

Mobile WiMAX™ Manufacturing Test with the N8300A

*The First Steps Towards
Test Automation*

Application Note



Table of Contents

Introduction	3
The challenge	3
Path to manufacturing	3
Production automation for high demand	3
Programming and applications	3
The Test Plan	4
Test setup	6
Basic settings	7
Phase 1 – Basic check (“Alive”)	8
Phase 2 – Output power calibration (“Tx Cal”)	9
Phase 3 – Received signal strength indicator (“Rx Cal”)	13
Phase 4 – Spectral mask and power verification (“Tx Verify”)	15
Phase 5 – Receiver sensitivity and maximum input level (“Rx Verify”)	19
Phase 6 – Modulation accuracy (“Tx Quality”)	22
Error Handling and Programming Issues	25
Error checking	25
Pacing tests	25
Timeouts	26
Advanced error handling	27
Test Speed	28
Maximizing chipset control	28
Parallel test	28
Other	28
Glossary	29
Appendix	30
Appendix A – DUT command list	30
Appendix B – Additional measurement results	31
Appendix C – Building sequence files	32
Appendix D – Configuring the N8300A uplink map	34
Sales and Service	36

Introduction

The challenge

Mobile WiMAX manufacturing test is a challenge faced by many companies wishing to ramp to volume production. Choosing the right test equipment is important for achieving this quickly and efficiently. The N8300A wireless networking test set is specifically designed for WiMAX™ manufacturing test. It is recognized as being an optimized one box tester (OBT) that delivers fast test speed for RF test. This kind of RF manufacturing test requires devices be put in test modes to facilitate transmitter and receiver testing, without the use of protocol. Fast measurement speed allows manufacturers to reduce cost and optimize test plans.

Path to manufacturing

The N8300A leverages Agilent Technologies' leading research and development applications and algorithms to deliver consistency in test throughout a WiMAX test lifecycle. The N8300A offers a wide range of measurements, providing flexibility for test and troubleshooting. This allows a production engineer to relate measurements to other test equipment and provide traceability back to R&D. This is particularly important in prototype production testing before volume ramps.

Production automation for high demand

Rapid test speed, high end specifications, and a wide choice of measurements are clearly of great importance to the manufacturing engineer. However, having specified and purchased the equipment, the next goal is to integrate the tester with the device to be tested. Getting started quickly with the instrument is paramount in order to reduce time to market and costs in setup. The N8300A front panel display and controls provide measurement capability out of the box so that initial characterization is straightforward.

The next problem to address is full automation in a production context. The provision of a good programming interface to the instrument is one factor that must be considered. A remote interface that is easily integrated into all programming environments with a logical command structure helps aid this task. The N8300A's compatibility with the Agilent MXG vector signal generator and MXA signal analyzer instruments means that code can be easily transferable from R&D through to production.

Programming and applications

This document describes how to take the first steps toward WiMAX manufacturing test using the N8300A wireless networking test set. It begins by outlining a series of key phases in test and the test setup. Each phase of the example test plan is then described in detail and shows how a test plan is implemented programmatically.

The purpose of the application note is to help explain which measurements are required to test an 802.16e mobile subscriber station (MS) in manufacturing. Moreover, it shows how to execute standard commands for programmable instruments (SCPI) with the N8300A instrument. By focusing on real customer requirements and problems, this application note allows any WiMAX production or application engineer to get started with ease. This is aided in part by the industry standard SCPI interface of the N8300A, which is widely recognized as being a benefit to RF test/instrumentation engineers and similar to other Agilent instruments.

SCPI programming examples are provided for each specific phase of the example test plan that might be executed in manufacturing of WiMAX devices. The test plan itself is general and not designed to fit any one particular device or test requirement. However, the key tests presented will undoubtedly be a consideration for any WiMAX manufacturer and the example code can be easily tailored for specific requirements. In order to introduce the N8300A SCPI commands and relate them to the test phase, pseudo code provides a way of illustrating how a test sequence could be structured and what functions it might call to operate on both the instrument and device under test (DUT).

The example test plan is not designed to address specific requirements of the WiMAX Forum's® radio conformance test (RCT) or protocol conformance test (PCT). It is primarily intended to check that the device has been manufactured properly and is functioning correctly. Naturally, the tests chosen in the manufacturing test plan will relate to previous design choices and tests. Conformance to PCT, RCT, and other regulatory standards is a wider issue that is not addressed specifically by manufacturing test.

The Test Plan

This section explains the test plan that will be undertaken in this document. The test plan is broken down into stages explaining why each phase is performed and what benefit is achieved.

Figure 1 shows how a WiMAX test plan may be structured. The first column contains a name that is often associated with each test phase. Each test phase (second column) has a minimum of one RF

parameter (third column) that is measured and compared versus a required specification or limit. The N8300A measurement (fourth column) indicates which N8300A measurement can deliver the RF variable that is required for analysis. This test plan is intended to be generic and will require changes to suit particular devices. However, it is a logical approach to take and breaks down the test plan for discussion.

Name	Test phase	Primary RF test parameter	N8300A measurement
Alive	Basic check	Power (dBm)	Waveform
Tx Cal	Output power calibration	Power (dBm)	Waveform
Rx Cal	Received signal strength indicator calibration	Received signal strength indicator (dBm)	N/A – taken from DUT control (stimulus from N8300A)
Tx Verify	Spectral mask and power verify	Failed spectrum (%)	Spectral mask
Rx Verify	Receiver sensitivity and maximum input level	Packet error rate (%)	N/A – taken from DUT control (stimulus from N8300A)
Tx Quality	Modulation accuracy	Relative constellation error (%)	Modulation analysis

Figure 1. WiMAX test plan example

The Test Plan (continued)

The “Alive” test provides a quick check as to whether a device is able to function to a basic level before beginning proper test or calibration. The “Tx Cal” calibrates the transmitter path of the device. The “Rx Cal” calibrates the receiver path. The “Tx Verify” verifies the operation of the transmitter following calibration. The “Rx Verify” verifies the operation of the receiver following calibration. Finally, the “Tx Quality” checks the quality of the transmitter having already been calibrated and verified.

The order of the test phases is important and must match the device’s characteristics. Checking whether the device can communicate and perhaps output a signal is a good way to eliminate inoperable devices. This is one of the initial steps. Calibrating both transmitter and receiver determines whether the transmitter and receiver paths are operational. It also forms a basis for further analysis and verification. Depending upon the device quality and characteristics, some manufacturers may choose

to do calibration and verification sequentially. It depends on where it is thought the test problems lie and how to expose them fastest within the test plan. Checking modulation quality at the end ensures that the device is functional and passing RF specifications with bursted signals.

Before starting measurements, typically those done through remote programming should be accomplished using a single measurement operation (not continuous). The following command can be used to setup a single measurement operation:

```
INIT:CONTinuous OFF
```

The single¹/continuous state affects all measurements.

¹ Single refers to a measurement operation and not acquisitions. For example, if averaging is switched on in a single measurement operation, the instrument must obtain a number of captures as defined in the averaging. For example, if averaging is set to 10, then 10 different acquisitions must be made to finish the measurement operation.

The Test Plan (continued)

Test setup

The N8300A and device setup is shown in Figure 2. The N8300A has one RF input/output port and one RF output port. The output port provides the downlink signals into the device. Signals from the DUT that are to be tested are sent to the input/output port of the N8300A.

In Figure 2, the green line indicates a signal being output from the source of the N8300A (downlink). The blue line represents a signal from the device that inputs into the N8300A (uplink). These colors will be used later in the programming examples to make it clear when a command is operating the source or analyzer in the N8300A.

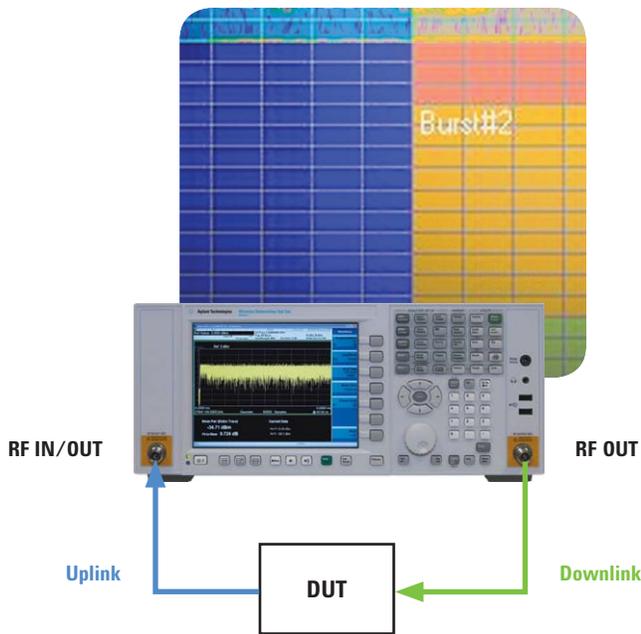


Figure 2. N8300A and device test setup

The following assumptions have been made about the WiMAX device being tested²:

1. Test modes exist to facilitate transmitter and receiver testing
2. No use of protocol is required
3. The device requires a downlink signal in order to transmit on the uplink
4. It can generate a signal at a known power level and frequency
5. It can report a count of the number of downlink sub-frames successfully received
6. It can report the RSSI of the received signal

² Alternative setups are explored in the *Test Speed* section.

