
1 of 2			2 of 3

to a single query from the control program, eliminating the need for multiple queries and their associated overhead.

The following short BASIC program illustrates a phase error measurement that makes 10 measurements, uses RF rise triggering, and the FETCH command to retrieve minimum, maximum, and average results from the test set.

```
10 OUTPUT 714; "SETUP:PFER:COUNT 10"  
20 OUTPUT 714; "SETUP:PFER:TRIG:SOURCE RISE"  
30 OUTPUT 714; "INITIATE:PFER"  
40 OUTPUT 714; "FETCH:PFER:RMS:MIN?;MAX?;AVER?"  
50 ENTER 714; Rms_min, Rms_max, Rms_avg
```

Notice how the SETUP subsystem allows you to customize the measurement exactly how you want it. This example uses only 5 lines of code and runs in less than 300 ms. Compared with previous products, obtaining these results on the 8960 is roughly 20 times faster and requires significantly less coding.

High-Level Measurements— More powerful than ever before

While the previous example illustrated how easy it is to do a complex measurement, there are many other advantages to high-level measurements. A step that may appear missing from the previous example is **measurement calibration**.

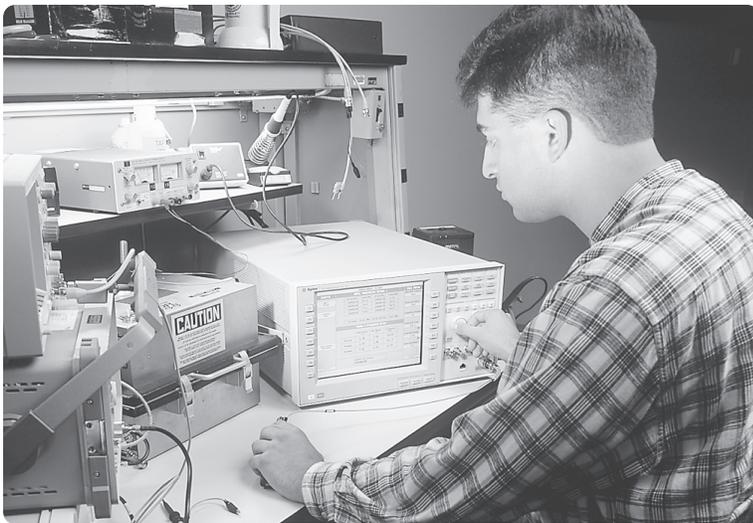
This was not an oversight or simplification—this is the way the 8960 Series 10 is designed. **All required calibration is performed automatically as part of the high-level measurement.** Each time a high-level measurement is initiated, the test set determines if any calibrations are required, and if so, the calibration is automatically performed. These calibrations are virtually instantaneous and require no information or external connections from the user. This greatly simplifies the programmer's task, and ensures that each measurement result is calibrated and accurate.

Another important feature of high-level measurements is **built-in statistical analysis**. A common desire in manufacturing is to take multiple measurements for quality control. Imagine you need to run 100 TX Power measurements and need to know the minimum, maximum, average, and standard deviation of the

100 measurements. For most instruments, this would require an external control program. The external control program would be required to record data and calculate the average and standard deviation. **On the 8960 Series 10, whenever multiple-measurements are run, statistical results are automatically calculated and easily available to the user.** The following short program illustrates a TX Power measurement that makes 100 measurements and returns minimum, maximum, average, and standard deviation to the control program.

```
10 OUTPUT 714; "SETUP:TXP:COUNT 100"  
20 OUTPUT 714;"INITIATE:TXP"  
30 OUTPUT 714;"FETCH:TXP:POW:ALL?"  
40 ENTER 714;Avg, Min, Max, S_Dev
```

Notice this example on the 8960 Series 10 uses just 4 lines of code and will run in about 600 ms. For most previous generation test sets, this same functionality would require dozens of lines of code and take at least 10 times as long.



Synchronization

A significant programming challenge for the test engineer is to synchronize the computer, the measurement equipment, and the device-under-test. Many of the events involved in testing a radio—such as setting up a call, changing channels, or waiting for signals to settle, are asynchronous events where the user has very little control or knowledge of the timing. In the past, RF test equipment provided little help, and the programmer was forced to use WAIT statements for synchronization. Using WAIT statements is inefficient and is not robust.

Realizing that simplifying the synchronization task would simplify programming and improve efficiency, the designers of the 8960 Series 10 have provided programming features to solve synchronization problems.

Sequential Command Syntax

Consider a simple process like changing a traffic channel and then starting a measurement. Obviously you don't want to start the measurement too soon and get unsettled or erroneous measurements. Likewise you don't want your program to spend excessive time waiting. What you want to do is know exactly when the traffic channel has changed and immediately begin your measurements. Some simple examples will illustrate how the 8960 Series 10 solves this problem.

