LUMILOOP

User's Manual

LSPM 1.0

9kHz – 6(12)GHz
Triple High-Speed Power Meter

Contents

1 System Overview

The Triple High-Speed Power Meter LSPM enables continuous RF power measurements with high resolution, high speed and low noise. LSPM is not limited to the acquisition of quasi-static power levels. With its sampling rate of 2 MSamples/s LSPM is able to measure rapidly changing RF signals. The Power Meter's large dynamic range and fast pulse response make it ideally suited for pulsed and swept signal analysis.

Every LSPM comes with a complete set of calibration data for high linearity and fine-grained frequency compensation. Temperature stability is guaranteed by actively controlling the power sensors' temperature. The high-accuracy in-house calibration data can be further enhanced by external accredited calibration laboratories.

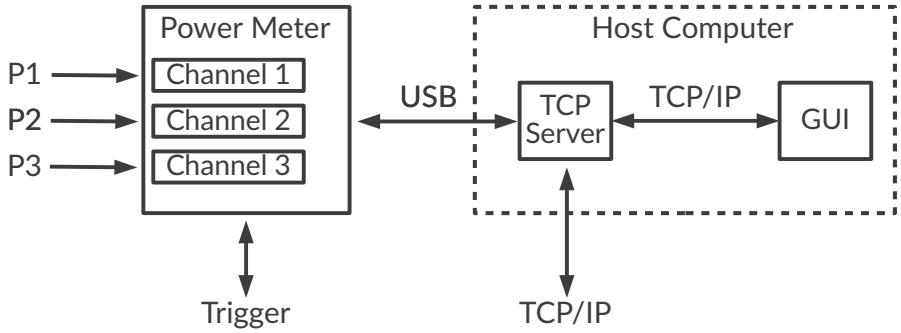


Figure 1: *LSPM system block diagram*

As shown in Figure 1 LSPM consists of up to three channels which record power values using logarithmic power detectors and dedicated analog-to-digital converters. The digital sample values are transmitted to a host computer via USB. External trigger signals are handled by the LSPM's hardware. The LSPM TCP Server handles USB communication, application of calibration data, data post-processing and communication with client programs via TCP/IP SCPI commands, e.g. the LSPM Graphical User Interface (GUI).

Moreover, the TCP Server receives SCPI commands via its standard input and replies via its standard output. This enables simple manual configuration and verification of the LSPM's operation by typing/pasting commands into a command window. The feature can also be used to connect the TCP Server to third party software using a standard IO pipe.

Multiple LSPMs may be connected to one TCP Server instance to form a Multi-Power Meter System consisting of synchronized Power Meters.

1.1 Data Acquisition and Processing

The software delivered with the LUMILOOP LSPM 1.0 Power Meter consists of:

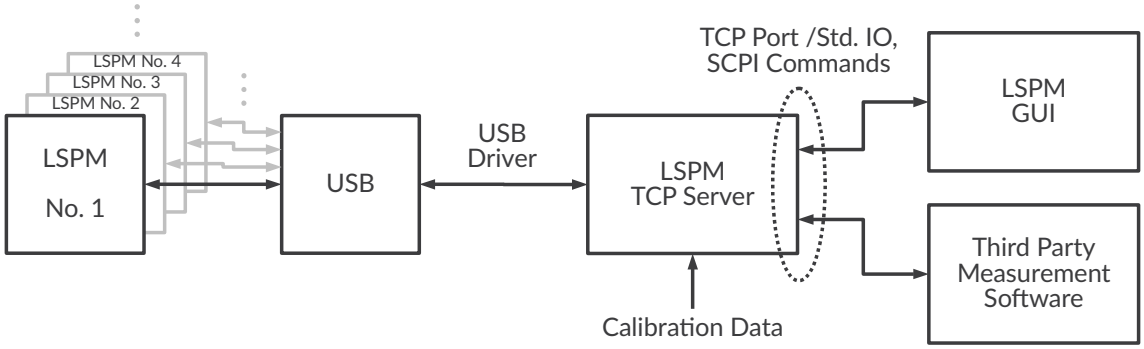


Figure 2: *LSPM software block diagram*

- the LSPM USB driver,
- the LSPM TCP Server and
- the LSPM Graphical User Interface (GUI).

As shown in Figure 2 one instance of the LSPM TCP Server communicates with all Power Meters connected to the host computer via USB 2.0. The TCP Server configures the Power Meter, streams all power values and auxiliary data, applies calibration data to the received power values, handles trigger events, performs sensor data buffering and post-processing. The TCP Server provides an exhaustive set of SCPI commands for simple and reliable text-based integration into test and measurement automation solutions, see Section 9. Concurrent access to all Power Meters is offered to up to 32 TCP clients.

All third party EMC software accesses the LSPM TCP Server through SCPI commands, examples are given in Section 4.3.2, 4.4.2, etc. When operating multiple TCP client programs in parallel, the user is responsible for avoiding undesired interference. For example, one client must not change the measurement mode while another client relies on a different setting. Generally, concurrent access to the LSPM TCP Server is discouraged except for debugging purposes.

The LSPM GUI detailed in Section 4 is a graphical user interface for configuring and monitoring all device settings as well as measuring and logging data. As any client the GUI connects to a local or remote TCP Server.

1.2 Power Waveforms

In addition to the continuous measurement of RF power and output of the current power value with the optional application of a low-pass filter, the LSPM TCP-Server is able to record a user specified length of power values starting from a specific point in time, i.e. power waveforms. Subsequent signal analysis, such as Radar pulse detection, frequency/power sweep evaluation and statistical analysis, are based on previously recorded power waveforms.

Power waveforms are recorded upon receiving a trigger event. They can be queried until being replaced by a newly recorded power waveform. The waveform is recorded relative to a trigger event.

Trigger events can be generated by software, by crossing a specified power value threshold or by externally generated trigger signals. See Section 4.4 on page 29 for more details.

Note that power waveforms require memory in proportion to the recorded duration of time, i.e. approximately 12 MB per second. For a single probe this limits the maximum waveform length to approximately 2.5 minutes on a 32 bit system.

1.3 Sweep Analysis

The LSPM TCP Server is able to analyze frequency and level sweeps recoded in power waveforms. Information about the sweep's timing and frequency steps must be provided as sweep parameters. Each step of the sweep is evaluated individually, taking into account its frequency and settling times. For each part of the waveform that corresponds to a specific generator step the calibration data is newly computed and applied in agreement with the associated frequency for the respective generator step. The LSPM TCP Server returns the averaged power value for each step. Typically, the signal generator and Power Meter are synchronized using a hardware trigger line when acquiring power waveforms for sweep analysis for precise timing control. See Section 4.4.5 on page 34 for more details.

1.4 Statistical Analysis

The LSPM TCP Server is able to perform a statistical evaluation of power data originating from one or more Power Meters. Thus reducing the programming effort, communication overhead, memory requirements and CPU load. Both scalar statistics values and histogram-like distributions are accessible through the LSPM TCP Server. Figure 3 shows a simplified data flow diagram for both continuous and triggered statistics. See Section 4.5 on page 36 for more details.

1.4.1 Continuous Statistics

Continuous statistics are based on three histograms generated from all incoming power values. There is one histogram for each power input. Histograms are created at a resolution of 1/200 dB. Data collection is enabled and disabled using dedicated signal lines described in Section 2.1. Continuous statistics consume a minimal amount of memory since timing information is discarded in the process of creating histogram data. Consequently, continuous statistics can be recorded for arbitrary durations of time. Scalar statistics values and histogram-like distributions use statistics snapshots, i.e. copies of the continuously updated histogram, created at specific times. Snapshot creation for multiple Power Meters is synchronized using dedicated signal lines detailed in Section 2.1. Hardware-based snapshot creation ensures that Multiprobe statistics will not be distorted by latencies introduced by buffering, the USB bus or the operating system's data processing. Histogram-like distributions and associated power values can be queried at a lower level of detail than the 1/200 dB default of the snapshot histograms. See Section 4.5.2 and 4.5.4 for details.

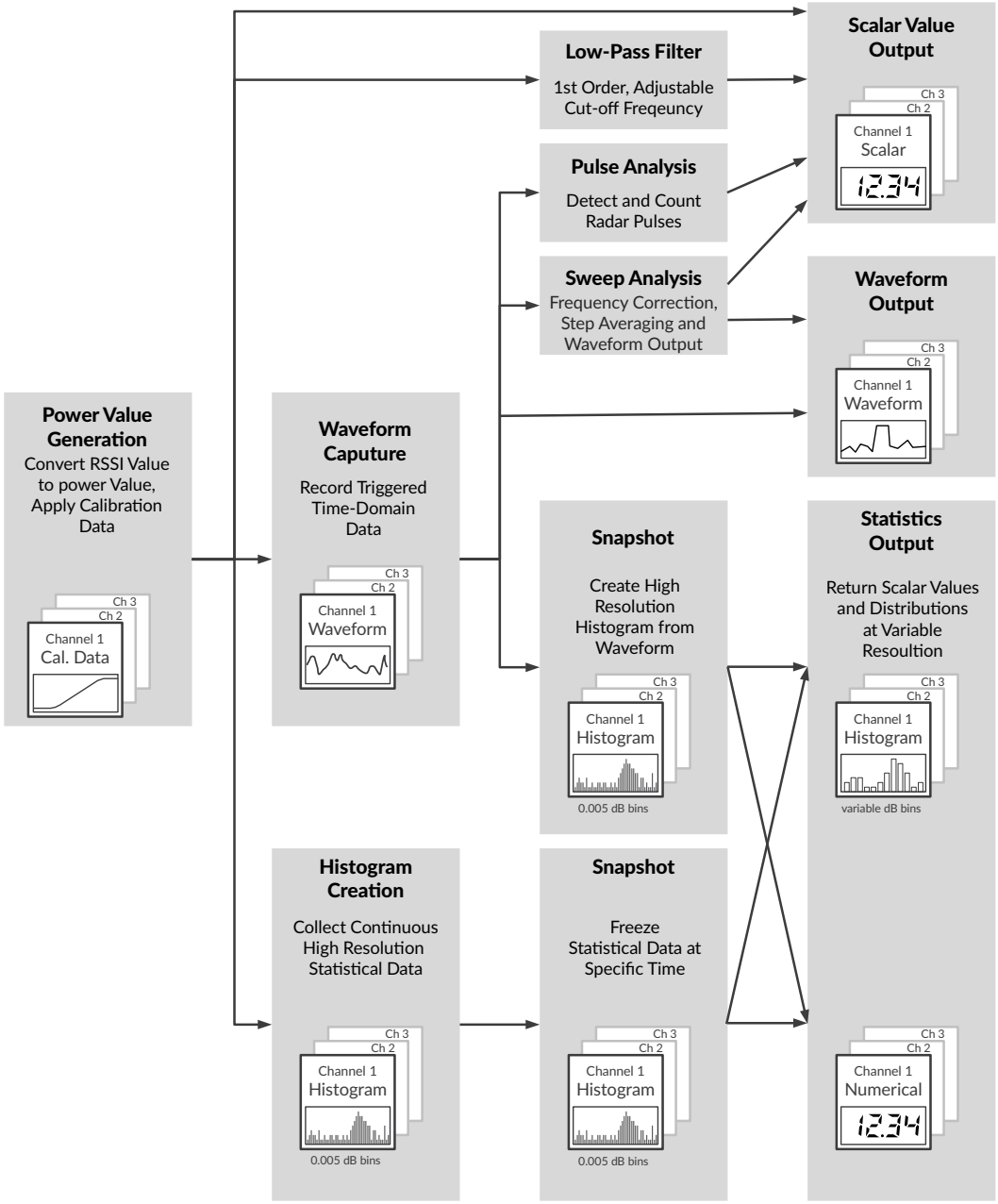


Figure 3: *Data flow diagram of power value processing*

1.4.2 Triggered Statistics

Triggered statistics create snapshot histograms from triggered waveforms. Consequently, a set of power waveforms must be recorded in full before statistics evaluation can take place. When compared to continuous statistics, triggered statistics have the advantage of preserving timing information in the form of the triggered waveforms. However, triggered statistics also have a number of disadvantages relative to continuous statistics:

- Triggered statistics require memory in proportion to the length of time to be evaluated and the number of Power Meters in a system, see Section 1.2 for more details.
- For the same duration of time creating a snapshot histogram from triggered waveforms is significantly slower than a continuous statistics snapshot. This is due to the fact that triggered statistics must evaluate all samples in the recorded waveforms while continuous statistics merely require a copy of the continuously updated histogram for every snapshot.
- Triggered statistics are only available for the recorded waveforms as a whole, continuous statistics may take statistics snapshots as data is being recorded.
- Triggered statistics may introduce significant delays in TCP Server to client communication, especially when recording large waveforms.

It is therefore generally preferable to rely on continuous statistics when timing information is not essential.

1.5 Stream Recording

The LSPM TCP Server is able to save a stream of continuously recorded power values directly to disk. This can be done for one or multiple Power Meters. The Power Meter's trigger signals can be employed to synchronize multiple streams. In this case one LSProbe or LSPM acts as the stream master, sending one synchronization pulse for every 500 µs. All other Power Meters act as stream slaves and use the master's synchronization pulses to ensure the sampling rate dictated by the master. See Figures 9, 10 and 11 for the required physical connections to the BNC connector and Ext1 RJ45 socket. See Section 4.6 on page 40 for more details.

2 LSPM Hardware

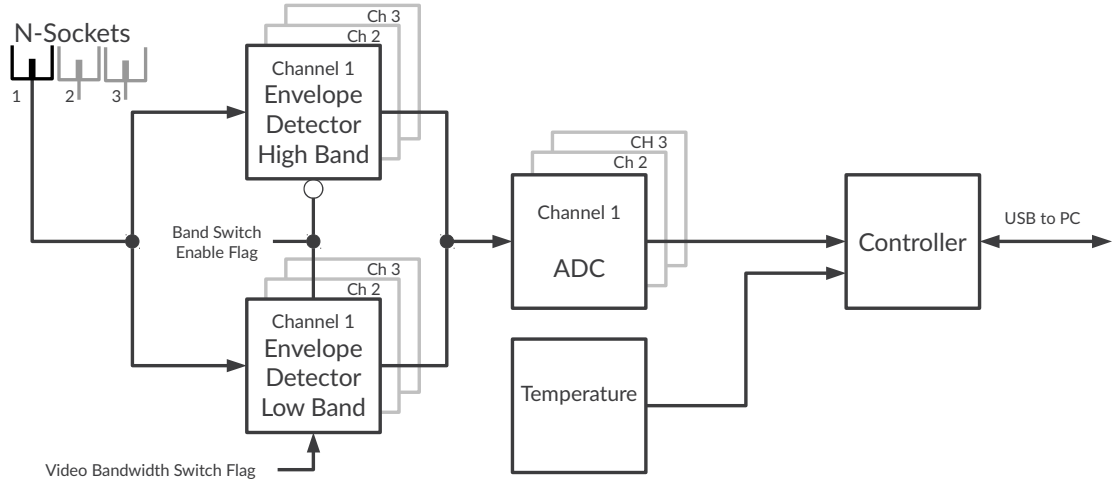


Figure 4: *Power Meter block diagram*

Figure 4 shows the simplified block diagram of the Power Meter. Three 50 Ω N sockets feed the power signals to dedicated detectors. Each channel has got one dedicated low-band and one dedicated high-band logarithmic field strength detector, of which one is selectable at a time using a mode flag (see Table 1). The low-band stretches from 9 kHz to 400 MHz, the high-band from 30 MHz to 6(12[2]) GHz.

Power Meter video bandwidth is defined as the -3 dB cut-off frequency of the first order low pass filter operating on the envelope of each logarithmic power detector. For accurate operation the detected frequency should be at least ten times larger than the video bandwidth. The low-band detector offers a choice between low and high video bandwidth, i.e. approximately 500 Hz and 1 MHz selectable using a mode flag (see Table 1). The high-band detector has a fixed video bandwidth of approximately 3 MHz.

For each axis a 14 bit analog-to-digital converter (ADC) is used to digitize the detected signal level.

The Power Meter offers four modes. Each mode is characterized by a specific frequency range, video bandwidth, sampling rate and sample timing, as listed in Table 1.

In mode 0 only the high band envelope detector is active. In mode 2 and 3 only the low band envelope detector is active. In mode 1 the suitable detector is selected according to the set frequency. Mode 1 spans the Power Meter's entire frequency range but requires approximately 1 ms of additonal settling time when crossing 30 MHz.

When using SCPI Commands or third party EMC software special care must be taken to ensure an appropriate mode setting and low-pass filtering for the measurement task at hand.

[2] Operation up to 12 GHz with reduced performance

Table 1: *LSPM measurement modes overview*

Mode	Minimum Frequency	Maximum Frequency	Video Bandwidth	Sampling Rate
0	30 MHz	$6(12^2)$ GHz	3 MHz	2 MS/s
1	9 kHz 30 MHz	$29.\overline{9}$ MHz $6(12^2)$ GHz	500 Hz 3 MHz	2 MS/s
2	10 MHz	400 MHz	1 MHz	2 MS/s
3	9 kHz	400 MHz	500 Hz	2 MS/s

The Power Meter's housing contains:

- up to three power detectors,
- a thermoelectric, i.e. Peltier, temperature controller for maintaining a constant power detector temperature,
- a BNC trigger input or output connector for synchronization and
- a USB 2.0 interface connecting to the host computer.

As shown in Figure 5 the main switch is located on the left side of the front panel. In "0" position, it disconnects the Power Meter's external 5 V supply. The right side of the front panel is occupied by the N sockets and the air outlet of the detector temperature controller, the latter must not be obstructed. Two labeled LED indicators display the LSPM's operating state as follows:

Figure 5: *Computer Interface front panel*

POWER (GREEN)

Flashing
Main switch is on, Power Meter is inactive.

Continuously on
USB connection to TCP Server has been established.

Continuously off
Main switch is off, mains adapter is disconnected or Power Meter firmware is compromised.

TEMP (RED)

Continuously off
Temperature is being controlled within the power detectors' optimum operating temperature range.

Continuously on
Temperature of the power detectors is above its optimum operating range, detectors are being cooled.

Flashing
Temperature of the power detectors is below its optimum operating range, detectors are being heated.

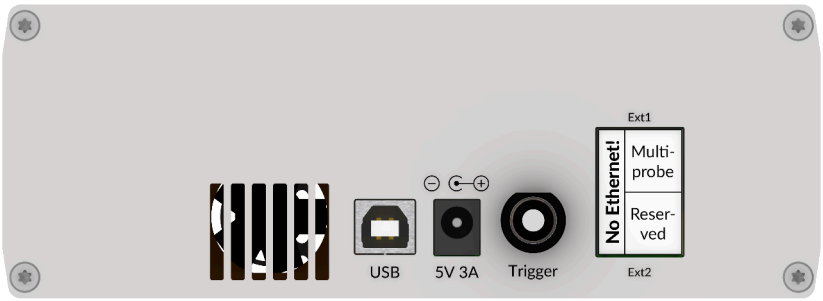


Figure 6: *Power Meter back panel*

The Power Meter's back panel shown in Figure 6 contains the air inlet of the power detector temperature controller and must not be obstructed.

The following electrical connectors are located at the bottom edge of the back panel, left to right:

USB
USB B connector attaching the Power Meter to the host computer.

5 V 3 A
External DC power supply, barrel jack 2.1/5.5 mm.

TRIGGER
: Trigger output or input BNC connector using 5 V CMOS logic levels.

EXT 1
: RJ45 extension connector for Multi-Power Meter or Multiprobe Systems. ***No Ethernet interface!***

EXT 2
: Reserved for future use. ***No Ethernet interface, do not connect!***

2.1 Multi-Power Meter and Multiprobe Systems

Continuous power statistics for Multi-Power Meter or continuous E-field statistics for Multiprobe setups require a hardware link via the "Ext 1" extension connector of every Power Meter. For systems with two Power Meters or a Power Meter and a compatible E-field Probe a straight shielded RJ45 (EIA/TIA 568) patch cable is sufficient, see Figure 7. For larger Multiprobe Systems a Multiprobe Interconnect and Power Supply unit as shown in Figure 8 connects to every Power Meter and field probe's Computer Interface. Note that the barrel plug patch cables used for supplying the Power Meter and field probe Computer Interfaces via the Multiprobe Interconnect and Power Supply unit are not shown.

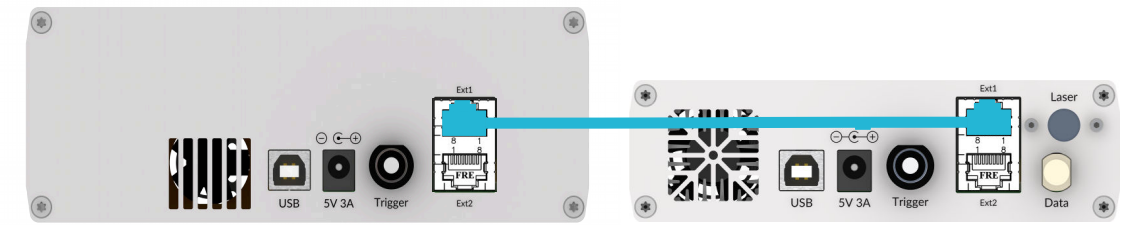


Figure 7: *Connection of one Power Meter and one E-field probe*

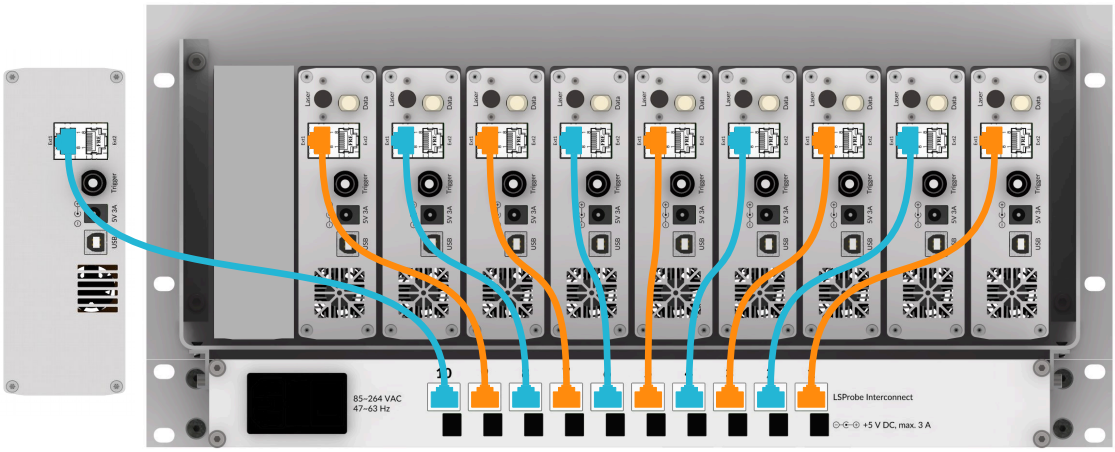


Figure 8: *Multiprobe connections for one Power Meter and nine E-field Probes.*

In a Multiprobe System one Power Meter or LSProbe Computer Interface is configured as the statistics master. The Multiprobe master controls continuous statistics collection via dedicated enable and snapshot lines. The hardware link also carries lines indicating the master/slave status of each Power Meter and Computer Interface in a Multiprobe System. There are dedicated indicator LEDs on the front of the Multiprobe Interconnect and Power Supply unit. The master is indicated by a flashing LED. Multiprobe slaves' LEDs are on continuously if the statistics collection is turned on. Continuously off LEDs indicate that statistics collection is off.

Upon enumeration the first devices are automatically set to master status. Therefore, the default Power Meter statistics master or the default LSProbe CI statistics master must be set to slave.

2.2 Trigger Inputs and Outputs

The Power Meter features two independent trigger inputs and outputs. The BNC connector on the back of the Power Meter uses a single-ended 5 V CMOS logic trigger signal. Figures 9 and 10 show the basic point-to-point setup for using an external device as either trigger target or trigger source. The external device can be another Power Meter or an electrically compatible third-party device.

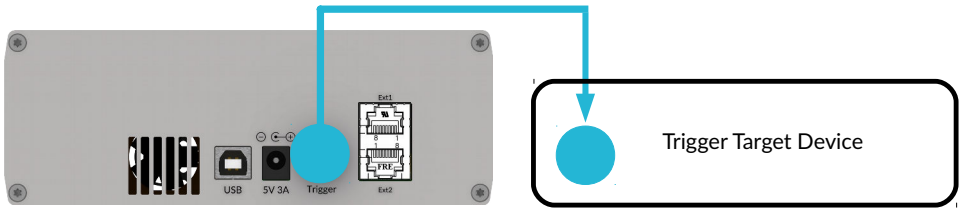


Figure 9: *External trigger output using BNC connector*

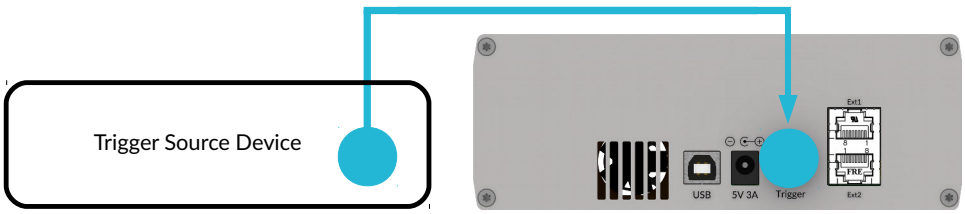


Figure 10: *External trigger input using BNC connector*

The Ext1 RJ45 socket on the back of the Power Meter uses a differential 3.3 V CMOS logic trigger signal. This signal can be used to exchange trigger signals in a Multiprobe setup containing two or more Power Meters or field probe Computer Interfaces as shown in Figure 11. When a Multiprobe Power Supply Unit is connected all Power Meters and Computer Interfaces automatically share a pair of dedicated differential logic lines. The Ext1 RJ45 sockets are not recommended for use with third-party devices.

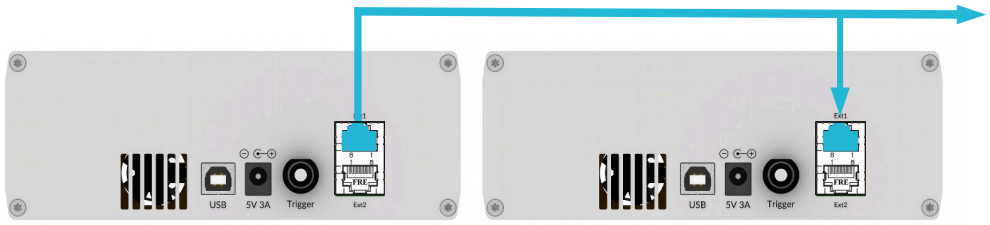


Figure 11: *External trigger input and output using Ext1 RJ45 sockets*

3 LSPM Software

3.1 LSPM TCP Server and GUI Installation

Software installation consists of the following steps:

1. Run the TCP Server and GUI installer. Follow the displayed installation instructions carefully.
2. Copy the supplied calibration data into the directory selected during TCP Server and GUI installation. E.g. if the calibration data zip-file is `C:\Program Files (x86)\LSPM_1.0\cal\` and the serial number of the Power Meter is `42`, please copy the zip-file named `sn42.zip` from the installation medium into the `cal` directory. Repeat the procedure for all Power Meters connected to the host computer.
3. Install the LabView RTE as detailed in Section 3.3.

The TCP Server host name, the TCP Server port as well as paths for data storage and calibration files are set via system-wide environment variables which are set during TCP Server installation. Variables may also be modified manually or be overwritten by user variables:

`LSPM_HOST`
: Set to the hostname or IP address of the computer running the TCP Server. This variable is used by the LSPM GUI only.

`LSPM_PORT`
: Set to the desired TCP port of the LSPM TCP Server. This variable is used by the LSPM TCP Server and GUI, both default to port 10,001 if the variable is not defined.

`LSPM_CAL_PATH`
: Points to the directory containing the calibration data zip-files or sub-directories for all attached Power Meters. This variable is used by the LSPM TCP Server only.

`LSPM_SAVE_PATH`
: Points to the directory that saved files and settings will be stored in. This variable is used by the LSPM TCP Server and GUI.

`LSPM_UPDATE_CHECK`
: Enable and disable checking for software updates on the LUMILOOP homepage when the LSPM GUI is run. This variable is used by the LSPM GUI only. If set to 1 update checking is enabled. If the variable is not set or set to zero update checking is disabled.

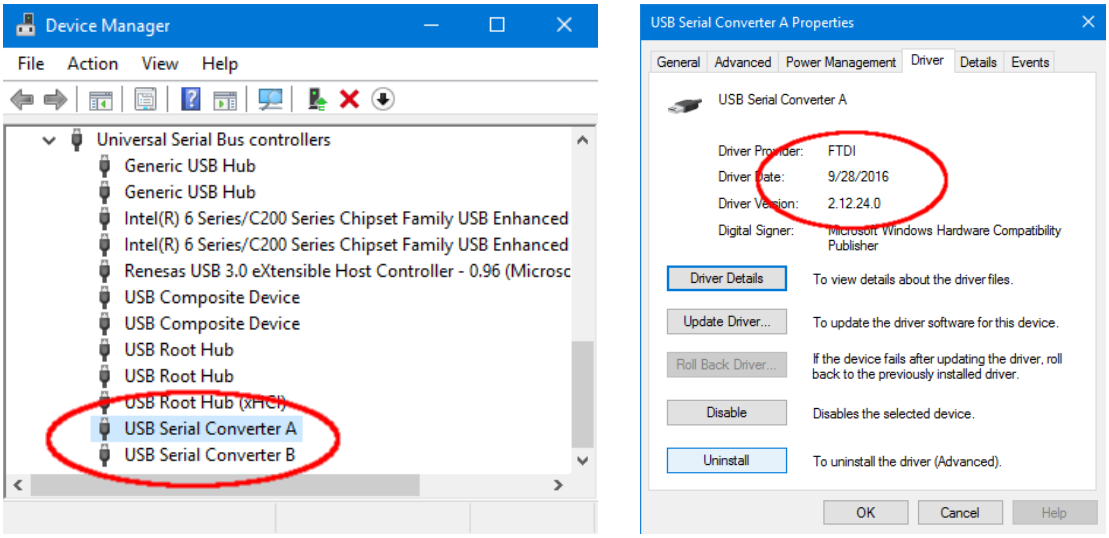
If the LSPM TCP Server and LSPM GUI are supposed to run on different host computers, the installer must be run on both systems. The environment variable `LSPM_HOST` of the system running the LSPM GUI must be set to point to the host computer running the LSPM TCP Server. Moreover, the environment variable `LSPM_PORT` must be set identically.

3.2 USB Driver Installation

If the host computer has access to the online Microsoft Windows Update, the drivers should be installed automatically when a Computer Interface is connected and powered-up for the first time.

Note that for normal operation no internet access is required.

After successful driver installation the Device Manager will list the Computer Interface as "USB Serial Converter A" and "USB Serial Converter B" as shown in Figure 12(a) and (b). Note that the device naming is generic and references neither LUMILOOP nor LSPM. However, this does not affect the proper operation of the E-field probe.



