



ROHDE & SCHWARZ

Geschäftsbereich
Messtechnik

Bedienhandbuch

Leistungsmesskopf (AVG)

R&S NRP-Z11

10 MHz ... 8 GHz / 200 pW ... 200 mW

1138.3004.02

R&S NRP-Z21

10 MHz ... 18 GHz / 200 pW ... 200 mW

1137.6000.02

Printed in the Federal
Republic of Germany

Registerübersicht

Datenblatt

Sicherheitshinweise
Qualitätszertifikat
EU-Konformitätserklärung
Support-Center-Adresse
Liste der R&S-Niederlassungen

Register

- | | |
|---|--|
| 1 | Kapitel 1: Inbetriebnahme |
| 2 | Kapitel 2: Virtueller Leistungsmesser |
| 3 | Kapitel 3: Manuelle Bedienung |
| 4 | für Erweiterungen vorgesehen |
| 5 | Kapitel 5: Fernbedienung – Grundlagen |
| 6 | Kapitel 6: Fernbedienung – Befehle |
| 7 | für Erweiterungen vorgesehen |



Zertifikat-Nr.: 2002-36

Hiermit wird bescheinigt, dass der/die/das:

Gerätetyp	Identnummer	Benennung
NRP	1143.8500.02	Leistungsmesser
NRP-B1	1146.9008.02	Test Generator
NRP-B2	1146.8801.02	Zweiter Messeingang
NRP-B5	1146.9608.02	3. und 4. Messeingang
NRP-B6	1146.9908.02	Messeingänge Rückseite
NRP-Z3	1146.7005.02	USB Adapter
NRP-Z4	1146.8001.02	USB Adapter
NRP-Z11	1138.3004.02	Leistungsmesskopf
NRP-Z21	1137.6000.02	Leistungsmesskopf

mit den Bestimmungen des Rates der Europäischen Union zur Angleichung der Rechtsvorschriften der Mitgliedstaaten

- betreffend elektrische Betriebsmittel zur Verwendung innerhalb bestimmter Spannungsgrenzen (73/23/EWG geändert durch 93/68/EWG)
- über die elektromagnetische Verträglichkeit (89/336/EWG geändert durch 91/263/EWG, 92/31/EWG, 93/68/EWG)

übereinstimmt.

Die Übereinstimmung wird nachgewiesen durch die Einhaltung folgender Normen:

EN61010-1 : 1993 + A2 : 1995
EN55011 : 1998 + A1 : 1999
EN61326 : 1997 + A1 : 1998 + A2 : 2001

Bei der Beurteilung der elektromagnetischen Verträglichkeit wurden die Störaussendungsgrenzwerte für Geräte der Klasse B sowie die Störfestigkeit für Betrieb in industriellen Bereichen zugrunde gelegt.

Anbringung des CE-Zeichens ab: 2002

ROHDE & SCHWARZ GmbH & Co. KG
Mühlldorfstr. 15, D-81671 München

München, den 27. Juni 2002

Zentrales Qualitätsmanagement FS-QZ / Becker

Inhaltsübersicht

1	Inbetriebnahme	1.1
	Auspacken	1.1
	Anschließen	1.1
	Betrieb am Grundgerät R&S NRP	1.2
	Anschließen des Messkopfes an das Grundgerät R&S NRP	1.2
	Anschließen des Messkopfes an das Messobjekt	1.2
	Betrieb an einem PC	1.2
	Hardware- und Software-Voraussetzungen	1.2
	Betrieb über aktiven USB-Adapter R&S NRP-Z3	1.4
	Betrieb über passiven USB-Adapter R&S NRP-Z4	1.5
	Anschließen des Messkopfes an das Messobjekt	1.5

Bilder

Bild 1-1	Gesamt verfügbaren Strom eines USB-Anschlusses anzeigen.....	1.3
Bild 1-2	Konfiguration mit dem aktiven USB-Adapter R&S NRP-Z3	1.4
Bild 1-3	Wechseln des Primäradapters	1.4
Bild 1-4	Konfiguration mit dem passiven USB-Adapter R&S NRP-Z4	1.5

1 Inbetriebnahme



Beachten Sie genau die folgenden Hinweise, um Schäden am Gerät auszuschließen, insbesondere wenn Sie den Messkopf zum ersten Mal in Betrieb nehmen.

Auspacken

Entnehmen Sie den Messkopf der Verpackung und überprüfen Sie, ob die Lieferung vollständig ist. Untersuchen Sie alle Teile sorgfältig auf Beschädigungen. Wenn Sie irgendwelche Beschädigungen finden, dann verständigen Sie bitte unverzüglich das zuständige Transportunternehmen und heben Sie alle Verpackungsteile zur Wahrung Ihrer Ansprüche auf.

Die Originalverpackung sollten Sie auch für den späteren Transport und Versand des Messkopfes benutzen.



