Transient Optical Power Measurements with the N7744A and N7745A

Application Note
Version 2.0

Introduction

Measuring optical power level changes, to determine fiberoptic switching times or to observe transient fluctuations from fiber movement or network reconfiguration, goes beyond the design of most fiberoptic power meters. These instruments are generally designed for calibrated determination of optical power levels that are constant or change in synchronization with other instruments. The typical sample rates like 10 kHz, data capacity of perhaps 100,000 samples, and data transfer speed to the controller are often insufficient for general time-dependent measurements. Instead alternative setups, like a fast optical-to-electrical converter combined with an oscilloscope, have been used and described in standards. These often sacrifice optical power calibration, involve additional integration effort, and are likely implemented with an over-dimensioned scope bandwidth.

The N7744A 4-port and N7745A 8-port optical power meters now offer the performance to make these measurements with a small self-contained programmable instrument that is used together with a controller computer.
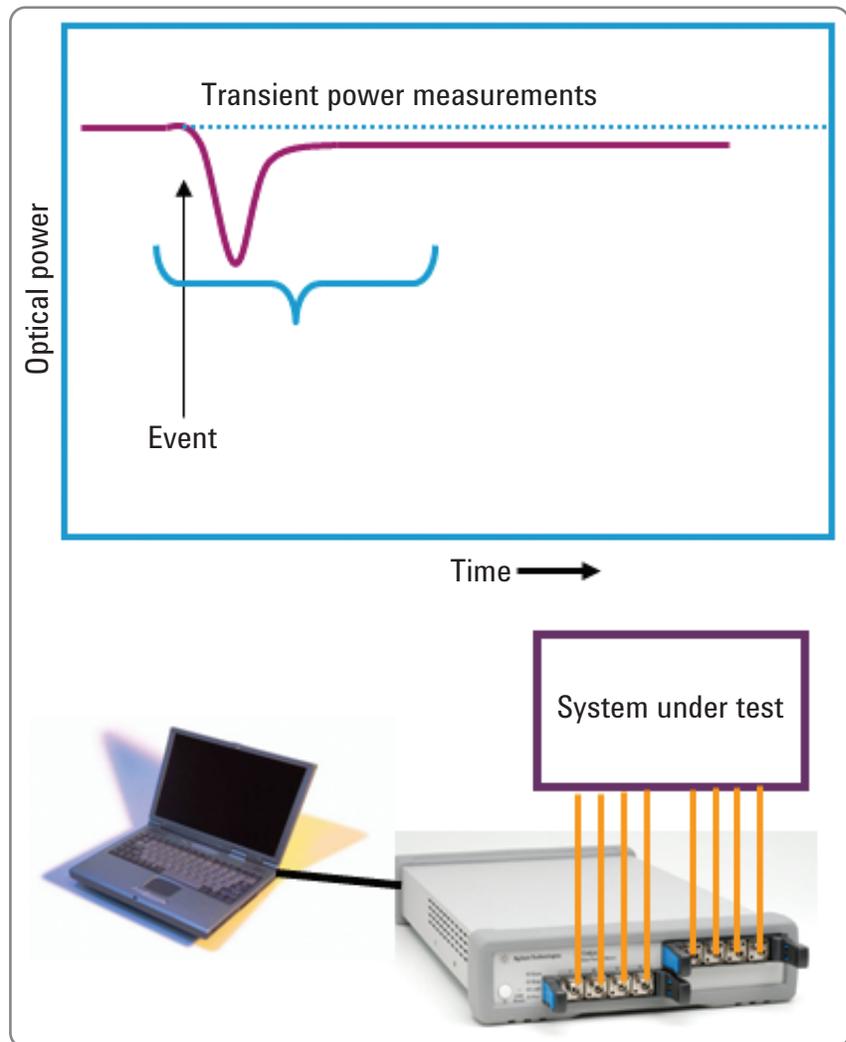

Figure 1. Schematic transient power measurement.

These power meters accurately log optical power at selectable sample rates up to 1 MHz, store up to 2 million samples per port, allow fast data transfer via USB or LAN and support simultaneous measurement and data transfer for continuous power monitoring without interruption. This application note describes the basic details to program these measurements.

This document has been revised to include new commands introduced with firmware version 1.183 for the instruments.

Agilent Technologies

Logging Functionality Basics

The measurement of time-dependent signals is realized with the easy-to-use logging function of the optical power meters. The logging function is set up by choosing the number of logging samples, N, and the averaging time of each sample, t. The logging measurement is then started with a programming command or an electrical trigger. The instrument can be configured to make the complete logging measurement of N samples or individual samples when triggered. For logging time-dependence, the measurement will usually be configured for logging all samples without pause over a total time Nt.