(a) Device Manager USB device

(b) Current FTDI USB driver version

Figure 12: *Correctly installed FTDI USB driver*

If the automated install fails, or if the host computer has no Internet connection, execute the FTDI USB driver installer `CDM v2.12.28 WHQL Certified.exe` contained in the the LSPM installation path's `lib` directory as shown in Figure 13.

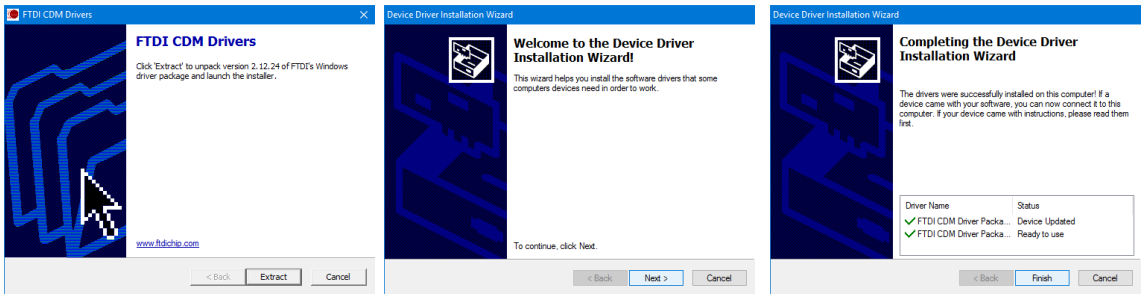


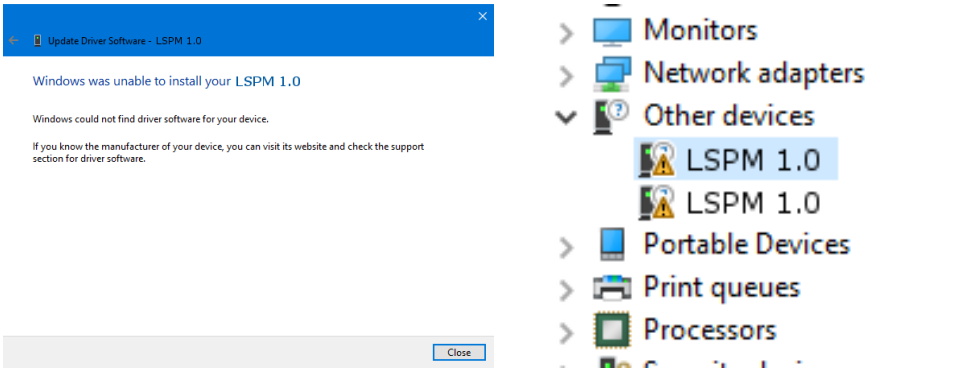
Figure 13: *Manual FTDI USB device driver installation*

It is strongly advised to observe the following recommendations:

- Plug the Power Meter directly into the computer. Do not use a USB hub or docking station.
- Do not connect other high bandwidth USB devices to the same USB root hub. In rare cases this may reduce read performance significantly, resulting in unreliable operation and eventual loss of measurement data.
- Especially, do not operate the Power Meter on a USB port where a USB graphics adapter is installed or was previously installed. The USB graphics driver may disturb communications even if the hardware is no longer attached.

3.2.1 Troubleshooting USB Driver Installation

If no Power Meter and no other FTDI hardware have previously been connected to the computer and automatic Windows driver installation is deactivated or no Internet connection is available the error message shown in Figure 14(a) will be displayed. In this case the Device Manager's "Other devices" section will give an output similar to Figure 14(b), listing the Power Meters' USB end points as unknown devices.



(a) Installation failure message (b) Device Manager output

Figure 14: *USB device driver failure messages*

Make sure that you are using the most recent FTDI USB driver. The driver version at the time of writing is 2.12.24. Using out-of-date USB drivers may result in improper operation.

Check the driver version by opening the Device Manager and extending the "USB-Controller" category. Right-click "USB Serial Converter A" and select "Properties". Open the "Driver" tab to view the FTDI USB driver version as shown in Figure 12(b). If an older version of the FTDI driver has been installed click on "Uninstall". In the following dialog, make sure to check "Uninstall the driver software for this device" as shown in Figure 15, failing to do so will prevent driver updates.

Make sure to repeat the uninstall process for all Power Meter devices using old FTDI USB driver versions. Note that the process must be executed for both "USB Serial Converter A" and "USB Serial Converter B" of every Power Meter. Power-cycle each Power Meter and repeat the installation procedure as described above in Section 3.2.

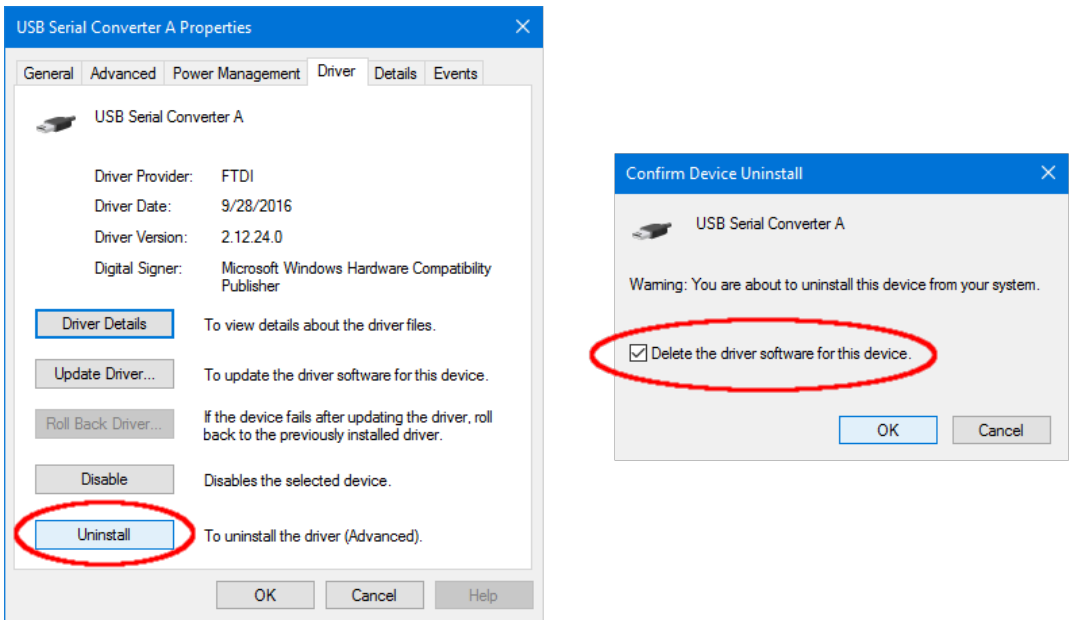


Figure 15: *Uninstalling the FTDI USB driver*

3.3 LabVIEW Run Time Environment Installation

The LSPM GUI requires an installed 32 bit LabVIEW 2012 Run Time Environment or 32 bit LabVIEW 2012 Development System. If either software package is already installed on the host computer, no further steps are required. Otherwise the LabVIEW 2012 Run Time Environment needs to be installed from an installation medium or via download from `http://www.ni.com/download/labview-run-time-engine-2012/3433/en/`.

4 Measuring Power

4.1 Getting Ready to Measure

4.1.1 Making Electrical Connections

When installing the LUMILOOP LSPM 1.0 Power Meter for the first time make the following electrical connections:

1. Connect the supplied mains adapter.
2. Connect the Power Meter to the host computer using the supplied USB cable.
3. Optionally, connect the Power Meter to any trigger sources or sinks via the BNC trigger connector.

Switch on the Power Meter setting the front panel switch to "1" and observe the green power LED starting to flash.

4.2 Power Meter Start-Up and Mode Selection

4.2.1 Starting the LSPM TCP Server

Carefully follow all instructions in the previous sections. Start the LSPM TCP Server and ensure correct operation by verifying that the green power LED is constantly on and the LSPM TCP Server has enumerated all connected Power Meters, listing their serial numbers similar to Figure 16. The LSPM TCP Server will open inside a terminal window displaying status information, debugging output and error messages.

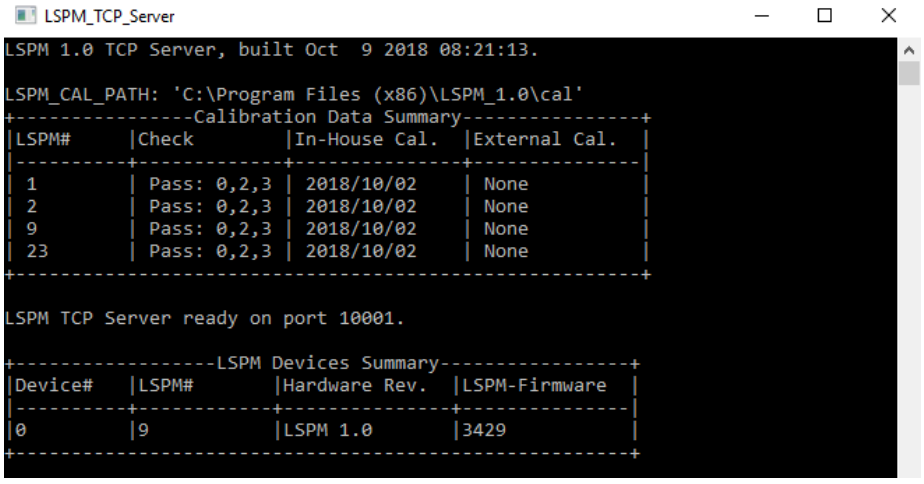


Figure 16: *LSPM TCP Server terminal window*

Upon start-up the LSPM TCP Server will display a tabular summary of available in-house and external calibration data. During Power Meter enumeration the firmware is loaded onto every device. The

TCP Server executable incorporates the firmware image required for proper operation. The version number is displayed inside the LSPM GUI's "Connection" tab.

After Power Meter enumeration the LSPM TCP Server will list all detected Power Meters with their respective serial numbers and firmware revision numbers.

If Power Meters are added to or removed from the host computer after starting the LSPM TCP Server the server will detect these events, update the respective Power Meters's firmware and display an updated Power Meter overview table. Re-enumeration may also be forced by sending a »*RST« SCPI command.

Since the LSPM TCP Server needs to open a TCP port, the system's firewall may ask for permission for network access. Access must be granted to operate the LSPM TCP Server (see Figure 17).

Figure 17: *Microsoft Windows Firewall requesting TCP port access permissions*

4.2.2 General Notes on the LSPM GUI

The LSPM GUI has two modes of operation: Basic Mode and Expert Mode. It will always start in Basic Mode as shown in Figure 18. Expert Mode as shown in Figure 20 can be toggled using the check box control "Expert Mode" in the lower right corner of the main window.

The LSPM GUI is intended as an easy to use demonstration software for all LSPM capabilities. The LSPM GUI is designed in such a way that it will not issue any configuration commands to the LSPM TCP Server unless the user changes a setting using one of the controls. This feature allows for running the LSPM TCP Server in parallel with any third-party EMC software and observing all Power Meter settings and measurement results. This feature is especially useful during third party EMC software integration and function testing.

When there is a mismatch between a setting of the LSPM TCP Server and the expected setting of the GUI, the text of the control element will turn red. Black control element text indicates that the TCP

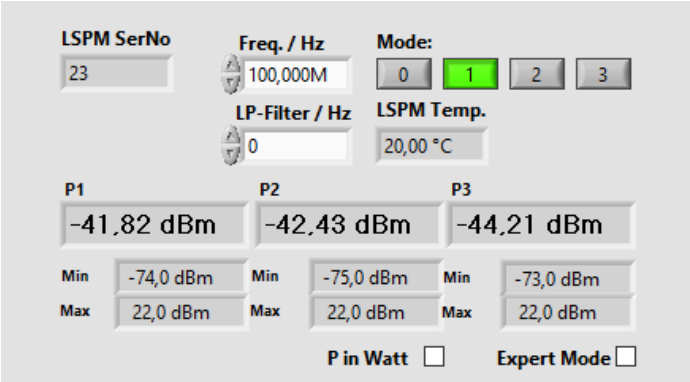


Figure 18: *LSPM Basic GUI*

(a) Match (b) Mismatch

Figure 19: *Frequency control for matching (a) and mismatching (b) TCP Server and GUI settings*

Server's settings are in sync with the GUI. Figure 19 shows the behavior of the GUI for matching and mismatching frequency and mode settings.

4.2.3 Mode Selection Using the GUI

For accurate power measurements the signal's frequency must be specified using the "Freq./Hz" input field. Values are entered in Hertz, SI unit prefixes may be used, e.g. "1.8G" for 1.8 GHz. The actual decimal separator is determined by the host computer's system language. When selecting a mode via the buttons "0" through "3" the user must ensure that "Frequency/Hz" is appropriate for the selection. Frequency values outside a mode's supported frequency range will not be accepted by the entry field. See Table 1 on page 15 for a list of supported modes, their frequency ranges and sampling rates. Hovering the mouse pointer over any of the mode selection buttons of the GUI will display a tool-tip containing a brief description of the mode.

4.2.4 Mode Selection Using SCPI Commands

After establishing the TCP/IP connection the SCPI commands »:SYSTem:SERial? [<MPMeter>]« and »:SYSTem:SERial <Value>« can be used to query all enumerated Power Meters and set the serial number of the Power Meter to be accessed. If only a single Power Meter is attached to the host computer it will be selected automatically and its serial number can be queried using »:SYSTem:SERial? [<MPMeter>]«. »:SYSTem:SERial? [<MPMeter>]« can be used with the MPMeter parameter set to zero to list all enumerated Power Meters.

The desired mode of the selected Power Meter is set using »:SYSTem:MODe <Mode>[,<MPMeter>]«. Refer to Table 1 on page 15 for a list of valid modes. Set the operating frequency in Hertz using »:SYSTem:FREQuency <Frequency>[,<MPMeter>]«. Frequencies outside a mode's supported frequency range will be diverted to the nearest supported frequency. »:SYSTem:FREQuency? [<MPMeter>]« can be used to verify the frequency setting.

4.3 Continuous Power Measurements

While the LSPM Power Meter is capable of exceptionally high speed measurements it is also able to perform high precision measurements of quasi-static electric fields. For continuous power measurements the TCP Server receives all field strength values, applies calibration data and performs low-pass filtering if configured accordingly.

4.3.1 Continuous Measurements Using the GUI

As shown in Figure 20 continuous measurements are configured through the "Log" tab. Power values are displayed both textually and graphically. Values are polled perpetually, resulting in an update rate in the order of 100 samples per second. The actual polling frequency depends on the speed of the host computer and the speed of the network connection. The calibrated range of power values is displayed textually below the channel 1, 2 and 3 power values. For single and dual channel versions of the Power Meter "NaN" will be displayed for unpopulated channels.

"Graph Length" determines the maximum number of samples displayed in the plot at the bottom of the window. It will show at most this many most recent values, older values will be discarded. To display the elapsed time instead of sample indexes for graphs' x-axis and graph length select the "Display x-axis as time" check box. The plot can be paused using "Pause Graph" and cleared using the "Clear" button. By way of the "Log" tab the logging of continuously polled values and of the Trigger, Radar, Sweep and Statistics data can be turned on and off individually. If enabled, newly arriving data of the specific subsystem gets immediately written to a log file. The button "Quick Save" offers a shortcut for the logging of the currently viewed data without previously having enabled the log feature. The data gets logged to a file whose prefix is determined by the "File Prefix" text input field. Log files are saved according to the value of `LSPM_SAVE_PATH`. The Power Meter serial number, a date and time string and a csv file suffix will be appended to every newly created log file. See Section 10.1.1 on page 115 for file format details.

Low pass filter settings are made through the "LP-Filter" numeric input field. A value of 0 disables low-pass filtering. A non-zero value sets the -3 dB cut-off frequency for the first order low-pass filter used for power values, for example 10 Hz. When changing the frequency the low-pass filtered power value will be updated directly, i.e. low-pass filtered values will see a step response instead of a slewing of values.

The controls below the plot pane offer shortcuts for enabling and disabling automatic y-axis scaling, logarithmic and linear display of the y-axis, choosing between sample index and time as the unit

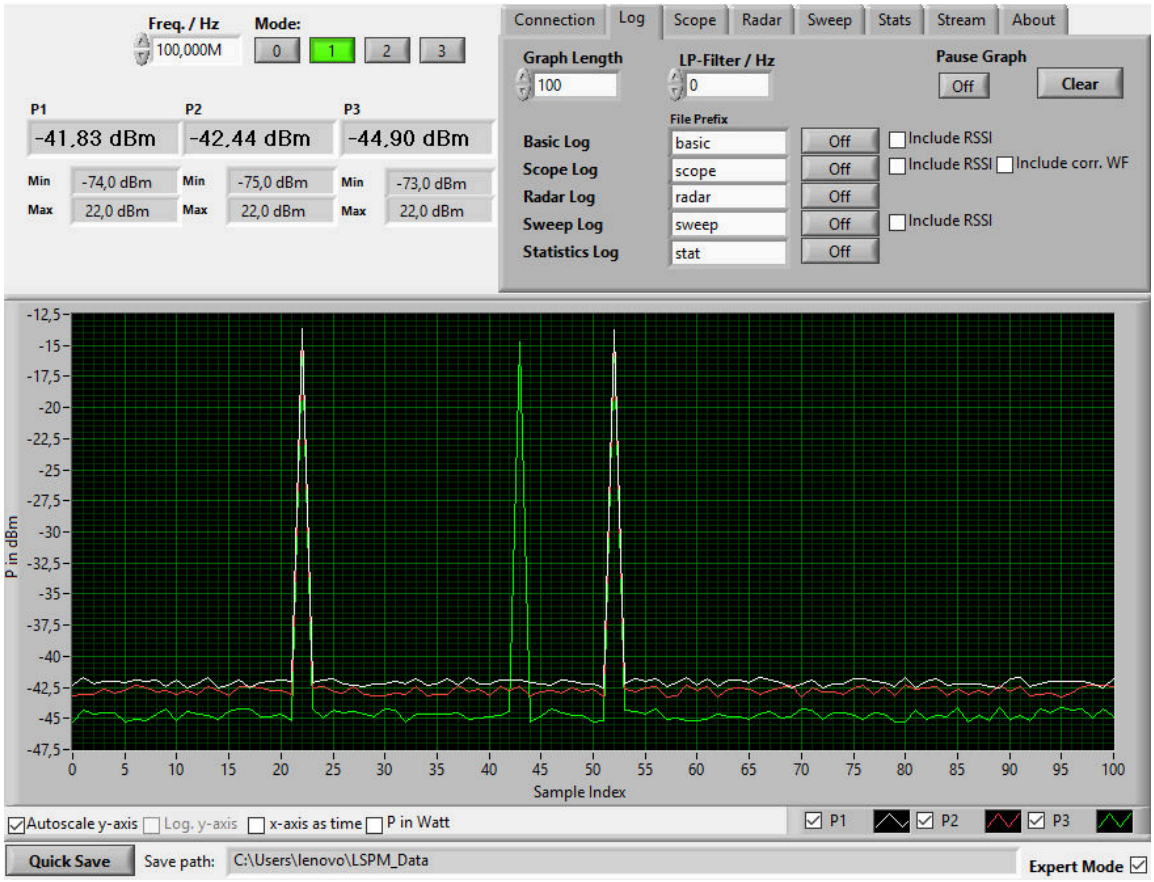


Figure 20: *LSPM GUI data logger view*

for the graphs' x-scale and choosing between displaying power in dBm or Watts. The plot pane also supports adjusting the x- and y-axis by clicking the first or last index of the axis labels and setting its value. Individual graphs can be disabled by unchecking the graph labels at the bottom of the plot pane. Right-clicking the plot pane exposes a number of generic functions including data export to the host computer's clipboard.

4.3.2 Continuous Measurements Using SCPI Commands

After setting up mode and frequency as described in Section 4.2.4 the low-pass filter frequency is configured through »:MEASure[:Power]:LPFrequency <Frequency>[,<MPMeter>]«. A synchronized set of power values can be queried through »:MEASure[:Power]:ALL? [<MPMeter>]«. This is the recommended method ensuring that values have been acquired at the same time. Power values can be queried individually through

- »:MEASure[:Power]:P[1]? [<MPMeter>]«,
- »:MEASure[:Power]:P2? [<MPMeter>]« and
- »:MEASure[:Power]:P3? [<MPMeter>]«.

4.4 Triggered Power Measurements

Triggered Power Measurements allow the user to take full advantage of the LSPM's exceptionally high speed measurements. Waveform acquisition can be triggered by software, using an external trigger signal or edge-sensitive power level triggering.

4.4.1 Power Waveform Acquisition Using the GUI

Figure 21 shows the LSPM GUI in power scope mode which is entered by selecting the "Scope" tab. In power scope mode the textual display shows the averaged power values for the displayed waveforms in addition to the calibrated range of the power values displayed textually below the channel 1, 2 and 3 values.

The "Trigger Source" drop-down box is used for selecting rising or falling edge external BNC, external RJ45, channel 1, 2 or 3 power value triggering. For the latter the threshold value in dBm or Watts can be set via "P1/P2/P3 Threshold". By changing "Mode" automatic free-running triggering, normal event-based triggering and one-shot single triggering can be selected. "Trigger State" displays the condition of the trigger system. The length of the acquired waveform is determined by the "Trigger Length" numeric input field, setting the number of samples for each triggered waveform. "Trigger Begin" sets the start of the saved waveform relative to the position of the trigger event. "Trigger Begin", "Trigger Length" and the graph's x-axis can be displayed/entered as time values by selecting the "Display x-axis as time" checking box. Time values are displayed according to the sampling rates in Table 1 on page 15. The "Arm" and "Force" buttons serve to prepare the trigger system and

to force triggering regardless of actual trigger events. The "BNC Trigger Output" and "RJ45 Trigger Output" drop down menus are used to enable trigger signal output via the Power Meter's BNC connector and "Ext1" RJ45 socket, and set their respective signal polarities. Triggers can be output either when encountering a trigger event, including forced triggering, or for synchronization triggering as described in Section 9.6.31.

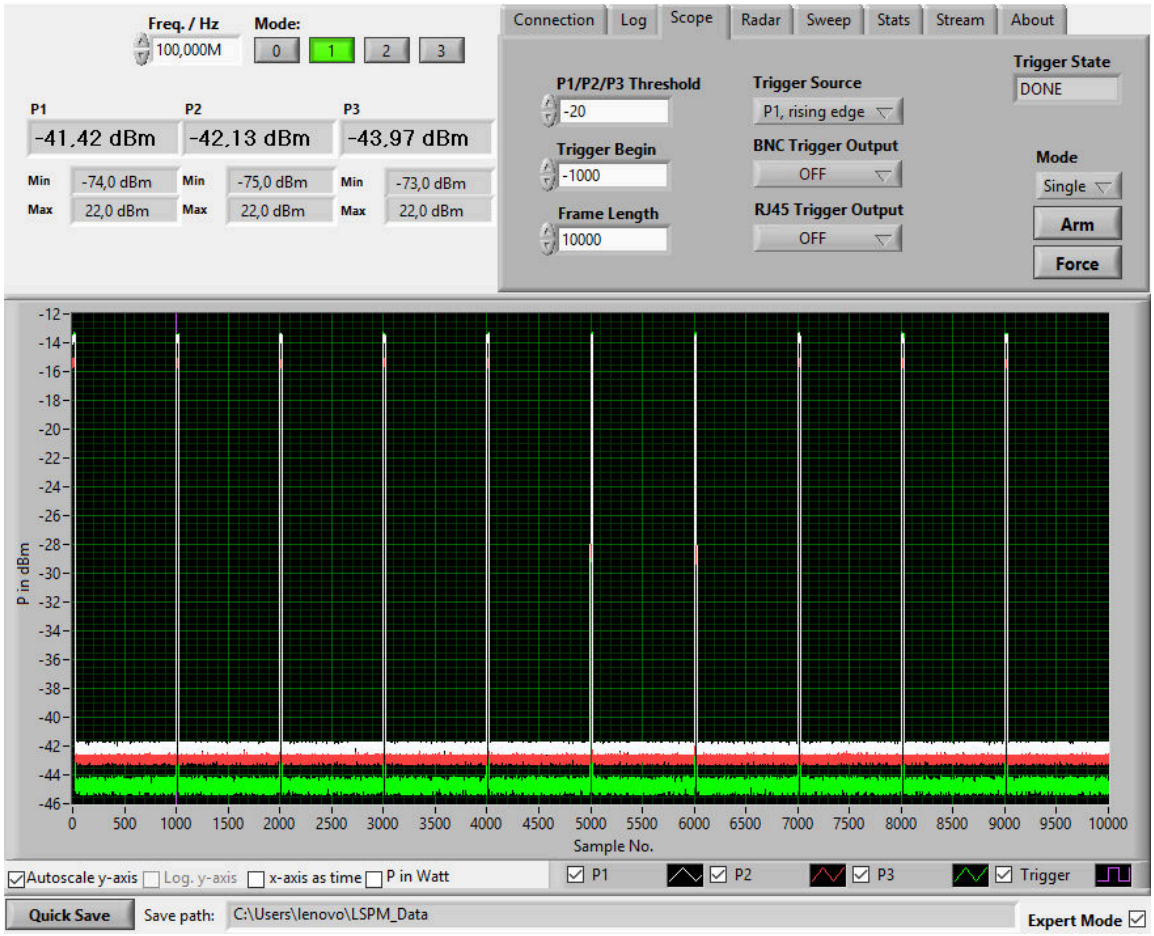


Figure 21: *LSPM GUI, Scope tab*

4.4.2 Power Waveform Acquisition Using SCPI Commands

The state of the trigger system is queried using »:TRIGger:STATe? [<Timeout>,<MPMeter>]«. The configuration of triggered measurements must take place in IDLE state. Waveform query must take place when the trigger system is in DONE state. The SCPI commands »:TRIGger:CLear [<MPMeter>]«, »:TRIGger:ARM [<MPMeter>]« and »:TRIGger:FORce [<MPMeter>]« are used for directly manipulating the state of the trigger system. Figure 22 shows all valid trigger states and state transitions.

The trigger source and polarity are set using »:TRIGger:SOURce <Source>[,<MPMeter>]« and »:TRIGger:FALLing <0/1>[,<MPMeter>]«. If power value triggering is to be performed the trig-

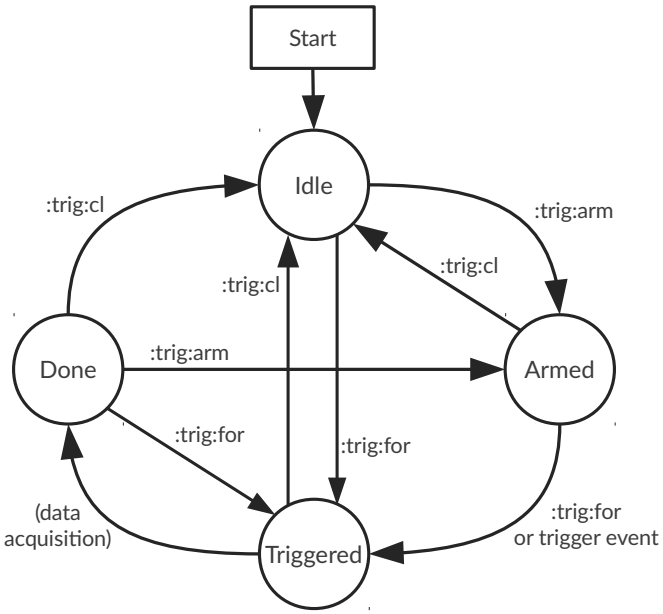


Figure 22: *Trigger system states and state transitions*

ger level is set using »:TRIGger:LEVel <Level>[,<MPMeter>]«. The trigger length is set using »:TRIGger:LENgth <Length>[,<MPMeter>]« and »:TRIGger:BEGin <Index>[,<MPMeter>]«. The corresponding query commands are »:TRIGger:LENgth? [<MPMeter>]« and »:TRIGger:BEGin? [<MPMeter>]«. The progess of waveform acquisition can be checked using »:TRIGger:PROgress? [<MPMeter>]«.

Trigger output is configured using :TRIGger:OUTput <0/1>[,<MPMeter>], :TRIGger:INVert <0/1>[,<MPMeter>], :TRIGger:SYNC <0/1>[,<MPMeter>] for the BNC connector and :TRIGger:BPOUTput <0/1>[,<MPMeter>], :TRIGger:BPINVert <0/1>[,<MPMeter>] and :TRIGger:BPSYNC <0/1>[,<MPMeter>] for the RJ45 connector. In DONE state the waveform values can be queried using the commands

- »:TRIGger[:WAVeform][:Power]:P[1]? [<MPMeter>]«,
- »:TRIGger[:WAVeform][:Power]:P2? [<MPMeter>]« and
- »:TRIGger[:WAVeform][:Power]:P3? [<MPMeter>]«.

Average waveform values can be queried using »:TRIGger[:WAVeform][:Power]:ALL? [<MPMeter>]«. The »:TRIGger[:WAVeform][:Power]:BINary? [<MPMeter>]« command is available for fast and computationally efficient waveform readout.

4.4.3 Radar Pulse Measurements Using the GUI

The LSPM 1.0 Power Meter is able to scan a power waveform for pulses in order to find their positions and averaged power values. See Section 4.4.1 for details regarding triggered value acquisition. In all modes pulses above a specified threshold can be detected if they are longer than 0.5 µs and at

least 0.5 µs apart. Pulse measurements in mode 1 below 30 MHz and 3 are not recommended due to their lower video bandwidths.

For pulses containing one or two samples the pulse's power value is defined as the larger of the two values. For pulses containing at least three samples the pulse's power value is defined as the arithmetic mean of all but the first and last sample value of the pulse, Figure 23 denotes these samples using squares.

Pulses starting before the beginning of a waveform will only be detected if they contain at least two samples. The last sample of the pulse is always discarded and the pulse's power in this special case is defined as the arithmetic mean of all preceding power values. Pulses ending after the end of a waveform are handled in the same way, discarding the pulse's first sample value instead of the pulse's last sample value.