Der Messkopf enthält Bauelemente, die durch elektrostatische Entladungen zerstört werden können. Vermeiden Sie es deshalb, den Innenleiter des HF-Anschluss-Steckers zu berühren, und öffnen Sie den Messkopf nicht.

Anschließen



Um elektromagnetische Störungen zu vermeiden, darf der Messkopf nur geschlossen betrieben werden. Es dürfen nur geeignete, abgeschirmte Kabel verwendet werden.

Überschreiten Sie niemals die maximal zulässige HF-Leistung. Schon kurzzeitige Überlastungen können zur Zerstörung des Messkopfes führen.

In vielen Fällen wird es genügen, den HF-Anschluss-Stecker handfest anzuziehen. Bei hohen Anforderungen an die Messgenauigkeit ist es notwendig, den HF-Anschluss-Stecker mit einem Drehmomentschlüssel anzuziehen, dessen nominales Drehmoment 1,36 Nm (12" lbs) betragen sollte.

Betrieb am Grundgerät R&S NRP

Anschließen des Messkopfes an das Grundgerät R&S NRP

Der Messkopf kann an das Grundgerät R&S NRP im laufenden Betrieb angeschlossen werden. Der Schnittstellenstecker muss mit der roten Farbmarkierung nach oben in eine der Messkopfbuchsen des Grundgerätes R&S NRP eingesteckt werden. Nach dem Anschließen wird der Messkopf vom Grundgerät R&S NRP erkannt und initialisiert.

Anschließen des Messkopfes an das Messobjekt

Der Messkopf R&S NRP-Z11/-Z21 verfügt über einen N-Stecker und kann damit an alle üblichen N-Buchsen angeschlossen werden. Bringen Sie unter leichtem Druck und ohne zu verkanten den N-Stecker mit dem Gegenstück zusammen und drehen Sie die Überwurfmutter des N-Steckers fest (Rechtsgewinde).

Betrieb an einem PC

Hardware- und Software-Voraussetzungen

Für einen Betrieb des Messkopfes an einem PC über Schnittstellenadapter müssen folgende Voraussetzungen erfüllt sein:

- Der PC muss über einen USB-Anschluss verfügen.
- Das PC-Betriebssystem muss den USB unterstützen. Dies ist der Fall für Windows™ 98, Windows™ ME, Windows™ 2000, Windows™ XP oder aktuellere Versionen des Windows™-Betriebssystems.
- Die in der mitgelieferten Software NRP Toolkit enthaltenen USB-Gerätetreiber müssen installiert sein.

Sind diese Voraussetzungen erfüllt, kann der Messkopf mit einem geeigneten Anwendungsprogramm wie dem im NRP Toolkit enthaltenen Programm NrpFlashup (enthält die Module Power Viewer, USB Terminal, Firmware Update und Update S-Parameters) betrieben werden.

Das Installationsprogramm für das NRP Toolkit startet automatisch beim Einlegen der im Lieferumfang befindlichen CD-ROM. Das weitere Vorgehen ist selbsterklärend.

Der Messkopf kann auf zwei Arten mit Strom versorgt werden:

- *self-powered* von einem separaten Netzteil über den aktiven USB-Adapter R&S NRP-Z3,
- *bus-powered* vom PC oder einem USB-Hub mit eigener Stromversorgung (*self-powered hub*) über den aktiven USB-Adapter R&S NRP-Z3 oder den passiven USB-Adapter R&S NRP-Z4.

Da der Messkopf als *high-power device* klassifiziert ist, ist nicht gewährleistet, dass er von jedem Laptop oder Notebook im *bus-powered*-Betrieb mit Strom versorgt werden kann. Um sicherzugehen, sollten Sie vorher die an den USB-Anschlüssen verfügbare Stromstärke ermitteln:



- im Windows™-Startmenü den Menüpunkt **Einstellungen – Systemsteuerung**, wählen
- **System** -Icon wählen
- die Registerkarte **Hardware** wählen
- durch Mausklick auf den gleichnamigen Button den Geräte-Manager starten
- Eintrag **USB-Controller** öffnen (alle USB-Controller, Hubs und USB-Geräte sind hier aufgeführt)
- auf **USB-Root-Hub** (in einigen deutschen Windows-Versionen wird auch die Bezeichnung **USB-Stamm-Hub** benutzt) doppelklicken oder im Kontextmenü (über die rechte Maustaste zu erreichen) **Eigenschaften** wählen
- die Registerkarte **Strom** (Bild 1-1) wählen. Ist der Hub selbstversorgend, und der unter **Hubinformationen** aufgeführte verfügbare Strom beträgt 500 mA pro Anschluss, so können high power devices angeschlossen werden.