The Test Plan (continued)

Basic settings

The commands that are presented in *The Test Plan* section relate to specific measurement parameters and operations. Before they can be used successfully, there are some basic settings that must be addressed.

Input power range

One of the most important settings is the N8300A input power range, which prepares the instrument to receive at a required input power level. Power range is a basic setting that must be configured for all measurements. Normally the range is set 12 to 15 dB higher than the average power being transmitted by a device to accommodate OFDM signals, which have high peak-to-average ratios. An example SCPI command is as follows:

```
:SENSe:POWer:RANGe -30 dBm
```

Trigger type

The following measurement categories are available on the N8300A:

1. Waveform – provides time domain traces and measurements such as average and peak power
2. Spectral emissions mask (SEM) – for spectral analysis of the output from devices versus known limits
3. Power stat CCDF – advanced analysis to provide the power statistics of a signal
4. Modulation analysis – provides a range of demodulation analysis including EVM calculations
5. Adjacent channel power (ACP) – measures the ratio of power present in adjacent channels

Independent trigger controls are available for each measurement category. Trigger options are free run, external, and RF burst. For example, it is possible to have a free run trigger configured for power measurements under the waveform measurements, and have a burst trigger configured for any modulation analysis measurement (such as RCE and preamble frequency error).

Trigger level

For some measurements the N8300A is required to measure bursted signals. This can be set using the RF burst trigger option. This requires a trigger level to be set. Normally a trigger level 10 to 15 dB below the average power is recommended. An example SCPI command is as follows:

```
:TRIGger:SEQuence:RFBurst:LEVel:ABSolute -45 dBm
```

The following diagram summarizes how the power range and trigger level might be set for measurements (trigger level is only required for bursted signals). The input power range and trigger level of the N8300A is based on the expected average power transmitting from the device.

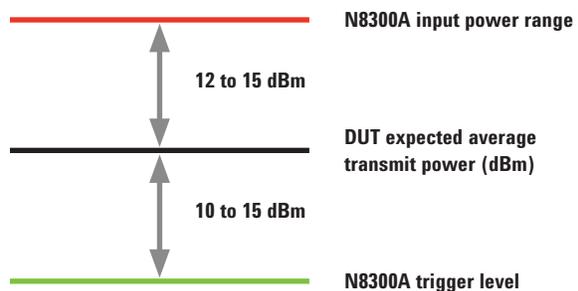


Figure 3. Setting power range and trigger level for an expected power level

Frequency

Setting the frequency is a basic requirement for any transmitter or receiver measurement. For example:

```
:SENSe:FREQuency:CENTer 2.42 GHz
```

This example sets the frequency to 2.42 GHz for both the N8300A analyzer and N8300A source. This sets up the downlink and uplink frequency to the same channel.

The Test Plan (continued)

Phase 1 – Basic check (“Alive”)

An “Alive” test is a very general test that is designed to be a quick check for basic RF operation. This is likely to be an RF test that would follow tests already completed on current, voltage, and digital connectivity to see if the device can respond to commands. It was stated previously in the introduction that, for this example test plan, it is assumed the device requires a downlink signal in order to transmit on the uplink. This implies that the chipset software can synchronize the device to the downlink signal from the N8300A. This capability means that the Alive test can also be a check that a synch condition is being satisfied. This is determined by the chipset software.

Test procedure

```

N8300A: selectSegmentFile( 16E_10M_DL_PUSC_MULT )
N8300A: setSampleRate( 22.4 MHz )
N8300A: setRuntimeScaling( 85 )
N8300A: setSourceTriggerSource( BUS )
N8300A: setSourceTriggerMode( CONTINUOUS )
N8300A: switchArbOn()
DUT3: setFrequency( 2.422 GHz )
N8300A: setFrequency( 2.422 GHz )
N8300A: setOutputPower( -30 dBm )
N8300A: setOutputState()
DUT: readSynchCondition()

```

The test procedure above sets up the source to transmit continuously on the downlink. The device control software then checks that it can synchronize the device to that downlink signal.

³ See *Appendix A – DUT command list* for a full list of DUT PSEUDO code references used in the examples.

N8300A SCPI commands⁴

Operation	SCPI command
selectSegmentFile()	:SOURce:RADio:ARB:WAVEform “WFM1:16E_10M_DL_PUSC_MULT”
setSampleRate()	:SOURce:RADio:ARB:CLOCK:SRATe 22.4 MHz
setRuntimeScaling()	:SOURce:RADio:ARB:RSCaling 85
setSourceTriggerSource()	:SOURce:RADio:ARB:TRIGger:SOURce BUS
setSourceTriggerMode()	:SOURce:RADio:ARB:TRIGger:TYPE CONTInuous
switchArbOn()	:SOURce:RADio:ARB:STATE ON
setFrequency()	:SENSe:FREQuency:CENTer 2.422 GHz
setOutputPower()	:SOURce:POWer:LEVel:IMMEDIATE:AMPLitude -30 dBm
setOutputState()	:SOURce:OUTPut:STATe ON

The SCPI commands above setup the generator to transmit to the device. The detail of each command for the N8300A source is discussed in more detail in the receiver tests in the *Phase 3* and *Phase 5* sections of this application note.

⁴ As with all examples throughout the application note, typical parameters have been inserted to demonstrate how to use each command (parameters such as file names, frequencies, and power levels).

The Test Plan (continued)

Phase 2 – Output power calibration (“Tx Cal”)

The output stage of a conventional WiMAX device contains a power amplifier with means of controlling output power. By writing to the output power control register, the actual output power of the device can be controlled. The purpose of the calibration phase is to build up a table that maps the value in the power control register to the actual output power of the device measured in dBm by the N8300A test equipment. This table can then be used by the DUT to work out what power control value it should use if it is asked to generate a specific power level.

There is likely to be a frequency dependency in the power amplifier. It may be the case that a value of 80 in the power control register causes the amplifier to generate -20 dBm at 2.4 GHz, but only -27 dBm at 3.5 GHz. Thus the table normally has to have two dimensions – one for the power control value and one for frequency. These values are normally written to EEPROM memory on the device at the end of phase 3 (having completed receiver calibration as well as transmitter) so that the device can use them in the future to generate accurate power levels.

At this very early stage of test it is likely that the test engineer has little evidence to confirm the device can transmit, so it is probably best to get the

device to generate a continuously-modulated signal. This has the benefit of the instrument using a free-run trigger. So if the device is completely broken, the instrument does not have to initiate a measurement, then detect no signal after a period of time, and finally produce a timeout error that then must be handled by the test executive software.

In order to make the power measurement it is necessary to pick a measurement duration. The longer the acquisition, the longer the measurement takes. However, longer acquisitions should produce more accurate results. A measurement duration should be chosen to suit the DUT. In this case a value of 500 μ s is chosen. This is a short acquisition, keeping the test time down. Optimization such as reducing capture acquisition time is recommended once parametric results have been analyzed and monitored on a large scale so that any changes can be monitored carefully for measurement accuracy.

During the design phase of a WiMAX device, data will have been collated to provide an idea of the power levels expected for a given power control value. If the power reading is significantly different from that which was expected for a given power control value, a faulty device will be detected very quickly.

The Test Plan (continued)

Phase 2 – Output power calibration (“Tx Cal”) (continued)

Test procedure

The procedure to setup the downlink signal on the N8300A is virtually the same as shown in *Phase 1*. The analyzer and source frequency is configured later (as it is being looped):

```
N8300A: selectSegmentFile( 16E_10M_DL_PUSC_MULT )
N8300A: setSampleRate( 22.4 MHz )
N8300A: setRuntimeScaling( 85 )
N8300A: setSourceTriggerSource( BUS )
N8300A: setSourceTriggerMode( CONTINUOUS )
N8300A: switchArbOn()
N8300A: setOutputPower( -50 dBm )
```

The procedures above can now be extended to allow the N8300A to make power measurements:

```
N8300A: setPowerOnlyMode()
N8300A: setFreeRunTriggerSource( IMMEDIATE )
N8300A: setMeasurementDuration( 500 us )
N8300A: setOutputState()

foreach frequency F in { list of frequencies }
  DUT: setFrequency( F )
  N8300A: setFrequency ( F )

  foreach DUTPowerControlValue PCV in { list of
                                         powerControlValues }
    DUT: setPowerControlValue( PCV )
    N8300A: setPowerRange ( expected power level in dBm )
    N8300A: measurePower()
    N8300A: checkNotOverrange()
```

The test procedures above set the N8300A to measure average power only (in order to optimize the instrument for test speed), setting the trigger to free-run and setting the acquisition length. The device then transmits on a particular frequency as determined by an entry in an array of frequencies. In each case the N8300A analyzer also matches the frequency of the device. The device control software then writes to the power control register to generate a power level (as determined by a value in an array of power control values). The N8300A is setup to receive at the expected power level from the device, measures the power, and checks that the analog to digital converter is not overloaded (ensuring valid results for accurate measurements).