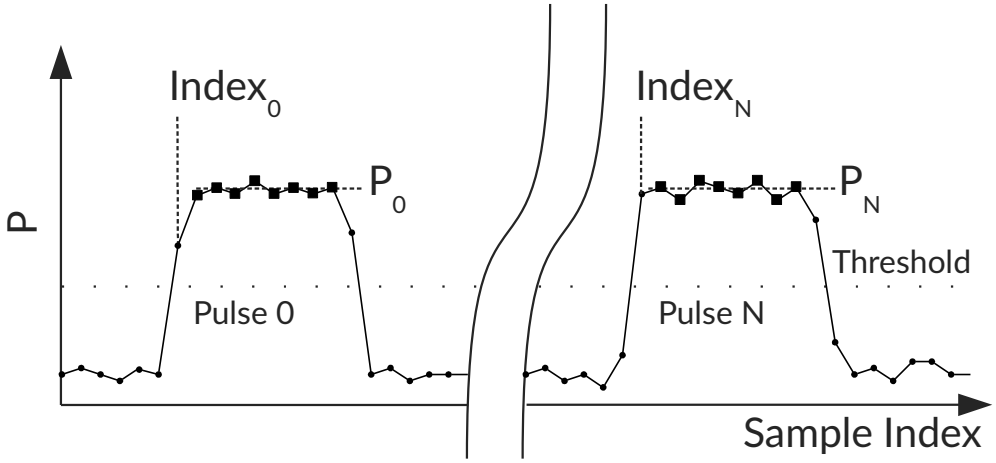


Figure 23: *Principle of radar pulse detection*

In order to enable pulse detection and evaluation within the LSPM GUI select the "Radar" tab and activate "Enable Radar" as shown in Figure 24. Averaged power values for all axes are displayed for channel 1, 2 and 3. Averaged values over all pulses are shown on top, the start index of individual pulses and their averaged power value are displayed below. If no pulses are detected the averaged power values will display "NaN". The sample index denotes the position of the pulse within the waveform, it represents the first sample exceeding the given pulse threshold. The selector field directly above "Index" can be used to step through individual pulses, the first value has an index of zero. An invalid index will display "NaN" for sample index and power value. By selecting the "Display x-axis as time" check box the "Index" values of pulses and x-axis labels of the waveform graph will be displayed in seconds. If the radar subsystem is turned off the "Pulses" field will be set to zero and all other fields be cleared.

The "Arm" and "Force" buttons are duplicated from the "Scope" tab to enable waveform acquisition

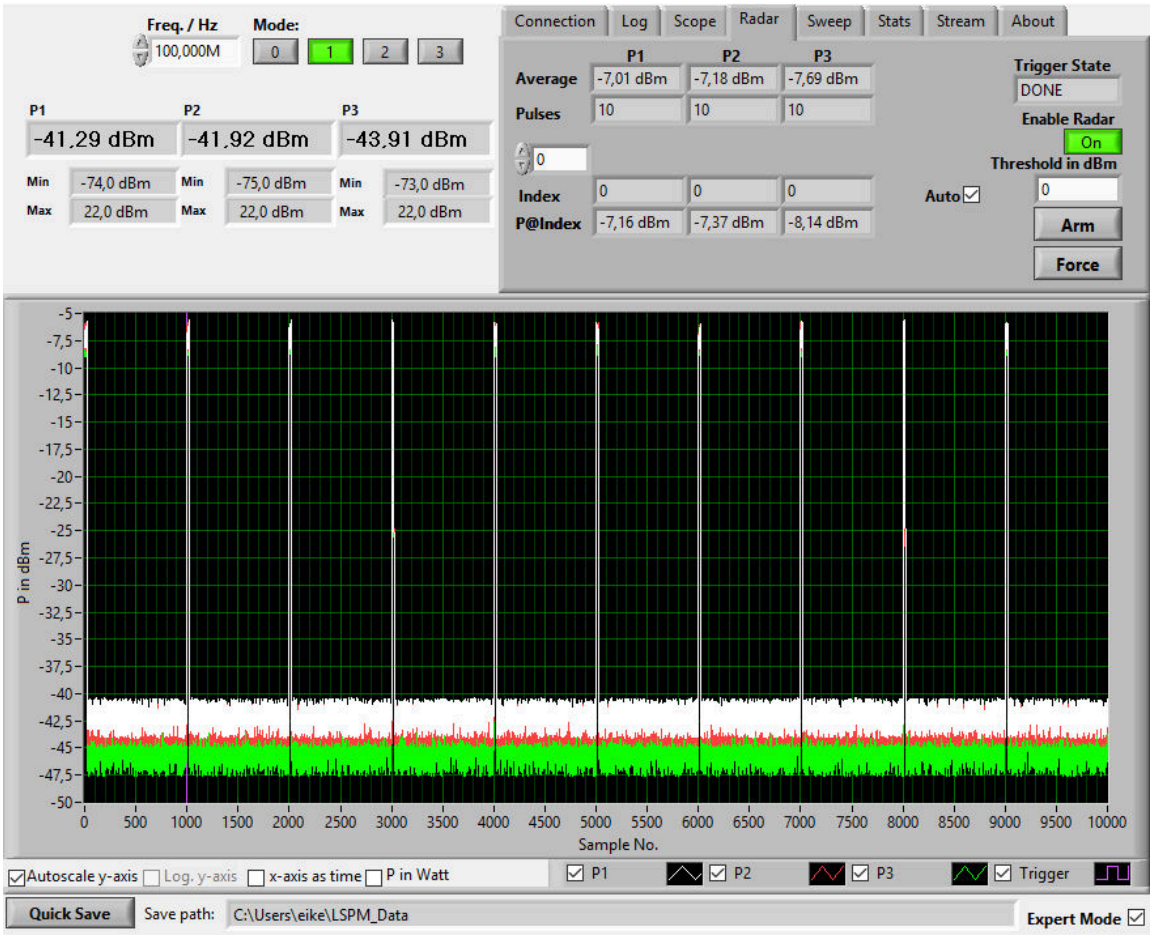


Figure 24: *LSPM GUI, Radar tab*

without switching to the “Scope” tab. All trigger settings are performed via the “Scope” tab. Pulse evaluation is repeated for every update of the power waveform and with every change in the threshold value.

The threshold value for all channels is set via the "Auto" checkbox and the “Threshold” input field. If the "Auto" checkbox is enabled, the arithmetic mean of maximum and minimum power found in the present waveform is used as a threshold. Both minimum and maximum power value are accessible via the statistics subsystem, see Section 4.5. If "Auto" is disabled, the "Threshold" input field sets the threshold value in dBm or Watts. Figure 23 shows the relationship of power waveform values, pulse indices and pulse values.

The number of detected pulses can differ for each channel. The numeric fields “Pulses” give the number of pulses found for each channel.

4.4.4 Radar Pulse Measurements Using SCPI Commands

See Section 4.4.2 for the details of triggered value acquisition. Pulse detection requires the trigger system to be in DONE state. Pairs of sample index and maximum pulse power values can be obtained through the

- »:TRIGger[:WAVeform]:RADar:P[1]? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:RADar:P2? [<MPMeter>]« and
- »:TRIGger[:WAVeform]:RADar:P3? [<MPMeter>]«

commands. Averaged maximum pulse power values can be queried using »:TRIGger[:WAVeform]:RADar:ALL? [<MPMeter>]«. All commands accept an optional parameter specifying the threshold power value. If the parameter is absent or the automatic threshold flag is set to 1, a threshold of 50% relative to the waveform's maximum and minimum value will be used. The automatic threshold flag is set via »:TRIGger[:WAVeform]:RADar:THreshold:AUTO <0/1>[,<MPMeter>]« and can be queried by »:TRIGger[:WAVeform]:RADar:THreshold:AUTO? [<MPMeter>]«. Radar pulse evaluation can be performed multiple times using different threshold values.

4.4.5 Sweep Measurements Using the GUI

The LSPM 1.0 Power Meter is able to evaluate a power waveform containing a level or frequency sweep, thus enabling sped up measurements. See Section 4.4.1 for details regarding triggered value acquisition. One or more sections of the waveform, which align with the sweep's steps, are analyzed independently, yielding a set of averaged power values for each section. Waveform sections have a uniform length and spacing relative to each other. They are spaced in such a way to guarantee well-defined conditions for each section and must take into account the switching and settling characteristics of the setup under test.

In order to enable sweep evaluation within the LSPM GUI select the “Sweep” tab and activate “Enable Sweep”. Sweep evaluation requires the trigger system to be in DONE state. The “Trigger State”

indicator field, the "Force" and the "Arm" buttons are identical with the "Scope" tab. Frequency sweep re-evaluation is performed upon every evaluation parameter change and power waveform update.

The timing of the sweep is configured using the fields "Step", giving the length of a sweep step, "Begin", giving the start of averaging relative to the start of the step, and "End", giving the end of averaging relative to the start of the step. Evaluation starts with the first sample of the power waveform, allowing to shift the waveform by means of a different value for "Trigger Begin" in the "Scope" tab.

There are four frequency sweep modes selectable under "Mode":

- "FIXED" uses the frequency set under "Freq./Hz".
- "LIN" selects a linear frequency sweep, defined by its start frequency "Freq. Begin" in Hertz, the number of frequency steps "Freq. Count" and the linear frequency increment "Freq. Step" in Hertz.
- "LOG" selects a logarithmic frequency sweep, defined by its start frequency "Freq. Begin" in Hertz, the number of frequency steps "Freq. Count" and the incremental factor between two steps "Freq. Step", e.g. 1.1 for increasing the by 10% from one step to the next.
- "LIST" allows adding an arbitrary frequency to a list of frequency values by clicking the "Add" button. The list can be reset using the "Clear" button.

If there are more sweep steps than frequency steps the frequency list will be evaluated from the beginning after reaching its last value.

The total number of sweep steps is displayed in the field "Steps". The selector field next to "Index" is used to select individual sweep steps. The center of each step relative to the trigger event is displayed in the field "Index". The corresponding frequency is shown under "Freq@Index". Averaged power values for the selected sweep step are displayed at the bottom. Selecting an invalid sweep step will display "NaN" for all corresponding values.

By selecting the "Display x-axis as time" check box the waveform's x-axis as well as the fields "Step", "Begin", "End" and "Index" are expressed in seconds instead of samples.

4.4.6 Sweep Measurements Using SCPI Commands

See Section 4.4.2 for the details of triggered waveform acquisition. To set the sweep step length and start/end index of averaging within each step the SCPI commands »:TRIGger[:WAVeform]:SWeep:TStep <TStep>[,<MPMeter>]«, »:TRIGger[:WAVeform]:SWeep:TBegin <TBegin>[,<MPMeter>]« and »:TRIGger[:WAVeform]:SWeep:TEnd <TEnd>[,<MPMeter>]« are used. The sweep mode is set via »:TRIGger[:WAVeform]:SWeep:MODe <Mode>[,<MPMeter>]«. For linear and logarithmic sweeps the start frequency, number of frequency steps and frequency increment is set via »:TRIGger[:WAVeform]:SWeep:BEGin <Freq>[,<MPMeter>]«, »:TRIGger[:WAVeform]:SWeep:COUnt <Count>[,<MPMeter>]« and »:TRIGger[:WAVeform]:SWeep:STEP <Step>[,<MPMeter>]«. A list of arbitrary frequencies can be created incrementally via the »:TRIGger[:WAVeform]:SWeep:LIST:ADD <Freq>[,<MPMeter>]« command. »:TRIGger[:WAVeform]:SWeep:LIST:CLear [<MPMeter>]« is used to clear the arbitrary frequency list.

In any mode »:TRIGger[:WAVeform]:SWeep:LIST? [<MPMeter>]« returns the list of frequencies in accordance with the selected sweep mode. The command »:TRIGger[:WAVeform]:SWeep:IDX? [<MPMeter>]« returns the center sample indices of all sweep steps in the waveform. The averaged power and RSSI values for each step are queried using the commands

- »:TRIGger[:WAVeform]:SWeep[:Power]:P[1]? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:SWeep[:Power]:P2? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:SWeep[:Power]:P3? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:SWeep[:Power]:ALL? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:SWeep:RSsi:P[1]? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:SWeep:RSsi:P2? [<MPMeter>]« and
- »:TRIGger[:WAVeform]:SWeep:RSsi:P3? [<MPMeter>]«.

Frequency corrected trigger waveforms can be obtained using

- »:TRIGger[:WAVeform]:SWeep:WPower:P[1]? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:SWeep:WPower:P2? [<MPMeter>]«,
- »:TRIGger[:WAVeform]:SWeep:WPower:P3? [<MPMeter>]« and
- »:TRIGger[:WAVeform]:SWeep:WPower:ALL? [<MPMeter>]«.

The binary SCPI command :TRIGger[:WAVeform]:SWeep:BINary? Wave,[<MPMeter>] is available for efficient value readout.

4.5 Power Statistics

Two types of power statistics are available for the LUMILOOP LSPM 1.0 Power Meter, continuous power statistics as described in Section 4.5.1 and triggered power statistics based on acquired waveforms as described in Section 4.5.3. Continuous power statistics evaluate all measured power values from statistics enabling to statistics snapshot creation. Continuous power statistics are collected in the background and can be performed over arbitrary periods of time. Triggered power statistics evaluate only power values of waveforms in memory. All SCPI commands of the statistics subsystem apply to both continuously collected and triggered power values.

4.5.1 Continuous Power Statistics using the GUI

All statistics functions are controlled via the the "Stats" tab of the LSPM GUI, see Figure 25. See Section 4.5 for an explanation of the operating principle.

As explained in Section 2.1 continuous power statistics use a physical connection for enabling statistics collection and snapshot creation. Consequently, one master Power Meter or field probe Computer Interface must be set up for continuous power statistics. This also applies to single probe systems where the Power Meter must always be configured to be the statistics master. The first enumerated Power Meter will be configured automatically as the statistics master, all others will be

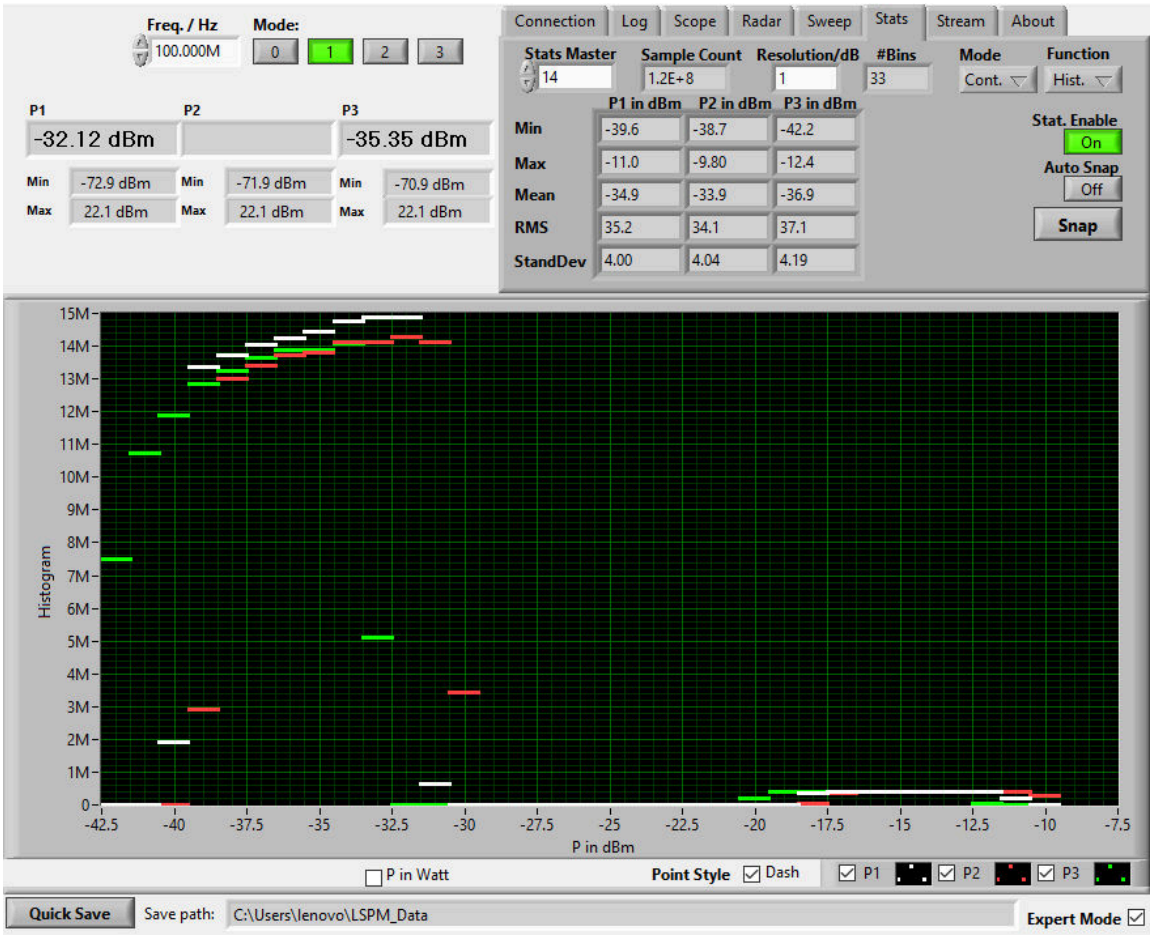


Figure 25: *LSPM GUI, Statistics tab*

configured as statistics slaves. A different Power Meter may be configured as the statistics master via the "Stats Master" drop-down list.

In order to enable continuous power statistics in the LSPM GUI select "Continuous" from the "Mode" drop-down list, this will enable additional controls for continuous power statistics.

The "Stat. Enable" button is used to enable and disable continuous power statistics collection via the statistics master Power Meter. Continuous statistics data can be viewed in the form of statistics snapshots as described in Section 4.5. A statistics snapshot can be created manually by clicking on the "Snap" button. A statistics snapshot will also be created when statistics collection is disabled via the "Stat. Enable" button. Automatic statistics snapshot creation is enabled via the "Auto Snap" button, instructing the GUI to trigger a new statistics snapshot after each update of the statistics display. The total number of samples used for the most recent statistics snapshot is displayed in the "Sample Count" display field. For Multi-Power Meter and Multiprobe systems this number may vary slightly due to minimally different local clock frequencies of each power and E-field detector.

All output is based on the most recent statistics snapshot. See the SCPI command reference in Section 9.7 for a detailed description of scalar and histogram-like statistics values.

The table displayed in the "Stats" tab lists all scalar statistics values for the currently selected Power Meter. Maximum, minimum, arithmetic mean, root mean square and standard deviation are displayed for channels 1, 2 and 3.

Histogram-like statistics are selected via the "Function" drop-down list. Available choices are histogram, discrete relative probability distribution, discrete cumulative probability distribution and discrete complementary cumulative probability distribution.

Histogram-like statistics are displayed in the main plot area. The style of the displayed graphs is adjustable via the "Point Style" checkbox below the plot area. If selected, dashes stretching each power value bin will be displayed, otherwise one point will be plotted for every power value bin.

The power value resolution in dB can be set via the "Resolution" input field, determining the size of the power value bins. Its smallest permissible value, yielding the maximum power value resolution, is 0.005 dB. The number of bins resulting from the set power value resolution and power value distribution is shown in the "#Bins" display field.

4.5.2 Continuous Power Statistics using SCPI Commands

Continuous statistics and triggered statistics are accessible using a common set of SCPI commands. An SCPI command's function is determined by the parameter "Triggered". If set to 0 an SCPI command applies to continuous statistics, if set to 1 an SCPI command applies to triggered statistics. For most commands the parameter "Triggered" is optional, making the SCPI command default to continuous statistics.

The continuous statistics master Power Meter is set by selecting it using »:SYSTem:SERial <Value>« followed by setting its master status to one using »:STATistics:MAster <State>«. The master/slave status of any Power Meter may be queried using »:STATistics:MAster? [<MPMeter>]«.

Continuous power statistics collection is started by issuing »:STATistics:ENable <State>[,<MPMeter>]« with the parameter State set to one for the statistics master Power Meter. Statistics snapshots will be generated on receiving either »:STATistics:SNAPshot [<Triggered>][,<MPMeter>]« or »:STATistics:ENable <State>[,<MPMeter>]« with the parameter State set to zero. The snapshot counter will be incremented by one for every new snapshot, the counter(s) can be queried using »:STATistics:COUnt? [<MPMeter>]«. Using this query enables snapshot synchronization since snapshot query and execution are inherently asynchronous for continuous E-field statistics. Enabling continuous statistics will reset the snapshot counter to zero. »:STATistics:SAMples? [<Triggered>][,<MPMeter>]« returns the number of samples used for the most recent statistics snapshot.

Scalar statistics values can be read using the commands described in Sections 9.7.12 through 9.7.31. Histogram-like statistics values are returned by the commands described in Sections 9.7.33 through 9.7.44.

The resolution for histogram-like values is set using »:STATistics:RESolution <Resolution>[,<MPMeter>]«. The resulting number of bins, the offset of the bin with the smallest power value and the center power value of each bin can be queried via »:STATistics:HISTogram:SIZe? [<Triggered>][,<MPMeter>]«, »:STATistics:HISTogram:OFFset? [<Triggered>][,<MPMeter>]« and »:STATistics:Power? [<Triggered>][,<MPMeter>]« respectively. All statistics values are also available in binary format, see »:STATistics:BINary? [<Triggered>][,<MPMeter>]« for details.

4.5.3 Triggered Power Statistics using the GUI

Triggered power statistics use waveform data for building the scalar and histogram-like values discussed in the previous section. See Section 4.4.1 for a description of waveform acquisition. Triggerd power statistics do not rely on the physical connections required for continuous power statistics. For triggered power statistics there is no statistics master, no statistics enable function and no hardware-based snapshot feature.

In order to access triggered power statistics in the LSPM GUI select "Triggered" from the "Mode" drop-down list, this will enable additional controls for triggered power statistics.

A statistics snapshot based on the most recently acquired waveforms can be created manually by clicking on the "Snap" button. Automatic statistics snapshot creation is enabled via the "Auto Snap" button, making the GUI take a new triggered snapshot upon receiving a new power value waveform. The "Arm" and "Force" buttons are provided for ease of use and are identical in function to the buttons described in Section 4.4.

Triggered Statistics data is viewed in the form of statistics snapshots as described in Section 4.5. All scalar and histogram-like values are controlled and displayed as described in the previous sections. When displaying triggered statistics data the "Function" drop-down list may additionally be set to "Scope" for more convenient viewing of scalar statistics values and triggered waveforms at the same time.

4.5.4 Triggered Power Statistics using SCPI Commands

Triggered statistics SCPI commands require the parameter Triggered to be set to 1 for all related SCPI commands. Statistical evaluation requires a valid set of triggered waveforms, see Section 4.4.2 for details about waveform acquisition.

Snapshot histograms are generated using »:STATistics:SNAPshot [<Triggered>][,<MPMeter>]«. There is no trigger snapshot counter. Instead, triggered snapshot generation is performed synchronously making statistics values available immediately after issuing the SCPI snapshot command.

Scalar and histogram-like statistics values can be obtained using the same SCPI commands as described in Section 4.5.4 with the parameter Triggered set to 1 at all times.

4.6 Stream Recording

The LSPM 1.0 Power Meter supports recording power values at the Power Meter's full sampling rate for virtually unlimited durations of time. Recording time is only limited by the disk space available.

Stream files are stored in the path specified by the `LSPM_SAVE_PATH` environment variable and adhere to the naming convention in Section 10.1.6. File names start with an arbitrary prefix string followed by the Power Meter's serial number and a unique time stamp. A new stream file or set of stream files, containing an updated time stamp, will be created every 1,000,000,000 samples.

The Stream Recording feature stores power data in a binary format in order to reduce both disk space and CPU load. The binary file format is detailed in Section 10.1.6. Binary stream files can be converted into CSV files using the "Bin2csv" conversion tool described in Section 10.1.7.

4.6.1 Stream Recording Using the GUI

To configure and perform stream recordings select the "Stream" tab of the LSPM GUI, as depicted in Figure 26.

If synchronization between multiple Power Meters or with Computer Interfaces is required, make sure to connect the appropriate signal lines. Set the synchronization Power Meter using the "Stream Master" drop-down list, the setting defaults to the first enumerated Power Meter.

The number of samples to be recorded and the number of power values to be skipped after every recorded sample are set via the "StreamLength" and "Stream Skip Count" numeric input fields. Make sure to enable the "ALL" button when using multiple Power Meters.

The synchronization source can be selected via the "Stream Sync" drop-down list. The stream file prefix is set via the "Stream Prefix" input field. The number of samples recorded during a stream recording session is displayed in the "StreamProgess" field. If the "Stream Length" input field is set to a non-zero value pressing the "Stream Enable" button will initiate the recording of the set number of samples. A stream recording may be terminated via the "Stream Enable" button before the set

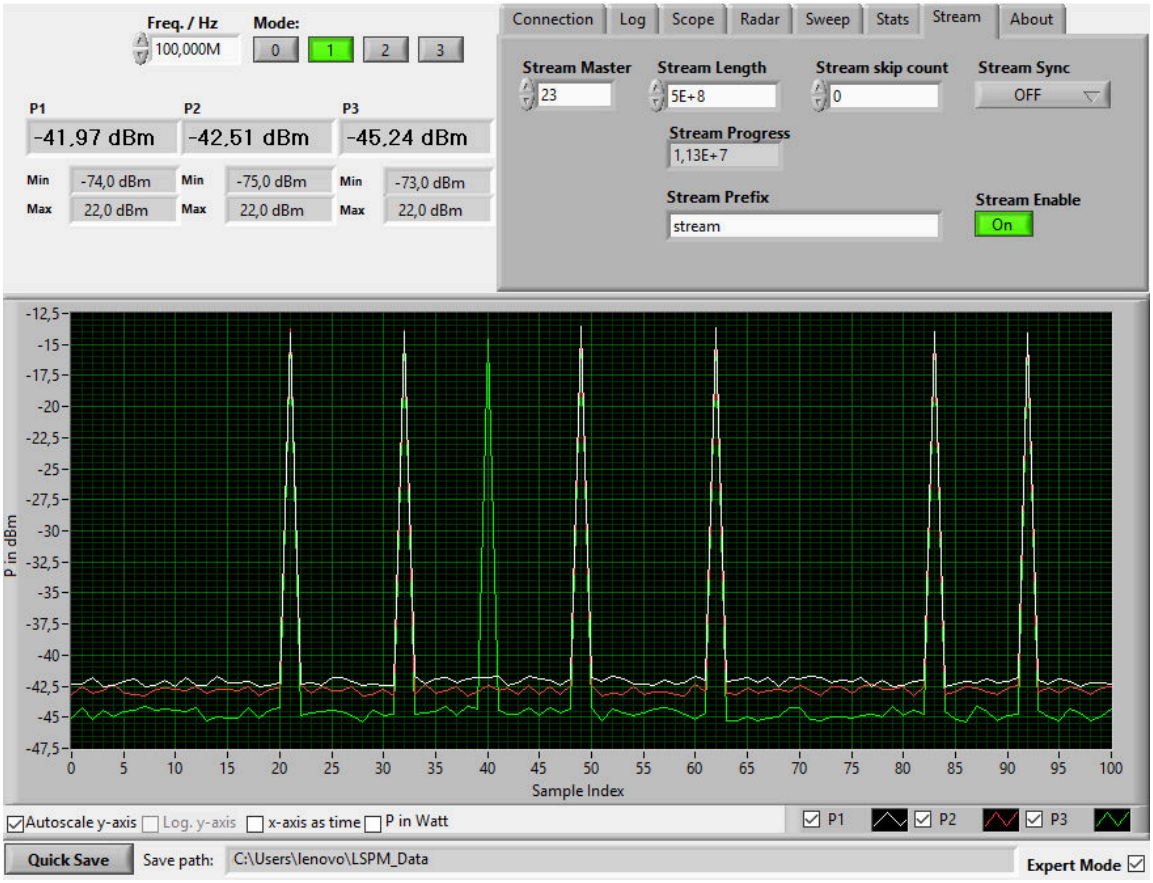


Figure 26: *LSPM GUI, Stream tab for stream recording*

number of samples has been reached. If the "Stream Length" input field is set to zero stream recoding must be terminated manually via the "Stream Enable" button. Manual termination of a stream recording can lead to stream files containing a disparate number of samples.

4.6.2 Stream Recording Using SCPI Commands

Stream synchronization can be configured using the :STReam:SYNC <Sync>[,<MPMeter>] SCPI command. When synchronization is disabled the stream master Power Meter can be changed by setting the former master Power Meter to slave, selecting the new Power Meter and setting it to stream master using the :STReam:MAster <State> and :SYSTem:SERial <Value> commands.

To set the maximum number of samples to be recorded use the :STReam:LENgth <Length>[,<MPMeter>] command. If set to a non-zero number stream recording will terminate automatically after reaching the maximum number of samples.

The stream data rate can be reduced using the :STReam:SKIp? [<MPMeter>] command, specifying the number of skipped samples following each stored sample. The :STReam:PREfix <String>[,<MPMeter>] command can be used to change the stream files' prefix string.

Stream recording is initiated and terminated using the :STReam:ENable <State>[,<MPMeter>]. The progress of stream recording can be monitored using the :STReam:PROgress? [<MPMeter>] command.

4.7 Saving Log Files using the GUI

The "Log" tab contains controls for the file name prefix of LSPM GUI log files. The prefix applies to both one-shot and continuous logging, see Figure 27. Separate log files are created for the "Log", "Scope", "Radar", "Sweep" and "Statistics" tabs. One-shot log files are created using the "Quick Save" button in the lower left corner of the GUI window. The "Quick Save" button will create a new log file for the currently active tab and Power Meter of the LSPM GUI. Continuous logging is enabled in the "Log" tab using the buttons next to the file name prefixes. It saves the respective data for all enumerated Power Meters to disk. For scope and sweep logging every new set of data will be written to a separate file. For all other log data continuous logging will append new sets of data to the respective log file as long as continuous logging is enabled.

Log files are stored in the path specified by the environment variable `LSPM_SAVE_PATH`. The path is displayed in the lower right corner of the GUI window. Log file names consist of the file name prefix extended by the Power Meter's serial number and a timestamp as detailed in Section 10.

Logging of RSSI values, i.e. raw ADC sample values, for all three channels can be enabled by selecting the "include RSSI" check box. The feature is available for "Basic Log", "Scope Log" and "Sweep Log" files.