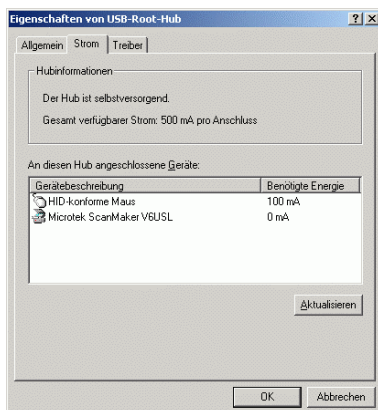


Bild 1-1 Gesamt verfügbaren Strom eines USB-Anschlusses anzeigen

Fragen Sie im Zweifelsfall den Hersteller, ob der USB-Anschluss Ihres Laptops oder Notebooks den Betrieb von *high power devices* zulässt.

Betrieb über aktiven USB-Adapter R&S NRP-Z3

Bild 1-2 zeigt die Konfiguration mit dem aktiven USB-Adapter R&S NRP-Z3, der es auch ermöglicht, ein Triggersignal für die Modi *Timeslot* und *Scope* zuzuführen. Dabei ist es unkritisch, in welcher Reihenfolge die Kabelverbindungen hergestellt werden.

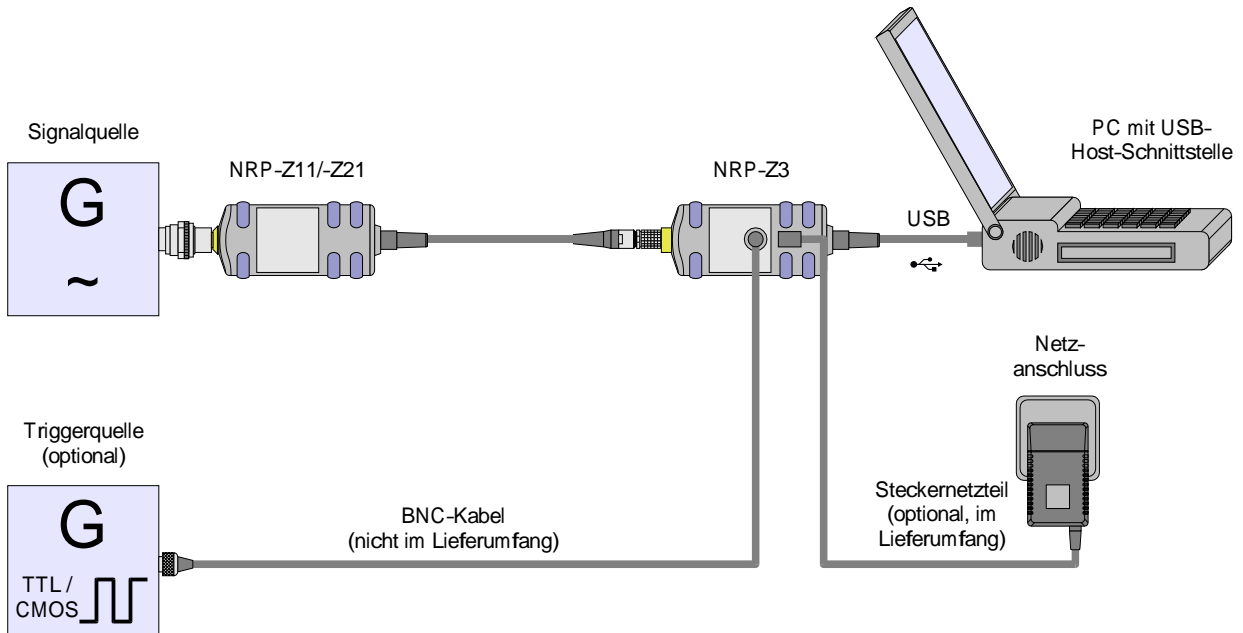


Bild 1-2 Konfiguration mit dem aktiven USB-Adapter R&S NRP-Z3

Das Steckernetzteil für den R&S NRP-Z3 kann an einer Einphasen-Wechselspannung mit einer Nennspannung von 100 V bis 240 V und einer Nennfrequenz von 50 Hz bis 60 Hz betrieben werden. Es stellt sich automatisch auf die Höhe der Netzspannung ein. Ein manuelles Umschalten ist nicht notwendig.

Dem Steckernetzteil liegen vier Primäradapter (für Europa, U.K., USA und Australien) bei, um den Anschluss an die entsprechenden Netzsteckdosen zu ermöglichen. Um den Primäradapter zu wechseln, werden keinerlei Werkzeuge benötigt. Er wird von Hand herausgezogen, und ein anderer Adapter wird eingeschoben, bis er einrastet (Bild 1-3).