The Test Plan (continued)

Phase 2 – Output power calibration (“Tx Cal”) (continued)

N8300A SCPI commands

Operation	SCPI command
selectSegmentFile()	:SOURce:RADio:ARB:WAVeform “WFM1:16E_10M_DL_PUSC_MULT”
setSampleRate()	:SOURce:RADio:ARB:CLOCK:SRATe 22.4 MHz
setRuntimeScaling()	:SOURce:RADio:ARB:RSCaling 85
setSourceTriggerSource()	:SOURce:RADio:ARB:TRIGger:SOURce BUS
setSourceTriggerMode()	:SOURce:RADio:ARB:TRIGger:TYPE CONTInuous
switchArbOn()	:SOURce:RADio:ARB:STATE ON
setOutputPower()	:SOURce:POWer:LEVel:IMMEdiate:AMPLitude -50 dBm
setPowerOnlyMode()	:SENSe:WAVeform:PONLY:STATe ON
setFreeRunTriggerSource()	:SENSe:WAVeform:TRIGger:SOURce IMMEdiate
setMeasurementDuration(500)	:SENSe:WAVeform:SWEep:TIME 500 us
setOutputState()	:SOURce:OUTPut:STATe ON
setFrequency(F)	:SENSe:FREQuency:CENTer 2.422 GHz
setPowerRange()	:SENSe:POWer:RANGe -30 dBm
measurePower()	:READ:WAVeform1?
checkNotOverrange()	:STATus:QUESTionable:INTegrity?

The first seven commands are similar to what was used before in *Phase 1* for the downlink signal.

The Test Plan (continued)

Phase 2 – Output power calibration (“Tx Cal”) (continued)

The next SCPI command shows how to set the Power Only mode, which only returns average power (also referred to as mean power). If this was not switched on, then a set of seven results would be returned by the READ:WAVEform1? command: sample time, mean power, mean power averaged, number of samples, peak to mean ratio, maximum power, and minimum power. For most production requirements the mean power result from the READ:WAVEform1? command is all that is required and this can be returned in the quickest possible time in the Power Only mode.

Note that the trigger is set to free run and the measurement duration to 500 μ s. The source is then switched on in order to transmit on the downlink. The N8300A source and analyzer frequency is set.

The power range is also set accordingly. There is no requirement to setup more complicated triggering because this measurement assumes a continuously modulated signal under test being output from the DUT.

The :STATus:QUEStionable:INTEgrity? command is used to check whether an over-range condition has occurred in the measurements. The command returns a decimal value of the sum of the bits in the questionable integrity event register. So, if bit 4 of this register is set for an over-range condition, then decimal 16 is returned (HEX 0x10). A decimal 0 means that no errors are present, which also means no over-range condition occurred:

Table 1. Questionable status register check for over-range

7	6	5	4	3	2	1	0
			0				

The Test Plan (continued)

Phase 3 – Received signal strength indicator (“Rx Cal”)

The received signal strength indicator (RSSI) value indicates the strength of the downlink sub-frame in dBm, as measured by the DUT. Just as with transmitter calibration, the DUT will normally need to be supplied with a set of signals of varying power levels and frequencies from which a table in EEPROM can be populated with information. This allows the DUT to convert from a unitless internal signal strength value to a calibrated result in dBm.

One important consideration is how frequently the DUT calculates the RSSI value and is read back by the controlling PC. If it is not known exactly when the DUT is going to calculate the RSSI, a signal must be sent for a period long enough to guarantee that the RSSI will have been recalculated. It is common to send a downlink signal one thousand times. So for a 5 ms WiMAX frame this will take 5 s to complete in the instrument. If the number of packets can be reduced at no expense of accuracy, then test time can be reduced.

Note that if the DUT calculates RSSI on every downlink sub-frame, this calibration phase can be very fast.

Test procedure

```
N8300A: selectSequenceFile( 16E_10M_DL_PUSC_MULT )
N8300A: setSampleRate( 22.4 MHz )
N8300A: setRuntimeScaling( 85 )
N8300A: setAlcMarkerRouting( M4 )
N8300A: setSourceTriggerSource( BUS )
N8300A: setSourceTriggerMode( SINGLE )
N8300A: setOutputState()
N8300A: switchArbOn()

foreach frequency F in { list of frequencies }
  DUT: setFrequency( F )
  N8300A: setFrequency( F )

  foreach sourcePowerLevel PL in { list of powerLevels }
    N8300A: setOutputPower( PL )
    N8300A: triggerSequence()
    wait() // wait until RSSI update guaranteed
  DUT: readRSSIvalue()
```

The preceding example is very similar to the receiver sensitivity and maximum input level phase covered later in the document. (See *Phase 5*.) In this test, the variable being read from the device is the RSSI value. The first eight commands setup the source to transmit a particular signal for the downlink. The code then loops by each channel required for test, setting both the DUT and N8300A to the correct frequency. For each frequency channel the code loops the N8300A source power as determined by an entry in array of power levels. Once the power level is set in each case, a sequence is triggered to playback the downlink signal a fixed number of times. The device then provides the RSSI value which is retrieved by the test software after the device receives a required number of bursts.

The Test Plan (continued)

Phase 3 – Received signal strength indicator (“Rx Cal”) (continued)

N8300A SCPI commands

Operation	SCPI command
selectSequenceFile()	:SOURce:RADio:ARB:WAVEform “SEQ:16E_10M_DL_PUSC_MULT”
setSampleRate()	:SOURce:RADio:ARB:CLOCK:SRATe 22.4 MHz
setRuntimeScaling()	:SOURce:RADio:ARB:RSCaling 85
setAlcMarkerRouting()	:SOURce:RADio:ARB:MDESTination:ALCHold M4
setSourceTriggerSource()	:SOURce:RADio:ARB:TRIGger:SOURce BUS
setSourceTriggerMode()	:SOURce:RADio:ARB:TRIGger:TYPE SINGLE
setOutputState()	:SOURce:OUTPut:STATe ON
switchArbOn()	:SOURce:RADio:ARB:STATE ON
setFrequency(F)	:SENSe:FREQuency:CENTer 2.422 GHz
setOutputPower(PL)	:SOURce:POWer:LEVel:IMMEDIATE:AMPLitude –65 dBm
triggerSequence()	:SOURce:TRIGger

The first command uses a sequence file to playback a signal a fixed number of times (not playing a signal continuously as in phases 1 and 2). The :SOURce:RADio:ARB:WAVEform “SEQ:16E_10M_DL_PUSC_MULT” command selects a downlink sequence file for playback. It assumes that the file is in volatile memory. If it was not, a file can be copied from non-volatile using the following command:

```
:SOURce:MMEMory:COpy “NVWFM:10M_DL_PUSC_MULT”, “WFM1:10M_DL_PUSC_MULT”
```

The sampling rate, run time scaling, and marker routing is explicitly configured in the SCPI commands above. However, for the test waveforms provided with the N8300A, it is sufficient to select them for playback without configuration as these files contain the correct default settings.

The run time scaling value sets the waveform amplitude as a percentage of full scale. Marker 4 is routed to the automatic level control hold function. Marker 4 is the default marker used for the ALC as per Agilent Signal Studio products used for producing waveform files. The triggering type sets the waveform’s response to a trigger – in this case to playback once.

Triggering is also set to operate over the bus. This implies that one of the interfaces such as USB or LAN will enable triggering using the :SOURce:TRIGger command when it is required.

It is assumed that the N8300A sequence contains enough downlink bursts to calculate the RSSI in the device. The N8300A source is then switched on, the frequency setup, the output power set and the sequence triggered to begin playback to the device.

The Test Plan (continued)

Phase 4 – Spectral mask and power verification (“Tx Verify”)

Phase 4 contains two types of measurements in the N8300A: power (as with phase 2) and spectral mask. The power verification is meant to verify the success of the Tx Cal in phase 2. Within phase 2, the device is calibrated across a number of points in the operational range. The power verification may check the power output over a number of different frequencies and also at different power levels, potentially interpolating between the exact power levels calibrated previously.

A spectral analysis of the signal is required to ensure that the device outputs a signal that fits standards and regulatory requirements. This measurement is optimized for speed and is an absolute mask as required in order to meet such specifications. The test is normally done at maximum output of the device.

Note that the N8300A contains additional flexibility within the spectral emission mask (SEM) and adjacent channel power (ACP) measurements for more detailed testing. This additional flexibility is likely to be relevant prototype testing and verification testing. The following example is a standard mask completed over a fixed integration bandwidth.