For completeness, note that the instruments also have a stability function that performs similarly, but with a programmable dwell time between samples. This is used for measuring longer term changes in optical power, as for source stability tests, and is not discussed here further.

The N7744A and N7745A multiport power meters, MPPM, can perform this logging simultaneously on optical signals from up to 8 fibers. The averaging time can be chosen between 1 µs and 10 s, and up to 1 million samples can be taken. During the logging, a wide dynamic range can be recorded, exceeding 60 dB for averaging times of 100 µs or more, and the power range maximum can be chosen between –30 dBm and +10 dBm in 10 dB steps.
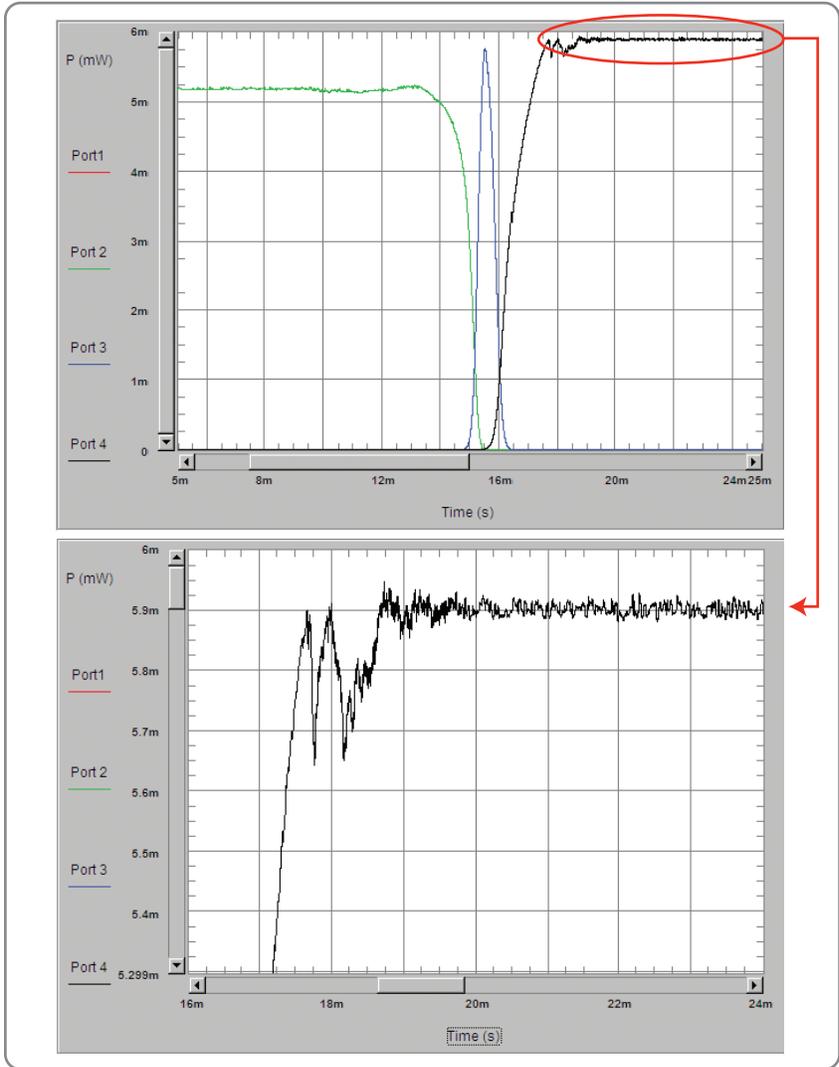

Figure 2. Switch test measurement showing the transfer of power from Port 2 to Port 4, with transient cross-talk at Port 3. The zoomed image in Figure 2(b) shows how the switch bounces and settles at the new port.

The MPPM can also be configured to begin a new logging measurement of N samples as soon as the previous measurement finishes. The existing results can be uploaded to the controller computer during the new measurement. This set of functionality provides two methods for making transient measurements, which we label here as triggered logging and continuous logging methods.