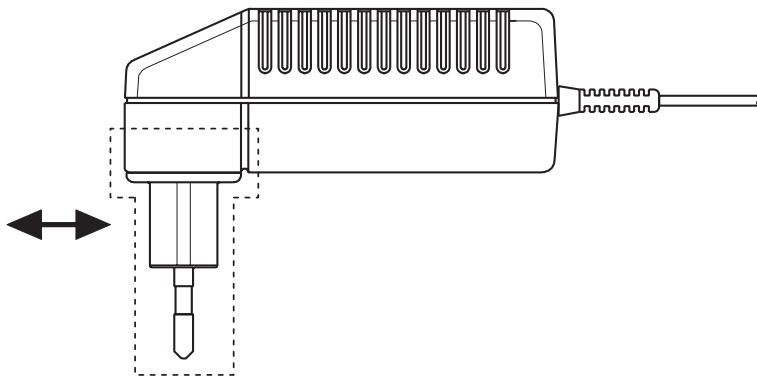


Bild 1-3 Wechseln des Primäradapters

Das Steckernetzteil ist kurzschlussfest und zusätzlich intern abgesichert. Ein Sicherungswechsel oder Öffnen ist nicht möglich.



Das Steckernetzteil ist nur zum Gebrauch in Innenräumen bestimmt.

Beachten Sie den Temperaturbereich von 0°C bis 50°C.

Lassen Sie ein durch Kondenswasser feucht gewordenes Steckernetzteil trocknen, bevor Sie es an die Netzspannung anschließen.

Betrieb über passiven USB-Adapter R&S NRP-Z4

In Bild 1-4 ist der Messaufbau zusammengestellt. Dabei ist es unkritisch, in welcher Reihenfolge die Kabelverbindungen hergestellt werden.

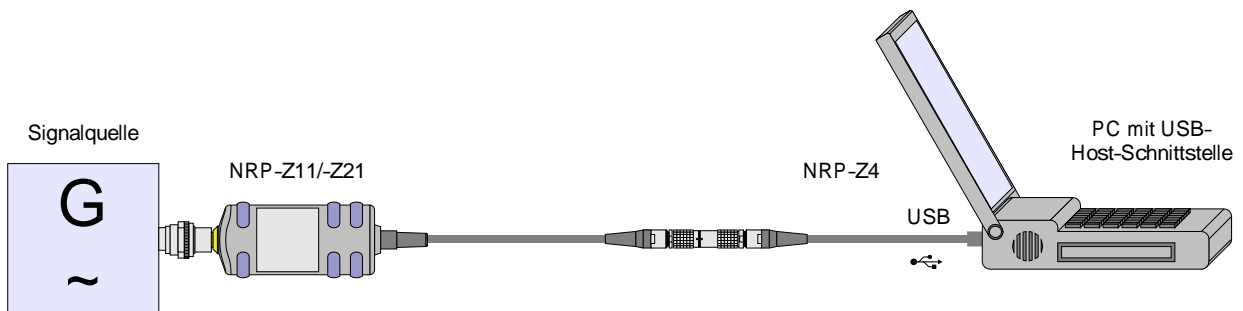


Bild 1-4 Konfiguration mit dem passiven USB-Adapter R&S NRP-Z4

Anschließen des Messkopfes an das Messobjekt

Zum Anschließen des Messkopfes an das Messobjekt siehe Abschnitt „Betrieb am Grundgerät R&S NRP“.

Inhaltsübersicht

2	Virtueller Leistungsmesser	2.1
	Übersicht	2.1
	Menüs.....	2.3

Bilder

Bild 2-1 Virtuelles Messgerät **Power Viewer** 2.1

Tabellen

Tabelle 2-1 Tasten des virtuellen Leistungsmessers 2.2
Tabelle 2-2 Eingabefelder des virtuellen Leistungsmessers 2.2

2 Virtueller Leistungsmesser

Auf der dem Messkopf beiliegenden CD-ROM befindet sich das Programm **NrpFlashup**, mit dem sich der Messkopf bei Betrieb an einem PC unter Windows™ steuern lässt. Es besteht aus mehreren Programm-Modulen, die zentral über den Windows™-Startmenü-Eintrag **NRP Toolkit** gestartet werden können.

Dieser Abschnitt beschreibt das Programm-Modul **Power Viewer**. Dabei handelt es sich um einen virtuellen Leistungsmesser, der den Funktionsumfang des Messkopfes allerdings nur zu einem kleinen Teil ausnutzt. Dafür ist es schon nach sehr kurzer Einarbeitungszeit möglich, die mittlere Leistung von modulierten Signalen zu messen.

Die anderen in **NrpFlashup** enthaltenen Module werden in Abschnitt 3 des Betriebshandbuches (Module **Terminal** und **Update S-Parameters**) bzw. im Servicehandbuch (Modul **Firmware Update**) behandelt.

Übersicht

Starten Sie den virtuellen Leistungsmesser über den Startmenü-Eintrag **NRP Toolkit – Power Viewer**. Es erscheint das **Power Viewer**-Programmfenster (Bild 2-1).