Test procedure

```
N8300A: selectSegmentFile( 16E_10M_DL_PUSC_MULT )
N8300A: setSampleRate( 22.4 MHz )
N8300A: setRuntimeScaling( 85 )
N8300A: setAlcMarkerRouting( M4 )
N8300A: setSourceTriggerSource( BUS )
N8300A: setSourceTriggerMode( SINGLE )
N8300A: switchArbOn()
N8300A: setOutputState()
N8300A: setOutputPower( -50 )
```

```
N8300A: setMode2ms()
N8300A: setModeDownlinkRatio( 0 )
N8300A: setPowerTriggerSource( RF BURST )
N8300A: setMaskTriggerSource( RF BURST )
N8300A: setTriggerSlope( POSITIVE )
N8300A: setTriggerDelay( -20 us )
N8300A: setTriggerDelayState()
N8300A: setTriggerHoldoff( 50 us )
N8300A: setTriggerHoldoffState()
N8300A: setPowerOnlyMode()
N8300A: setMeasurementDuration( 500 us )
```

```
foreach frequency F in { list of frequencies }
  N8300A: setFrequency( F )
  DUT: setFrequency( F )
```

```
foreach DUTPowerLevel DPL in { list of dBmPowerLevels }
  DUT: setPowerLevel( DPL )
  N8300A: setPowerRange( PR )
  N8300A: setTriggerLevel( TL )
  N8300A: initPower()
  N8300A: triggerMeasurement()
  N8300A: measComplete()
  N8300A: fetchPowerResult()
  N8300A: checkNotOverrange()
  if (DPL == MaxPower)
    // verify results are valid
  N8300A: initSpectralMeasurement()
  N8300A: triggerMeasurement()
  N8300A: measComplete()
  N8300A: fetchSpectralResults()
  N8300A: CheckNotOverrange()
```

The test procedure loops by frequency and device power level for the power verification, and frequency only for the spectral mask test. The trigger level changes as the device is sending the uplink signal at different levels to the N8300A to validate the transmission range. When the highest power is reached a spectral mask test is performed.

The Test Plan (continued)

Phase 4 – Spectral mask and power verification (“Tx Verify”) (continued)

N8300A SCPI commands

Operation	SCPI command
selectSegmentFile()	:SOURce:RADio:ARB:WAVeform “WFM1:16E_10M_DL_PUSC_MULT”
setSampleRate()	:SOURce:RADio:ARB:CLOCK:SRATe 22.4 MHz
setRuntimeScaling()	:SOURce:RADio:ARB:RSCaling 85
setAlcMarkerRouting()	:SOURce:RADio:ARB:MDEStination:ALCHold M4
setSourceTriggerSource()	:SOURce:RADio:ARB:TRIGger:SOURce BUS
setSourceTriggerMode()	:SOURce:RADio:ARB:TRIGger:TYPE SINGLE
switchArbOn()	:SOURce:RADio:ARB:STATE ON
setOutputState()	:SOURce:OUTPut:STATe ON
setOutputPower(-50)	:SOURce:POWer:LEVel:IMMEDIATE:AMPLitude -50
setMode2ms()	:SENSe:RADio:STANdard:FDURation 2 ms
setModeDownlinkRatio()	:SENSe:RADio:STANdard:DRATio 0
setPowerTriggerSource()	:SENSe:WAV:TRIGger:SOURce RFBurst
setMaskTriggerSource()	:SENSe:SMASK:TRIGger:SOURce RFBurst
setTriggerSlope()	:TRIGger:SEQuence:SLOPe POSitive
setTriggerDelay(-20)	:TRIGger:SEQuence:RFBurst:DELay -20 us
setTriggerDelayState()	:TRIGger:SEQuence:RFBurst:DELay:STATe ON
setTriggerHoldoff(50)	:TRIGger:SEQuence:HOLDoff 50 us
setTriggerHoldoffState()	:TRIGger:SEQuence:HOLDoff:STATe 1
setPowerOnlyMode()	:SENSe:WAVeform:PONLy:STATe ON
setMeasurementDuration(500)	:SENSe:WAVeform:SWEp:TIME 500 us
setFrequency(F)	:SENSe:FREQuency:CENTer 2.422
setPowerRange(PR)	:SENSe:POWer:RANGe -15
setTriggerLevel(TL)	:TRIGger:SEQuence:RFBurst:LEVel:ABSolute -45 dBm
initPower()	:INIT:WAVeform1
triggerMeasurement()	:SOURce:TRIGger
measComplete()	STATus:OPERation:CONDition? // When the result has bit 4 (decimal 16 / HEX 0x10) clear, the measurement has completed.
fetchPowerResult()	:FETCh:WAVeform1?
checkNotOverrange()	:STATus:QUEStionable:INTegrity?
initSpectralMeasurement()	:INIT:SMASK
triggerMeasurement()	:SOURce:TRIGger
measComplete()	STATus:OPERation:CONDition? // When the result has bit 4 (decimal 16 / HEX 0x10) clear, the measurement has completed.
fetchSpectralResults()	:FETCh:SMASK?
checkNotOverrange()	:STATus:QUEStionable:INTegrity?

The Test Plan (continued)

Phase 4 – Spectral mask and power verification (“Tx Verify”) (continued)

The first nine SCPI commands setup the downlink signal. The signal is required to be played back from the N8300A source whenever the device needs to be stimulated and then the transmit measurements can be made on the resultant uplink signal returned from the DUT. The signal is set to be played back once. Note that the RF is set to on and the downlink signal will not playback until triggered.

The next four **SENSe** commands setup the analyzer. The command `:SENSe:RADio:STANdard:FDURation 2 ms` sets the frame duration parameter specifying the OFDMA frame duration in units of time. The `:SENSe:RADio:STANdard:DRATio 0 normal` sets the ratio of the downlink sub-frame length to the frame length. Both the spectral mask and power measurements are then configured for burst triggering. When testing an RF burst trigger it is necessary to consider a few options:

1. Trigger level – a power level in dBm. This is the power that causes the measurement to be triggered
2. Trigger slope – in order to analyze the uplink sub-frame a positive trigger slope is chosen
3. Trigger delay – for most accurate results it is best if the analysis algorithm can “see” both edges of the burst. A trigger delay of, for example, say $-20 \mu\text{s}$ permits this
4. Trigger holdoff – this prevents the measurement from re-triggering during the burst. Switching this “on” ensures that the measurement will not trigger unless the observed input level has been below the trigger level for the period of time specified in the trigger holdoff setting

It is possible to reduce the amount of time that is captured and analyzed by setting the frame duration value to the shortest period that completely captures the uplink sub-frame. Uplink sub-frames are typically 1.2 ms, so a 2 ms frame duration is a good choice. If a 2 ms capture is performed starting $20 \mu\text{s}$ before the start of the uplink, the N8300A shall see zero percent of the downlink. As a result the downlink ratio must be set to 0%.

Note the position of the `:SOURce:TRIGger` command in relation to the `INIT` command. `INIT` should be issued before triggering the source output in order to arm the N8300A analyzer. The output of the N8300A source with a downlink signal then causes the device to respond with an uplink signal.

Before using the `FETCh` command for the power measurement, it is extremely important to check that the measurement was triggered and returned valid data. This is done using the `:STATus:OPER:COND?` query. By polling the instrument with this query, when the result has bit 4 (decimal 16 or HEX 0x10) clear, the measurement has completed.

Table 2. Status operation register check for operation complete

7	6	5	4	3	2	1	0
			0				

The Test Plan (continued)

Phase 4 – Spectral mask and power verification (“Tx Verify”) (continued)

This is an essential part of coding the instrument to make sure that measurement timeouts for burst triggering are handled correctly. For example, if this command was not used and a faulty device never produced an expected signal, a FETCH command would be requesting data that was not available. A *OPC? query cannot be used to determine whether a measurement has completed. For more information on the use of this command and error handling refer to the *Error Handling and Programming Issues* section.

The spectral mask measurement follows the same structure as the burst power measurement – an INIT command, trigger of the source, a check on the measurement completion, a FETCH of the result, and then a check on the integrity of the result. The spectral mask result returns a percentage. The percentage value is the proportion of frequency in the measured spectrum that has failed the mask.

The spectral mask measurement assumes that the mask has already been configured or it uses the default settings. The mask can be changed using the following SCPI command:

```
CALC:SMASK:MDEF <mask points>
```

To create a custom mask, the parameter mask points must be setup in the form of an array (comma delimited) of the offset frequency (from the center) and the absolute amplitude pairs be connected. For example:

```
CALC:SMASK:MDEF -30.0e6,-37.0,-20.0e6,-37.0,-15.0e6,-28.58,-15.0e6,-25.0,-11.0e6,-25.0,-10.0e6,-13.0,-5.05e6,-13.0,-5.05e6,9.91e37,5.05e6,9.91e37,5.05e6,-13.0,10.0e6,-13.0,11.0e6,-25.0,15.0e6,-25.0,15.0e6,-28.58,20.0e6,-37.0,30.0e6,-37.0
```

Loss compensation can be added for the spectral mask measurement using the SCPI command:

```
CALCulate:SMASK:LCOMpensatio
```

For example, the following command accounts for 20 dbm loss:

```
:CALCulate:SMASK:LCOMpensatio -20
```

Within the spectral mask measurement it is also possible to return all amplitude and frequency points as required.