Triggered logging is used to measure a fixed number of samples, starting from a time chosen by software or an electrical signal to synchronize with the event to be measured. This is most useful when the timing of the event to be measured is also controlled, as for setting a switch or shutter, changing an attenuator, or blocking an input signal to an amplifier or ROADM (reconfigurable add/drop multiplexer). Since 1 million samples can be stored per port, a single logging measurement is usually sufficient. In fact as described later, the data buffers can be used to extend this to 2 million, allowing for example up to 2 s of measurement at 1 Ms/s. Thus all ports can be measured with high temporal resolution over relatively long times without limitation due to the data transfer. The multiple ports of the instrument make it easy to watch, for example, all output ports of a switch during reconfiguration. Measurements like described in the IEC standard 61300-3-21 for switching time and bounce time or transient characterization of optical amplifiers can be accomplished with this method.
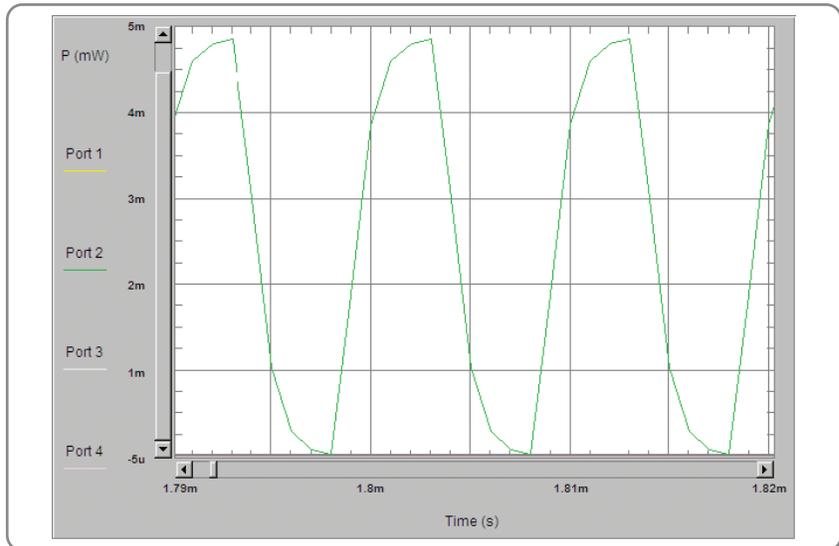

Figure 3. An example measuring a DFB laser modulated ON/OFF at 100 kHz, using 1 µs sampling.

Continuous logging is especially useful for recording events with unpredictable timing as well as for keeping a very large number of samples. A typical application would be the measurement described in IEC 61300-3-28 for transient loss, where the power from fibers is monitored for change due to mechanical disturbances. This method can be programmed using the same logging function mentioned above, with the extension that the complete logging sequence is repeated multiple times. This is made possible with full-size data buffers that allow logging to continue while the previous data is uploaded. Logging can continue indefinitely as long as the time required to upload the mN samples from m ports does not exceed the time Nt to make the measurement. The current limit is above 500,000 samples per second. The streaming data to the PC can be monitored for transient events to trigger saving and analyzing data from before and after the event, as well as used for strip-chart style graphics or other post-processing. For such real-time processing while data is being gathered, multi-threaded programming is useful to avoid interruption of the data stream, as now available in Agilent VEE 9.0 and higher.
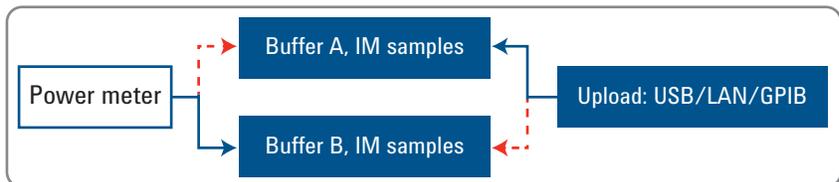

Figure 4. Functional diagram of buffer use for simultaneous measurement and data transfer.

Triggered logging

Triggered logging applications are easily implemented using the built-in logging functionality of the power meters. The programming descriptions below are based on the fundamental SCPI commands for the instruments. Programming can also be facilitated using the IVI-COM driver of the LXI-compliant N7744A & N7745A. It is also possible to use the popular 816x VXI Plug&Play driver, which is particularly comfortable for graphical programming environments like VEE, but this limits the number of logging samples to 100,000 per port and the available choices (but not range) of averaging times somewhat, for legacy reasons. Please refer to the programming guide for details to the programming commands.

A basic setup for a measurement is shown schematically in Figure 1. Here a PC is used both to configure the device under test and to control the N7745A. Each fiber output of the device is connected to one of the power meter ports. The PC can be connected to the MPPM via LAN, USB or GPIB. In this case, the PC will synchronize the timing between switching and logging with programming commands. Alternately, an electrical trigger from the device to the BNC trigger input of the MPPM could be used to start the logging.