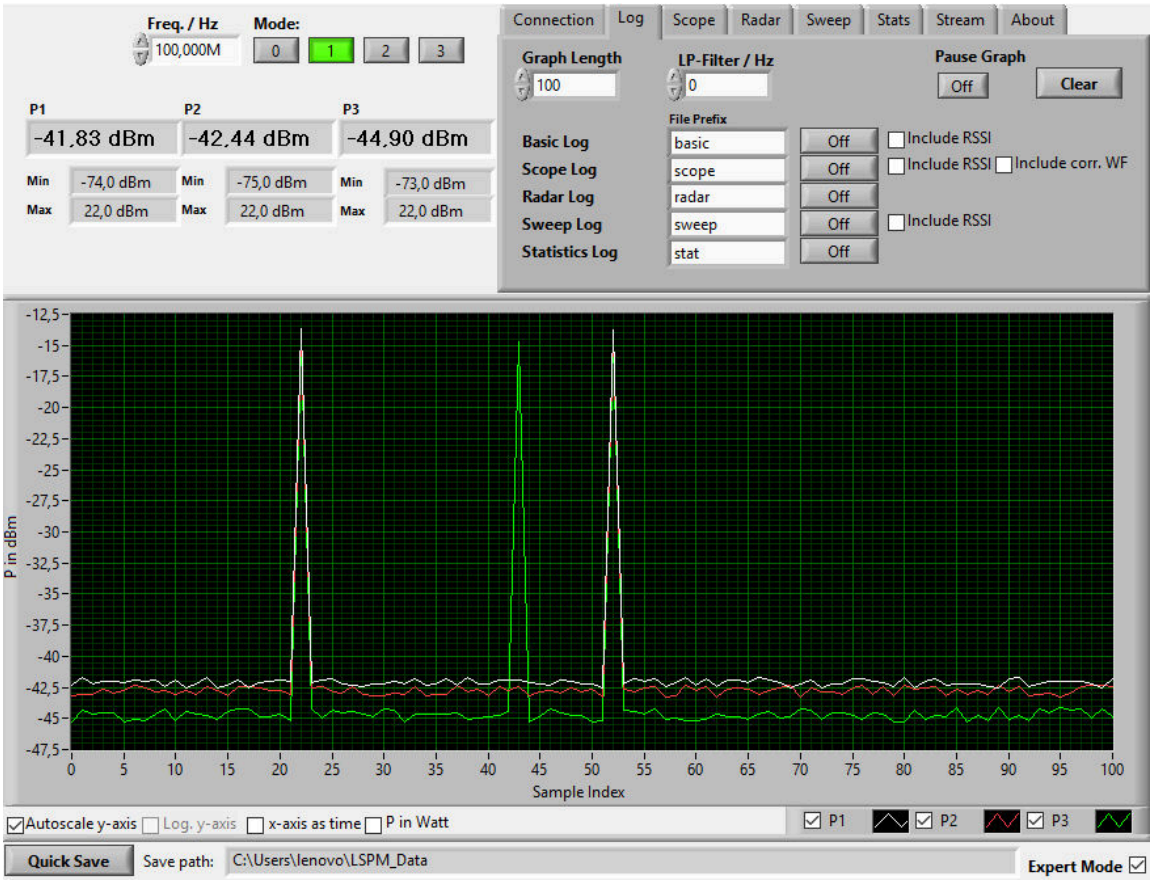


Figure 27: *LSPM GUI, Data Logger tab*

5 Third Party EMC Software

This section describes the setup of the LSPM 1.0 Power Meter in third party EMC test automation software. Third party support files are installed in separate directories of the `lib` sub-directory of the LSPM 1.0 install path.

Before running any third party EMC test automation software follow the hardware setup instructions detailed in Section 4.1 on page 24 and start the TCP Server as described in Section 4.2.1 on page 24.

Starting the TCP Server, mode setting and starting third party software can be automated by the "LSPM-EMC-Starscript.pl" Perl script in the `bin` sub-directory of the LSPM 1.0 install path. "Strawberry Perl" has to be installed in order to use the start script. In Line 28 and 29 the paths for the LSPM TCP-Server and the third party software must be set appropriately. The measurement mode is defined in line 30, with a default value of 1. After closing the third party software the TCP-Server will be terminated as well.

The LSPM GUI is not required when using third party EMC test automation software, with the notable exception of setting the mode. However, the LSPM GUI may be run in parallel as long as it is used solely for monitoring and no settings are changed using the LSPM GUI, i.e. the user must not modify any LSPM GUI controls, e.g. mode, frequency, low-pass filter frequency, etc. The LSPM GUI is designed in such a way that it will not apply any settings on its own.

5.1 EMC32

LUMILOOP recommends using the most recent version of the R&S EMC32 measurement software since EMC32 integration is always tested against the most recent EMC32 release. The LSPM 1.0 Power Meter is supported by EMC32 version 10.3 and later.

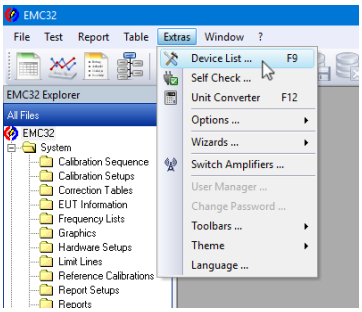
Copy all device configuration files ending in `DeviceConfiguration` from the `EMC32-10.3` sub-directory of the LSPM installation path's `lib` directory to the EMC32 program data path's `Configuration\Power Meters` sub-directoy, a typical location is `C:\ProgramData\EMC32\Configuration\Power Meters`.

5.1.1 CW fields

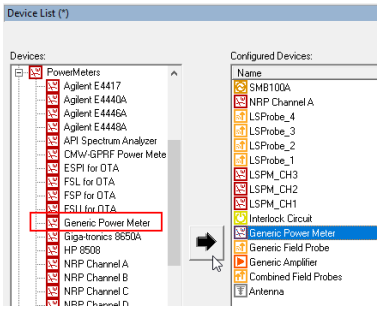
To set up the LSPM 1.0 Power Meter in EMC32 run the TCP Server, then start EMC32.

Open the EMC32 Device List via "Extras→Device List" in the menu bar as shown in Figure 28(a). In the "Device List" window select "Generic Power Meter" from the "Devices:" list's "Power Meters" category and create a new "Configured Device" by clicking on the right-pointing arrow in the center as shown in Figure 28(b). This will create a new entry named "Generic Power Meter". Optionally add a second and a third "Generic Power Meter" entry by clicking the right-pointing arrow an additionally one or two times. The first Power Meter will be used for channel 1 power value queries,

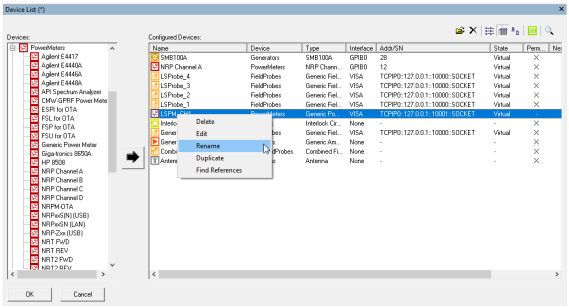
the second for channel 2 and the third for channel 3. Rename the first "Generic Power Meter" entry to "LSPM_Ch1" and the optional second and third entries to "LSPM_Ch2" resp. "LSPM_Ch3" via "right-click→Rename" as shown in Figure 28(c).



(a) Opening the Device List



(b) Adding Generic Power Meter(s)



(c) Renaming Power Meters

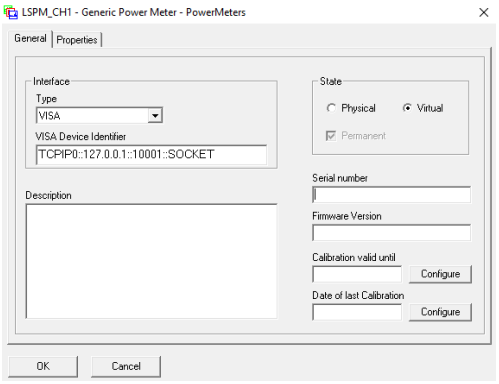
Figure 28: *Adding the LSPM 1.0 Power Meter in EMC32*

Use "right-click→Edit" to open the Generic Power Meter's settings. In the "General" tab shown in Figure 29(a) edit "VISA Device Identifier" to configure the IP address and TCP port of the LSPM TCP Server. The identifier string has the generalized format "TCPIP0::<IP address>::<TCP Port>::SOCKET". Usually the LSPM TCP Server is run on the same computer and listening to the default TCP port 10,001. Consequently, the default identifier string is "TCPIP0::127.0.0.1::10001::SOCKET". All other settings in the "General" tab are optional and may be left unchanged.

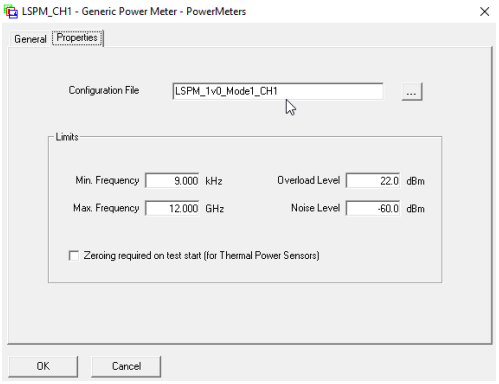
Select the "Properties" tab shown in Figure 29(b) through (d), edit all parameters as detailed in Table 2. Select the appropriate file for "Configuration File" located in `...\ProgramData\EMC32\Configuration\Power Meters`. The low pass filter value in the `.DeviceConfiguration` files can be adjusted by the user to accommodate longer or shorter settling times.

To finish the setup restart EMC32 and open the EMC32 Device List via "Extras→Device List". Use "right-click→Edit" for the "LSPM 1.0" entry appropriate for the measurement task and change "State" from "Virtual" to "Physical". This will prompt EMC32 to connect to the LSPM TCP Server. After establishing the correct mode the "Serial number" value turns blue.

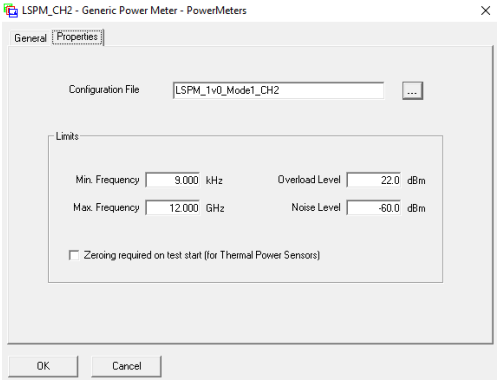
For multiple Power Meters, the configuration files have to be adapted so a specific LSPM gets se-



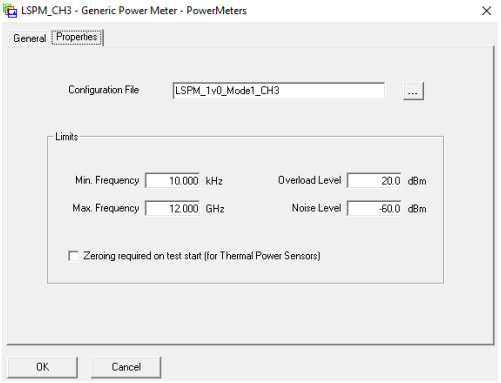
(a) General Settings



(b) Channel 1 Power Meter Properties



(c) Channel 2 Power Meter Properties



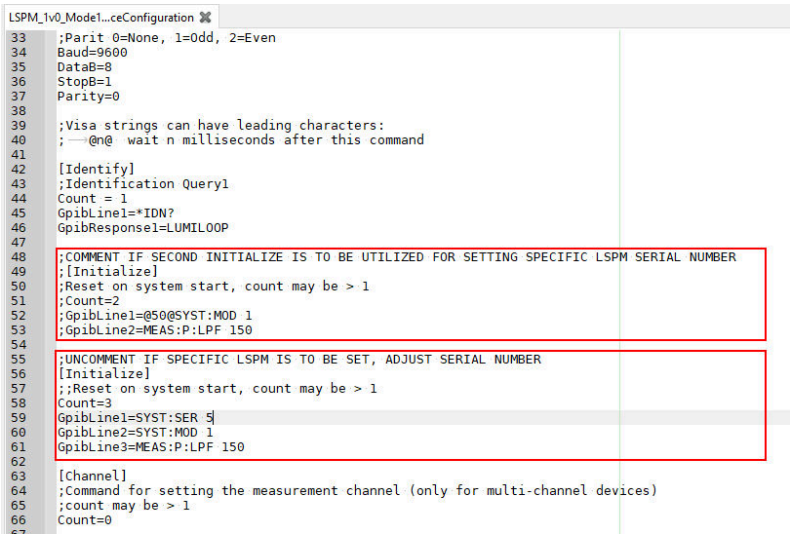
(d) Channel 3 Power Meter Properties

Figure 29: *Configuring LSPM 1.0 Power Meters in EMC32*

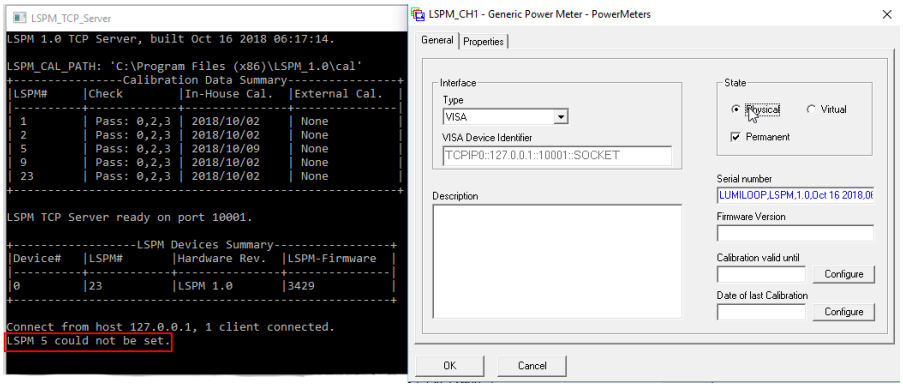
Table 2: *EMC32 Property tab values for Channel 1 Power Meter*

Setting	Channel 1
Min. Frequency	9.000 kHz
Max. Frequency	12.000 GHz
Overload Level	22 dBm
Noise Level	-60 dBm
Zeroing required on test start	unchecked
Configuration File	LSPM_1v0_Mode1_CH1.Device-Configuration

lected by a EMC32 client connection for communication. The first "[Initialize]" block encompassing line 49 to 53 has to be commented. The second "[Initialize]" block encompassing line 55 to 61 must not be commented out. An additional SCPI command is send to the LSPM TCP Server setting the active Power Meter. The specific serial number is set in line 59. After changing "State" from "Virtual" to "Physical" check the TCP-Server window. If the setting of the active LSPM failed, an error message will be printed to the TCP Server window, as depicted in Figure 30(b). If



(a) Modification of LSPM_1v0_Mode1_CH1.Device-Configuration file



(b) Check LSPM TCP Server for error messages

Figure 30: *Adapting LSPM device configuration files for selecting specific LSPM*

5.2 BAT-EMC

The BAT-EMC test automation software supports CW and pulsed power measurements. BAT-EMC requires a DLL file, make sure that "FieldP_Lumliloop_LS12.dll" is present in the BAT-EMC directory "...\BAT-EMC\BAT-EMS".

Please import the provided "One input Power Meter" equipment models listed in Table 3 by right-

clicking on "One input Power Meter" inside the "Equipment" tree structure as shown in Figure 31.

Table 3: *BAT-EMC equipment model files for CW and pulsed powers*

CW	Pulsed
LSPM_1.0_CW1.xml	LSPM_1.0_Pulse1.xml
LSPM_1.0_CW2.xml	LSPM_1.0_Pulse2.xml
LSPM_1.0_CW3.xml	LSPM_1.0_Pulse3.xml

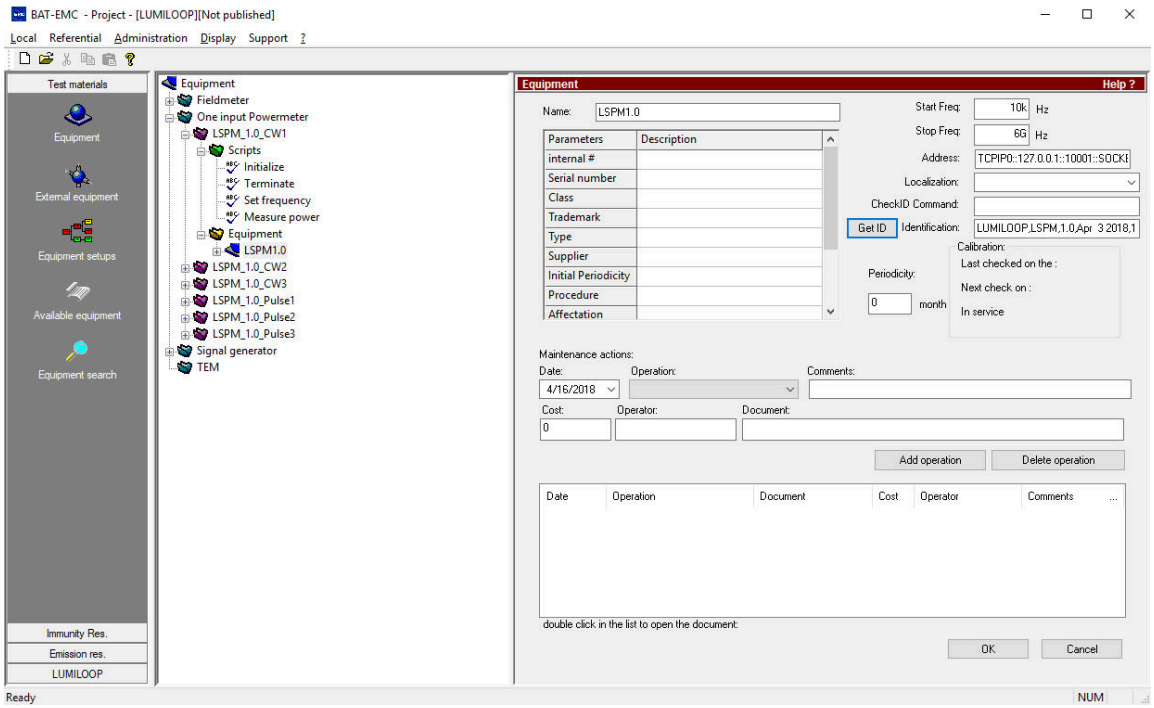


Figure 31: *BAT-EMC Equipment editor, network configuration*

If the IP address and/or the port number of the LSPM connection differ from the default values of localhost and port 10,001 go to the "Equipment" subsection and change the "Address" input field appropriately, see also Figure 31.

BAT-EMC requires the LSPM 1.0 Power Meter to be configured before performing measurements. Before starting BAT-EMC start the LSPM TCP Server and optionally LSPM GUI to set a mode other than 1 as described in Sections 4.2.1 through 4.2.4.

5.2.1 CW power measurements

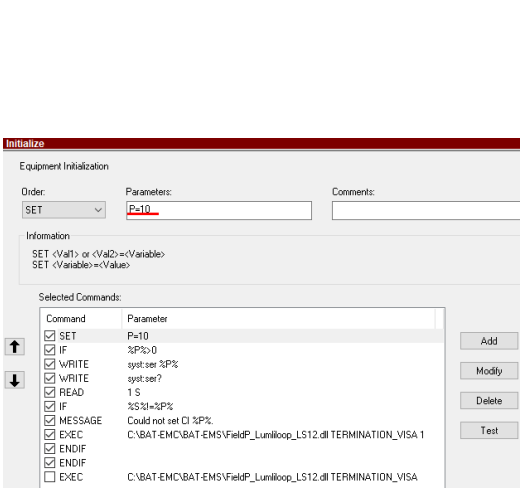
The One input Power Meter model "LSPM_1.0_CW1" handles all communication including setting/checking the operating frequency as well as retrieving channel 1, channel 2 and channel 3 power

values. The Power Meter returns the channel 1 power value and stores the power values of channel 2 and channel 3 in global variables named "AB" and "AC". The two One input Power Meter models "LSPM_1.0_CW[2/3]" retrieve the global variables and return the respective channels' power values. When measuring channel 1, channel 2 and channel 3 power values, the "LSPM_1.0_CW1" Power Meter model must be called first.

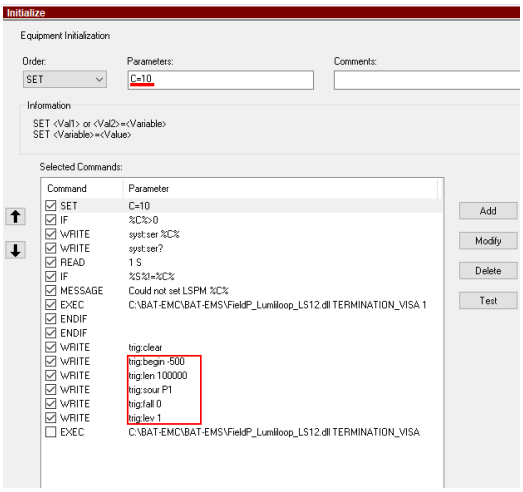
5.2.2 Pulsed power measurements

The "One input Power Meter" model "LSPM_1.0_Pulse1" handles all communication including setting/checking the operating frequency as well as retrieving channel 1, channel 2 and channel 3 power values. Additionally, the "One input Power Meter" model includes commands for trigger subsystem configuration, trigger detection and radar pulse property retrieval. The "One input Power Meter" returns the channel 1 power value and stores the power values of channel 2 and 3 using global variables named "AB" and "AC". The two "One input Power Meter" models "LSPM_1.0_Pulse[2/3]" read these global variables and return the respective channels' power values. When measuring channel 1, channel 2 and channel 3 power values, the "LSPM_1.0_Pulse1" "One input Power Meter" model must be called first.

The trigger subsystem's configuration can be modified via the "One input Power Meter" model's "Initialize" script, the relevant commands are shown inside the red frame in Figure 32(b). See Section 4.4.2 for more information about the trigger subsystem's SCPI commands.



(a) LSPM_1.0_CW1 Initalize script



(b) LSPM_1.0_Pulse1 Initalize script

Figure 32: *Configuring the Initialize scripts for LSPM 1.0 CW and pulse measurements in BAT-EMC*

Figure 33 shows the "Measure level" script of the "LSPM_1.0_Pulse" "One input Power Meter" model. Every call to the "Measure level" script arms the trigger subsystem, waits for the trigger subsystem to acquire a waveform, checks the number of pulses and queries the averaged pulsed power values.

The default setting of the "LSPM_1.0_Pulse" "One input Power Meter" model is suitable for the GMW-3097 standard. It will record 400,000 samples (see the trigger length setting in Figure 32(b)),

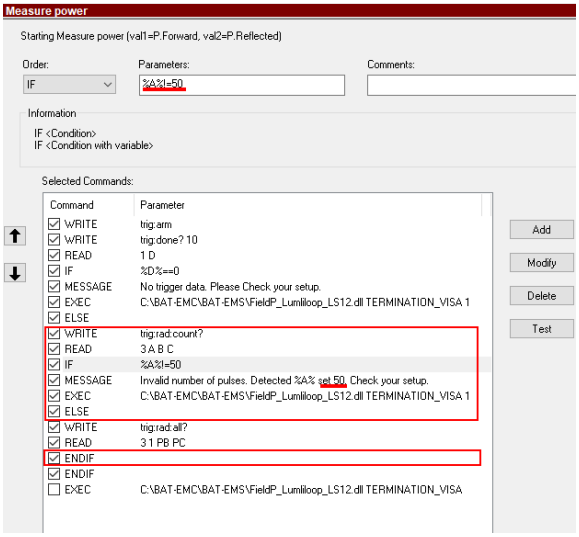


Figure 33: *Addressing a specific Power Meter using BAT-EMC*

verify that there are a total of 50 pulses (see the topmost red frame in Figure 33) and retrieve the averaged pulse channel 1, channel 2 and channel 3 power values. The trigger source, trigger length and expected pulse count can be modified by editing the respective scripts. Pulse count checking can be disabled by unchecking all boxes in the two lower red frames in Figure 33.

If more than one Power Meter are attached to the host computer set the variable PM in Figure 32 to a value other than zero. Doing so will enable the setting and verification of the Power Meter's serial number. The Figure 32 demonstrates setting the Power Meter serial number to 10, the serial number must be changed to match the desired Power Meter's serial number. The variable is available for both CW and pulsed "One input Power Meter" models.

For monitoring multiple Power Meters in parallel create a copy of all "One input Power Meter" models by right-clicking on the "One input Power Meter" models and choosing "duplicate". Adjust all model names and Power Meter serial number variable settings appropriately.

6 Virtual Power Meters

The LSPM TCP Server is capable of instantiating virtual Power Meters including the simulation of arbitrary power patterns.

Virtual Power Meters can replace physical meters and signal generators during measurement setup preparation, feature demonstration, third party EMC software development and off-line signal analysis, including multi-Power Meter setups.

The following virtual Power Meter properties can be can be configured:

- Power Meter serial number,
- Power Meter channel 1, 2 and 3 power value, see pattern description below.

Virtual Power Meter serial numbers must be unique, i.e. must not duplicate the serial numbers of any physical or virtual units.

Power value patters channel 1, 2 and 3 are simulated by summing up RSSI patterns of the following types:

CW (continuous wave)
: Generates constant RSSI values.

Noise
: Generates random RSSI values with a configurable maximum amplitude whose time-average is zero.

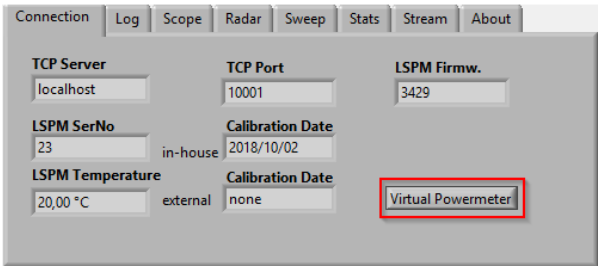
Pulse
: Generates periodically pulsed RSSI values whose OFF-value is zero and whose ON-value is configurable for each channel. The pattern's ON-time and period are configurable and apply to all channels.

List
: Generates a sequence of arbitrary RSSI values, optionally calculated from a list of power values, using the present mode and operating frequency. The sequence is repeated indefinitely.

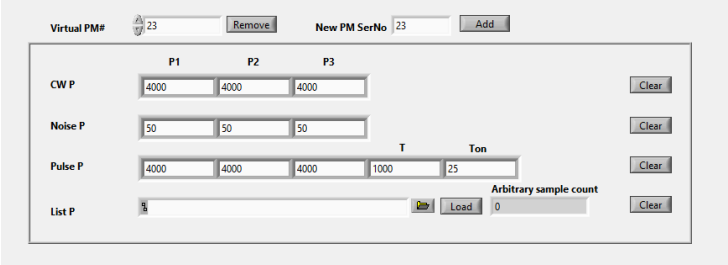
Virtual Power Meters support all modes. Triggering is supported with the exception of external trigger input and output. Virtual Power Meters support the collection of continuous statistics. By default virtual Power Meters are configured as statistics slaves. Statistics snapshots are recorded when the statistics master Power Meters triggers a snapshot, the master Power Meter may be physical or virtual. Note that a virtual statistics master cannot control physical statistics slave Power Meter.

6.1 Controlling Virtual Power Meters Using the GUI

To open the "Virtual Power Meter Control Panel" press the "Virtual Power Meter" button in the "Connection" tab as shown in Figure 34.



(a) LSPM GUI



(b) Virtual Power Meters Control Panel

Figure 34: *Opening the Virtual Power Meters Control Panel of the LSPM GUI*

Click the "Add" button to add a new virtual Power Meter with the specified serial number. The "Virtual PM#" ring lists all connected virtual Power Meters by their serial numbers. Click the "Remove" button to disconnect the currently selected Power Meter.

The selected virtual Power Meter can be configured using the controls in the frame below. The power pattern of channel 1, 2 and 3 are set using the controls inside the frame below. Settings take effect as soon as they are entered. The "Clear" buttons can be used to reset the corresponding settings to their default values.

Power value lists can be loaded from Scope-Log CSV files whose format is described in Section 10.1.2 on page 116. The file name is selected via the "List P" control. Clicking "Load" will append all power values to the virtual Power Meter's list buffer. The list can be emptied using the corresponding "Clear" button.

6.2 Controlling Virtual Power Meters Using SCPI Commands

Virtual Power Meters are added using »:VIRTual:CONnect [<SER>]«. »:VIRTual:SERial?« lists the serial numbers of all virtual Power Meters. The currently selected virtual Power Meter can be removed using »:VIRTual:DISConnect« The virtual Power Meter's serial number and parametric power patterns are set/queried using the following SCPI commands:

- »:VIRTual:CW <RSSI1>,<RSSI2>,<RSSI3>«/»:VIRTual:CW?«,
- »:VIRTual:NOIse <NOISE1>,<NOISE2>,<NOISE2>«/»:VIRTual:NOIse?« and
- »:VIRTual:PULse [<RSSI1>],[<RSSI2>],[<RSSI3>],[<T>],[<Ton>]«/»:VIRTual:PULse?«.

Arbitrary power values are appended to the virtual Power Meter's list using »:VIRTual:LIST <RSSI1_1>,<RSSI2_1>,<RSSI3_1>[,...,<RSSI1_N>,<RSSI2_N>,<RSSI3_N>]« for arbitrary RSSI, »:VIRTual:PLIST <P1_1>,<P2_1>,<P3>[,...,<P1_N>,<P2_N>,<P3_N>]« for arbitrary power values. The complete list of power values is queried using »:VIRTual:LIST?«. »:VIRTual:LCNt?« returns the number of samples in the list. »:VIRTual:LCLear« clears the list of values.

7 Power Meter Calibration

LSPM 1.0 uses in-house and external calibration files for calculating accurate power values based on the ADC values generated by each of the Power Meter's channels, see Figure 4 on page 14 for a principle block diagram of the Power Meter.

In-house linearity and frequency compensation files whose format is detailed in Section 10.3.1, contain the relationship between detector input power and returned ADC value. They cover all channels, modes and frequencies. The LSPM TCP Server interpolates between these recorded data points to obtain a linearity-compensated detector characteristic for each channel.

The reference power value for the in-house power value calibration is established using a transfer standard, i.e. a reference Power Meter. Using a transfer standard results in an increased calibration uncertainty and represents no accredited procedure. Therefore, LSPM supports the inclusion of externally generated, accredited calibration data.

External power calibration files whose format is detailed in Section 10.3.2, contain the power value errors in Decibels for a constant input power and desired number of frequencies. Typically, 0 dBm is used.

A different power value can be chosen as long as the power level can be generated for the entire calibration frequency range. Though it is recommended to use the same set of calibration frequencies for both external and internal power calibration, it is permissible to use a different set of frequencies. The LSPM TCP Server will interpolate or extrapolate the calibration factors as needed.

7.1 Calibration Data Acquisition

External power calibration is always performed in mode 0 and 3 against the in-house power calibration data, recording the actual and indicated power values. The usage of preexisting external calibration data is prevented by issuing a »:CALibration:EXTernal <Value>[,<MPMeter>]« command with the parameter Value set to zero, this command is mandatory.

The LSPM TCP Server will create a calibration log file when instructed to do so by the »:CALibration:LOGging <Value>« SCPI command, Value must be set to one. The command must be issued via the same TCP/IP connection as all other calibration commands. Log files are saved to the directory specified by the `LSPM_SAVE_PATH` environment variable. Calibration logs can be imported automatically using the CalImport tool detailed in the next section.

For maximum accuracy the Power Meter should be powered-up at least 5 minutes before the first measurement, allowing the Power Meter's temperature to settle.

7.2 Automated Calibration Data Import

External power calibration CSV files need to be derived from the log files acquired during the accredited calibration procedure. The LSPM software installation includes the "CalImport" program

for simple, quick and error-free calibration data import. The default file format and naming conventions are stated below. If your calibration setup cannot meet these conventions contact LUMILOOP for inclusion of a suitable import function in the "CalImport" program.