The Test Plan (continued)

Phase 5 – Receiver sensitivity and maximum input level (“Rx Verify”)

By this phase of the test plan it has been proven that the transmitter can generate signals. It is now time to find out whether the receiver can receive bursted signals and to verify the receiver calibration.

This test checks that the receiver successfully receives downlink packets that are sent to it at a known power level and frequency from the tester. Normally this would be done in two parts:

1. At a very lower power level, to ensure that the device sensitivity is as it should be
2. At the highest power level the device would be expected to be able to receive successfully (typically -30 dBm), sometimes known as saturation

The N8300A uses segment files and sequence files in this process. A segment file is a file which contains I/Q pairs representing a waveform (in this case the waveform is that of a single downlink sub-frame). A sequence file allows for playback of one or more segment files a set number of times.

For receiver sensitivity testing, normally a 64QAM downlink signal is used since it tests the accuracy of the receiver more than any other modulation format. Typically, a sequence file will be built which results in the repetition of a transmission of a sub-frame, or segment, for example, one thousand times.

Test procedure

```
N8300A: selectSequenceFile( 16E_10M_UL_PUSC_MULT )
N8300A: setSampleRate( 22.4 MHz )
N8300A: setRuntimeScaling( 85 )
N8300A: setAlcMarkerRouting( M4 )
N8300A: setSourceTriggerSource( BUS )
N8300A: setSourceTriggerMode( SINGLE )
N8300A: switchArbOn()
N8300A: setOutputState()
```

```
foreach frequency F in { list of frequencies }
  N8300A: setFrequency( F )
  N8300A: setOutputPower( lowest_P ) // set lowest power level
  N8300A: triggerSequence()
  While ( NOT sequenceComplete() )
    Wait();
  DUT: readPacketCount() // read packets RX
  if (readPacketCount() < threshold) // if not acceptable
    fail() // record fail

  N8300A: setOutputPower( highest_P ) // set highest power level
  N8300A: triggerSequence()
  While ( NOT sequenceComplete() )
    Wait();

  DUT: readPacketCount()
  if (readPacketCount() < threshold)
    fail()
```

The test procedure begins by setting up the N8300A’s source. First, a sequence file is selected, and then the sampling rate and runtime scaling will need to be setup.

Having selected the trigger source, the source is then setup to trigger once – this will playback the sequence file once. The N8300A arbitrary waveform generator is switched on and then the RF is enabled. Next, the N8300A is set to transmit on a particular frequency as determined by an entry in an array of frequencies. The output power level is also set in the instrument. With the instrument fully ready to transmit, it is triggered to playback. The test code must then check to see when the sequence has completed to know when to ask the device for the packet count it has successfully received.

The N8300A output is set to the highest signal level and the test repeated to check the DUT can also receive at this level.

The Test Plan (continued)

Phase 5 – Receiver sensitivity and maximum input level (“Rx Verify”) (continued)

N8300A SCPI commands

Operation	SCPI command
selectSequenceFile()	:SOURce:RADio:ARB:WAVeform “SEQ:16E_10M_UL_PUSC_MULT”
setSampleRate()	:SOURce:RADio:ARB:CLOCK:SRATe 22.4 MHz
setRuntimeScaling()	:SOURce:RADio:ARB:RSCaling 85
setAlcMarkerRouting()	:SOURce:RADio:ARB:MDEStination:ALCHold M4
setSourceTriggerSource()	:SOURce:RADio:ARB:TRIGger:SOURce BUS
setSourceTriggerMode()	:SOURce:RADio:ARB:TRIGger:TYPE SINGLE
switchArbOn()	:SOURce:RADio:ARB:STATE ON
setOutputState()	:SOURce:OUTPut:STATe ON
setFrequency(F)	:SENSe:FREQuency:CENTer 2.422 GHz
setOutputPower(P)	:SOURce:POWer:LEVel:IMMEDIATE:AMPLitude -65
triggerSequence()	:SOURce:TRIGger
sequenceComplete()	:SOURce:RADio:ARB:DONE?

The Test Plan (continued)

Phase 5 – Receiver sensitivity and maximum input level (“Rx Verify”) (continued)

The run time scaling value sets the waveform amplitude as a percentage of full scale. Marker 4 is routed to the automatic level control hold function. The triggering type sets the waveform’s response to a trigger – in this case to playback the sequence once. Triggering is also set to operate over the bus. This implies that one of the interfaces such as USB or LAN will enable triggering using the `:SOURCE:TRIGger` command when it is required. The arbitrary waveform generator is then enabled and then the RF enabled.

Depending on the device and test plan, it might not be required to do both receiver sensitivity and maximum input level testing at every frequency, rather to only perform sensitivity at some frequencies, and maximum input level at others. Some manufacturers may choose to continually step the power down and check the packet error rate in order to search for a more accurate degree of the actual sensitivity level. This is more a sensitivity search and it is typically a long test. It might be regarded as another test phase entirely such as Rx Quality.

Depending on the linearity of the power sensor in the WiMAX device’s receiver, the signals generated at this stage of the process may be enough to give the data that RSSI calibration needs.

There are a number of different ways to determine when to ask the device to return the received packet count. The best way to do this is to program the test code to use the N8300A to determine when the device should return the PER result. The command `:SOURCE:RADio:ARB:DONE?` returns 0 if a sequence is playing or 1 if the sequence is not playing. In a program, a “while” loop can be used to wait on a “1” being returned. Then the device can be asked for a PER result.

There are other ways to determine when the device needs to be polled for a result, but they tend to be less reliable. A timing loop could be used or the device could count up to a value. However, these approaches are likely to encounter problems. For example, timing would have to be modified if source waveforms were changed. Also, a device counting towards a value is not a good idea if the device does not receive the correct number of packets expected (so a timeout may occur). The latter two approaches are more successful if the N8300A is simply asked to transmit continually and not for a set number of packet count. Given the ability of the test set to playback a fixed number of packets using sequence files and being able to determine when that operation is complete, this is the optimum method for any reliable receiver test.

The Test Plan (continued)

Phase 6 – Modulation accuracy (“Tx Quality”)

Before embarking on this phase of the test plan, the values determined in the calibration phases completed previously should have been written to EEPROM in the DUT. By now, a lot is known about the device:

1. It can be instructed to generate signals at a power level in dBm that we request
2. It can report the power level of a received signal in dBm
3. It can receive downlink sub-frames successfully at the highest and lowest signal levels required

The last task is to ensure that when the DUT generates a signal at a known (calibrated) power level that it is being generated cleanly enough that a receiving base station will be able to decode the signal. The primary measurement of modulation accuracy is relative constellation error (RCE) which is often referred to as error vector magnitude (EVM). It is known that the device should be able to generate signals of a given level so it is possible to optimize triggering and so speed up the measurement in some cases.

It is normal to set the DUT to generate an uplink sub-frame at a high power level. This works especially well given that high-power transmissions tend to generate the worst RCE results.

When testing an RF burst trigger it is necessary to consider a few options:

1. Trigger level – a power level in dBm. This is the power that causes the measurement to be triggered
2. Trigger slope – in order to analyze the uplink sub-frame, a positive trigger slope is chosen
3. Trigger delay – for most accurate RCE results it is best if the analysis algorithm can see both edges of the burst. A trigger delay of $-20\ \mu\text{s}$ permits this
4. Trigger holdoff – this prevents the measurement from re-triggering during the burst. Switching this on ensures that the measurement will not trigger unless the observed input level has been below the trigger level for the period of time specified in the trigger holdoff setting

It is possible to reduce the amount of time that is captured and analyzed by setting the frame duration value to the shortest period that completely captures the uplink sub-frame. Uplink sub-frames are typically 1.2 ms, so a 2 ms frame duration is a good choice. If a 2 ms capture is performed starting $20\ \mu\text{s}$ before the start of the uplink, the N8300A shall see zero percent of the downlink. As a result the downlink ratio must be set to 0%.