The basic programming structure is illustrated below. An explanation of the programming follows.

1.	Initialization: for all ports n	
	SENSn:FUNC:STAT LOGG,STOP	cleaning up, in case the last run didn't finish
2.	Configuration: for each port n	
	SENSn:POW:GAIN:AUTO 0	optional, disable Auto Gain if required
	SENSn:POW:RANG rDBM	choose range r
	SENSn:POW:UNIT 1	fastest upload with units Watts
	TRIGn:INP CME	to start complete logging on trigger
	SENSn:FUNC:PAR:LOGG N,t us	enter logging samples N and avg. time t
3.	Enable Logging: for each port n	
	SENSn:FUNC:STAT LOGG,STAR	starts logging, but waits for trigger
4.	Trigger Logging externally or with command	
	INIT1:IMM	triggers logging for all "started" ports
5.	Check for measurement finished	
	SENSn:FUNC:STAT?	requires evaluation of STRING for COMplete
6.	Get Logging results for each port n	
	SENSn:FUNC:RES?	data in binary block form, Intel byte ordering

Figure 5. EDFA transients: 2-wavelength method.

- It can be a good idea to clear any previous measurement functions like Step 1. In programming the multiport instrument, it is useful to keep in mind that the ports are not completely independent instruments and that logging with the same parameters and timing is expected for each active port. Thus the program begins with a loop through each port to be used, setting the measurement parameters in Step 2 before the logging function is started.
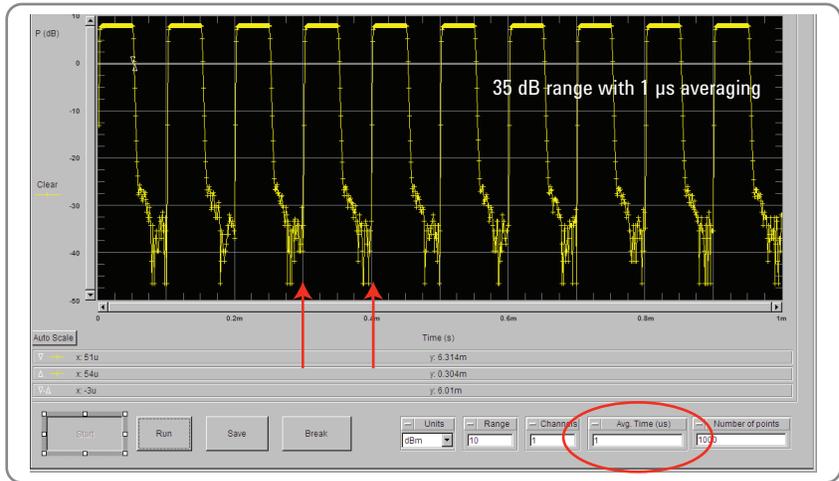
- The Auto Gain functionality is used to increase the sensitivity within a fixed power range by adjusting the gain in real time. With some modulated signals, especially if the averaging time is longer than the modulation period, this produces irregular results and should be disabled. This setting must precede setting of the averaging time or logging parameters.

- The power range will be constant during the logging and can be set in advance for each port. If autoranging is active before the logging is started, then logging will continue in the current power range.

- The ports can be configured to return results in units of Watts, dBm or relative to a reference value in dB. This is an enhancement from the 816x-series power meters, which always return results from logging in W. For fastest data transfer performance, choosing W is still preferred and conversion to dBm or dB is faster on the controller PC. Performance is also fastest in the default absolute units mode (sens$n$:pow:ref:stat 0).

Figure 6. DFB modulated at 10kHz, logging started with modulation clock trigger.

- The reaction of the instrument to an input trigger can be configured for the application. If configured to ignore, then the logging will begin as soon as the logging start command is given. If configured for complete measurement, the logging function can be started for each port and then the instrument waits for a software or electrical trigger before beginning the measurement. This is convenient for coordination with external events. Finally the instrument can be configured for a single measurement on input trigger, in which case each trigger results in a single sample on each port. This is primarily used for synchronization with other instruments, like swept-wavelength tunable lasers, and is not discussed further here.

- Then the logging parameters are set up with the desired number of samples and the averaging time. These should match for all active ports.

- Step 3, once the ports have been configured, the logging can be started on each port. In the case of multiple ports and ignored triggering, the final port's start command is effectively also the start for all previously started logging ports.