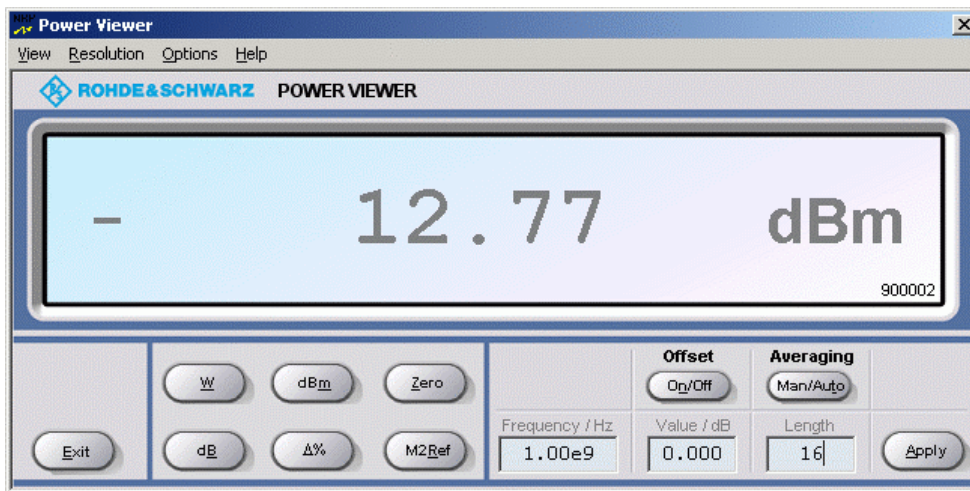


Bild 2-1 Virtuelles Messgerät **Power Viewer**

Den größten Teil des Programmfensters belegt das Messwert-Display. Hier werden Messwert, Einheit und zusätzliche Informationen zum Status des Messkopfes angezeigt. Rechts unten wird die Seriennummer des Messkopfes eingeblendet. Außerdem enthält das Programmfenster grafisch animierte Buttons und Eingabefelder (siehe Tabelle 2-1 und Tabelle 2-2).

Tabelle 2-1 Tasten des virtuellen Leistungsmessers

Button	Funktion	Tastenkombination
Exit	Beendet das Programm. Dabei werden die aktuellen Einstellungen gespeichert und beim nächsten Programmstart wieder hergestellt.	Alt + E
W	Schaltet die Anzeigeeinheit auf Watt.	Alt + W
dBm	Schaltet die Anzeigeeinheit auf dBm.	Alt + M
Zero	Löst einen Nullabgleich des Messkopfes aus.	Alt + Z
dB	Schaltet die Anzeigeeinheit auf Dezibel. Dabei wird das Verhältnis des Messwertes zum Referenzwert angezeigt.	Alt + B
Δ%	Schaltet die Anzeigeeinheit auf Prozent. Dabei wird die relative Abweichung des Messwertes vom Referenzwert angezeigt.	Alt + %
M2Ref	Definiert den aktuellen Messwert als Referenzwert für die relativen Anzeigeeinheiten Dezibel und Prozent.	Alt + R
Offset On/Off	Schaltet die Offsetkorrektur des Messkopfes ein oder aus. Bei ausgeschalteter Offsetkorrektur ist das Eingabefeld Offset/dB grau hinterlegt.	Alt + N
Averaging Man/Auto	Schaltet die automatische Bestimmung des Averaging-Faktors (Auto-Averaging) ein oder aus. Bei eingeschaltetem Auto-Averaging ist das Eingabefeld Length grau hinterlegt, dabei wird der aktuell ermittelte Averaging-Faktor angezeigt.	Alt + T
Apply	Übernimmt geänderte Zahlenwerte in den Eingabefeldern Frequency/Hz , Value/dB und Length und überträgt sie an den Messkopf.	Alt + A oder Eingabetaste

Tabelle 2-2 Eingabefelder des virtuellen Leistungsmessers

Eingabefeld	Funktion
Frequency/Hz	Frequenz des HF-Trägers in Hertz.
Value/dB	Dämpfung eines dem Messkopf vorgeschalteten Vierpols in dB. Hier sind Werte von -100 bis 100 zulässig. Die Offsetkorrektur muss mit der Taste Offset On/Off aktiviert worden sein, damit dieses Eingabefeld editiert werden kann.
Length	Länge des Averaging-Filters (= Averaging-Faktor). Hier sind Werte von 1 bis 65536 zulässig. Das Averaging muss mit der Taste Averaging Man/Auto auf manuelles Averaging umgeschaltet worden sein, damit dieses Eingabefeld editiert werden kann.

Bei der Eingabe in Eingabefelder kann auch das wissenschaftliche Zahlenformat verwendet werden. Unzulässige Werte werden mit einer Fehlermeldung quittiert. Damit ein geänderter Zahlenwert an den Messkopf übermittelt wird, muss die Eingabe unbedingt mit dem Button **Apply** oder der Eingabetaste abgeschlossen werden!

Menüs

Auf weniger häufig benötigte Funktionen kann über die Menüleiste zugegriffen werden.

View

Display Refresh Rate

Öffnet einen Dialog zur Anpassung der Display-Aktualisierungsrate. Eingegeben wird die Zeit in Millisekunden zwischen zwei Display-Aktualisierungen. Die Voreinstellung ist 200 ms.