Test procedure

```
N8300A: selectSegmentFile( 16E_10M_UL_PUSC_MULT )
N8300A: setSampleRate( 22.4 MHz )
N8300A: setRuntimeScaling( 85 )
N8300A: setSourceTriggerSource( BUS )
N8300A: setSourceTriggerMode( SINGLE )
N8300A: switchArbOn()
N8300A: setOutputState()
N8300A: setOutputPower( -50 )
N8300A: setMode2ms()
N8300A: setModeDownlinkRatio( 0 )
N8300A: setTriggerSource( RF BURST )
N8300A: setTriggerDelay( -20 us )
N8300A: setTriggerDelayState()
N8300A: setTriggerSlope( POSITIVE )
N8300A: setTriggerHoldoff( 50 us )
N8300A: setTriggerHoldoffState()
```

```
foreach frequency F in { list of frequencies }
  DUT: setFrequency( F )
  N8300A: setFrequency( F )
```

```
foreach DUTPowerLevel DPL in { list of dBmPowerLevels }
  DUT: setPowerLevel( DPL )
  N8300A: setPowerRange( PR )
  N8300A: setTriggerLevel( TL )
  N8300A: initEVMmeasurement()
  N8300A: triggerMeasurement()
  N8300A: measComplete()
  N8300A: fetchEVMresults()
  N8300A: checkNotOverrange()
```

The test procedure loops by frequency and power level, changing the downlink signal to the DUT. The trigger level does not need to be changed as the device is sending the uplink signal at a known high level to the N8300A.

The Test Plan (continued)

Phase 6 – Modulation accuracy (“Tx Quality”) (continued)

N8300A SCPI commands

Operation	SCPI command
selectSegmentFile()	:SOURce:RADio:ARB:WAVeform "WFM1:16E_10M_DL_PUSC_MULT"
setSampleRate()	:SOURce:RADio:ARB:CLOCK:SRATe 22.4 MHz
setRuntimeScaling()	:SOURce:RADio:ARB:RSCaling 85
setAlcMarkerRouting()	:SOURce:RADio:ARB:MDEStination:ALCHold M4
setSourceTriggerSource()	:SOURce:RADio:ARB:TRIGger:SOURce BUS
setSourceTriggerMode()	:SOURce:RADio:ARB:TRIGger:TYPE SINGLE
switchArbOn()	:SOURce:RADio:ARB:STATE ON
setOutputState()	:SOURce:OUTPut:STATe ON
setOutputPower (-50)	:SOURce:POWer:LEVel:IMMediate:AMPLitude -50
setMode2ms()	:SENSe:RADio:STANdard:FDURation 2 ms
setModeDownlinkRatio()	:SENSe:RADio:STANdard:DRATio 0
setTriggerSource()	:SENSe:EVM:TRIGger:SOURce RFBurst
setTriggerLevel(-45)	:TRIGger:SEQuence:RFBurst:LEVel:ABSolute -45
setTriggerDelay(-20)	:TRIGger:SEQuence:RFBurst:DELay -20 us
setTriggerDelayState()	:TRIGger:SEQuence:RFBurst:DELay:STATe
setTriggerSlope()	:TRIGger:SEQuence:SLOPe POSitive
setTriggerHoldoff(50)	:TRIGger:SEQuence:HOLDoff 50 us
setTriggerHoldoffState()	:TRIGger:SEQuence:HOLDoff:STATe 1
<hr/>	
setFrequency(F)	:SENSe:FREQuency:CENTer 2.422 GHz
<hr/>	
setPowerRange(PR)	:SENSe:POWer:RANGe -15
setTriggerLevel(TL)	:TRIGger:SEQuence:RFBurst:LEVel:ABSolute -45 dBm
initEVMmeasurement()	:INIT:EVM
triggerMeasurement()	:SOURce:TRIGger
measComplete()	:STATus:OPERation:CONDition? // When the result has bit 4 (decimal 16 / HEX 0x10) clear, the measurement has completed.
readEVMresults()	:FETCh:EVM?
checkNotOverrange()	:STATus:QUEStionable:INTegrity?

The Test Plan (continued)

Phase 6 – Modulation accuracy (“Tx Quality”) (continued)

Configuration of the modulation accuracy measurement is very similar to that of the spectral mask and power verification. First, a downlink signal is setup by selecting a waveform segment file and choosing to play it back once from the N8300A. Frame duration and the downlink ratio is set. For the RF burst trigger, trigger slope, trigger delay, and trigger holdoff are configured. For each loop of the device frequency and device power level, the frequency, power range, and trigger levels are set.

The `:SOURce:TRIGger` command is again used after the `INIT` command. `INIT` should be issued before triggering the source output in order to arm the N8300A analyzer. The output of the source with a downlink signal then causes the device to respond with an uplink signal. The `:STATus:OPER:COND?` query is used to poll the instrument to know when the result has bit position four (decimal 16 or HEX 0x10) clear and so the measurement has completed. The `:FETCh:EVM?` query returns a full range of modulation measurements too long to list in this quick guide. Please refer to the N8300A help for a full listing, which includes results such as data burst information and frequency error.

The results obtained can also be compared with limits. In the case of EVM, it is possible to set pass/fail limits for particular coding rates for the RMS RCE measurement.

The N8300A has the ability to auto-detect a FCH/DL-MAP and UL-MAP of a downlink signal. This is useful for setting up the N8300A for subsequent uplink measurements. This operation is not something to be considered for manufacturing code, but is used more for bench-top automation. For further details please see *Appendix D*.

It should be noted that there are a number of advanced settings that may be configured for RCE measurements. When comparing RCE measurements with other test equipment and result sets, consider settings such as:

1. Whether or not the pilot was included in RCE calculations
2. Symbol timing adjust settings
3. Whether the analyzer was set to track amplitude changes in the pilot sub-carriers
4. Whether the analyzer was set to track phase changes in the pilot sub-carriers
5. Whether the analyzer was set to track timing changes in the pilot sub-carriers
6. The type of equalizer training set

Error Handling and Programming Issues

Error checking

Making correct use of the N8300A error queue (the `:SYST:ERR?` command) allows for errors to be read back effectively into a test environment. The error queue can indicate several different types of errors, the most common of which include:

- Incorrect instrument setting (e.g. the FCH of a signal could not be detected)
- Critical error situations during test
- Non-critical measurement errors during the course of a test (also known as measurement advisory messages). Although they are reported as errors, they do not necessarily mean that the test results are non-valid. An example would be setting a variable out of range and then the instrument has to clip to the range end.

As a single SCPI transaction or measurement event can cause multiple errors, the error queue should be continually read (polled) until +0, "No error" is returned. Once that condition is met, then any errors that have been gathered should be analyzed and appropriate action should be taken. Test code should validate all errors by polling `SYST:ERR?` then ignore (or passively log) any of those non-critical errors.

Pacing tests

A good practice is to use the following process to pace the progress of a test sequence:

1. Configure the measurement
2. Initiate the measurement (using various `:INIT` commands)
3. Wait for a condition flag it to return +1 indicating that the test is complete
4. Gather the measurement results

The command used to check for the completion of a measurement (the condition flag) is extremely important. By using the `:STATus:OPER:CONDition?` it is possible to:

1. Take action to detect an early fail and abort testing during the course of a test sequence
2. Avoid lengthy timeouts and handle timeouts effectively if the DUT is not responding correctly. This mostly applies to measurements using RF burst triggering. Timeouts are very important and so the next section covers this

Error Handling and Programming Issues (continued)

Timeouts

N8300A tests can exceed the default timeout settings of software development applications and/or test executive. If the status sub-system does not detect a problem then a VISA timeout will be triggered. If a timeout is not handled correctly then the test executive can hang or abort.

Users should:

- a) During development, use long enough timeouts to understand the timing of SCPI request/response of the various actions they are requesting the N8300A to perform
- b) Once the timing has been understood, that information should be used to dynamically change the timeout length during the course of test
- c) Provide robust routines within the test code to handle any timeouts (i.e. if a timeout occurs what should they do – abort the test, or try to re-test, etc?)

The most robust method of pacing the progress of a test is through the use of the N8300A STATUS sub-system, and in the case of receiver test, the use of the `:SOURce:RADio:ARB:DONE?` command.

Polling a status register in the status sub-system is a necessity for many of the tests. For example, here is an example from a C# program showing how to make an EVM measurement, wait for it to complete, and then fetch the measurement:

```
Send_Command( ":INIT:EVM" );

// Wait for measurement to complete.
do
{
    Status = int.Parse( Send_Query( "STAT:OPER:COND?" ),
                       System.Globalization.CultureInfo.InvariantCulture );
} while ( Status != 0x10 );

// Fetch back the results now that we know the measurement
// has completed.
Result = Send_Query( ":FETch:EVM?" );
```

The example above will continue to loop until the measurement is completed. In order to avoid a long timeout, another condition should be added to the while loop to check the progress of a timer. A timer should be used to make sure that the while loop does not loop infinitely if the operation never gets to finish. If it does expire then the error should be then handled appropriately.

Error Handling and Programming Issues (continued)

Advanced error handling

Using service requests is another approach to handling errors. They offer a way of leaving the instrument to handle the error using interrupts. It avoids the controller requiring continuous involvement and performance penalties. Interrupts allow for servicing of other code and services while waiting for an event to happen. Interrupts also ensure that the controller is told immediately when a condition change has occurred. With a polling approach, the program has to repeatedly read registers. Service requests can also be used to have a generic approach to all measurements, independent of a specific status register that you need to check (poll). This means that you are not constantly polling the instrument or sending commands across the bus. However, not all programming environments will support interrupts and exact implementation also varies depending upon the communication interface. For a simple and single purpose program, some will find polling less complex.