- If the triggering has been set to complete measurement, then the actual logging can be started with a software command as in Step 4, or an electrical input trigger. This can be coordinated with the activation of the device or event to be tested.

- The logging will be completed after the time Nt, and the status can be queried to confirm that it is complete. It should normally be sufficient to check this for a single port like Step 5. After the first comma in the string will be either PRO for in progress or COM for complete. The index query can also be used, like in the next programming example.

- When complete, the logging results are read out for each port. The data are transferred in binary block format by the SCPI command query. The data has Intel byte format, as described in the programming guide. In VEE programs, this is most easily accommodated by setting the Byte Ordering to LSB in the advanced instrument properties. The IVI-COM or VXI PnP drivers provide convenient handling of the binary data.

5

Continuous logging

Programming for continuous logging only requires a few changes to the routine described above. An additional Loop parameter is used to enable repetitions of the logging measurement. For each repetition as the logging finishes, the results are kept in the buffer and a new loop is started. Default behavior without repetition, as for the above triggered logging, is set with the Loop value 1. A positive integer value, n, results in n repetitions. For continuously logging over an indefinite period, the value is set to 0. The looping continues until stopped by the program. To support this feature, a new input trigger mode has been introduced with the firmware version 1.183. In this mode for multi-measurement triggering (MME), the command to start logging causes the instrument to wait for an input trigger, but then successive loops proceed without additional triggers. The CME mode requires a separate trigger for each loop.

In a special case, the Loop value can be set to make 2 logging measurements for double the number of samples. The results can then be read out from the buffers after the measurements, so the sampling rate is not limited to the data transfer rate. This **double logging** thus provides an extension of the triggered logging method for more points. The SCPI commands for reading the buffered results are described in the programming guide.

  :SENSe[n][:CHANnel[m]]:FUNCtion: RESult:BUFA?

  :SENSe[n][:CHANnel[m]]:FUNCtion: RESult:BUFB?

A programming structure for continuous logging is shown below. Since this is a minor variation of the previous method, most of the comments above also apply here.

|		
1.	**Initialization: for all ports *n***	
	SENS*n*:FUNC:STAT LOGG,STOP	cleaning up, in case the last run didn't finish
2.	**Set Loop Parameter**	
	SENS1:FUNC:LOOP 0	for unlimited loops
3.	**Configuration: for all ports *n***	
	SENS*n*:POW:GAIN:AUTO 0	optional, disable Auto Gain if required
	TRIG*n*:INP MME	to start multiple logging on 1 trigger
	SENS*n*:POW:RANG rDBM	choose range *r*
	SENS:POW:UNIT 1	fastest upload with units Watts
	SENS*n*:FUNC:PAR:LOGG *N,t* us	enter logging samples *N* and avg. time *t*
4.	**Enable Logging: for all ports *n***	
	SENS*n*:FUNC:STAT LOGG,STAR	starts logging, but waits for trigger
5.	**Trigger Logging externally or with command**	
	INIT1:IMM	triggers logging for all "started" ports
6.	**Check for new result available**	
	SENS1:FUNC:RES:INDex?	check for Index incremented when each loop finishes
7.	**Get Logging results for all ports**	
	SENSn:FUNC:RES?	data in binary block form, Intel byte ordering
8.	**Stop Loops when finished**	
	SENS1:FUNC:LOOP 1	
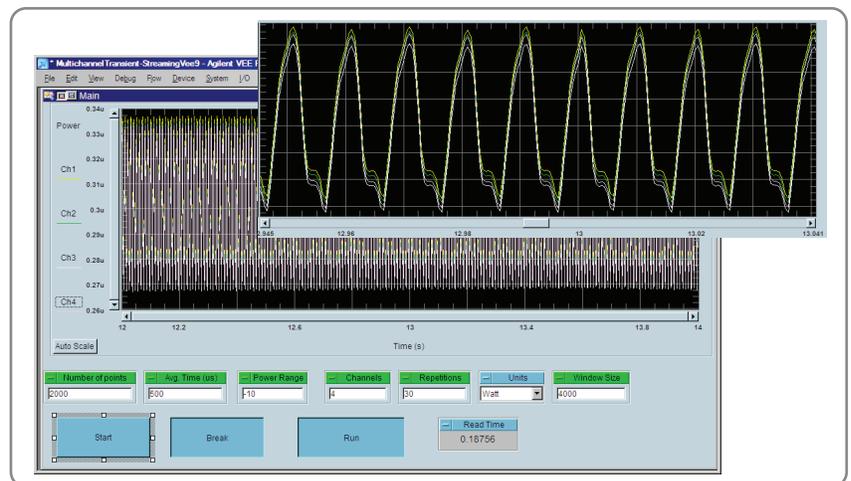
Figure 7. Repeated logging (fluorescent lighting).