Colours

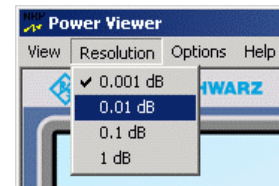
Result
Unit
Edit
Button

Öffnet einen Dialog zur Auswahl der Vordergrundfarbe für

- das Messergebnis,
- die Einheit,
- den Text in den Zahlenfeldern bzw.
- die Tastenbeschriftung.

Resolution

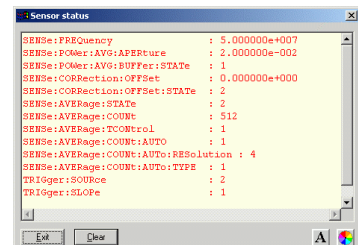
Erlaubt es, die gewünschte Auflösung der Messwertdarstellung einzustellen. Eine höhere Auflösung führt bei aktiviertem Auto-Averaging zu einem größeren Averaging-Faktor und damit zu einer längeren Einschwingzeit des Messergebnisses.



Options

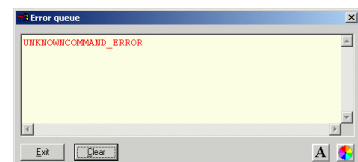
Read Sensor Status ...

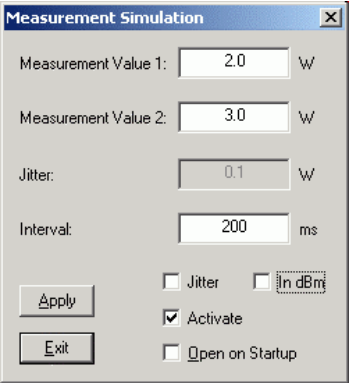
Liest den aktuellen Status des Messkopfes aus. Es wird eine Parameterliste ausgegeben.



Read Error Queue ...

Liest die Fehlerqueue aus. Alle seit dem letzten Aufruf aufgetretenen Fehlermeldungen werden zeilenweise ausgegeben. Sind Fehler aufgetreten, dann wird durch ein Häkchen vor diesem Menüeintrag darauf hingewiesen.



Simulation ...	<p>Erlaubt es, die Funktionalität des virtuellen Leistungsmessers auch ohne angeschlossenen Messkopf auszuprobieren. Die Anzeige wechselt zwischen Measurement Value 1 u. Measurement Value 2 im Abstand Interval hin und her. Mit Hilfe der Checkbox Activate wird die Simulation sofort aktiviert.</p>	
Reset Sensor	<p>Initialisiert den Messkopf. Dabei bleibt ein vorher erfolgter Nullabgleich erhalten.</p>	
Help	<p>Contents Öffnet das Inhaltsverzeichnis zur Online-Hilfe.</p> <p>About Zeigt u. a. Informationen zur verwendeten Programmversion an.</p>	

Inhaltsübersicht

3	Manuelle Bedienung	3.1
	Programmmodul "Terminal"	3.1
	Wichtigste Bedienelemente.....	3.1
	Menüs.....	3.3
	Programmmodul "Firmware Update"	3.6
	Programmmodul "Update S-Parameters"	3.6
	Grundlagen	3.6
	Vorgehensweise.....	3.9

Bilder

Bild 3-1 Senden von Befehlen über Eingabefeld **Input** 3.1
Bild 3-2 Senden von Befehlen über Command Files 3.2
Bild 3-3 Dialogfenster für das Laden einer S-Parameter-Tabelle 3.9
Bild 3-4 Dialogfenster zum Laden einer Kalibrierdatensatz-Sicherungskopie 3.10

Tabellen

Tabelle 3-1 Beschreibung der dem Eingabefeld **Input** zugeordneten Buttons 3.2
Tabelle 3-2 Beschreibung der dem Listenfeld **Command File** zugeordneten Buttons 3.2
Tabelle 3-3 Beschreibung der dem Ausgabefeld **Output** zugeordneten Buttons 3.3
Tabelle 3-4 Unsicherheiten des S-Parameter-Messplatzes (Beispiel) 3.7
Tabelle 3-5 Interpolierte Unsicherheiten der Frequenzstützstellen der S-Parameter (Beispiel) 3.7

3 Manuelle Bedienung

Im vorigen Abschnitt wurde auf das im Lieferumfang enthaltene Programmmodul "Power Viewer" eingegangen, womit sich die wohl häufigste Funktion eines Leistungsmessers – das Messen der mittleren Leistung eines nahezu beliebig modulierten HF-Signals – auf einfache Weise bewerkstelligen lässt. Im Lieferumfang befinden sich weitere Programmmodule, die sich über das Startmenü starten lassen. Im Startmenü finden sich die folgenden Einträge:

- **Power Viewer:** Virtueller Leistungsmesser. Die Funktion dieses Moduls ist in Abschnitt 2 ausführlich beschrieben.
- **Terminal:** Programmmodul zum Senden von Befehlen und Befehlsfolgen an den Messkopf und zum Anzeigen der vom Messkopf gelieferten Messwerte, Statusmeldungen und sonstigen Daten.
- **Firmware Update:** Programmmodul zum Update der Messkopf-Firmware.
- **Update S-Parameters:** Programmmodul zum Laden einer S-Parameter-Tabelle in den Messkopf.