8 SCPI Communication Basics

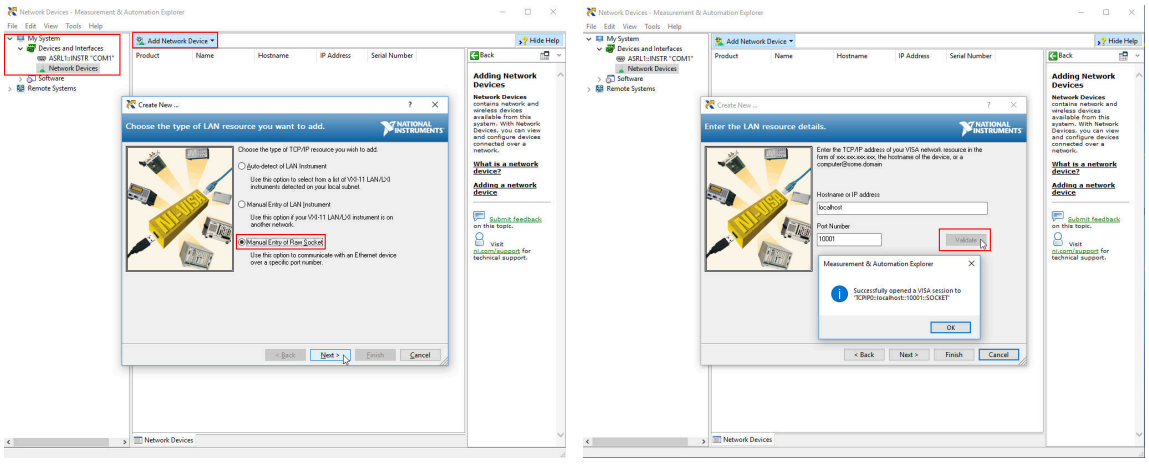
The LSPM TCP Server provides a convenient text command-based interface to Power Meter measurement data, it supports up to 32 concurrent TCP/IP client connections. All commands sent to the TCP Server are ASCII text commands which terminated by a newline (\n), carriage return (\r) or semicolon (;) character or any combination thereof. Replies sent by the TCP Server in reply to queries are single lines of text terminated by a carriage return character followed by a newline character (\r\n). Binary replies deviate from this convention, see the individual commands' descriptions in Section 9 for further details.

This section gives examples of communication with the TCP Server using standard libraries and utilities.

8.1 National Instruments VISA

NI VISA is a cross-platform library for unified communication with measurements connected via GPIB, serial port, Network socket, etc. NI VISA handles all low-level configuration and provides buffered bidirectional IO streams. This sections explains how to configure a socket connection to the LSPM 1.0 TCP Server using the debug tool provided with the NI VISA library and how to test it. NI VISA needs be downloaded from the "National Instruments" homepage and installed first.

Open the NI VISA Measurement and Automation Explorer (NI MAX). Add a new network device by selecting the subsection"Network devices" of "Devices and Interfaces" next click on "Add Network Device". As shown in Figure 35(a), select "Manual Entry of Raw Socket" and click "Next". As shown in Figure 35(b) enter the correct "Hostname or IP" the TCP "Port Number", click "Validate" to connect to the LSPM TCP Server. Both NI MAX and the TCP Server's output will indicate a successful connection. Click "OK" and "Finish" to return to the NI MAX main window.



(a) (b)

Figure 35: *Connection to LSPM TCP Server through NI MAX*

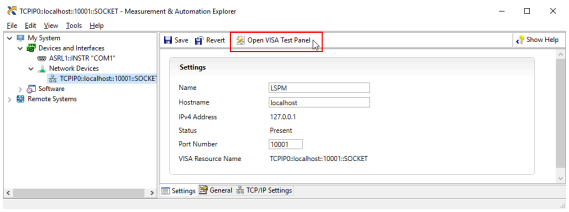
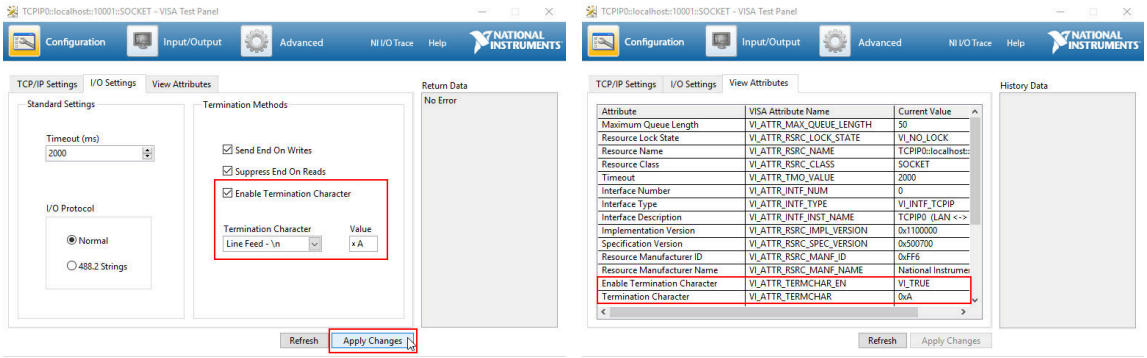


Figure 36: *Starting NI VISA Test Panel*

Right-click on the newly created network device and select "OPEN VISA Test Panel" as shown in Figure 36. No changes are required in the "TCP/IP Settings" tab. Set "Enable Termination Character" in the "IO Settings" tab, click "Apply Changes" and observe the return data output as shown in Figure 37(a). This step needs to be repeated for every NI VISA Input/Output debug session. The "View Attributes" tab in Figure 37(b) shows shows the VISA parameters "VI_ATTR_TERMCHAR_EN" set to "VI_TRUE" and the "VI_ATTR_TERMCHAR" attribute set to "0xA". When using the NI VISA library for connecting to the LSPM TCP Server make sure to set all VISA parameters identically.



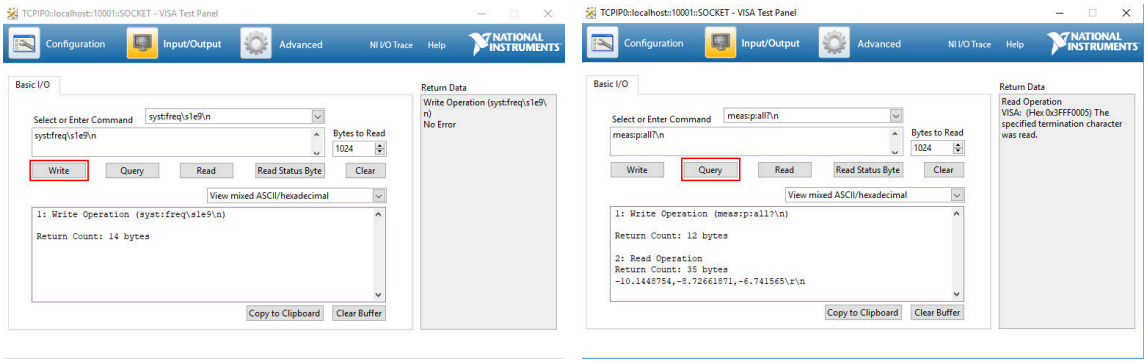
(a) (b)

Figure 37: *Configuring VISA TCP/IP socket parameters*

Click on "Input/Output" to start testing NI VISA communication. Clicking "Query" will retrieve the identification string using the "*idn?\n" command. As shown in Figure 38 the frequency is set to a specific value by entering and writing the "syst:freq 1e9\n" command. Set the command to "meas:p:all?\n" and click "Query" to obtain three power values. Make sure no errors are produced at any time.

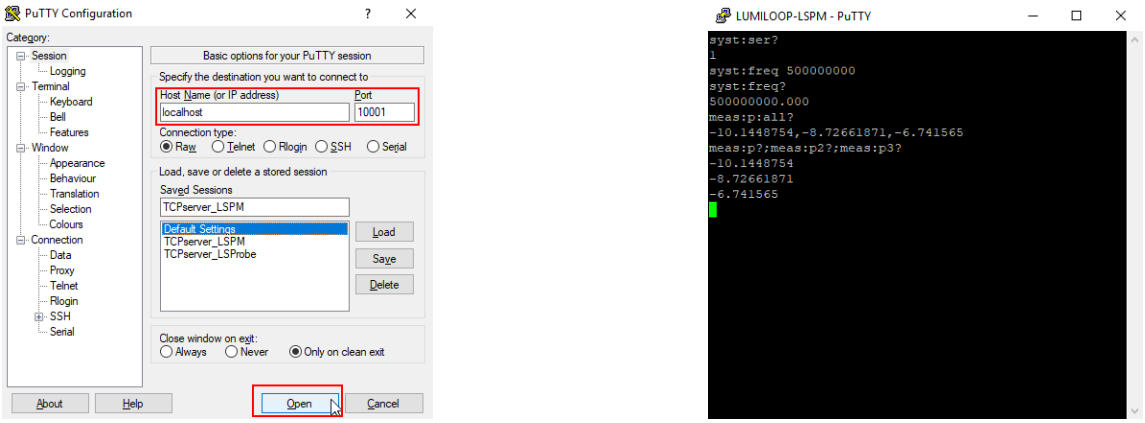
8.2 Raw TCP socket communication using PuTTY

Run PuTTY and enter the host name or IP address and the TCP port number. Set "Connection type" to "Raw" as shown in Figure 39(a). Optionally, save the session configuration for later use. Click "Open" to start the terminal session. Figure 39(b) shows the terminal window. Enter commands and press Return when done. Query commands will generate one reply line each. Multiple commands may be sent in rapid succession by separating them by semicolons.



(a) Setting the frequency (b) Power value query

Figure 38: *Testing NI VISA LSPM TCP Server writes and queries*



(a) Session settings (b) LSPM TCP-Server session

Figure 39: *Using PuTTY*

9 SCPI Command Reference

9.1 Multi-Power Meter Behavior

Most SCPI commands support the optional MPMeter parameter determining the Multi-Power Meter behavior of the respective command. The following MPMeter parameters are available:

EMPTY
: If the MPMeter parameter is omitted the command will only be executed for the presently active Power Meter.

0
: If the MPMeter parameter is set to zero the command is executed for all enumerated Power Meters. Power Meters will be enumerated and displayed sorted by their serial numbers starting with the smallest value.

N
: If the MPMeter parameter is set to a number N that is greater than zero the command will be executed for all Power Meters of the specified Multi-Power Meter setup. Multi-Power Meter systems are defined using the :MPMeter:SERial <MPMeter>,<SN1>[,<SN2>,...,<SNN>] SCPI command, the identifier N can be any positive, non-zero integer value. Use :MPMeter:SERial? <MPMeter> to query the Multi-Power Meter system.

When any SCPI command is executed for more than one Power Meter, i.e. the MPMeter parameter refers to multiple Power Meters, the command behaves as if executed for each Power Meter of the specified Multi-Power Meter system successively. The order of execution is the same as the serial numbers returned by »:MPMeter:SERial? <MPMeter>«. All output will be joined on a single line by replacing line breaks between the output of different Power Meters with commas. For the sake of brevity the return value descriptions in the following sections do not explicitly state the return values for Multiprobe calls if they conform to the format explained above.

9.2 Generic Commands

9.2.1 *CLS

Clear all status registers and structures, e.g. error queue.

9.2.2 *ESE <ESR>

Set event status enable register. This feature is currently not implemented.

PARAMETERS:
: Integer value for event status register.

9.2.3 *ESE?

Query event status enable register. This feature is currently not implemented.

RETURN VALUE:
Returns the integer value of the event status register.

9.2.4 *ESR?

Query the most recent error status register value. The error will be removed from error queue.

RETURN VALUE:
Value of most recent errors in error queue.

9.2.5 *IDN?

Query TCP Server identification string.

RETURN VALUE:
Comma-separated string, consisting of maker, product name, product version, TCP server build date and TCP server build time, e.g. »LUMILOOP,LSPM,1.0,Jun 2 2018,08:07:06«.

9.2.6 *OPC

Set operation complete flag after the completion of the previously sent command. This feature is currently not implemented.

9.2.7 *OPC?

Query operation complete flag. This feature is currently not implemented.

RETURN VALUE:
Always 1.

9.2.8 *RST

Reset TCP server. This will close all previously opened Power Meters, rescan the USB bus and open all detected Power Meters. This will perform a power-on reset of all Power Meters.

The TCP server will print enumeration status information to its standard error output.

9.2.9 *SRE <int>

Set service request enable register. This feature is currently not implemented.

PARAMETER:
Integer value of service request enable register.

9.2.10 *SRE?

Query service request enable register. This feature is currently not implemented.

RETURN VALUE:
Always 0.

9.2.11 *STB?

Query status byte. Note that only bit 2 is currently implemented.

RETURN VALUE:
The returned integer value contains the following status flags:

BIT 0
Unused bit.

BIT 1
Protection event flag, currently not implemented.

BIT 2
Error/Event queue message available.

BIT 3
Questionable status, currently not implemented.

BIT 4
Message available, currently not implemented.

BIT 5
Standard event status register, currently not implemented.

BIT 6
Service request, currently not implemented.

BIT 7
Operation status flag, currently not implemented.

9.2.12 *TST?

Initiate self test and return test result. This feature is currently not implemented.

RETURN VALUE:

0 on success and 1 on passing the self test.

9.2.13 *WAI

Wait for the completion of the previously issued command. This feature is currently not implemented.

9.3 :SYSTem Commands

9.3.1 :SYSTem:WAIT <mSec>

Wait for mSec milli seconds.

9.3.2 :SYSTem:ERRor[:NEXT]?

Query most recent entry in system error queue and remove entry from error queue.

RETURN VALUE:

Returns comma-separated error and error message string enclosed in quotes, e.g. »0,"No error".«

9.3.3 :SYSTem:ERRor:COUNt?

Query number of entries in system error queue.

RETURN VALUE:

Number of values in error queue.

9.3.4 :SYSTem:SERial <Value>

Select active Power Meter by serial number.

PARAMETER:

One serial number out of the list returned by :SYSTem:SERial? [<MPMeter>] with the MPMeter parameter set to 0.

9.3.5 :SYSTem:SERial? [<MPMeter>]

Query serial number of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

Unsigned integer-valued serial number of selected Power Meter or comma-separated list of Power Meter serial numbers for Multi-Power Meter systems. If no Power Meters have been enumerated the command will return NAN.

9.3.6 :SYSTem:MAKer? [<MPMeter>]

Query maker identification string of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

Name string of the device maker, e.g. »LUMILOOP«.

9.3.7 :SYSTem:DEVice? [<MPMeter>]

Query device identification string of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

Name string of the device, e.g. »LSPM«.

9.3.8 :SYSTem:VERSion? [<MPMeter>]

Query device version string of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

Version string of the device, e.g. »1.0«.

9.3.9 :SYSTem:REVision? [<MPMeter>]

Query firmware revision of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

An unsigned integer value indicating the firmware's revision number is returned.

9.3.10 :SYSTem:DEBUG <Value>

Set value of debug flags in debug register. Doing so makes the TCP output debug information to standard error.

PARAMETER:

Unsigned integer value containing flags for debugging purposes. Setting a bit to 1 enables debug output, setting a bit to 0 disables debug output. The bit positions of the debug flags are defined as follows:

BIT 0

Enable memory consumption information.

BIT 1

Enable timing information.

BIT 2

Enable echoing of all incoming TCP server messages.

BIT 3

Enable echoing of all outgoing TCP server messages.

BIT 4

Enable timing and data throughput information for USB communications.

BIT 5

Enable information about USB burst and FIFO function calls.

BIT 6

Currently not in use.

BIT 7

Enable calibration data and interpolation information.

BIT 8

Enable trigger information.

BIT 9

Enable timing information for parsing of SCPI commands.

BIT 10
: Enable streaming information.

BIT 11
: Enable statistics information.

BIT 12
: Enable time output for correction data read in.

BIT 13
: Enable external calibration log file information.

BIT 14
: Enable FPGA firmware loading information.

BIT 15
: Enable Device polling information.

E.g. enable SCPI command input and output debugging by issuing »:syst:debug 12«.

9.3.11 :SYSTem:DEBUG?

Query value of debug flags in debug register.

RETURN VALUE:
: Unsigned integer value containing the debug flags. See :SYSTem:DEBUG <Value> for the description of the individual debug flags.

9.3.12 :SYSTem:MODe <Mode>[,<MPMeter>]

Set Power Meter operating mode of one or multiple Power Meters.

PARAMETERS:
: The unsigned integer parameter Mode specifies the Power Meter operating mode as described in Table 1 on page 15. Valid values are 0, 1, 2 and 3.

  The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.3.13 :SYSTem:MODe? [<MPMeter>]

Get Power Meter operating mode of one or multiple Power Meters.

PARAMETER:
: The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:
: An unsigned integer mode value as described in Table 1 on page 15 will be returned.

9.3.14 :SYSTem:FREQuency <Frequency>[,<MPMeter>]

Set frequency for frequency-compensated operation of one or multiple Power Meters.

PARAMETERS:

The first floating point-valued parameter sets the desired frequency. If the frequency exceeds the calibrated frequency range for the mode set via :SYSTem:MODe <Mode>[,<MPMeter>] the frequency will be forced to the nearest calibrated frequency.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. E.g. »:syst:freq 1e9,0« will set the compensation frequency to 1 GHz for all enumerated Power Meters, »:syst:freq 2e9« will set the compensation frequency to 2 GHz for the currently selected Power Meter.

9.3.15 :SYSTem:FREQuency? [<MPMeter>]

Query frequency for frequency-compensated operation of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

A floating point value rounded to three decimal places indicating the set compensation frequency will be returned.

9.3.16 :SYSTem:FREQuency:MINimum? [<MPMeter>]

Query minimum calibrated frequency for frequency-compensated operation of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

A floating point value indicating the minimum calibrated compensation frequency of the current mode will be returned.

9.3.17 :SYSTem:FREQuency:MAXimum? [<MPMeter>]

Query maximum calibrated frequency for frequency-compensated operation of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

A floating point value indicating the maximum calibrated compensation frequency of the current mode will be returned.

9.4 :CALibration Commands

9.4.1 :CALibration:LOGging <Value>

Enable or disable logging of external calibration information for current TCP/IP connection.

PARAMETER:

Enable logging of LSPM information for the currently active socket client by setting Value to 1, disable logging by setting Value to 0.

If Value is set to 1 the SCPI queries

- :MEASure[:Power]:P[1]? [<MPMeter>]
- :MEASure[:Power]:P2? [<MPMeter>]
- :MEASure[:Power]:P3? [<MPMeter>]
- :MEASure[:Power]:ALL? [<MPMeter>]

issued over the same TCP/IP connection append one line to the log file in the `LSPM_SAVE_PATH` directory for each Power Meter. See Section 10.2 for details about the log file format.

9.4.2 :CALibration:LOGging?

Query status of logging of external calibration data for current TCP/IP connection.

RETURN VALUE:

An unsigned integer value indicating if the logging status for the current TCP/IP connection is set will be returned. If external calibration data is being logged the command returns 1, otherwise 0 will be returned.

9.4.3 :CALibration:EXTernal <Value>[,<MPMeter>]

Enable or disable application of external calibration data of one or multiple Power Meters. After start-up the application of external calibration data is enabled.

PARAMETERS:

Enable application of external calibration data by setting Value to 1, disable application by setting Value to 0.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.4.4 :CALibration:EXTernal? [<MPMeter>]

Query application of external calibration data of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

An unsigned integer value indicating if external calibration data is being applied or not will be returned. If external calibration data is being applied the command returns 1, otherwise zero will be returned. NAN will be returned if no Power Meter is connected.

9.4.5 :CALibration:CERTificate? [<MPMeter>]

Query external calibration certificate string of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

An external calibration certificate string as given in the external calibration data CSV file will be returned. If the external calibration data CSV file does not contain a certificate string, no Power Meter is connected or there is no valid external calibration data "undefined" will be returned.

9.4.6 :CALibration:TStamp? [<MPMeter>]

Query in-house and external calibration time stamps of one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

A pair of unsigned integer values indicating the time stamp of in-house calibration and external calibration will be returned. Time stamps are expressed as the number of seconds since Jan 1 1904 00:00:00. NAN will be returned if no Power Meter is connected or there is no valid in-house or external calibration data. NAN will also be returned for the external calibration time stamp if external calibration data has been disabled using :CALibration:EXTernal <Value>[,<MPMeter>].

9.5 :MEASure Commands

9.5.1 :MEASure[:Power]:TCold? [<MPMeter>]

Query the common cold plate temperature of the power sensors for one or multiple Power Meters. Temperature is controlled at 20 °C by a Peltier cooler for temperature-independent sensor opera-

tion.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the float-valued cold plate temperature in °C.

9.5.2 :MEASure[:Power]:VPeltier? [<MPMeter>]

Query Peltier cooler voltage for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the float-valued Peltier cooler voltage in Volts.

9.5.3 :MEASure[:Power]:IPeltier? [<MPMeter>]

Query Peltier cooler current for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns the float-valued Peltier cooler current in Amperes.

9.5.4 :MEASure[:Power]:P[1]? [<MPMeter>]

Query channel 1 power value for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns the float-valued channel 1 power value in dBm. Values are low-pass filtered as described in :MEASure[:Power]:LPFrequency <Frequency>[,<MPMeter>]. -100 will be returned if the power sensor of the respective channel is not present. NAN will be returned if there is no valid calibration data.

9.5.5 :MEASure[:Power]:MINP[1]? [<MPMeter>]

Query minimum value of channel 1 power range for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the float-valued minimum calibrated channel 1 power value in dBm. The value is determined using the calibration data for a given frequency and mode. NAN will be returned if there is no valid calibration data. -100 will be returned if the power sensor of the respective channel is not present.

9.5.6 :MEASure[:Power]:MAXP[1]? [<MPMeter>]

Query maximum value of channel 1 power range for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the float-valued maximum calibrated channel 1 power value in dBm. The value is determined using the calibration data for a given frequency and mode. -100 will be returned if the power sensor of the respective channel is not present. NAN will be returned if there is no valid calibration data.

9.5.7 :MEASure[:Power]:P2? [<MPMeter>]

Query channel 2 power value for one or multiple Power Meters, see :MEASure[:Power]:P[1]? [<MPMeter>] for a description of parameters and return values.

9.5.8 :MEASure[:Power]:MINP2? [<MPMeter>]

Query minimum value of channel 2 power range for one or multiple Power Meters, see :MEASure[:Power]:MINP[1]? [<MPMeter>] for a description of parameters and return values.

9.5.9 :MEASure[:Power]:MAXP2? [<MPMeter>]

Query maximum value of channel 2 power range for one or multiple Power Meters, see :MEASure[:Power]:MAXP[1]? [<MPMeter>] for a description of parameters and return values.

9.5.10 :MEASure[:Power]:P3? [<MPMeter>]

Query channel 3 power value for one or multiple Power Meters, see :MEASure[:Power]:P[1]? [<MPMeter>] for a description of parameters and return values.

9.5.11 :MEASure[:Power]:MINP3? [<MPMeter>]

Query minimum value of channel 3 power range for one or multiple Power Meters, see :MEASure[:Power]:MINP[1]? [<MPMeter>] for a description of parameters and return values.

9.5.12 :MEASure[:Power]:MAXP3? [<MPMeter>]

Query maximum value of channel 3 power range for one or multiple Power Meters, see :MEASure[:Power]:MAXP[1]? [<MPMeter>] for a description of parameters and return values.

9.5.13 :MEASure[:Power]:ALL? [<MPMeter>]

Query channel 1, channel 2 and channel 3 power values for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a list of float-valued channel 1, channel 2 and channel 3 power values in dBm, in this order. -100 will be returned if the power sensor of the respective channel is not present. NAN will be returned if there is no valid calibration data.

9.5.14 :MEASure[:Power]:MINALL? [<MPMeter>]

Query minimum value of channel 1, channel 2 and channel 3 calibrated power range for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a list of float-valued channel 1, channel 2 and channel 3 minimum calibrated power values in dBm, in this order. -100 will be returned if the power sensor of the respective channel is not present. NAN will be returned if there is no valid calibration data.

9.5.15 :MEASure[:Power]:MAXALL? [<MPMeter>]

Query maximum value of channel 1, channel 2 and channel 3 calibrated power range for one or multiple Power Meters, see :MEASure[:Power]:MINALL? [<MPMeter>] for details.

9.5.16 :MEASure[:Power]:LPFrequency <Frequency>[,<MPMeter>]

Set power low-pass filter -3 dB cut-off frequency for one or multiple Power Meters.

PARAMETER:

Float value specifying the -3 dB cut-off frequency for the first order power low-pass filter in Hertz. The filter is applied to calibrated and uncalibrated power values. Setting the value to 0 Hz disables low-pass filtering.

The second optional unsigned integer parameter MPMeter is described in Section 9.1.

9.5.17 :MEASure[:Power]:LPFrequency? [<MPMeter>]

Query power low-pass filter -3 dB cut-off frequency for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a float value specifying the -3 dB cut-off frequency for the first order power low-pass filter in Hertz. A value to 0 indicates that low-pass filtering is disabled.

9.5.18 :MEASure:RSsi:P[1]? [<MPMeter>]

Query RSSI value for channel 1 for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned integer value representing the uncalibrated 14 bit ADC value acquired by the channel 1 RSSI chip, i.e. received signal strength indicator, used to detect the power. The value is low-pass filtered using the same low-pass filter as described in :MEASure[:Power]:LPFrequency <Frequency>[,<MPMeter>]. 0 will be returned if the power sensor of the respective channel is not present.

9.5.19 :MEASure:RSsi:P2? [<MPMeter>]

Query RSSI value for channel 2 for one or multiple Power Meters, see :MEASure:RSsi:P[1]? [<MPMeter>] for a description of parameters and return values.

9.5.20 :MEASure:RSsi:P3? [<MPMeter>]

Query RSSI value for channel 3 for one or multiple Power Meters, see :MEASure:RSsi:P[1]? [<MPMeter>] for a description of parameters and return values.

9.5.21 :MEASure:RSsi:All? [<MPMeter>]

Query RSSI value for channel 1,2 and 3 for one or multiple Power Meters, see :MEASure:RSsi:P[1]? [<MPMeter>] for a description of parameters and return values.

9.6 :TRIGger Commands

9.6.1 :TRIGger:COUnt? [<MPMeter>]

Query number of detected trigger events for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

An integer-valued number of trigger events recorded is returned.

9.6.2 :TRIGger:BEGin <Index>[,<MPMeter>]

Set index of first sample of power waveform for one or multiple Power Meters.

PARAMETERS:

The first parameter is the integer-valued position of the beginning of the power waveform relative to the position of the trigger. E.g. »:trig:beg 0« will record samples starting at the trigger position, »:trig:beg -100« will record samples starting 100 samples before the trigger position, »:trig:beg 100« will record samples starting 100 samples after the trigger position.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.3 :TRIGger:BEGin? [<MPMeter>]

Query index of first sample of power waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an integer-valued position of the beginning of the power waveform relative to the position of the trigger, corresponds to the first parameter of :TRIGger:BEGin <Index>[,<MPMeter>]. If executed for multiple Power Meters the command returns a list of integer-valued positions of the beginning of the power waveform relative to the position of the trigger for each Power Meter of the respective list.

9.6.4 :TRIGger:LENgth <Length>[,<MPMeter>]

Set length of the power waveform for one or multiple Power Meters.

PARAMETERS:

The unsigned integer-valued parameter of the command specifies the length of a power waveform. E.g. »:trig:len 100« will record 100 samples starting at the sample index specified by :TRIGger:BEGin <Index>[,<MPMeter>].

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.5 :TRIGger:LENgth? [<MPMeter>]

Query number of samples in power waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned integer-valued length of the power waveform, corresponding to the length set by :TRIGger:LENgth <Length>[,<MPMeter>]. If executed for multiple Power Meters the command returns a list of unsigned integer-valued lengths of the power waveform lengths for each Power Meter of the respective list.

9.6.6 :TRIGger:PROgress? [<MPMeter>]

Query trigger progress of current power waveform acquisition for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the unsigned integer-valued number of samples that have been saved to the power waveform buffer. If executed for multiple Power Meters the command returns a list of lengths for each Power Meter of the respective list.

9.6.7 :TRIGger:STATe? [<Timeout>,<MPMeter>]

Query the state of the trigger system for one or multiple Power Meters.

PARAMETER:

The first optional float valued parameter Timeout sets the maximum number of seconds to wait until encountering a DONE state. If the trigger state is equal to DONE, if Timeout is set to zero, or if Timeout is omitted the command will return immediately.

The second optional unsigned integer parameter MPMeter is described in Section 9.1. The second parameter always requires the first parameter to be set.

RETURN VALUE:

The command returns a string value giving the state of the trigger system, valid return values are IDLE, ARMED, TRIGGERED, DONE. See Figure 22 on page 31 for reference.

9.6.8 :TRIGger:DONE? [<Timeout>,<MPMeter>]

Query the done state of the trigger system for one or multiple Power Meters.

PARAMETER:

The first optional float valued parameter Timeout sets the maximum number of seconds to wait until encountering a DONE state. If the trigger state is equal to DONE, if Timeout is set to zero, or if Timeout is omitted the command will return immediately.

The second optional unsigned integer parameter MPMeter is described in Section 9.1. The second parameter always requires the first parameter to be set.

RETURN VALUE:

The command returns an unsigned integer value indicating if the trigger state is done. If the trigger state is IDLE, ARMED or TRIGGERED the return value is 0, if it is DONE the return value will be 1.

9.6.9 :TRIGger[:WAVeform][:Power]:P[1]? [<MPMeter>]

Query channel 1 power values of power waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma-separated list of float-valued channel 1 power values of the power waveform in dBm. -100 will be returned if the power sensor of the respective channel is not present. NAN will be returned if there is no valid calibration data or if the trigger system state is not equal to DONE.

9.6.10 :TRIGger[:WAVeform][:Power]:P2? [<MPMeter>]

Query channel 2 power values of power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform][:Power]:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.11 :TRIGger[:WAVeform][:Power]:P3? [<MPMeter>]

Query channel 3 power values of power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform][:Power]:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.12 :TRIGger[:WAVeform][:Power]:ALL? [<MPMeter>]

Query channel1, channel 2 and channel 3 power values averaged over the present power waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma-separated list of three float values giving the averaged channel 1 power, channel 2 power and channel 3 power in this order. -100 will be returned if the power sensor of the respective channel is not present. NAN will be returned if there is no valid calibration data or if the trigger system state is not equal to DONE.

9.6.13 :TRIGger[:WAVeform]:RSsi:P[1]? [<MPMeter>]

Query channel 1 RSSI values of a waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a list of unsigned integer values representing the uncalibrated 14 bit ADC value acquired by the channel 1 RSSI chip, i.e. received signal strength indicator, used to detect the power level. 0 will be returned if the power sensor of the respective channel is not present. NAN will be returned if the trigger system state is not equal to DONE.

9.6.14 :TRIGger[:WAVeform]:RSsi:P2? [<MPMeter>]

Query channel 2 RSSI values of a waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:RSsi:P[1]? [<MPMeter>] for details about the parameter and return values.

9.6.15 :TRIGger[:WAVeform]:RSsi:P3? [<MPMeter>]

Query channel 3 RSSI values of a waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:RSsi:P[1]? [<MPMeter>] for details about the parameter and return values.