The SCPI used to setup the service requests is as follows. This can be done for all measurements and so this SCPI could be incorporated into a function in code. This example is for an EVM measurement.

```
*CLS
*ESE           // sets bit 0 in the standard event status
               // register corresponding to operation
               // complete
*SRE 32       // set the standard event status summary
               // bit in the service request enable register
INIT:EVM      // perform the measurement
*OPC          // wait for operation to complete
*STB?         // query the status byte register without
               // clearing it
```

If the operation is complete then decimal 96 is returned ($32 + 64 = 96$). The 64 is the numeric value of the request service summary bit that is enabled for any service request. The reading of the service request will differ depending upon the programming language. For example, C# could use events and VEE could use the service request handler.

Test Speed

Maximizing chipset control

For this application note it was assumed that the DUT has to receive a downlink signal before an uplink signal is transmitted to the N8300A. If it is possible to make the DUT generate signals without a downlink sub-frame being present, it will result in a faster test.

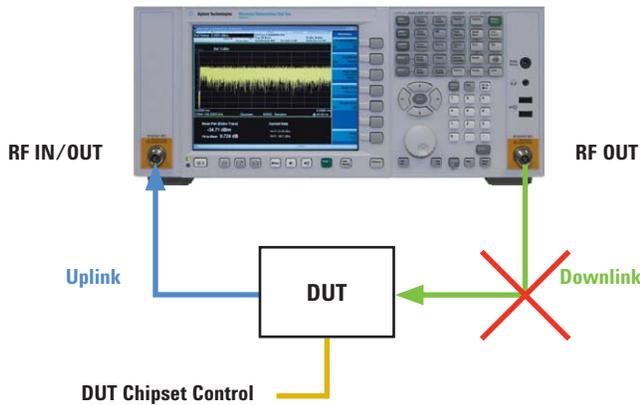


Figure 4. N8300A and device setup using chipset control with no downlink stimulus

Parallel test

The N8300A can support different frequencies for an analyzer and source. For example,

```
:SENSe:FREQuency:CENTer 2.42 GHz
```

sets the frequency for both sense and source. If the frequency control is then set to manual:

```
:SOURce:FREQuency:CONTRol MAN
```

it is possible to set the source frequency separately. The next command sets the source frequency to 2.47 GHz:

```
:SOURce:FREQuency[:MANual] 2.47 GHz
```

With separate RF frequencies and input/output connectors to the instrument, the possibility exists to do transmitter test and receiver test in parallel. This assumes that the device is able to work in this manner. For example, the N8300A transmits on the downlink to the DUT receiver where PER and RSSI can be extracted, while the device transmits on the uplink to the N8300A for analysis.

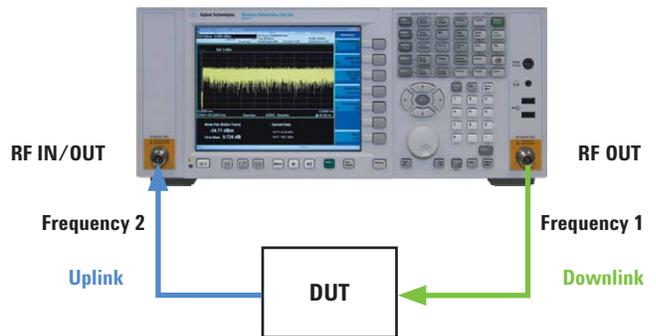


Figure 5. N8300A and device setup with separate source and analyzer frequencies

Other

The N8300A display screen can be switched off using the following command and makes a slight increase in test speed:

```
:DISPlay:ENABLE OFF
```

Glossary

ACP	Adjacent channel power, sometimes referred to as adjacent channel leakage ratio (ACLR), is a measure of the transmitter energy that is leaking into an adjacent or alternate channel
DCD	Downlink channel descriptor
DL-MAP	Downlink map
DUT	Device under test (the device being tested)
EVM	Error vector magnitude
MS	Mobile station
PCT	Protocol conformance test
RCE	Relative constellation error
RCT	Radio conformance test
Segment file	A file containing a representation of a waveform, which can be played by the N8300A source
Sequence file	A file containing one or more pairs of (segment file name, count), which is used to cause the segment file to be generated a specific number of times
UCD	Uplink channel descriptor
UL-MAP	Uplink map

Appendix

Appendix A – DUT command list

For reference, the following pseudo commands were used throughout the test procedures in *The Test Plan* section as a means of describing what a device must do to be tested at each stage.

DUT pseudo command reference/operation	Description
setFrequency(F)	Sets the frequency (F) of the device
readSynchCondition()	Determines whether the device has successfully synchronized to the downlink signal from the N8300A
setPowerControlValue(PCV)	Sets a transmit output level in the device according to an array of power control values (PCV is manufacturer-specific, uncalibrated, unitless) which can subsequently be calibrated by the instrument
readRSSIvalue()	Queries the received signal strength indication from the device
readPacketCount()	Queries successful packets received
SetPowerLevel(DPL)	Sets a transmit output power level following a request for a particular dBm power level (where DPL is a dBm value part of an array). This command is only valid once transmit calibration has been completed (see setPowerControlValue(PCV) above)

Appendix (continued)

Appendix B – Additional measurement results

The following measurements may be required for completing the six phases of the test plan depending upon device and test requirements.

Tx

IQ quadrature error

This parameter may be required for calibration.

:FETCh:EVM21?

For this measurement to return valid results it is necessary to have a sufficient number of symbols and sub-channels, otherwise -999 is returned.

IQ gain imbalance

This parameter may be required for calibration.

:FETCh:EVM21?

For this measurement to return valid results it is necessary to have a sufficient number of symbols and sub-channels, otherwise -999 is returned.

Rx

Generating CW

The following commands may be required for receiver calibration. The SCPI below will provide a CW tone at the requested power (0 dBm) and frequency (2.42 GHz).

Operation	SCPI command
Provide a CW tone at the requested power (0 dBm) and frequency (2.42 GHz)	:SOURce:RADio:ARB:WAVEform "WFM1:<any file name>" :SOURce:FREQuency:CONTRol MANual :SOURce:FREQuency:MANual 2.42 GHz :SOURce:OUTPut:STATe ON :SOURce:POWer:LEVel:IMMEDIATE:AMPLitude 0 :SOURce:RADio:ARB:CLOCK:SRATe 22.4 MHz :SOURce:RADio:ARB:STATe OFF :SOURce:RADio:ARB:TRIGger:TYPE CONTInous :SOURce:RADio:ARB:TRIGger:TYPE:CONTInous:TYPE FREE

The above can be replicated on the front panel by pressing [**Source**], selecting any waveform, selecting a power level, a frequency, turning the RF output on, and leaving the arbitrary waveform generator off.

Appendix (continued)

Appendix C – Building sequence files

In *phase 3*, a sequence file was used to allow a fixed number of repetitions of a waveform to be played. This section describes how a sequence file can be generated containing 1,000 repetitions of a single segment file.

Requirements

In order to create the sequence file for playback of the signal, the following files are required:

1. Segment file – the signal to be repeated
2. Marker file – these files are used for event triggering
3. Header file – a header file defines playback parameters such as sampling rate and scaling

Putting the segment file into the instrument

The first stage is to put the arbitrary waveform generator (AWG) segment file into the N8300A. This can either be performed by SCPI by transferring a BinBlock containing the binary data of the segment, or, using a USB memory drive/dongle to transfer the file.

To use the USB memory drive/dongle, put the segment file onto the root of the drive with a file extension of “.WFU.” The segment file should be complemented with both a marker (“.MKR”

extension) and header file (“.HDR” or “.HEADER”). Plug the USB drive into the N8300A while the WiMAX application is running. You will then be shown a screen like the following:

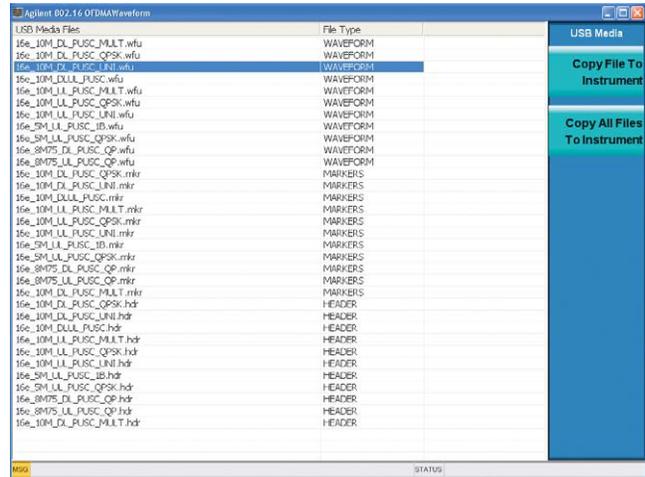


Figure 6. Putting segment, marker, and header files into the N8300A

Click **Copy File To Instrument**, and the files will be copied to the instrument’s volatile waveform memory.