- Continuous logging is activated by setting the Loop value to 0.

- All active ports are configured for the desired logging.

- The time for each logging loop is Nt. This determines the update rate for any data graphing or processing on the PC. This time should not be too short, so that it provides enough time to upload the data. For a given averaging time, both the measurement time and actual data upload time increase with the number of samples, but there is additional overhead time for each individual upload. A logging time of 1 s is often a good choice.

- The MME input triggering mode can be used so that after the first trigger, each loop starts as soon as the previous one finishes.

- After configuration, the logging function is started for each port. If input triggering is ignored, the function begins with this command. When MME is used, the logging does not begin until an input trigger is received.

- As soon as the N samples have been logged, another measurement loop begins and the data can be uploaded. The program checks for the data availability in Step 6 with the query for the result index on Port 1. This index has value 0 until the first results are available, indicated with value 1, and is subsequently again incremented with each additional loop.
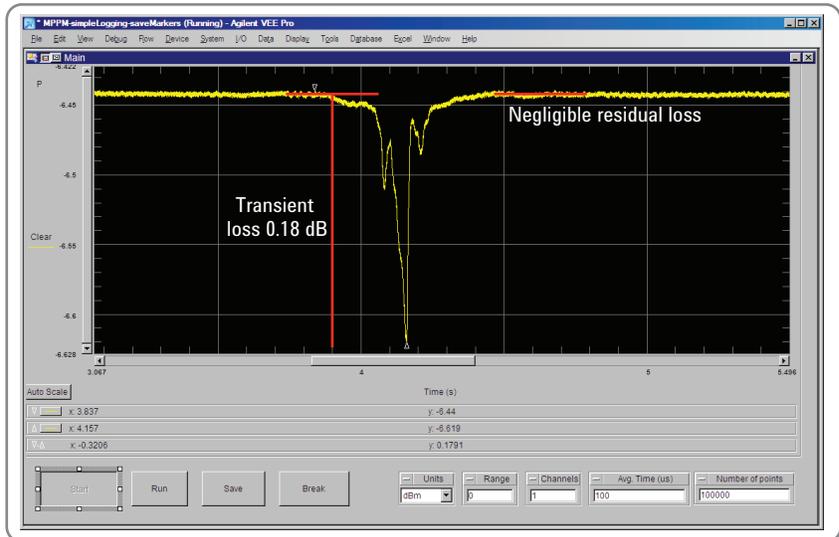

Figure 8. Transient due to fiber bending.

- The data are uploaded in the same way as with triggered logging. The values for N, t, and the number of ports should be chosen so that this can be completed before the next loop is completed.

- The program can graph or analyze each segment of data from a loop as it arrives. In a case where the upload time is similar to the measurement time, Nt, the controller PC is almost continuously uploading data. This does not require high CPU usage, so it is valuable to structure the program for multiple threads to allow parallel data processing.

- For example, with Agilent VEE version 9.0, the steps 6 and 7 can be placed in a thread object that places the data in variables for access by a separate processing thread.

- Besides graphing, processing can be used to detect events that cause the power to deviate more than a chosen threshold value. Sufficient data around this event can then be stored and reported. Events can be analyzed for magnitude and duration, overshoot, undershoot and resulting power offset, or rise and fall time, as appropriate. Long series of data without events can be reduced in size by averaging to record long-term stability.

Agilent Email Updates

www.agilent.com/find/emailupdates
Get the latest information on the products and applications you select.

LXI

www.lxistandard.org
LAN eXtensions for Instruments puts the power of Ethernet and the Web inside your test systems. Agilent is a founding member of the LXI consortium.

Agilent Channel Partners

www.agilent.com/find/channelpartners
Get the best of both worlds: Agilent's measurement expertise and product breadth, combined with channel partner convenience.

Agilent
Advantage
Services

Agilent Advantage Services is committed to your success throughout your equipment's lifetime. We share measurement and service expertise to help you create the products that change our world. To keep you competitive, we continually invest in tools and processes that speed up calibration and repair, reduce your cost of ownership, and move us ahead of your development curve.

www.agilent.com/find/advantageservices

Agilent Electronic Measurement Group
KEMA Certified
ISO 9001:2008
Quality Management System

www.agilent.com/quality

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

Agilent Technologies