This first example shows how a user might synchronize changing a channel and then starting a measurement using older test equipment.

```
10 OUTPUT Old_test_set; "CALL:TCH 100"  
20 WAIT 2 !Seconds  
30 OUTPUT Old_test_set; "MEAS:POWER"
```

Old method for synchronizing the start of measurement

In this example, the WAIT statement needs to be as long as the *slowest* radio. This means that for all other radios, the

program is wasting valuable time. Also, when a new radio is introduced, the WAIT statements need to be re-optimized to work properly with the new radio.

The 8960 Series 10 solves this problem with a simple "**SEquential**" command syntax. The following example illustrates use of the SEquential syntax.

```
10 OUTPUT Agilent8960; "CALL:TCH:SEQ 100"  
20 OUTPUT Agilent8960; "INIT:TXPOWER"
```

Agilent 8960 method for synchronizing the start of measurement

Notice that the WAIT statement is completely gone. The syntax on line 10 instructs the test set to initiate the channel change, and hold off on initiating the power measurement until the radio is safely on the new channel. Regardless of how fast or slow the radio, the test set uses protocol messages to automatically pace the events with the correct timing. This guarantees accurate and repeatable results. Also, notice the simplicity of writing and maintaining this code. New radios can have dramatically different timing characteristics, and the 8960 Series 10 will always test them efficiently and reliably.

INITIATE:DONE

In older test equipment, the only way to determine when results were ready was to query the instrument and hope the results were available. If the result was not available, the IEEE bus and the computer were hung-up waiting for the measurement to complete. Not only would this tie up the test system, but since measurements would sometimes *never* complete, the test engineer was required to write complex error recovery routines that included timeouts, clearing the IEEE bus, and aborting unfinished measurements.

In the 8960 Series 10, a unique command called “**INITIATE:DONE?**” queries the test set to determine if any measurement results are ready. This command does not hold the IEEE bus, and will always return a simple response like “TXP” indicating which measurement result is ready, “WAIT” indicating that the 8960 Series 10 is still working on a measurement, or “NONE” indicating that no measurements are running. The control software can continue to re-query until the test set indicates the result is ready. The following example illustrates this concept.

```
10 OUTPUT 714; "INIT:TXP;PFER" ! Initiates 2 measurements
20 OUTPUT 714; "INIT:DONE?"
30 ENTER 714; Meas_done$
40 IF Meas_done$ = "TXP" then GOSUB Read_txp_values
50 IF Meas_done $ = "PFER" then GOSUB Read_pfer_values
60 IF Meas_done $ = "WAIT" then GOTO 20
```

Measurement Timeout

In a manufacturing environment, the device-under-test (DUT) is often a new phone which has never been previously tested. Many scenarios could cause the DUT to be broken or so far out of alignment that the test set is unable to make a valid measurement. Within the 8960 Series 10, each high-level measurement contains a parameter called “**Measurement Timeout.**” This parameter instructs the test set to abort the measurement if it does not complete within a specified time. Once the measurement aborts, the test set returns a value indicating that the result is not available, and sets a measurement integrity indicator to indicate “Measurement Timed Out.” This feature provides a very predictable response from the test set, which makes it quite straightforward for the software programmer to deal with broken or uncalibrated radios.

Measurement Integrity

By now, you may be starting to get an idea about how high-level measurements in the 8960 Series 10 can greatly simplify your code development and maintenance. Since many complex tasks and algorithms are now part of the

test set, your control software is much smaller and much more straightforward. A final feature of high-level measurements that makes it easy to determine the quality of a measurement is the “Measurement Integrity” indicator. As part of each high-level measurement, the algorithm in the test set analyzes many factors and determines if the result is valid or questionable. Suppose you have a mobile phone that is working correctly, except that the output level is a few dB too high. When you try to measure this signal, you unknowingly overdrive the input to the test set, and your measurement results indicate problems. Unaware of the real problem, you spend valuable time troubleshooting a false failure. This example illustrates a common cause of “No Trouble Finds.”

This problem is solved in the 8960 Series 10 with a measurement integrity value. **For each result, or set of results, a measurement integrity value is generated.** For most measurements, the integrity value will be 0, indicating that the results are accurate and valid. If the test set detects a problem, the integrity value will be non-zero, giving the user an indication of the problem detected. Typical responses could be “Signal too large,” “Signal too small,” or “Burst too short” for bursted RF measurements.