9.6.16 :TRIGger[:WAVeform][:Power]:BINary? [<MPMeter>]

Query channel 1, channel 2 and channel 3 power values of power waveform in binary format for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

Binary data block followed by a carriage return, new-line sequence. The first four bytes specify the number of bytes of the binary data block which will be returned following the first four bytes.

For each Power Meter P a chunk of binary data will be sent, one for each referenced Power Meter. All values are encoded in little endian format. Data is ordered as follows:

POWER METER NUMBER

32 bit unsigned integer value giving the serial number of the corresponding Power Meter. If the Power Meter P is not defined the Power Meter serial number and sample count is set to zero and the binary data block ends.

SAMPLE COUNT

32 bit unsigned integer value giving the number of samples S in the waveform of the corresponding Power Meter. The following binary data will contain S values for each of the following values. If there is no valid calibration data or if the trigger system state is not equal to DONE the number of samples is set to zero and the binary data block ends.

CHANNEL 1 POWER

S 32 bit single precision floating point values giving a list of channel 1 power values in dBm of the power waveform, see :TRIGger[:WAVeform][:Power]:P[1]? [<MPMeter>].

CHANNEL 2 POWER

S 32 bit single precision floating point values giving a list of channel 2 power values in dBm of the power waveform, see :TRIGger[:WAVeform][:Power]:P2? [<MPMeter>].

CHANNEL 3 POWER

S 32 bit single precision floating point values giving a list of channel 3 power values in dBm of the power waveform, see :TRIGger[:WAVeform][:Power]:P3? [<MPMeter>].

CHANNEL 1 RSSI

S 16 bit unsigned integer values, encoded as 32 bit single precision floating point values, giving a list of channel 1 RSSI values in LSB of the waveform, see :TRIGger[:WAVeform]:RSsi:P[1]? [<MPMeter>].

CHANNEL 2 RSSI

S 16 bit unsigned integer values, encoded as 32 bit single precision floating point values, giving a list of channel 2 RSSI values in LSB of the waveform, see :TRIGger[:WAVeform]:RSsi:P2? [<MPMeter>].

CHANNEL 3 RSSI

S 16 bit unsigned integer values, encoded as 32 bit single precision floating point values, giving a list of channel 3 RSSI values in LSB of the waveform, see :TRIGger[:WAVeform]:RSsi:P3? [<MPMeter>].

9.6.17 :TRIGger[:WAVeform][:Power]:BINWait? [<Timeout>,<MPMeter>]

Query power component values of power waveform in binary format for one or multiple Power Meters. If data is available for all queried Power Meters a :TRIGger:CLear [<MPMeter>] and :TRIGger:ARM [<MPMeter>] will be sent automatically.

PARAMETER:

The first optional float valued parameter Timeout sets the maximum number of seconds to wait until encountering a DONE state. If the trigger state is equal to DONE, if Timeout is set to zero, or if Timeout is omitted the command will return immediately.

The second optional unsigned integer parameter MPMeter is described in Section 9.1. The second parameter always requires the first parameter to be set.

RETURN VALUES:

See :TRIGger[:WAVeform][:Power]:BINary? [<MPMeter>] for a description of the command's return values.

9.6.18 :TRIGger:SOURce <Source>[,<MPMeter>]

Set trigger source for triggered operation for one or multiple Power Meters.

PARAMETER:

String parameter without quotes specifying the trigger source. Valid values are:

SOFT

Software trigger, triggering must occur by means of the :TRIGger:FORce [<MPMeter>] command.

EXT

External trigger using BNC socket, polarity configured using :TRIGger:INVert <0/1>[,<MPMeter>].

EXT2

External trigger using RJ45 socket, polarity configured using :TRIGger:BPINVert <0/1>[,<MPMeter>].

P1/P2/P3

Power level trigger using channel 1/2/3 power value, :TRIGger:LEVel <Level>[,<MPMeter>] and :TRIGger:FALLing <0/1>[,<MPMeter>] are used for configuration.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.19 :TRIGger:SOURce? [<MPMeter>]

Query trigger source for triggered operation for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a string value without quotes specifying the trigger source, see :TRIGger:SOURce <Source>[,<MPMeter>] for more details.

9.6.20 :TRIGger:ARM [<MPMeter>]

Arm trigger, the trigger system will change state from any other trigger state to ARMED if not in TRIGGERED state for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.21 :TRIGger:CLear [<MPMeter>]

Clear trigger, the trigger system will change state from any other trigger state to IDLE for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.22 :TRIGger:FORce [<MPMeter>]

Force trigger, the trigger system will change state to TRIGGERED if not already in state TRIGGERED, this is independent of the trigger source set by :TRIGger:SOURce <Source>[,<MPMeter>] for one or multiple Power Meters. This command is used for software triggering.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.23 :TRIGger:LEVel <Level>[,<MPMeter>]

Set the trigger power level for P1, P2 and P3 triggering for one or multiple Power Meters.

PARAMETER:

Floating point parameter specifying the power level in dBm. Triggering occurs if the power value crosses the set power level.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.24 :TRIGger:LEVel? [<MPMeter>]

Query the trigger power level for P1, P2 and P3 triggering for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns the trigger level as a float-valued power value in dBm. NAN will be returned if the Power Meter is off or in start-up.

9.6.25 :TRIGger:FALLing <0/1>[,<MPMeter>]

Set the direction for external, P1, P2 and P3 triggering for one or multiple Power Meters.

PARAMETERS:

Boolean value of either 0 or 1. If set to 0 the the rising edge of the external trigger signal or passing the threshold value in rising direction will bring the trigger system from the state ARMED to TRIGGERED. If set to 1 the falling edge will be used for triggering.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.26 :TRIGger:FALLing? [<MPMeter>]

Query the direction for external, P1, P2 and P3 triggering for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the boolean value giving the trigger edge direction, see :TRIGger:FALLing <0/1>[,<MPMeter>] for details.

9.6.27 :TRIGger:OUTput <0/1>[,<MPMeter>]

Enable or disable the output of a trigger signal via the Power Meter's BNC connector for one or multiple Power Meters.

PARAMETERS:

Boolean value of either 0 or 1. If set to 0 trigger output is disabled and the Power Meter's BNC connector can be used for trigger input. If set to 1 trigger output is enabled and the Power Meter's BNC connector cannot be used for trigger input.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.28 :TRIGger:OUTput? [<MPMeter>]

Query status of trigger output for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a boolean value giving the state of the Power Meters BNC trigger connector, see :TRIGger:OUTput <0/1>[,<MPMeter>] for details.

9.6.29 :TRIGger:INVert <0/1>[,<MPMeter>]

Set the polarity for trigger output via the Power Meter's BNC trigger connector for one or multiple Power Meters.

PARAMETERS:

Boolean value of either 0 or 1. If set to 0 trigger output uses a rising edge logic signal. If set to 1 a falling edge logic signal will be generated.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.30 :TRIGger:INVert? [<MPMeter>]

Query the polarity for trigger output via the Power Meter's BNC trigger connector for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a boolean value of either 0 or 1. See :TRIGger:INVert <0/1>[,<MPMeter>] for details.

9.6.31 :TRIGger:SYNC <0/1>[,<MPMeter>]

Enable or disable synchronization trigger output using the Power Meter's BNC trigger connector for one or multiple Power Meters. This function is useful for synchronizing signal generators or transmitters with the Power Meter.

PARAMETERS:

Boolean value of either 0 or 1. If set to 0 and external trigger output is enabled, output trigger signal as described in :TRIGger:OUTput <0/1>[,<MPMeter>] and :TRIGger:INVert

<0/1>[,<MPMeter>]. If set to 1 and trigger output is enabled, a logic edge will be generated synchronously with power value acquisition. If in mode 0 to 3 a 250 ns long pulse is generated once every 500 ns.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.32 :TRIGger:SYNC? [<MPMeter>]

Query the configuration synchronization trigger output for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a boolean value of either 0 or 1. See :TRIGger:SYNC <0/1>[,<MPMeter>] for details.

9.6.33 :TRIGger:BPOUTput <0/1>[,<MPMeter>]

Enable or disable the output of a trigger signal via the Power Meter's upper RJ45 connector for one or multiple Power Meters. See :TRIGger:OUTput <0/1>[,<MPMeter>] for parameters.

9.6.34 :TRIGger:BPOUTput? [<MPMeter>]

Query status of trigger output via the Power Meter's upper RJ45 connector for one or multiple Power Meters. See :TRIGger:OUTput? [<MPMeter>] for parameter and return value.

9.6.35 :TRIGger:BPINVert <0/1>[,<MPMeter>]

Set the polarity for trigger output via the Power Meter's upper RJ45 connector for one or multiple Power Meters. See :TRIGger:INVert <0/1>[,<MPMeter>] for parameters.

9.6.36 :TRIGger:BPINVert? [<MPMeter>]

Query the polarity for trigger output via the Power Meter's upper RJ45 connector for one or multiple Power Meters. See :TRIGger:INVert? [<MPMeter>] for parameter and return value.

9.6.37 :TRIGger:BPSYNC <0/1>[,<MPMeter>]

Enable or disable synchronization trigger output via the Power Meter's upper RJ45 connector for one or multiple Power Meters. See :TRIGger:SYNC <0/1>[,<MPMeter>] for parameters.

9.6.38 :TRIGger:BPSYNC? [<MPMeter>]

Query the configuration of the synchronization trigger output via the Power Meter's upper RJ45 connector for one or multiple Power Meters. See :TRIGger:SYNC? [<MPMeter>] for parameter and return value.

9.6.39 :TRIGger[:WAVeform]:RADar:THreshold <Value>[,<MPMeter>]

Set channel 1, channel 2 and channel 3 radar threshold value.

PARAMETERS:

Float value specifying the threshold for radar pulse detection in dBm. The value applies to the SCPI commands :TRIGger[:WAVeform]:RADar:P[1]? [<MPMeter>], :TRIGger[:WAVeform]:RADar:P2? [<MPMeter>], :TRIGger[:WAVeform]:RADar:P3? [<MPMeter>], :TRIGger[:WAVeform]:RADar:ALL? [<MPMeter>] and :TRIGger[:WAVeform]:RADar:COUnt? [<MPMeter>]. The default threshold value is 0 dBm. The value will only be used if automatic threshold calculation is disabled, see :TRIGger[:WAVeform]:RADar:THreshold:AUTO <0/1>[,<MPMeter>].

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.40 :TRIGger[:WAVeform]:RADar:THreshold? [<MPMeter>]

Query channel 1, channel 2 and channel 3 radar threshold value.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

See :TRIGger[:WAVeform]:RADar:THreshold:AUTO <0/1>[,<MPMeter>] for return value.

9.6.41 :TRIGger[:WAVeform]:RADar:THreshold:AUTO <0/1>[,<MPMeter>]

Enable/disable channel 1/2/3 radar pulse detection with automatic threshold value.

If set to one the arithmetic mean of minimum and maximum power level will be used as the pulse detection threshold. If set to zero the SCPI commands :TRIGger[:WAVeform]:RADar:P[1]? [<MPMeter>], :TRIGger[:WAVeform]:RADar:P2? [<MPMeter>], :TRIGger[:WAVeform]:RADar:P3? [<MPMeter>], :TRIGger[:WAVeform]:RADar:ALL? [<MPMeter>] and :TRIGger[:WAVeform]:RADar:COUnt? [<MPMeter>] will use the set threshold value parameter set by :TRIGger[:WAVeform]:RADar:THreshold <Value>[,<MPMeter>].

9.6.42 :TRIGger[:WAVeform]:RADar:THreshold:AUTO? [<MPMeter>]

Query channel 1/2/3 radar pulse detection with automatic threshold value.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

See :TRIGger[:WAVeform]:RADar:THreshold:AUTO <0/1>[,<MPMeter>] for return value.

9.6.43 :TRIGger[:WAVeform]:RADar:P[1]? [<MPMeter>]

Query channel 1 radar power values of a waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma-separated list of unsigned integer values giving the number of detected pulses, followed by index/power value pairs giving the sample position and power value of the radar pulse. The first value of each pair gives the sample index after crossing the threshold value in a rising fashion relative to the trigger position. The second value of each pair gives the power value of the respective pulse. For pulses containing one or two samples the pulse's power value is defined as maximum value, otherwise the pulse's power value is defined as the arithmetic mean of all but the first and last sample value of the pulse, see Figure 23. The number of pulses will be returned as zero and no index/power value pairs be returned if no pulses could be detected, there is no valid calibration data or the trigger system state is not equal to DONE.

9.6.44 :TRIGger[:WAVeform]:RADar:P2? [<MPMeter>]

Query channel 2 radar power values of a waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:RADar:P[1]? [<MPMeter>] for details.

9.6.45 :TRIGger[:WAVeform]:RADar:P3? [<MPMeter>]

Query channel 3 radar power values of a waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:RADar:P[1]? [<MPMeter>] for details.

9.6.46 :TRIGger[:WAVeform]:RADar:ALL? [<MPMeter>]

Query channel 1, channel 2 and channel 3 radar power values averaged over the present waveform, for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma-separated list of three float values giving the channel 1, channel 2 and channel 3 power levels averaged over all detected pulses, in this order. NAN will be returned if no pulses could be detected, there is no valid calibration data or the trigger system state is not equal to DONE.

9.6.47 :TRIGger[:WAVeform]:RADar:COUnt? [<MPMeter>]

Query number of detected channel 1, channel 2 and channel 3 power pulses in the present power waveform, for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma-separated list of unsigned integer values giving the number of detected pulses for the channel 1, channel 2 and channel 3 power levels. NAN will be returned if there is no valid calibration data or the trigger system state is not equal to DONE.

9.6.48 :TRIGger[:WAVeform]:SWeep:TStep <TStep>[,<MPMeter>]

Set number of samples per sweep step for one or multiple Power Meters.

PARAMETERS:

The unsigned, non-zero integer-valued parameter of the command specifies the number of samples per sweep step within the power waveform, dividing it into as many sections as will fit into the waveform, starting with the first sample of the waveform.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.49 :TRIGger[:WAVeform]:SWeep:TStep? [<MPMeter>]

Query number of samples per sweep step for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the unsigned integer-valued number of samples per sweep step, corresponding to the value set via :TRIGger[:WAVeform]:SWeep:TStep <TStep>[,<MPMeter>], the default value is 1000 samples. If executed for multiple Power Meters the command returns a list of values for each Power Meter of the respective list.

9.6.50 :TRIGger[:WAVeform]:SWeep:TBegin <TBegin>[,<MPMeter>]

Set the index of the first sample of the averaged portion of each sweep step for one or multiple Power Meters.

PARAMETERS:

The unsigned integer-valued parameter of the command sets the index of the first value used for averaging in each sweep step.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.51 :TRIGger[:WAVeform]:SWeep:TBegin? [<MPMeter>]

Query the index of the first sample of the averaged portion of each sweep step for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the unsigned integer-valued index of the first sample used for averaging in each sweep step, corresponding to the length set via :TRIGger[:WAVeform]:SWeep:TBegin <TBegin>[,<MPMeter>], the default value is 500. If executed for multiple Power Meters the command returns a list of indexes for each Power Meter of the respective list.

9.6.52 :TRIGger[:WAVeform]:SWeep:TEnd <TEnd>[,<MPMeter>]

Set the index of the last sample of the averaged portion of each sweep step for one or multiple Power Meters.

PARAMETERS:

The unsigned integer-valued parameter of the command sets the index of the last value used for averaging in each sweep step.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.53 :TRIGger[:WAVeform]:SWeep:TEnd? [<MPMeter>]

Query the index of the last sample of the averaged portion of each sweep step for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the unsigned integer-valued index of the last sample used for averaging in each sweep step, corresponding to the length set via :TRIGger[:WAVeform]:SWeep:TEnd <TEnd>[,<MPMeter>], the default value is 899. If executed for multiple Power Meters the command returns a list of indexes for each Power Meter of the respective list.

9.6.54 :TRIGger[:WAVeform]:SWeep:MODe <Mode>[,<MPMeter>]

Set frequency sweep mode for power waveform evaluation for one or multiple Power Meters.

PARAMETER:

String parameter without quotes specifying the sweep mode, valid values are FIXED, LIN, LOG and LIST. When set to LIN or LOG the sweep must be parametrized via the SCPI commands :TRIGger[:WAVeform]:SWeep:BEGin <Freq>[,<MPMeter>], :TRIGger[:WAVeform]:SWeep:STEP <Step>[,<MPMeter>] and :TRIGger[:WAVeform]:SWeep:COUnt <Count>[,<MPMeter>]. When set to LIST an arbitrary frequency list with a non-zero number of frequencies must be set via the SCPI command :TRIGger[:WAVeform]:SWeep:LIST:ADD <Freq>[,<MPMeter>]. When set to FIXED the frequency set via :SYSTem:FREQuency <Frequency>[,<MPMeter>] is used for all sweep steps.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.55 :TRIGger[:WAVeform]:SWeep:MODe? [<MPMeter>]

Query frequency sweep mode for power waveform evaluation for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a string value without quotes specifying the sweep mode, see :TRIGger[:WAVeform]:SWeep:MODe <Mode>[,<MPMeter>] for more details. If executed for multiple Power Meters the command returns a list of modes of all Power Meters of the respective list.

9.6.56 :TRIGger[:WAVeform]:SWeep:BEGin <Freq>[,<MPMeter>]

Set frequency of first sweep step for linear and logarithmic frequency sweeps for one or multiple Power Meters.

PARAMETERS:

The floating point-valued parameter of the command specifies the frequency of the first sweep step for linear and logarithmic sweeps in Hertz.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.57 :TRIGger[:WAVeform]:SWeep:BEGin? [<MPMeter>]

Query frequency of first sweep step for linear and logarithmic frequency sweeps for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the floating point-valued frequency of the first sweep step in Hertz for linear and logarithmic frequency sweeps, corresponding to the value set by :TRIGger[:WAVeform]:SWeep:BEGin <Freq>[,<MPMeter>]. The default value is 100 MHz. If executed for multiple Power Meters the command returns a list of floating point values indicating the frequency of the first sweep step for each Power Meters.

9.6.58 :TRIGger[:WAVeform]:SWeep:COUnt <Count>[,<MPMeter>]

Set number of frequency steps for linear and logarithmic frequency sweeps for one or multiple Power Meters.

PARAMETERS:

The unsigned integer-valued parameter of the command specifies the number of frequency steps of the frequency sweep.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.59 :TRIGger[:WAVeform]:SWeep:COUnt? [<MPMeter>]

Query number of frequency steps for linear and logarithmic frequency sweeps for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the unsigned integer-valued number of frequency steps for linear and logarithmic frequency sweeps set via :TRIGger[:WAVeform]:SWeep:COUnt <Count>[,<MPMeter>]. The default is 10 steps. If executed for multiple Power Meters the command returns a list of unsigned integer-valued number of frequency steps for each Power Meters.

9.6.60 :TRIGger[:WAVeform]:SWeep:STEP <Step>[,<MPMeter>]

Set the incremental frequency step for linear and logarithmic frequency sweeps for one or multiple Power Meters.

PARAMETERS:

The floating-point valued parameter of the command specifies the frequency increment. For linear frequency sweeps the parameter gives the frequency increment from one sweep step to the next in Hertz. For logarithmic frequency sweeps the parameter specifies the incremental factor from one sweep step to the next.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.61 :TRIGger[:WAVeform]:SWeep:STEP? [<MPMeter>]

Query the incremental frequency step for linear and logarithmic frequency sweeps for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the floating-point valued linear frequency increment or factor from one sweep step to the next for linear resp. logarithmic frequency sweeps set by :TRIGger[:WAVeform]:SWeep:STEP <Step>[,<MPMeter>]. The default value is 1.1. If executed for multiple Power Meters the command returns a list of floating point-valued numbers for each Power Meter.

9.6.62 :TRIGger[:WAVeform]:SWeep:LIST:ADD <Freq>[,<MPMeter>]

Append single frequency to the list of arbitrary sweep frequencies for one or multiple Power Meters.

PARAMETERS:

The floating-point valued parameter of the command specifies the frequency to be appended to the arbitrary frequency list in Hertz.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.63 :TRIGger[:WAVeform]:SWeep:LIST:CLear [<MPMeter>]

Clear the list of arbitrary sweep frequencies for one or multiple Power Meters.

PARAMETERS:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

9.6.64 :TRIGger[:WAVeform]:SWeep:LIST? [<MPMeter>]

Query the list of frequencies used for the sweep evaluation of a power waveform for one or multiple Power Meters.

PARAMETERS:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a float-valued list of frequency steps used for sweep evaluation. The number of frequency steps is the number of samples of the waveform divided by the number of samples per sweep step, rounded down to the nearest integer number, see :TRIGger:LENgth <Length>[,<MPMeter>] and :TRIGger[:WAVeform]:SWeep:TStep <TStep>[,<MPMeter>]. Frequency values are applied according to the set sweep mode and its parameters beginning with the start frequency for linear and logarithmic sweeps. For arbitrary list sweeps the list's values will be applied. A constant frequency is applied in fixed sweep mode. If the number of frequency steps that fit into the waveform exceeds the number of sweep steps the list of frequency steps will be applied repeatedly until a frequency value has been assigned to each sweep step. If the number of sweep steps is zero and/or the sweep step parameters TStep, TBegin and TEnd are invalid for the currently set trigger length or contradict each other the query will return NAN. If executed for multiple Power Meters the command returns a list of frequencies for each Power Meter.

9.6.65 :TRIGger[:WAVeform]:SWeep:IDX? [<MPMeter>]

Query the center indices of the averaged portions of each sweep step for one or multiple Power Meters.

PARAMETERS:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma separated, integer-valued list of index values of the power waveform. The indices give the arithmetic mean of the first and last index used for averaging of each sweep step, see :TRIGger[:WAVeform]:SWeep:TBegin <TBegin>[,<MPMeter>] and :TRIGger[:WAVeform]:SWeep:TEnd <TEnd>[,<MPMeter>]. The index is especially useful for overlaying power waveforms and averaged sweep values. If executed for multiple Power Meters the command returns a list of indices for each Power Meter.

9.6.66 :TRIGger[:WAVeform]:SWeep[:Power]:P[1]? [<MPMeter>]

Query averaged channel 1 power value for each sweep step of the power waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma-separated list of float-valued averaged channel 1 power values

for each sweep step of the power waveform in dBm. NAN will be returned if the trigger system's state is not equal to DONE, if there are no valid sweep frequencies or if there is no valid calibration data for this frequency step.

9.6.67 :TRIGger[:WAVeform]:SWeep[:Power]:P2? [<MPMeter>]

Query averaged channel 2 power value for each sweep step of the power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:SWeep[:Power]:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.68 :TRIGger[:WAVeform]:SWeep[:Power]:P3? [<MPMeter>]

Query averaged channel 3 power value for each sweep step of the power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:SWeep[:Power]:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.69 :TRIGger[:WAVeform]:SWeep[:Power]:ALL? [<MPMeter>]

Query averaged channel 1, channel 2 and channel 3 power values for each sweep step of the power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:SWeep[:Power]:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.70 :TRIGger[:WAVeform]:SWeep:RSsi:P[1]? [<MPMeter>]

Query averaged channel 1 RSSI value for each sweep step of the power waveform for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a comma-separated list of integer-valued averaged channel 1 RSSI values for each sweep step of the power waveform in LSB. NAN will be returned if the trigger system's state is not equal to DONE or if or if there are no valid sweep frequencies.

9.6.71 :TRIGger[:WAVeform]:SWeep:RSsi:P2? [<MPMeter>]

Query averaged channel 2 RSSI value for each sweep step of the power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:SWeep:RSsi:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.72 :TRIGger[:WAVeform]:SWeep:RSsi:P3? [<MPMeter>]

Query averaged channel 3 RSSI value for each sweep step of the power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:SWeep:RSsi:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.73 :TRIGger[:WAVeform]:SWeep:RSsi:ALL? [<MPMeter>]

Query averaged channel 1, channel 2 and channel 3 RSSI values for each sweep step of the power waveform for one or multiple Power Meters, see :TRIGger[:WAVeform]:SWeep:RSsi:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.74 :TRIGger[:WAVeform]:SWeep:WPower:P[1]? [<MPMeter>]

Query channel 1 power values of frequency corrected power waveform using the configured sweep frequencies for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

The command returns a list of float-valued channel 1 power values of the sweep frequency corrected power waveform in dBm. NAN will be returned if there is no valid calibration data, if the trigger system state is not equal to DONE or if there are no valid sweep frequencies.

9.6.75 :TRIGger[:WAVeform]:SWeep:WPower:P2? [<MPMeter>]

Query channel 2 power values of frequency corrected power waveform using the configured sweep frequencies for one or multiple Power Meters. See :TRIGger[:WAVeform]:SWeep:WPower:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.76 :TRIGger[:WAVeform]:SWeep:WPower:P3? [<MPMeter>]

Query channel 3 power values of frequency corrected power waveform using the configured sweep frequencies for one or multiple Power Meters. See :TRIGger[:WAVeform]:SWeep:WPower:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.77 :TRIGger[:WAVeform]:SWeep:WPower:ALL? [<MPMeter>]

Query channel 1, channel 2 and channel 3 power values of frequency corrected power waveform using the configured sweep frequencies for one or multiple Power Meters. See :TRIGger[:WAVeform]:SWeep:WPower:P[1]? [<MPMeter>] for details about the parameter and the return values.

9.6.78 :TRIGger[:WAVeform]:SWeep:BINary? Wave,[<MPMeter>]

Query averaged and unaveraged sweep corrected channel 1, channel 2 and channel 3 power values, and center indices of averaged sweep values in binary format for one or multiple Power Meters.

PARAMETERS:

The first mandatory integer-values parameter Wave controls the output of sweep corrected power waveform values. If set to 1 waveform output is enabled, if set to 0 waveform is disabled.

The second optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUES:

Binary data block followed by a carriage return, new-line sequence. The first four bytes specify the number of bytes of the binary data block which will be returned following the first four bytes.

For each Power Meter P a chunk of binary data will be sent, one for each referenced Power Meter. All values are encoded in little endian format. If the Power Meter P is not defined the Power Meter serial number, and generator step count are set to zero and the binary data block ends. Data are ordered as follows:

POWER METER NUMBER

32 bit unsigned integer value giving the serial number of the corresponding Power Meter.

GENERATOR STEP COUNT

32 bit unsigned integer value giving the number of sweep steps S1 in the power waveform of the corresponding Power Meter. If there is no valid calibration data, if the trigger system state is not equal to DONE or if there are no valid sweep frequencies S1 is set to zero and the binary data block ends.

SAMPLE COUNT

32 bit unsigned integer value giving the number of samples S2 in the sweep corrected power waveform of the corresponding Power Meter.

INDEX

S1 32 bit integer values giving a list of center index values for the averaged portion of each sweep step of the power waveform as described in :TRIGger[:WAVeform]:SWeep:IDX? [<MPMeter>].

FREQUENCIES

S1 32 bit single precision floating point values giving a list of the sweep frequency values for the averaged portion of each sweep step of the power waveform as described in :TRIGger[:WAVeform]:SWeep:LIST? [<MPMeter>].

CHANNEL 1 SWEEP

S1 32 bit single precision floating point values giving a list of averaged channel 1 power values in dBm of each sweep step within the power waveform, see :TRIGger[:WAVeform]:SWeep[:Power]:P[1]? [<MPMeter>].

CHANNEL 2 SWEEP

S1 32 bit single precision floating point values giving a list of averaged channel 2 power values in dBm of each sweep step within the power waveform, see :TRIGger[:WAVeform]:SWeep[:Power]:P2? [<MPMeter>].

CHANNEL 3 SWEEP

S1 32 bit single precision floating point values giving a list of averaged channel 3 power values in dBm of each sweep step within the power waveform, see :TRIGger[:WAVeform]:SWeep[:Power]:P3? [<MPMeter>].

CHANNEL 1 RSSI

S1 32 bit single precision floating point values, giving a list of averaged channel 1 RSSI values in LSB of each sweep step of the power waveform, see :TRIGger[:WAVeform]:RSsi:P[1]? [<MPMeter>].

CHANNEL 2 RSSI

S1 32 bit single precision floating point values, giving a list of averaged channel 2 RSSI values in LSB of each sweep step of the power waveform, see :TRIGger[:WAVeform]:RSsi:P2? [<MPMeter>].

CHANNEL 3 RSSI

S1 32 bit single precision floating point values, giving a list of averaged channel 3 RSSI values in LSB of each sweep step of the power waveform, see :TRIGger[:WAVeform]:RSsi:P3? [<MPMeter>].

CHANNEL 1

S2 32 bit single precision floating point values giving a list of sweep frequency corrected channel 1 power values in dBm of the power waveform, see :TRIGger[:WAVeform]:SWeep:WPower:P[1]? [<MPMeter>].

CHANNEL 2

S2 32 bit single precision floating point values giving a list of sweep frequency corrected channel 2 power values in dBm of the power waveform, see :TRIGger[:WAVeform]:SWeep:WPower:P2? [<MPMeter>].

CHANNEL 3

S2 32 bit single precision floating point values giving a list of sweep frequency corrected channel 3 power values in dBm of the power waveform, see :TRIGger[:WAVeform]:SWeep:WPower:P3? [<MPMeter>].

9.7 :STATistics Commands

9.7.1 :STATistics:MAster <State>

Set currently selected Power Meter to be the master or a slave Power Meter for continuous statistics collection. The active Power Meter is set using :SYSTem:SERial <Value>. By default the first enumerated Power Meter automatically becomes the continuous statistics master Power Meter.

PARAMETER:

Setting State to 1 makes the current Power Meter the master of the continuous statistics subsystem. A State of 0 makes the Power Meter a slave of the continuous statistics subsystem, i.e. continuous statistics will be controlled by a different Power Meter.

9.7.2 :STATistics:MAster? [<MPMeter>]

Query statistics subsystem master/slave status of the currently active Power Meter. By default the first enumerated Power Meter automatically becomes the continuous statistics master Power Meter.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned integer value containing the master/slave status of the currently active Power Meter. Slaves return 0, the master returns 1.

9.7.3 :STATistics:ENable <State>[,<MPMeter>]

Enable or Disable statistics acquisition for statistics subsystem master Power Meter. This command is only effective for Power Meters configured as the statistics subsystem master Power Meter, see :STATistics:MAster <State>. Enabling statistics acquisition resets the snapshot counter queried via :STATistics:COUnt? [<MPMeter>].

PARAMETERS:

Setting State to 1 activates statistics acquisition, setting State to 0 disables statistics acquisition for one or multiple Power Meters. Changing the state from disabled to enabled will reset and start statistics collection. Changing the state from enabled to disabled will trigger an automatic snapshot identical to issuing :STATistics:SNAPshot [<Triggered>][,<MPMeter>] and stop statistics collection, see also :STATistics:SNAPshot [<Triggered>][,<MPMeter>].