Programmmodul "Terminal"

Wichtigste Bedienelemente

Das USB-Terminal erlaubt es, Befehle und Befehlsfolgen an den Messkopf zu senden, und zwar auf zweierlei Weise:

- Die Befehle werden in das Eingabefeld **Input** eingegeben (Bild 3-1). Mehrere aufeinander folgende Befehle können zeilenweise untereinander stehen. In Tabelle 3-1 sind die dem Eingabefeld **Input** zugeordneten Buttons beschrieben.
- Die Befehle oder Befehlsfolgen werden in Befehlsdateien (*command files*) gespeichert. Diese Befehlsdateien lassen sich z. B. mit einem Texteditor erstellen und abspeichern. Danach kann beliebig oft darauf zurück gegriffen werden (Bild 3-2). In Tabelle 3-2 sind die dem Listenfeld **Command File** zugeordneten Buttons beschrieben.

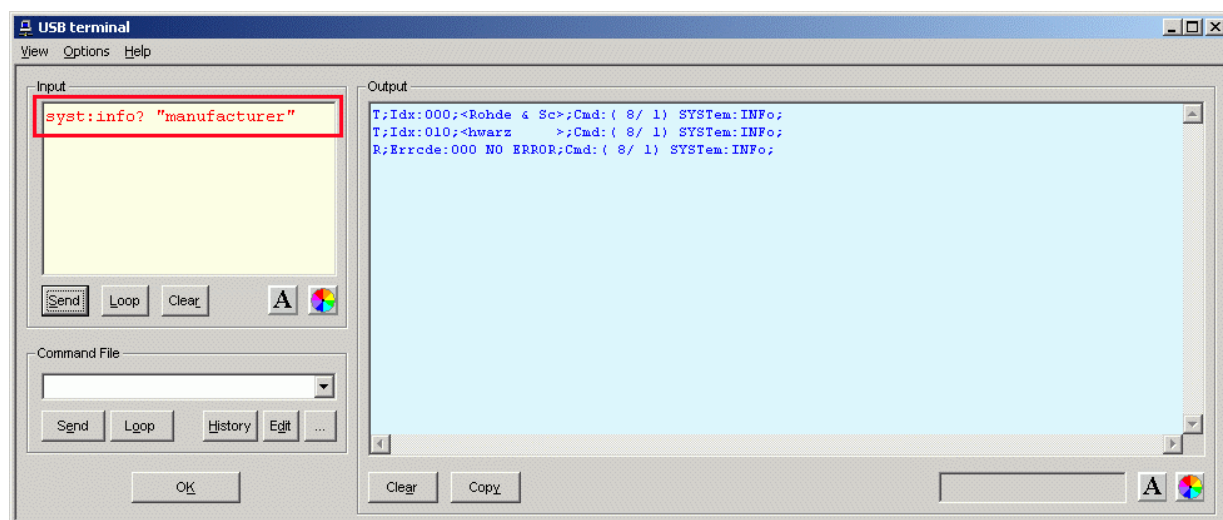


Bild 3-1 Senden von Befehlen über Eingabefeld **Input**

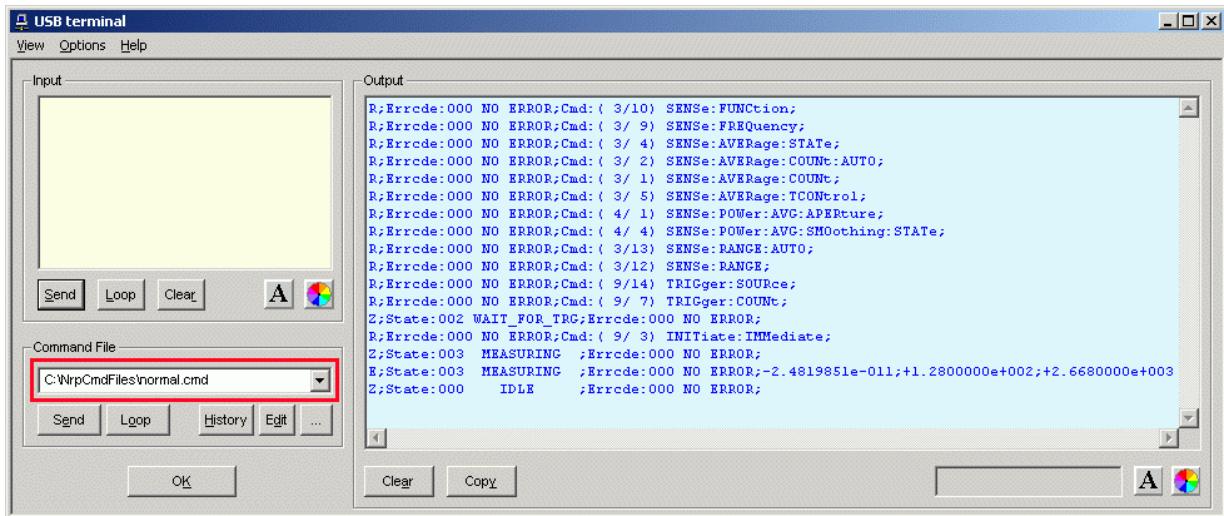


Bild 3-2 Senden von Befehlen über Command Files

Tabelle 3-1 Beschreibung der dem Eingabefeld **Input** zugeordneten Buttons

Button	Funktion	Tastenkombination
Send	Sendet den Inhalt des Eingabefeldes Input an den Messkopf.	Alt + S
Loop	Mit Loop wird der Befehl oder die Befehlsfolge zyklisch gesendet. Durch erneutes Betätigen wird das zyklische Senden beendet. Die Wiederholfrequenz wird über ein Dialogfenster, welches mit View - Loop ... geöffnet wird, eingestellt.	Alt + L
Clear	Löscht den Inhalt des Input -Textfeldes.	Alt + R
Schriftart-Button	Öffnet ein Dialogfenster zur Auswahl der Schriftart im Eingabefeld Input .	
Farbe-Button	Öffnet ein Dialogfenster zur Auswahl der Hintergrundfarbe des Eingabefeldes Input .	

Tabelle 3-2 Beschreibung der dem Listenfeld **Command File** zugeordneten Buttons

Button	Funktion	Tastenkombination
Send	Sendet den Inhalt der Befehlsdatei an den Messkopf.	Alt + E
Loop	Mit Loop wird der Befehl oder die Befehlsfolge zyklisch gesendet. Durch erneutes Betätigen wird das zyklische Senden beendet. Die Wiederholfrequenz wird über ein Dialogfenster, welches mit View - Loop ... geöffnet wird, eingestellt.	Alt + O