Because it is critical to know that your results are valid, each time you initiate a measurement, the test set automatically generates the integrity indicator. This makes it easy for the programmer and does not require any additional queries or commands from the program. The following example illustrates the TX Power measurement, and how the measurement integrity indicator could be used.

```
10 OUTPUT 714; "INIT:TXP"
20 OUTPUT 714; "FETCH:TXP?"
30 ENTER 714; Txp_integrity, Txp_power
40 IF Txp_integrity = 0 THEN
50     PRINT "Power is "; Txp_power
60 ELSE
70     GOSUB Determine_txp_problem
80 END IF
```

Debugging Tools and Documentation

Although the 8960 Series 10 is inherently easier to program than previous test equipment, finding the correct programming commands and getting them typed in without any syntax errors, can still be a challenge in writing your control software. At Agilent Technologies we recognized this problem, and with the 8960 Series 10, we now provide documentation and a built-in debugger that simplifies finding commands and getting them typed in correctly.

The built-in debugger is a command parser which detects and notifies the user of the exact point in a command string that has an error. For instance, suppose you try to type the command “SETUP:TXP:COUNT 10”, but instead

type the wrong command “SETUP:TXP:COUNT 10”. When this command is sent to the test set, the debugger would display the following error message:

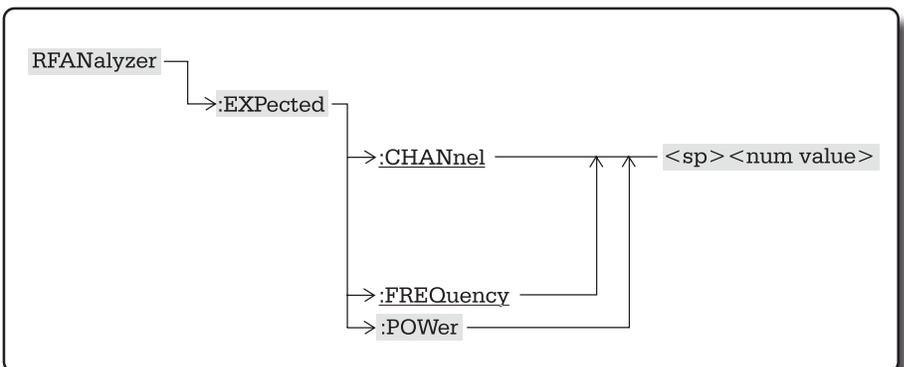
Undefined Header: SETUP:TXP:C0<err>UNT 10

Did you see the error? The <err> indicator clearly points out the position of the typing error. In the word COUNT, we accidentally typed in a zero instead of the letter O.

No matter how good your typing skills are, you still need to find the correct programming commands. With the 8960 Series 10, we provide documentation with “railroad diagrams,” which are graphical drawings that make it easy for programmers to build command syntax strings.

For speed and flexibility, the GPIB interface is decoupled from the manual user interface. This means the GPIB is not just an “automated button-pusher”; it is specifically designed with features and capabilities not needed by a front-panel user. Although these two interfaces are decoupled, it is often helpful to use the front panel to find a programming command. To help users find the correct railroad diagram (and programming command), a simple cross-reference is available which maps each front-panel key, menu, and field to the correct diagram.

Suppose one of the softkeys on the screen is labeled “Expected Power”. To find the correct command for setting this, simply look in the documentation index for “Expected Power”, and it will point you directly to the correct section of programming diagrams where you would find the following:



In addition to paper documentation, Agilent Technologies also provides manuals on CD-ROM and Web-based documentation.

Help is just a phone call away

Agilent Technologies provides a variety of services and solutions from basic training, to specialized consulting and custom test systems. Professionals at Agilent Technologies have a breadth of experience and depth of knowledge. They can provide help for almost any test, measurement, or manufacturing situation. For the 8960 Series 10, we have available on-line support services, as well as custom training on manual and remote operation. Please contact your Agilent Technologies representative for more information. They are an integral part of every product and will be glad to help you find out more about our products and services.

For more information about the Agilent 8960 Series 10
Wireless Communications Test Set visit our web site at:

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

For more information about Agilent Technologies test and measurement products, applications, services, and for a current sales office listing, visit our web site at:

<http://www.agilent.com/find/tmdir>

You can also contact one of the following centers and ask for a test and measurement sales representative.

United States:

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.
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(tel) 1 877 894 4414

Europe:

Agilent Technologies
Test & Measurement
European Marketing Organisation
P.O. Box 999
1180 AZ Amstelveen
The Netherlands
(tel) (31 20) 547 9999

Japan:

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

Latin America:

Agilent Technologies
Latin American 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
(tel) 1-800 629 485 (Australia)
(fax) (61 3) 9272 0749
(tel) 0 800 738 378 (New Zealand)
(fax) (64 4) 802 6881

Asia Pacific:

Agilent Technologies
24/F, Cityplaza One, 1111 King's Road
Taikoo Shing, Hong Kong
(tel) (852) 3197 7777
(fax) (852) 2506 9284

© Copyright Agilent Technologies 1998

All Rights Reserved

Technical data subject to change

Printed in U.S.A. 12/99

5968-7915E



Agilent Technologies

Innovating the HP Way