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.7.4 :STATistics:ENable? [<MPMeter>]

Query status of statistics acquisition for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned integer value containing the status of statistics acquisition. A value of 1 is returned when statistics acquisition is enabled, 0 is returned if statistics acquisition

is disabled. The enable state of the statistics subsystem is controlled by the statistics subsystem master, see :STATistics:MAster <State> and :STATistics:ENable <State>[,<MPMeter>]. All statistics subsystem slave Power Meters are controlled by the statistics master Power Meter, their return value is thus always be identical to the master Power Meter if connected correctly.

9.7.5 :STATistics:SNAPshot [<Triggered>][,<MPMeter>]

Create a snapshot of either continuously collected statistics, or of waveforms recorded by the trigger subsystem for one or multiple Power Meters.

PARAMETERS:

The optional integer-valued parameter Triggered selects the source for the statistics snapshot. If the parameter is omitted or set to 0 a snapshot of the continuously acquired statistics is created for subsequent analysis. This type of statistics snapshot is triggered by the Power Meter configured as the statistics subsystem master, see :STATistics:MAster <State>. Additionally, statistics acquisition must be enabled using :STATistics:ENable <State>[,<MPMeter>] before creating a snapshot.

If the parameter Triggered is set to 1 the most recently acquired triggered waveforms are analyzed to obtain a statistics snapshot for subsequent analysis. This kind of snapshot can only be created for one Power Meter or multiple Power Meters at a time, see the description of the second parameter below.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. If the parameter MPMeter is set the parameter Triggered is mandatory. For continuous statistics only Power Meters configured as the statistics master Power Meter will output a snapshot trigger signal.

9.7.6 :STATistics:COUnt? [<MPMeter>]

Return continuous statistics snapshot counter for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned integer value giving the number of snapshots taken for the selected Power Meter since the last enabling of statistics acquisition.

9.7.7 :STATistics:RESolution <Resolution>[,<MPMeter>]

Set resolution for histograms and distribution functions for one or multiple Power Meters.

PARAMETERS:

The float-valued parameter Resolution specifies the power value resolution in dB for all statistics query commands returning histograms and distribution functions. E.g. a value of 1.0 will output histograms with a bin size of 1 dB. Bins are aligned relative to and centered around 0 dBm. The smallest permissible value for Resolution is 0.005 dB, i.e. 1/200 dB.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.7.8 :STATistics:RESolution? [<MPMeter>]

Query resolution in dB for histograms and distribution functions for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a float value giving the power resolution in decibels for all statistics query commands returning histograms and distribution functions.

9.7.9 :STATistics:HISTogram:SIZe? [<Triggered>][,<MPMeter>]

Query number of bins for histograms and distribution functions for one or multiple Power Meters.

PARAMETERS:

If no parameter is provided or if Triggered is set to 0 the number of bins for the most recently created snapshot histograms and distribution functions based on continuous statistics acquisition is returned. If the parameter Triggered is set to 1 the number of bins for the most recently created snapshot histograms and distribution functions based on triggered waveforms is returned.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. If the parameter MPMeter is set the parameter Triggered is mandatory.

RETURN VALUE:

The command returns an unsigned integer value giving the number of histogram bins for the set resolution, see :STATistics:RESolution <Resolution>[,<MPMeter>]. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.10 :STATistics:HISTogram:OFFset? [<Triggered>][,<MPMeter>]

Query offset in dB relative to 0 dBm for all histograms and distribution functions for one or multiple Power Meters.

PARAMETERS:

If no parameter is provided or if Triggered is set to 0 the offset for the most recently created snapshot histograms and distribution functions based on continuous statistics acquisition is returned. If the parameter Triggered is set to 1 the offset for the most recently created snapshot histograms and distribution functions based on triggered waveforms is returned.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. If the parameter MPMeter is set the parameter Triggered is mandatory.

RETURN VALUE:

The command returns an unsigned integer value giving the offset for the set resolution, see :STATistics:RESolution <Resolution>[,<MPMeter>]. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.11 :STATistics:SAMples? [<Triggered>][,<MPMeter>]

Query number of sample values used for statistics acquisition for one or multiple Power Meters.

PARAMETERS:

If no parameter is provided or if Triggered is set to 0 the number of samples per axis used for the most recently created statistics snapshot based on continuous statistics acquisition is returned. If the parameter Triggered is set to 1 the number of samples per axis used for the most recently created statistics snapshot based on triggered waveforms is returned, i.e. the number of samples returned by :TRIGger:LENgth <Length>[,<MPMeter>].

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. If the parameter MPMeter is set the parameter Triggered is mandatory.

RETURN VALUE:

The command returns an unsigned 64 bit integer value giving the number of samples used to build the respective histogram. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.12 :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 minimum power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters.

PARAMETERS:

If no parameter is provided or if Triggered is set to 0 the channel 1 minimum power for the most recently created statistics snapshot based on continuous statistics acquisition is returned. If the parameter Triggered is set to 1 the channel 1 minimum power of the most recently created statistics snapshot based on triggered waveforms is returned.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. If the parameter MPMeter is set the parameter Triggered is mandatory.

RETURN VALUE:

The command returns a float-valued channel 1 minimum power in dBm. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.13 :STATistics:MINimum:P2? [<Triggered>][,<MPMeter>]

Query channel 2 minimum power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.14 :STATistics:MINimum:P3? [<Triggered>][,<MPMeter>]

Query channel 3 minimum power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.15 :STATistics:MINimum:ALL? [<Triggered>][,<MPMeter>]

Query channel 1, channel 2 and channel 3 minimum power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters.

PARAMETERS:

See :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] parameter description for details.

RETURN VALUES:

The command returns a comma-separated list of three float values in dBm giving the channel 1, channel 2 and channel 3 minimum power value for the most recently created statistics snapshot. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.16 :STATistics:MAXimum:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 maximum power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.17 :STATistics:MAXimum:P2? [<Triggered>][,<MPMeter>]

Query channel 2 maximum power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.18 :STATistics:MAXimum:P3? [<Triggered>][,<MPMeter>]

Query channel 3 maximum power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.19 :STATistics:MAXimum:ALL? [<Triggered>][,<MPMeter>]

Query channel 1, channel 2 and channel 3 maximum powers of the most recent statistics snapshot or triggered waveform, see :STATistics:MINimum:ALL? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.20 :STATistics:MEAN:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 arithmetic mean power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.21 :STATistics:MEAN:P2? [<Triggered>][,<MPMeter>]

Query channel 2 arithmetic mean power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.22 :STATistics:MEAN:P3? [<Triggered>][,<MPMeter>]

Query channel 3 arithmetic mean power of the most recent statistics snapshot for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.23 :STATistics:MEAN:ALL? [<Triggered>][,<MPMeter>]

Query channel 1, channel 2 and channel 3 arithmetic mean powers of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:ALL? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.24 :STATistics:RMS:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 root mean square power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.25 :STATistics:RMS:P2? [<Triggered>][,<MPMeter>]

Query channel 2 root mean square power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.26 :STATistics:RMS:P3? [<Triggered>][,<MPMeter>]

Query channel 3 root mean square power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.27 :STATistics:RMS:ALL? [<Triggered>][,<MPMeter>]

Query channel 1, channel 2 and channel 3 root mean square powers of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:ALL? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.28 :STATistics:SDEViation:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 standard deviation power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.29 :STATistics:SDEViation:P2? [<Triggered>][,<MPMeter>]

Query channel 2 standard deviation power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.30 :STATistics:SDEViation:P3? [<Triggered>][,<MPMeter>]

Query channel 3 standard deviation power of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.31 :STATistics:SDEViation:ALL? [<Triggered>][,<MPMeter>]

Query channel 1, channel 2 and channel 3 standard deviation powers of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:MINimum:ALL? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.32 :STATistics:Power? [<Triggered>][,<MPMeter>]

Query center power value of bins used by histograms and distribution functions for one or multiple Power Meters.

PARAMETERS:

If no parameter is provided or if Triggered is set to 0 the center power values of all bins for the most recently created statistics snapshot based on continuous statistics acquisition are returned. If the parameter Triggered is set to 1 the center power values of the most recently created statistics snapshot based on triggered waveforms are returned.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. If the parameter MPMeter is set the parameter Triggered is mandatory.

RETURN VALUES:

The command returns a comma-separated list of float-valued power values in dBm. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.33 :STATistics:HISTogram:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 histogram for one or multiple Power Meters.

PARAMETERS:

If no parameter is provided or if Triggered is set to 0 the channel 1 histogram for the most recently created statistics snapshot based on continuous statistics acquisition is returned. If the parameter Triggered is set to 1 the channel 1 histogram of the most recently created statistics snapshot based on triggered waveforms is returned.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1. If the parameter MPMeter is set the parameter Triggered is mandatory.

RETURN VALUES:

The command returns a comma-separated list of unsigned 64 bit integer values specifying the number of samples of a power value falling into the associated power value bins returned by :STATistics:Power? [<Triggered>][,<MPMeter>]. The bin size is specified by :STATistics:RESolution <Resolution>[,<MPMeter>]. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.34 :STATistics:HISTogram:P2? [<Triggered>][,<MPMeter>]

Query channel 2 histogram for one or multiple Power Meters, see :STATistics:HISTogram:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.35 :STATistics:HISTogram:P3? [<Triggered>][,<MPMeter>]

Query channel 3 histogram for one or multiple Power Meters, see :STATistics:HISTogram:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.36 :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 discrete relative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters.

PARAMETERS:

See :STATistics:HISTogram:P[1]? [<Triggered>][,<MPMeter>] parameter description for details.

RETURN VALUES:

The command returns a list of float-valued discrete relative probabilities of channel 1 power values. Each value is associated with a power value bin returned by :STATistics:Power? [<Triggered>][,<MPMeter>]. NAN will be returned if there is no valid statistics snapshot data or triggered waveform data.

9.7.37 :STATistics:PDF:P2? [<Triggered>][,<MPMeter>]

Query channel 2 discrete relative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.38 :STATistics:PDF:P3? [<Triggered>][,<MPMeter>]

Query channel 3 discrete relative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.39 :STATistics:CDF:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 discrete cumulative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:HISTogram:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.40 :STATistics:CDF:P2? [<Triggered>][,<MPMeter>]

Query channel 2 discrete cumulative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.41 :STATistics:CDF:P3? [<Triggered>][,<MPMeter>]

Query channel 3 discrete cumulative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.42 :STATistics:CCDF:P[1]? [<Triggered>][,<MPMeter>]

Query channel 1 discrete complementary cumulative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.43 :STATistics:CCDF:P2? [<Triggered>][,<MPMeter>]

Query channel 2 discrete complementary cumulative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.44 :STATistics:CCDF:P3? [<Triggered>][,<MPMeter>]

Query channel 3 discrete complementary cumulative probability distribution of the most recent statistics snapshot or triggered waveform for one or multiple Power Meters, see :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>] for description of parameters and return values.

9.7.45 :STATistics:BINary? [<Triggered>][,<MPMeter>]

Query all statistical values of the most recent statistics snapshot or triggered waveform in binary format for one or multiple Power Meters. This command can be used to reduce communications overhead when polling statistical values via software.

PARAMETERS:

See :STATistics:MINimum:P[1]? [<Triggered>][,<MPMeter>] parameter description for details.

RETURN VALUES:

Binary data block followed by a carriage return, new-line sequence. The first four bytes specify the number of bytes of the binary data block which will be returned following the first four bytes.

For each Power Meter P a chunk of binary data will be sent, one for each referenced Power Meter. All values are encoded in little endian format. Data are ordered as follows:

POWER METER NUMBER

32 bit unsigned integer value giving the serial number of the corresponding Power Meter. If the Power Meter P is not defined the Power Meter serial number and bin count is set to zero and the binary data block ends.

BIN COUNT

Three bytes specifying the number of bins N contained in the following binary data, the value is a 32 bit unsigned integer value. If there is no valid statistics snapshot data N will have a value of zero and no further data will be returned for the binary data chunk.

OFFSET

32 bit signed integer value as described in :STATistics:HISTogram:OFFset? [<Triggered>][,<MPMeter>].

SAMPLES

64 bit unsigned integer value as described in :STATistics:SAMples? [<Triggered>][,<MPMeter>].

RESOLUTION

32 bit single precision floating point value as described in :STATistics:RESolution? [<MPMeter>].

MINIMUM

Three 32 bit single precision floating point values as described in :STATistics:MINimum:ALL? [<Triggered>][,<MPMeter>].

MAXIMUM

Three 32 bit single precision floating point values as described in :STATistics:MAXimum:ALL? [<Triggered>][,<MPMeter>].

ARITHMETIC MEAN

Three 32 bit single precision floating point values as described in :STATistics:MEAN:ALL? [<Triggered>][,<MPMeter>].

ROOT MEAN SQUARE

Three 32 bit single precision floating point values as described in :STATistics:RMS:ALL? [<Triggered>][,<MPMeter>].

STANDARD DEVIATION

Three 32 bit single precision floating point values as described in :STATistics:SDEViation:ALL? [<Triggered>][,<MPMeter>].

BINS

N 32 bit single precision floating point values as described in :STATistics:BINary? [<Triggered>][,<MPMeter>].

HISTOGRAM, CHANNEL 1

N 64 bit unsigned integer values as described in :STATistics:HISTogram:P[1]? [<Triggered>][,<MPMeter>].

HISTOGRAM, CHANNEL 2

N 64 bit unsigned integer values as described in :STATistics:HISTogram:P2? [<Triggered>][,<MPMeter>].

HISTOGRAM, CHANNEL 3

N 64 bit unsigned integer values as described in :STATistics:HISTogram:P3? [<Triggered>][,<MPMeter>].

RELATIVE PROBABILITY, CHANNEL 1

N 32 bit single precision floating point values as described in :STATistics:PDF:P[1]? [<Triggered>][,<MPMeter>].

RELATIVE PROBABILITY, CHANNEL 2

N 32 bit single precision floating point values as described in :STATistics:PDF:P2? [<Triggered>][,<MPMeter>].

RELATIVE PROBABILITY, CHANNEL 3

N 32 bit single precision floating point values as described in :STATistics:PDF:P3? [<Triggered>][,<MPMeter>].

RELATIVE CUMULATIVE PROBABILITY, CHANNEL 1

N 32 bit single precision floating point values as described in :STATistics:CDF:P[1]? [<Triggered>][,<MPMeter>].

RELATIVE CUMULATIVE PROBABILITY, CHANNEL 2

N 32 bit single precision floating point values as described in :STATistics:CDF:P2? [<Triggered>][,<MPMeter>].

RELATIVE CUMULATIVE PROBABILITY, CHANNEL 3

N 32 bit single precision floating point values as described in :STATistics:CDF:P3? [<Triggered>][,<MPMeter>].

RELATIVE COMPLEMENTARY CUMULATIVE PROBABILITY, CHANNEL 1

N 32 bit single precision floating point values as described in :STATistics:CCDF:P[1]? [<Triggered>][,<MPMeter>].

RELATIVE COMPLEMENTARY CUMULATIVE PROBABILITY, CHANNEL 2

N 32 bit single precision floating point values as described in :STATistics:CCDF:P2? [<Triggered>][,<MPMeter>].

RELATIVE COMPLEMENTARY CUMULATIVE PROBABILITY, CHANNEL 3

N 32 bit single precision floating point values as described in :STATistics:CCDF:P3? [<Triggered>][,<MPMeter>].

9.8 :MProbe Commands

9.8.1 :MPMeter:SERial <MPMeter>,<SN1>[,<SN2>,...,<SNN>]

Define a Multi-Power Meter system by specifying one or multiple Power Meter serial numbers.

PARAMETERS:

The first unsigned integer parameter MPMeter sets the Multi-Power Meter system number for a Multi-Power Meter setup. MPMeter must be greater that zero.

The following unsigned integer parameters SN1 through SNN specify the Power Meter serial numbers of a Multi-Power Meter setup. The number of Power Meters in a Multi-Power Meter setup is independent of the Number of Power Meters specified and must be greater than zero. E.g. a four Power Meter setup can specify SN1 through SN3 or SN1 through SN8. Power Meter serial numbers must be set to one of the enumerated Power Meters queried via :SYSTem:SERial? [<MPMeter>], unknown Power Meter serial numbers will cause a vacancy in the Multi-Power Meter system.

9.8.2 :MPMeter:SERial? <MPMeter>

Query Power Meter serial number(s) for Multi-Power Meter systems. This command is an alias of :SYSTem:SERial? [<MPMeter>] when used with an MPMeter parameter.

PARAMETERS:

The unsigned integer parameter MPMeter specifies the Multi-Power Meter system number defined either automatically for setup number 0, or via :MPMeter:SERial <MPMeter>,<SN1>[,<SN2>,...,<SNN>] for setup numbers greater than zero.

RETURN VALUES:

Comma-separated list of unsigned integers indicating the Power Meter serial numbers for the Multi-Power Meter system specified by the parameter MPMeter. NAN will be returned instead of the Power Meter serial number if the specified Multi-Power Meter does not exist. NAN will be returned if the specified Multi-Power Meter system has not been configured.

9.9 :VIRTual Computer Interface Commands

9.9.1 :VIRTual:SERial?

Query serial numbers of connected virtual Power Meters.

RETURN VALUES:

Unsigned integer-valued comma-separated list of all connected virtual Power Meter serial numbers. If no virtual Power Meters have been connected the command will return NAN.

9.9.2 :VIRTual:CONnect [<SER>]

Connect a new virtual Power Meter.

PARAMETER:

The optional unsigned integer parameter Power Meter specifies the serial number of the virtual Power Meter. If omitted the default serial number is set to 1.

9.9.3 :VIRTual:DISConnect

Disconnect currently active Power Meter if it is a virtual Power Meter.

9.9.4 :VIRTual:CW <RSSI1>,<RSSI2>,<RSSI3>

Set channel 1, channel 2 and channel 3 CW RSSI value of the currently selected virtual Power Meter.

PARAMETERS:

The three unsigned integer parameters channel 1, channel 2 and channel 3 RSSI set the signal strength indicated by the virtual Power Meter's ADCs. The default values are 0.

9.9.5 :VIRTual:CW?

Query the channel 1, channel 2 and channel 3 RSSI values of the currently selected virtual Power Meter.

RETURN VALUES:

The command returns a comma-separated list of three unsigned integer values giving the channel 1, channel 2 and channel 3 RSSI values. NAN will be returned if the active Power Meter is not virtual.

9.9.6 :VIRTual:NOIse <NOISE1>,<NOISE2>,<NOISE2>

Set the maximum added noise amplitude of the currently selected virtual Power Meter.

PARAMETERS:

The three unsigned integer parameters channel 1, channel 2 and channel 3 NOISE set the maximum added RSSI value noise amplitude. The time-average of the values is zero. Ranges are distributed evenly between -1 times the given amplitudes and +1 times the given amplitudes. The default values are 0.

9.9.7 :VIRTual:NOIse?

Query the channel 1, channel 2 and channel 3 noise amplitude of the currently selected virtual Power Meter.

RETURN VALUES:

The command returns a comma-separated list of three unsigned integer values giving the maximum amplitude of added channel 1, channel 2 and channel 3 noise in LSB. NAN will be returned if the active Power Meter is not virtual.

9.9.8 :VIRTual:PULse [<RSSI1>],[<RSSI2>],[<RSSI3>],[<T>],[<Ton>]

Set the parameters of the virtual pulse signal for currently selected virtual Power Meter.

PARAMETERS:

RSSI1
: unsigned integer value setting the channel 1 RSSI pulse value

RSSI2
: unsigned integer value setting the channel 2 RSSI pulse value

RSSI3
: unsigned integer value setting the channel 3 RSSI pulse value

T
: unsigned integer value setting the pulse period expressed as a number of samples

TON
: unsigned integer value setting the ON-time at the beginning of reach pulse period expressed as a number of samples

9.9.9 :VIRTual:PULse?

Query the pulse parameters of the currently selected virtual Power Meter.

RETURN VALUES:
: The command returns a comma-separated list of five unsigned integer values as described in the parameter's description of :VIRTual:PULse [<RSSI1>],[<RSSI2>],[<RSSI3>],[<T>],[<Ton>]. NAN will be returned if the active Power Meter is not virtual.

9.9.10 :VIRTual:PLIST <P1_1>,<P2_1>,<P3>[,...,<P1_N>,<P2_N>,<P3_N>]

Append sets of channel 1, channel 2 and channel 3 power values to list of the currently selected virtual Power Meter.

PARAMETERS:
: Multiples of three float-valued power strengths, specifying channel 1, channel 2 and channel 3 power values. The command accepts up to tree times 256 values. The power values are converted to RSSI values using the currently set mode and frequency. RSSI values will not be adjusted if either of these parameters is changed. Power values exceeding the calibrated signal range will be limited to the maximum or minimum calibrated value.

9.9.11 :VIRTual:LIST <RSSI1_1>,<RSSI2_1>,<RSSI3_1>[,...,<RSSI1_N>,<RSSI2_N>,<RSSI3_N>]

Append sets of channel 1, channel 2 and channel 3 RSSI values to list of the currently selected virtual Power Meter.

PARAMETERS:

Multiples of three integer-valued RSSI, specifying the channel 1, channel 2 and channel 3 RSSI values. The command accepts up to tree times 256 values.

9.9.12 :VIRTual:LIST?

Query the list of arbitrary RSSI values the currently selected virtual Power Meter.

RETURN VALUES:

The command returns a comma separated, unsigned integer list of all channel 1, channel 2 and channel 3 RSSI values of the arbitrary power value list. NAN will be returned if the active Power Meter is not virtual or the list is empty.

9.9.13 :VIRTual:LCNt?

Query number of samples in arbitrary power value list of the currently selected virtual Power Meter.

RETURN VALUE:

Unsigned integer-valued number of power samples in arbitrary power value list. NAN will be returned if the active Power Meter is not virtual.

9.9.14 :VIRTual:LCLear

Clear arbitrary power value list of the currently selected virtual Power Meter.

9.10 :STReam Recording Commands

9.10.1 :STReam:MAster <State>

Set currently selected Power Meter to be the master or a slave Power Meter for stream recording.

PARAMETER:

Setting State to 1 makes the current Power Meter the master during stream recording. A State of 0 makes the Power Meter a slave during stream recording, i.e. stream synchronization is controlled by a different Power Meter.

9.10.2 :STReam:MAster? [<MPMeter>]

Query master/slave status of the currently active Power Meter during stream recording.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned integer value containing the stream recording master/slave status of the currently active Power Meter. Slaves return 0, the master returns 1.

9.10.3 :STReam:LENgth <Length>[,<MPMeter>]

Set number of samples to be recorded during stream recording for one or multiple Power Meters. The parameter can only be set when stream recording is inactive.

PARAMETERS:

The unsigned integer-valued parameter of the command specifies the number of consecutive samples to be streamed. E.g. »:STReam:len 100« will record 100 consecutive samples. Setting Length to zero configures indefinite streaming, i.e. streaming needs to be terminated by issuing a :STReam:ENable? [<MPMeter>] command.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.10.4 :STReam:LENgth? [<MPMeter>]

Query number of stream samples to be recorded for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the unsigned integer-valued number of samples to be streamed as specified using the :STReam:LENgth <Length>[,<MPMeter>] command. A value of zero indicates indefinite streaming. If executed for multiple Power Meters the command returns a list of unsigned integer-valued lengths of the number of streaming samples for each Power Meter of the list specified by the MPMeter parameter.

9.10.5 :STReam:ENable <State>[,<MPMeter>]

Enable or disable stream recording for one or multiple Power Meters.

PARAMETERS:

Setting State to 1 activates stream recording, creating a new stream file for each addressed

Power Meter. Setting State to 0 disables stream recording and closes the associated stream file(s). See Section 10.1.6 for details about the stream file format.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.10.6 :STReam:ENable? [<MPMeter>]

Query status of stream recording for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned integer value containing the stream recording status. A value of 1 is returned when stream data acquisition is enabled, 0 is returned if stream data acquisition is disabled.

9.10.7 :STReam:PROgress? [<MPMeter>]

Query number of samples in current stream recording for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns an unsigned 64 bit integer value giving the number of samples that have been recorded for the selected Power Meter since the start of stream recording.

9.10.8 :STReam:SKIp <Value>[,<MPMeter>]

Set number of stream samples to be skipped for one or multiple Power Meters. Parameter may only be set if stream data acquisition is disabled.

PARAMETERS:

The unsigned integer-valued parameter specifies the number of samples to be skipped after recording a sample during stream recording. E.g. »:stream:skip 100« will discard 100 samples in between two samples logged to a file. A value of zero indicates no skipping occurs, all samples are logged to file.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.10.9 :STReam:SKIp? [<MPMeter>]

Query number of stream samples to be skipped for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns the unsigned integer-valued number of samples to be skipped after recording a sample during stream recording, it corresponds to the value set using :STReam:SKIp <Value>[,<MPMeter>]. If executed for multiple Power Meters the command returns a list of unsigned integer-valued numbers for values to skip for each Power Meter of the respective list.

9.10.10 :STReam:PREfix <String>[,<MPMeter>]

Set file prefix for stream recording for one or multiple Power Meters. The parameter can only be set when stream recording is disabled.

PARAMETERS:

String parameter with quotes specifying the stream log file prefix. The string may not exceed 127 characters. The default value is set to "stream". E.g. »:str:pre "streamFile"« will result in a log files named "streamFile_PP_YYYYMMDD_hhmmss_msmsms.csv" to be saved when enabling stream recording. See Section 10.1.6 for a detailed description of stream file naming conventions and the stream file format.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.10.11 :STReam:PREfix? [<MPMeter>]

Query file prefix for stream recording for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a string value without quotes specifying the set stream log file prefix, see :STReam:PREfix <String>[,<MPMeter>] for more details. If executed for multiple Power Meters the command returns a list of string values containing the value for each Power Meter of the respective list.

9.10.12 :STReam:SYNC <Sync>[,<MPMeter>]

Set synchronization source for stream recording for one or multiple Power Meters.

PARAMETERS:

String parameter without quotes specifying the synchronization source, valid values are OFF, EXT and EXT2. When set to OFF stream recording will start immediately upon enabling using the »:STReam:ENable <State>[,<MPMeter>]« command. When set to EXT the Power Meter's BNC connector will be used to synchronize stream recording slaves with the stream recording master. When set to EXT2 the Power Meter's RJ45 socket will be used to synchronize stream recoding slaves with the stream recording master. When set to EXT or EXT2 the slave/master status set using the »:STReam:MAster <State>« command determines the input/output configuration of the respective connector.

The second, optional unsigned integer parameter MPMeter is described in Section 9.1.

9.10.13 :STReam:SYNC? [<MPMeter>]

Query synchronization source for stream recording for one or multiple Power Meters.

PARAMETER:

The optional unsigned integer parameter MPMeter is described in Section 9.1.

RETURN VALUE:

The command returns a string value without quotes specifying the set stream synchronization source, see :STReam:SYNC <Sync>[,<MPMeter>] for more details. If executed for multiple Power Meters the command returns a list of string values containing the value for each Power Meter of the respective list.

10 File Formats

All data used by the LSPM TCP Server and GUI are stored in the form of tabulator-separated plain ASCII text files. The uniform file extension is `.csv`. Lines are separated by newline characters (ASCII code `0xa`), columns are separated by tabulators.

Numbers are expressed as plain decimal integers, as floating point numbers using "." as the decimal separator, or in exponential format using "." as the decimal separator and "e" as the exponential separator, e.g. "1.2e3" encoding a value of 1,200.

In all examples given below "⊣" denotes a tabulator, ⏎ a newline character, and "↙" a line wrap indicating that the contents of the next line in this document belong to the same line of the `.csv` file.

10.1 LSPM GUI Log Files

The creation time and Power Meter serial number of all log files are saved by appending a time string formatted as "_N_YYYYMMDD_hhmmss_SSS" to a file's base name. "N","YYYY", "MM", "DD", "hh", "mm", "ss' and "SSS" denote Power Meter serial number, year, month, day, hour, minute, second and millisecond of log file creation with their respective number of digits. The number of digits used for the Power Meter serial number depends on the numeric value of the serial number.

The first line of all log files contains a header starting with a hash mark (#), which describes the contents of each column in the remainder of the file.

10.1.1 Basic Data Logger

The file format for all basic log files contains at least 7 columns described with their column headers and unit values in the table below.

Continuous logging will add one line for for every newly polled set of values. If more than one Power Meter are present one log file will be created for every Power Meter.

If a log file is created using the GUI's "Quick Save" button all presently displayed values of the active Power Meter will be saved to a newly created file.

Column	Header	Unit	Description
1	t	s	Timestamp expressed as a floating point number of seconds since 1904/01/01 00:00:00 UTC
2	Mode		Measurement mode, see Table 1, page 15
3	f	Hz	Compensation frequency
4	P1	dBm	channel 1 power
5	P2	dBm	channel 2 power
6	P3	dBm	channel 3 power

Column	Header	Unit	Description
7	fLpP	Hz	power value low pass filter frequency
8	RSSI1	LSB	channel 1 RSSI value, optional
9	RSSI2	LSB	channel 2 RSSI value, optional
10	RSSI3	LSB	channel 3 RSSI value, optional

EXAMPLE OF BASIC LOG FILE:

```
#t ⊣Mode ⊣f⊣P1 ⊣P2 ⊣P3 ⊣fLpP ⊣RSSI1⊣RSSI2⊣RSSI3↵
3606197249.102 ⊣1⊣100000000⊣-42.45547⊣-41.116783 ⊣-41.568943 ⊣0⊣↙
    6074 ⊣6074 ⊣6074↵
3606197249.245 ⊣1⊣100000000⊣-38.207194 ⊣-36.87114⊣-37.288121 ⊣0⊣↙
    6702 ⊣6702 ⊣6702↵
```

10.1.2 Power Scope Data Logger

The file format for all field scope log files contains at least five columns described with their column headers and unit values in the table below.

Continuous logging will create a new log file for every newly recorded set of waveforms. If more than one Power Meter is present one log file will be created for each Power Meter.

If a log file is created using the GUI's "Quick Save" button all presently displayed values of the active Power Meter will be saved to a newly created file.

Column	Header	Unit	Description
1	Mode		Measurement mode, see Table 1, page 15
2	f	Hz	Compensation frequency
3	P1	dBm	channel 1 power
4	P2	dBm	channel 2 power
5	P3	dBm	channel 3 power
6	RSSI1	LSB	channel 1 RSSI value, optional
7	RSSI2	LSB	channel 2 RSSI value, optional
8	RSSI3	LSB	channel 3 RSSI value, optional
9	f	Hz	Compensation frequency for sweep, optional
10	P1	dBm	channel 1 power for sweep, optional
11	P2	dBm	channel 2 power for sweep, optional
12	P3	dBm	channel 3 power for sweep, optional

EXAMPLE OF FIELD SCOPE LOG FILE:

```
#Mode⊣f⊣P1 ⊣P2 ⊣P3 ⊣RSSI1⊣RSSI2⊣RSSI3⊣f⊣P1 ⊣P2 ⊣P3↵
1⊣100000000⊣-82.007195 ⊣-83.007195 ⊣-75.007301 ⊣30 ⊣30 ⊣30 ⊣↙
   100000000⊣-82.007195 ⊣-83.007195 ⊣-75.007301↵
1⊣100000000⊣-82.007195 ⊣-83.007195 ⊣-75.007301 ⊣1030 ⊣1030 ⊣1030 ⊣↙
   200000000⊣-82.007195 ⊣-83.007195 ⊣-75.007301↵
```

10.1.3 Radar Data Logger

The file format for all radar log files contains at least 11 columns described with their column headers and unit values in the table below. The number of columns is dependent on the number of detected pulses for all channels. For every pulse a pair of columns consisting of the sample index of the start of the pulse and maximum power value for the pulse will be added to the radar log file. First, N1 value pairs for the channel 1 values will be added, followed by N2 and N3 value pairs for the other channels' values. The pulse counts N1, N2 and N3 are given in columns 9 through 11.