Appendix (continued)

Appendix C – Building sequence files (continued)

Alternatively, if you wish to use SCPI, you must send the segment file as a BinBlock. The SCPI command used is:

Operation	SCPI command
Send the segment file as a BinBlock	<code>:SOURce:MMEMory:DATA "WFM1:<filename>",<datablock></code>

The filename is the file that will be created on the N8300A. The data block is the BinBlock transfer over VISA of the file contents. The command above can be used for marker, header, and sequence files too. To change the destination of files on the N8300A, edit the prefix to the filename:

1. WFM1 – volatile waveform memory
2. NVWFM – non-volatile waveform memory
3. MKR1 – volatile marker memory
4. NVMKR – non-volatile marker memory
5. SEQ – sequence memory
6. HDR1 – header memory

Waveform files can be copied to non-volatile memory using the front panel controls in order for it to be stored and accessible in the future. The SCPI command to do this is:

Operation	SCPI command
Copy waveform files to non-volatile memory using the front panel controls	<code>:SOURce:MMEMory:COPIY "NVWFM:<source file name>","WFM1:<destination file name>"</code>

Building the sequence

A sequence file can be created from a segment file residing in non-volatile memory. The simplified SCPI is as follows:

Operation	SCPI command
Create a sequence file from a segment file residing in non-volatile memory	<code>:SOURce:RADio:ARB:SEQuence "<seq_name>",<segment-file>",<repeat-count>",<marker-type>"</code>
<i>Example:</i> Create a sequence "16E_PER_1000.seq" containing 1,000 repetitions of a segment file named "16E_10M_DL_PUSC_MULT," with no marker control	<code>:SOURce:RADio:ARB:SEQuence "16E_PER_1000.seq","16E_10M_DL_PUSC_MULT","1000","none"</code>
Copy sequence file to non-volatile memory if required	<code>:SOURce:MMEMory:COPIY "NVWFM:<source file name>","WFM1:<destination file name>"</code>

Appendix (continued)

Appendix D – Configuring the N8300A uplink map

A WiMAX uplink signal from a MS device does not contain the uplink map. This means that the N8300A cannot use this information to automatically configure settings to match the signal. There are two ways to configure the instrument correctly to overcome this:

1. Use the base station to auto-detect the UL-MAP from the downlink signal. By subsequently switching back to MS test (uplink analysis), the UL-MAP file can be stored in memory
2. Configure the instrument manually to match the MS uplink signal under test

Method 1: Auto-detect UL-MAP using DL signal

By analyzing the BS downlink signal first, the N8300A can acquire the UL-MAP. It can then use this information to successfully demodulate an uplink signal transmitted from a MS device.

Operation	SCPI command
Connect the N8300A to the base station downlink signal and configure the instrument to analyze a base station signal	:SENSe:RADio:STANdard:DEvice BTS
Select the modulation analysis measurement	:CONFigure:EVM
Select the correct 802.16e-2005 mode (the following assumes 10 MHz)	:SENSe:RADio:STANdard:Y05 B10M
Setup burst triggering to capture the bursts being transmitted from the base station, for example	:SENSe:EVM:TRIGger:SOURce RFBurst :TRIGger:SEQuence:RFBurst:LEVel:ABSolute -20 dBm :TRIGger:SEQuence:RFBurst:DELay -10 µs :TRIGger:SEQuence:RFBurst:DELay:STATe ON :TRIGger:SEQuence:HOLDoff 100 µs :TRIGger:SEQuence:HOLDoff:STATe ON
With the signal observed and configured in the instrument, set the zone definition to automatic	:CALCulate:EVM:MAP:SElect:TYPE AUTO
Select a map for burst analysis	:CALCulate:EVM:BURST:TYPE MAP
Now detect the map information automatically using the DL-MAP/UL-MAP and DCD/UCD from the base station	:CALCulate:EVM:MAP:DETECT
The MS can now be connected for test.	
Switching back to MS uplink analysis can be completed as follows	:SENSe:RADio:STANdard:DEvice MS
Select the correct 802.16e-2005 mode similar to what was configured above for the base station	:RADio:STANdard:Y05 B10M
Select the auto-detected map	:CALCulate:EVM:MAP:SElect:TYPE AUTO

You can now make an EVM measurement.

Appendix (continued)

Appendix D – Configuring the N8300A uplink map (continued)

Method 2: Manual configuration of uplink signal

The following example is based on using the supplied N8300A waveform file 16E_10M_UL_PUSC_QPSK.

Operation	SCPI command
First, set uplink analysis	:SENSe:RADio:STANdard:DEvIce MS
Select the modulation analysis	:CONFigure:EVM
Setup RF burst triggering	:SENSe:EVM:TRIGger:SOURce RFBurst :TRIGger:SEQuence:RFBurst:LEvEl:ABSolute -20 dBm :TRIGger:SEQuence:RFBurst:DELay -10 us :TRIGger:SEQuence:RFBurst:DELay:STATe ON :TRIGger:SEQuence:HOLDOff 100 μs :TRIGger:SEQuence:HOLDOff:STATe ON
Select the correct 802.16e-2005 mode (the following assumes 10 MHz). This changes a range of settings which includes FFT, BW ratio, guard interval, frame duration, and DL ratio	:SENSe:RADio:STANdard:Y05 B10M
Now set the ratio of uplink/downlink	:SENSe:RADio:STANdard:DRATio 0
Set the frame duration parameter to specify the OFDMA frame duration in units of time. You can use the radio standard or radio device preset to set the frame duration to a value specified in the OFDMA standard, or you can specify an arbitrary value manually in the frame duration parameter	:SENSe:RADio:STANdard:FDURation 2.0e-03
Set the zone definition, which defines the zone information. The following command sets custom	:CALCulate:EVM:MAP:SElect:TYPE CUST
Set a method of burst analysis, in this case single	:CALCulate:EVM:BURSt:TYPE SINGle
Set the zone interval to be used	:CALCulate:EVM:ZONE:INTerval 0
Set the number of the sub-channel logical interval for the burst region	:CALCulate:EVM:BURSt2:SUBChannel:INTerval 0
Set the sub-channel offset	:CALCulate:EVM:BURSt2:SUBChannel:OFFSet 0
Set the number of the OFDMA symbol interval for the burst position	:CALCulate:EVM:BURSt2:SYMBol:INTerval 0
Set the symbol offset	:CALCulate:EVM:BURSt2:SYMBol:OFFSet 0
Set the data modulation type to be used. This command sets 64QAM-3/4	:CALCulate:EVM:BURSt2:MODulation:TYPE QPSKR1BY2
Set wrapped or rectangle burst shapes. Here the burst shape is set to wrapped	:CALCulate:EVM:BURSt2:SHAPE WRAPped

You can now make an EVM measurement.

Remove all doubt

Our repair and calibration services will get your equipment back to you, performing like new, when promised. You will get full value out of your Agilent equipment throughout its lifetime. Your equipment will be serviced by Agilent-trained technicians using the latest factory calibration procedures, automated repair diagnostics and genuine parts. You will always have the utmost confidence in your measurements.

Agilent offers a wide range of additional expert test and measurement services for your equipment, including initial start-up assistance, onsite education and training, as well as design, system integration, and project management.

For more information on repair and calibration services, go to:

www.agilent.com/find/removealldoubt



Agilent Email Updates

www.agilent.com/find/emailupdates

Get the latest information on the products and applications you select.



Agilent Direct

www.agilent.com/find/agilentdirect

Quickly choose and use your test equipment solutions with confidence.



www.agilent.com/find/open

Agilent Open simplifies the process of connecting and programming test systems to help engineers design, validate and manufacture electronic products. Agilent offers open connectivity for a broad range of system-ready instruments, open industry software, PC-standard I/O and global support, which are combined to more easily integrate test system development.

“WiMAX” and “Mobile WiMAX” are trademarks of the WiMAX Forum. “WiMAX Forum” is a registered trademark of the WiMAX Forum.

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:

www.agilent.com/find/contactus

Americas

Canada	(877) 894-4414
Latin America	305 269 7500
United States	(800) 829-4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Thailand	1 800 226 008

Europe & Middle East

Austria	0820 87 44 11
Belgium	32 (0) 2 404 93 40
Denmark	45 70 13 15 15
Finland	358 (0) 10 855 2100
France	0825 010 700* *0.125 €/minute
Germany	01805 24 6333** **0.14 €/minute
Ireland	1890 924 204
Israel	972-3-9288-504/544
Italy	39 02 92 60 8484
Netherlands	31 (0) 20 547 2111
Spain	34 (91) 631 3300
Sweden	0200-88 22 55
Switzerland	0800 80 53 53
United Kingdom	44 (0) 118 9276201

Other European countries:

www.agilent.com/find/contactus

Revised: March 27, 2008

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc. 2008

Printed in USA, April 11, 2008

5989-8348EN