Continuous logging will create a new line in the log file for every newly recorded set of waveforms. If more than one Power Meter are present one log file will be created for every Power Meter.

If a log file is created using the GUI's "Quick Save" button all presently displayed values of the active Power Meter will be saved to a newly created file.

Column	Header	Unit	Description
1	t	s	Timestamp expressed as a floating point number of seconds since 1904/01/01 00:00:00 UTC
2	Mode		Measurement mode, see Table 1, page 15
3	f	Hz	Compensation frequency
4	P1	dBm	Arithmetic mean of all pulses' channel 1 power maximums
5	P2	dBm	Arithmetic mean of all pulses' channel 2 power maximums
6	P3	dBm	Arithmetic mean of all pulses' channel 3 power maximums
7	Samp		Number of samples in waveforms evaluated for radar detection
8	Th	V/m	Threshold value for radar pulse detection, set to zero for automatic detection threshold
9	CNT1		Number of pulses detected for channel 1 power
10	CNT2		Number of pulses detected for channel 2 power
11	CNT3		Number of pulses detected for channel 3 power
12...	IDX1N		Sample index of N^{th} channel 1 pulse
13...	P1N	dBm	N^{th} channel 1 pulse's maximum power value

Column	Header	Unit	Description
12...	IDX2N		Sample index of N^{th} channel 2 pulse
13...	P2N	dBm	N^{th} channel 2 pulse's maximum power value
12...	IDX3N		Sample index of N^{th} channel 3 pulse
13...	P3N	dBm	N^{th} channel 3 pulse's maximum power value

EXAMPLE OF RADAR LOG FILE:

```
#t ⊣Mode ⊣f⊣P1 ⊣P2 ⊣P3 ⊣Samp ⊣Th ⊣CNT1 ⊣CNT2 ⊣CNT3 ⊣IDX1N⊣P1N⊣IDX2N↙
   ⊣P2N⊣IDX3N⊣P3N⏎
3606213841.400 ⊣1⊣100000000⊣-22.839362 ⊣-26.087731 ⊣-17.762401 ⊣↙
   2500 ⊣-100 ⊣2⊣2⊣2⊣967⊣-22.839362 ⊣1967 ⊣-22.839362 ⊣967⊣↙
   -26.087731 ⊣1967 ⊣-26.087731 ⊣967⊣-17.762401 ⊣1967 ⊣-17.762401⏎
3606213974.097 ⊣1⊣100000000⊣-19.396083 ⊣NaN⊣-16.450567 ⊣2500 ⊣-20⊣1↙
   ⊣0⊣9⊣549⊣-19.396083 ⊣533⊣-16.338365 ⊣539⊣-14.785077 ⊣541⊣↙
   -19.441782 ⊣543⊣-15.66036⊣547⊣-16.509461 ⊣1534 ⊣-16.713666 ⊣↙
   1542 ⊣-14.569799 ⊣1545 ⊣-16.646755 ⊣1547 ⊣-17.389835⏎
```

10.1.4 Sweep Data Logger

The file format for all sweep log files contains at least six columns described with their column headers and unit values in the table below. One line is written for each sweep step.

Continuous logging will create a new log file for every newly recorded set of waveforms and change of a sweep parameter. If more than one Power Meter are present one log file will be created for every Power Meter.

If a log file is created using the GUI's "Quick Save" button all presently displayed values of the active Power Meter will be saved to a newly created file.

Column	Header	Unit	Description
1	Mode		Measurement mode, see Table 1, page 15
2	f	Hz	Compensation frequency
3	Index		Center sample index
4	P1	dBm	Averaged channel 1 power
5	P2	dBm	Averaged channel 2 power
6	P3	dBm	Averaged channel 3 power
7	RSSI1	LSB	Averaged raw channel 1 RSSI value, optional
8	RSSI2	LSB	Averaged raw channel 2 RSSI value, optional
9	RSSI3	LSB	Averaged raw channel 3 RSSI value, optional

EXAMPLE OF SWEEP LOG FILE:

```
#Mode⊣f⊣Index⊣P1 ⊣P2 ⊣P3 ⊣RSSI1⊣RSSI2⊣RSSI3⏎
1⊣100000000⊣699⊣-52.943813 ⊣-45.659973 ⊣-49.853455 ⊣4561 ⊣5422 ⊣↙
    4896⏎
1⊣450000000⊣1699 ⊣-52.845222 ⊣-45.333584 ⊣-49.958347 ⊣4575 ⊣5469 ⊣↙
    4881⏎
```

10.1.5 Statistics Data Logger

The file format for all statistics log files contains at least 26 columns described with their column headers and unit values in the table below. The number of columns is dependent on the number of power value bins in the recorded histogram, there is at least one bin in the histogram. For every bin three columns consisting of the number of channel 1, 2 and 3 samples detected for the corresponding bin will be added to the radar log file. The number of bins is specified in column number 23.

Continuous logging will add a new line to the log file for every newly recorded statistics snapshot. If more than one Power Meter are present, one log file will be created for every Power Meter.

If a log file is created using the GUI's "Quick Save" button all presently displayed values of the active Power Meter will be saved to a newly created file.

Column	Header	Unit	Description
1	t	s	Timestamp expressed as a floating point number of seconds since 1904/01/01 00:00:00 UTC
2	Mode		Measurement mode, see Table 1, page 15
3	f	Hz	Compensation frequency
4	Type		Statistics type, 0 for continuous statistics, 1 for triggered statistics
5	MIN1	dBm	Channel 1 power minimum
6	MIN2	dBm	Channel 2 power minimum
7	MIN3	dBm	Channel 3 power minimum
8	MAX1	dBm	Channel 1 power maximum
9	MAX2	dBm	Channel 2 power maximum
10	MAX3	dBm	Channel 3 power maximum
11	MEAN1	dBm	Channel 1 power arithmetic mean
12	MEAN2	dBm	Channel 2 power arithmetic mean
13	MEAN3	dBm	Channel 3 power arithmetic mean
14	RMS1	dBm	Channel 1 power root mean square
15	RMS2	dBm	Channel 2 power root mean square

Column	Header	Unit	Description
16	RMS3	dBm	Channel 3 power root mean square
17	SDEV1	dBm	Channel 1 power standard deviation
18	SDEV2	dBm	Channel 2 power standard deviation
19	SDEV3	dBm	Channel 3 power standard deviation
20	Samp		Number of samples used for statistics evaluation
21	Res	dB	power resolution for histogram output
22	Offs	dBm	power offset of minimum bin in histogram
23	Bins		Number of bins in power value histogram
24...	CNT1N		Number of channel 1 power values in N^{th} bin of histogram
25...	CNT2N		Number of channel 2 power values in N^{th} bin of histogram
26...	CNT3N		Number of channel 3 power values in N^{th} bin of histogram

EXAMPLE OF STATISTICS LOG FILE:

```
#t ⇥Mode ⇥f⇥Type ⇥MIN1 ⇥MIN2 ⇥MIN3 ⇥MAX1 ⇥MAX2 ⇥MAX3 ⇥MEAN1⇥MEAN2⇥↙
   MEAN3⇥RMS1 ⇥RMS2 ⇥RMS3 ⇥SDEV1⇥SDEV2⇥SDEV3⇥Samp ⇥Res⇥Offs ⇥Bins↙
    ⇥CNT1N⇥CNT2N⇥CNT3N↵
3606271892.999 ⇥1⇥100000000⇥0⇥-82.004997 ⇥-83.004997 ⇥-75.004997 ⇥↙
   10.995 ⇥10.995 ⇥10.995 ⇥-30.340851 ⇥-29.428646 ⇥-28.592386 ⇥↙
   44.246136⇥43.74614 ⇥41.94656 ⇥32.204865⇥32.367878⇥30.691845⇥↙
   22726830 ⇥5⇥-17⇥20 ⇥0⇥3406472⇥122056 ⇥330106 ⇥678243 ⇥914033 ⇥↙
   962578 ⇥1002801⇥1000027⇥1024993⇥1015284⇥1015284⇥1018058⇥1001650↙
   ⇥1014628⇥975764 ⇥1029706⇥981996 ⇥927903 ⇥4305248⇥3190100⇥37449⇥↙
   119282 ⇥313462 ⇥654664 ⇥891841 ⇥952869 ⇥994479 ⇥991705 ⇥1020832↙
   ⇥1011123⇥1009736⇥1020832⇥1004223⇥1014628⇥970212 ⇥1020180⇥979244↙
    ⇥919581 ⇥4610388⇥0⇥0⇥3453630⇥325945 ⇥651890 ⇥890454 ⇥951482 ⇥↙
   984770 ⇥980609 ⇥1013897⇥1002801⇥1008349⇥1015284⇥995946 ⇥1006300↙
   ⇥959108 ⇥1013236⇥975061 ⇥911259 ⇥4586809↵
3606271976.273 ⇥1⇥100000000⇥0⇥-82.004997 ⇥-83.004997 ⇥-75.004997 ⇥↙
   10.995 ⇥10.995 ⇥10.995 ⇥-30.341587 ⇥-29.429396 ⇥-28.593102 ⇥↙
   44.246834⇥43.746834⇥41.947224⇥32.205132⇥32.368137⇥30.692085⇥↙
   26576572 ⇥20 ⇥-4 ⇥6⇥4273970⇥3806834⇥4734618⇥4716471⇥4345745⇥↙
   4698934⇥4058244⇥3722490⇥4705422⇥4721137⇥4362165⇥5007114⇥4191248↙
   ⇥3716002⇥4668116⇥4685486⇥4357266⇥4958454↵
```

10.1.6 Stream Files in Binary Format

The format "PREFIX_SN_YYYYMMDD_hhmmss.bin" containing the following information is used for all binary stream files.

PREFIX
: File prefix set by the user, its default value is 'stream',

SN
: Serial number of the Power Meter which recorded the stream data.

YYYYMMDD_hhmmss
: Year, month, day of the month, hour, minute and second of the start of the stream file.

Stream files are optimized for small file size and low processor load. Their binary format contains power value lookup-tables and raw, unsigned integer RSSI values for all channels, including information about mode, temperature, frequency and skip count. Power value lookup-tables can be used off-line for converting RSSI values to power values.

All stream files start with a lookup-table block containing three-channel power value lookup-tables and auxiliary information. Power value information is stored in the form of RSSI data blocks. Lookup-table blocks start with a single byte set to a value of 254. They have a size of 196,623 bytes and the following structure:

Data	Offset	Size	Description
Identifier	0	1	Single byte set to 254
Mode	1	2	Power Meter mode, 16 bit, little endian, unsigned integer
Frequency	3	4	Power Meter frequency in Hertz, 32 bit, single precision, little-endian floating point value
Temperature	7	4	Power Meter temperature in °C, 32 bit, single precision, little-endian floating point value
Skip Count	11	4	Stream recording skip count, 32 bit, little endian, unsigned integer value
channel1[0]	15	4	channel 1 power value for RSSI=0, 32 bit, single precision, little-endian floating point value
channel2[0]	19	4	channel 2 power value for RSSI=0
channel3[0]	23	4	channel 3 power value for RSSI=0
channel1[1]	27	4	channel 1 power value for RSSI=1
...	...	4	...
channel3[16383]	196,619	4	channel 3 power value for RSSI=16383

Lookup-table blocks are followed by an arbitrary number of RSSI data blocks. Each block contains

simultaneous channel 1, 2 and 3 RSSI values and a dummy byte for one sampling instant. Each block has a a size of seven bytes and the following structure:

Data	Size	Description
unused	1	unused, constant value of 1, 8 bit unsigned integer
channel1 RSSI	2	channel1 RSSI value, 16 bit unsigned integer little-endian format
channel2 RSSI	2	channel2 RSSI value
channel3 RSSI	2	channel3 RSSI value

RSSI data blocks are followed by a lookup-table indicated by the header described above when the lookup table is modified, e.g. in response to a frequency or mode change of the Power Meter.

10.1.7 Stream Files in CSV Format

The Bin2Csv.exe program which is part of the LSPM Installer can be used to generate CSV files from binary stream files described in the previous section. CSV output files have the following format:

Column	Header	Unit	Description
1	Mode		Measurement mode, see Table 1, page 15, optional
2	f	Hz	Compensation frequency, optional
3	P1	dBm	channel 1 power
4	P2	dBm	channel 2 power
5	P3	dBm	channel 3 power
6	RSSI1	LSB	channel 1 raw RSSI value, optional
7	RSSI2	LSB	channel 2 raw RSSI value, optional
8	RSSI3	LSB	channel 3 raw RSSI value, optional
9	T	°C	Power Meter cold plate temperature, optional
10	Skip		Skip count used during stream recording, optional

The Bin2Csv.exe program accepts an arbitrary number of command line switches followed by the file name(s) of one or more binary stream files. One CSV file will be generated for every bin file, replacing the extension "bin" by "csv". Note that the Bin2Csv.exe program can be used for both E-field and power value stream files. The stream file type will be detected automatically. The following command line switches are supported for power stream files, redundant parameters will be ignored silently:

-h

display usage information and quit program,

-s

set sample index for the start of binary to CSV conversion, default is zero,

-e

set sample index for the end of binary to CSV conversion, default is last sample in bin file,

-l

set number of samples to be converted to CSV format, relative to start sample index if specified, defaults to all samples in bin file,

-M

enable optional mode column,

-F

enable optional frequency column,

-T

enable optional temperature column,

-r

enable optional RSSI value columns,

-S

enable optional skip count column.

»:MEASure[:Power]:P[1]? [<MPMeter>]«,

10.2 extCalLog TCP-Server Logger

For each TCP/IP connection with the exception of the LSPM GUI, the SCPI command :CALibration:LOGging <Value> enables/ disable logging of Power Meter status information and power values every time one of the following SCPI queries is sent by a client with its log flag set to 1:

- »:MEASure[:Power]:P[1]? [<MPMeter>]«
- »:MEASure[:Power]:P2? [<MPMeter>]«
- »:MEASure[:Power]:P3? [<MPMeter>]«
- »:MEASure[:Power]:ALL? [<MPMeter>]«.

The csv file format for all external calibration log files contains sixteen columns described by their column headers and unit values in the table below.

Column	Header	Unit	Description
1	RSSI1	LSB	channel 1 RSSI value
2	RSSI2	LSB	channel 2 RSSI value
3	RSSI3	LSB	channel 3 RSSI value
4	T	°C	Power Meter cold plate temperature
5	f	Hz	Compensation frequency
6	P1	dBm	channel 1 power

Column	Header	Unit	Description
7	P2	dBm	channel 2 power
8	P3	dBm	channel 3 power
9	Serno LSPM		Power Meter serial number
10	Mode		Measurement mode, see Table 1, page 15
11	FW LSPM		LSPM FPGA firmware version
12	FW Server		LSPM TCP Server firmware version
13	t	s	Timestamp expressed as a floating point number of seconds since 1904/01/01 00:00:00 UTC
14	cal:ext		external calibration enabled state, see :CALibration:EXTernal <Value>[,<MPMeter>]
15	fLpP	Hz	power value low pass filter frequency
16	123a		inidcator which SCPI query prompted logging of actual line

EXAMPLE OF EXTCALLOG FILE:

```
#RSSIx ⊣RSSIy⊣RSSIz⊣T⊣f⊣P1 ⊣P2 ⊣P3 ⊣SerNo LSPM ⊣Mode ⊣FW LSPM⊣FW ↙
    Server⊣t⊣cal:ext⊣fLpP ⊣123a⏎
```

10.3 Calibration Files

Each LSPM 1.0 Power Meter comes with a detailed set of in-house calibration files used for linearity compensation, frequency compensation and absolute power value calibration. Optionally, an external power calibration file can be added. Calibration files are stored in one directory per Power Meter, directory names consist of »sn« followed by the decimally coded Power Meter serial number without leading zeros. Calibration directories are stored in the directory specified via the `LSPM_CAL_PATH` environment variable. The CSV file conventions detailed in section 10 apply to all calibration files.

By default the calibration data folders are stored in the form of ZIP files consisting of »sn« followed by the decimally coded Power Meter serial number and the extension ».zip«. `LSPM_CAL_PATH` must not contain both a ZIP file and a directory for the same serial number.

The contents of all calibration files are protected against inadvertent modification by means of a decimally coded integer checksum in the last column of each calibration file's first line. The checksum is calculated by adding all ASCII code values of the calibration file starting with the first character of the second line of the respective calibration file.

10.3.1 In-House Linearity and Frequency Compensation Files

One in-house linearity and frequency file, or short LF file, exists for every in-house calibrated frequency and Power Meter mode. LF file names consist of »sn« followed by the decimally coded Power Meter serial number, »m« followed by the decimally coded mode number, »f« followed by the decimally coded frequency value in Hertz, followed by ».csv«.

The first line of LF files starts with a hash mark character (#) and gives context information for the LF file. It does not give column names for the data in the remainder of the CSV file. The first line's columns have the following contents:

Column	Unit	Description
1		Power Meter serial number as decimally coded integer value
2		Measurement mode, see Table 1, page 15
3	Hz	Measurement frequency
5		Time stamp of calibration for given frequency and mode
6		Checksum for rest of file as described in Section 10.3

The following lines of LF files have the following contents:

Column	Unit	Description
1	dBm	Power level used for given mode and frequency given in the first line of the file
2	LSB	channel 1 RSSI value for given frequency, mode and power value in first column
3	LSB	channel 2 RSSI value for given frequency, mode and power value in first column
4	LSB	channel 3 RSSI value for given frequency, mode and power value in first column

Calibration files for Power Meters equipped with less than three channels contain an RSSI constant value of zero for all power levels for the unavailable power detectors.

10.3.2 External Power Calibration Files

One external power calibration file or short EP file, exists for every externally calibrated Power Meter. An EP file contains correction factors expressed in Decibels, the correction factor will be applied to the base power value derived from the supplied internal calibration data. A value greater than zero will make the displayed power value larger, values smaller than zero lower the displayed power value. The EP file name consists of »sn« followed by the decimally coded Power Meter serial number followed by ».csv«.

The first line of an EP file starts with a hash mark character (#) and gives context information for the EP file, it does not give column names for the data in the remainder of the CSV file. The first line's

columns have the following contents:

Column	Unit	Description
1		Power Meter serial number as decimally coded integer value
2		Maximum time stamp during calibration
3		Checksum for rest of file as described in Section 10.3

The second line of EP files starts with a hash mark character (#) and contains a calibration certificate string that extends to the end of the second line. The subsequent lines of an EP file have the following contents:

Column	Unit	Description
1	Hz	Frequency for external power calibration
2	dB	channel 1 correction value for the given frequency
3	dB	channel 2 correction value for the given frequency
4	dB	channel 3 correction value for the given frequency

11 Specifications

Table 17: *LSPM Power Meter specifications*

Frequency Range	
Low Band	9 kHz ... 400 MHz
High Band	30 MHz ... 6 GHz (usable up to 12 GHz)
Analog Rise Time	
Low Band (Video BW 500 Hz)	1.9 ms
Low Band (Video BW 1 MHz)	770 ns
High Band (Video BW 3 MHz)	330 ns
Minimum Pulse Width	500 ns
VSWR	<1.2:1
Sampling Rate	2 MSample/s
Measurement Range & Dynamic Range	
Low Band	<-60 dBm ... >20 dBm (>80 dB)
High Band up to 4 GHz	<-70 dBm ... >20 dBm (>90 dB)
High Band 4 ... 6 GHz	<-50 dBm ... >20 dBm (>70 dB)
Amplitude Accuracy (0 dBm, CW)	0.1 dB
Linearity Error	0.15 dB
Temperature Stability	0.1 dB
Power Resolution	<0.1 dB (see plot below)
Channel Isolation	>50 dB
Damage Level	>30 dBm
PC Interface	USB 2.0
Application Software	LSPM TCP Server, LSPM GUI
Trigger Voltage	5 V
Trigger Connector	BNC
Input Voltage	5 V ± 5 %
Input Current	<3 A
Ambient Temperature	10 ... 40 °C
Dimensions (W x D x H)	165 x 142 x 61 mm^3
Certifications	CE

11.0.1 Typical Dynamic Range

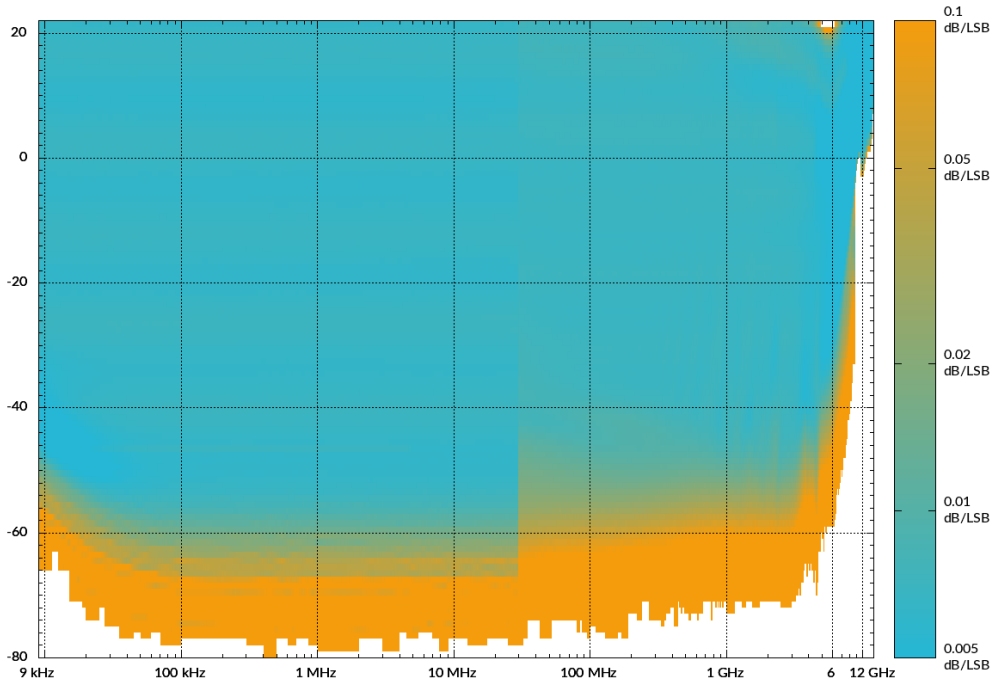


Figure 40: *Typical dynamic range, low and high band*

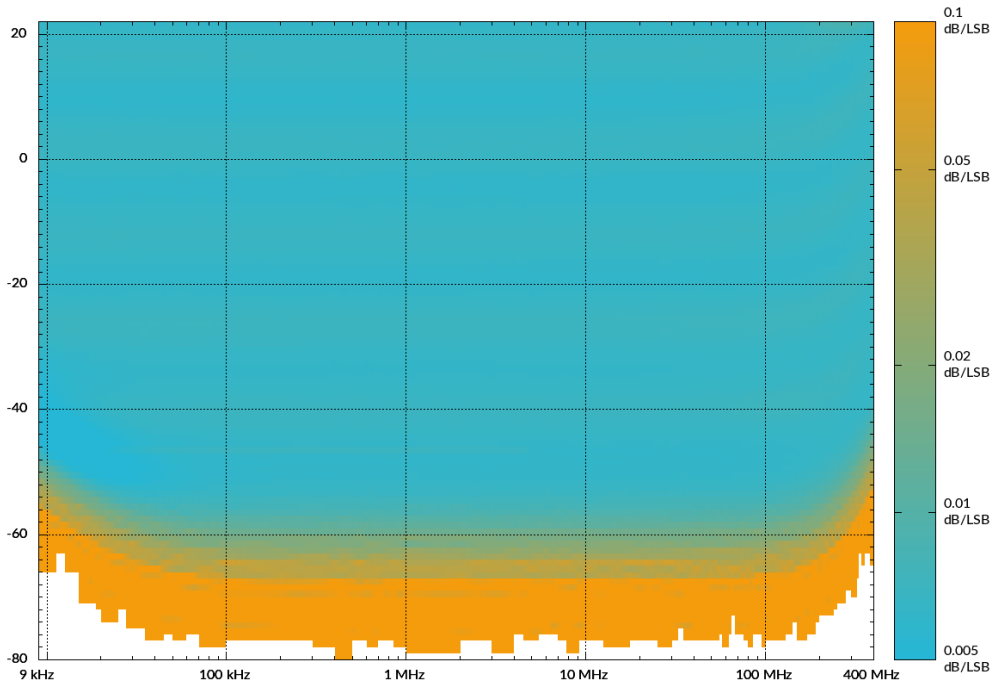


Figure 41: *Typical dynamic range, low band*

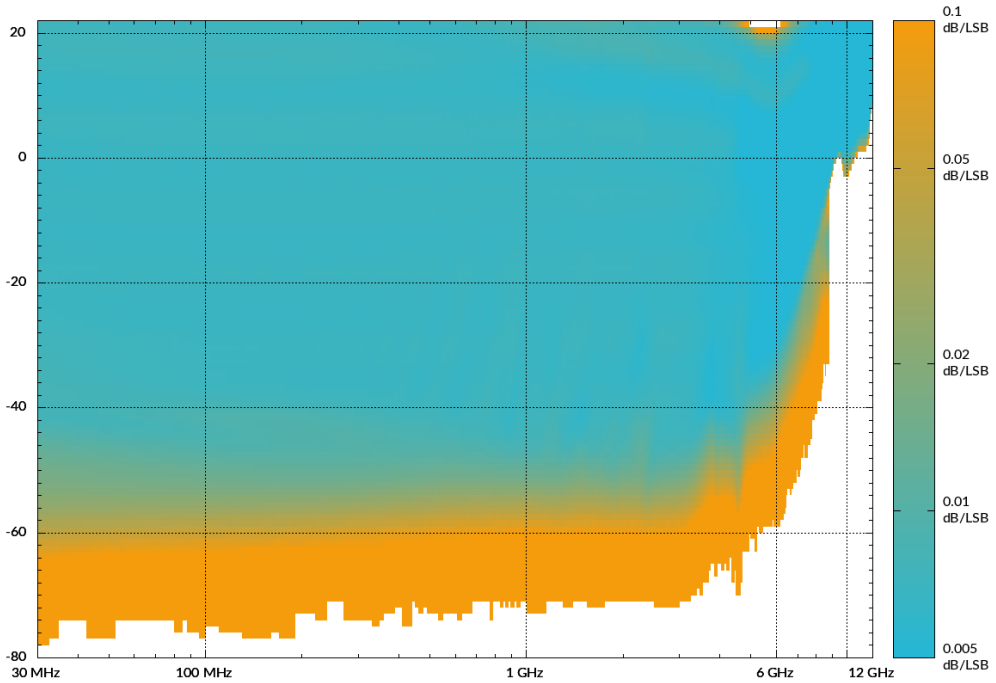


Figure 42: *Typical dynamic range, high band*

12 Warranty Conditions

1. The period of warranty shall start from the date of delivery of the product to the customer and shall cover a period of 24 months.
2. These warranty conditions apply to devices purchased in Germany. These conditions of warranty also apply if these devices are exported abroad and meet the technical requirements (e.g. voltage, frequency) for the respective country and which are suitable for the respective climatic and environmental conditions.
3. Every and all parts of the product are under LUMILOOP's warranty coverage against any defect that may occur during production, assembly and/or defective parts.
4. In case of repair within warranty period, the time spent on the repair work is added to the warranty period. Repair time of the product is maximum 20 (twenty) working days. A warranty event does not lead to a new warranty period. The warranty period for built-in spare parts ends with the warranty period for the entire device.
5. In case of failure of the product during warranty period, the producer or reseller company has to assign another product to the customer with similar features until completion of repair of the product.
6. Within the warranty period, if the product fails because of general material and workmanship, or mounting faults, it will be repaired without demanding any charge.
7. In case of any failure in the product, occurring at least four times in one year or six times within the warranty period, product replacement or refund is mandatory depending on the choice of the customer.
8. Free repair and product exchange obligations will be annulled under the following conditions:
   a) If the product becomes faulty due to use contrary to the terms or conditions stated in the user's manual,
   b) If the product has been opened, used, or previously repaired by unauthorized persons,
   c) Use of the product by plugging into inappropriate voltages or with faulty electric installation,
   d) If the product serial number has been altered or removed,
   e) If the fault or damage to the product occurred during the transportation outside of the responsibility of LUMILOOP GmbH,
   f) A break or scratch to the product's exterior while in the customer's possession,
   g) Damage from chemical and electrochemical effects of water
   h) When our product is damaged due to use with spare parts, accessories or devices purchased from other companies which are not original parts.
   i) Those damages caused by natural disasters such as fire, lightning, flood, earthquake, etc.
9. A short report prepared by the LUMILOOP GmbH will determine whether the damage was caused by improper use.
10. Customers are required to initially report any conflicts between themselves and an authorized reseller to the address below:

Gostritzer Str. 63
01217 Dresden
Germany
Phone: +49 (0)351 85097870
E-mail: info@lumiloop.de

EC DECLARATION OF CONFORMITY

We, LUMILOOP GmbH,
Gostritzer Str. 63,
01217 Dresden,
GERMANY,

declare under sole responsibility that the:

Model / Part Number: LSPM 1.0
Model / Part Name: RF Power Meter
Date of Declaration: October 12, 2018

to which this declaration relates, meets the requirements and is in conformity with the relevant EC Directives listed below using the relevant section(s) of the following EC harmonized standards and other normative documents:

Applicable Directives:

2014/35/EU (Low Voltage Directive)

2014/30/EU (EMC Directive)

2011/65/EU (RoHS)

Applicable harmonized standards and/or other normative documents:

EN 61010-1:2010	Safety requirements for electrical equipment for measurement, control, and laboratory use Part 1: General requirements
EN 61326-1:2013	Electrical equipment for measurement, control and laboratory use – EMC requirements Part 1: General requirements

Authorized Signatories:

LUMILOOP GmbH
Eike Suthau, Technical Director

This declaration attests the compliance with the stated directives. It does not imply any assurance of characteristics.

14 Revision History

2018/10/16

- Initial release.

2018/11/27

- Update of dynamic range plots.
- Fix spelling.
- Expected format of external calibration files by TCP-Server fixed.
- LSPM GUI indicates external calibration enabled/disabled state via date string.