History	Öffnet ein Fenster zum Editieren der Befehlsdateinamen im Command File -Listenfeld.	Alt + H
Edit	Öffnet die ausgewählte Befehlsdatei im Windows™-Texteditor.	Alt + D
...	Öffnet einen Datei-Öffnen-Dialog zur Auswahl einer Befehlsdatei.	

Steht am Anfang einer Befehlszeile ein Tabulator, Leerzeichen oder Sonderzeichen, so wird diese Zeile als Kommentar behandelt und nicht an den Messkopf gesendet.

Die vom Messkopf zurückgelieferten Messwerte, Parameter und Statusinformationen werden im Ausgabefeld **Output** angezeigt.

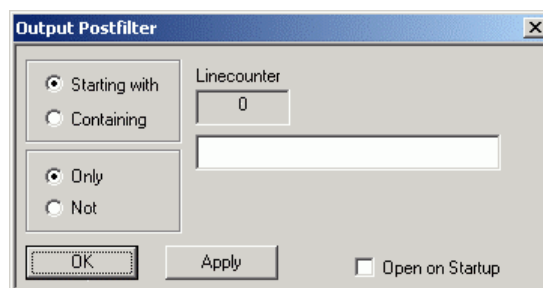
Tabelle 3-3 Beschreibung der dem Ausgabefeld **Output** zugeordneten Buttons

Button	Funktion	Tastenkombination
Clear	Löscht den Inhalt des Output-Textfeldes.	Alt + A
Copy	Kopiert den gesamten Inhalt des Output-Textfeldes in die Zwischenablage. (Es ist auch möglich, mit dem Maus-Cursor einen Teil der Ausgaben im Output-Fenster zu markieren und über Strg + C oder Betätigen der rechten Maustaste und anschließende Wahl des Menüpunktes Kopieren im sich öffnenden Kontextmenü in die Zwischenablage zu kopieren.)	Alt + Y
Schriftart-Button	Öffnet ein Dialogfenster zur Auswahl der Schriftart im Output-Textfeld.	
Farbe-Button	Öffnet ein Dialogfenster zur Auswahl der Hintergrundfarbe des Output-Textfeldes.	

Das USB-Terminal wird durch Klicken auf den **OK**-Button geschlossen.

Menüs

View Post Filter ... Öffnet den Dialog **Output postfilter**. Damit ist es möglich, die im Empfangspuffer gespeicherten Zeilen nach verschiedenen Kriterien zu filtern.



Filterkriterien:

Only + Starting with: Nur die Zeilen, die mit der eingegebenen Zeichenkette beginnen ...

Not + Starting with: Nur die Zeilen, die nicht mit der eingegebenen Zeichenkette beginnen ...

Only + Containing: Nur die Zeilen, die die eingegebene Zeichenkette enthalten ...

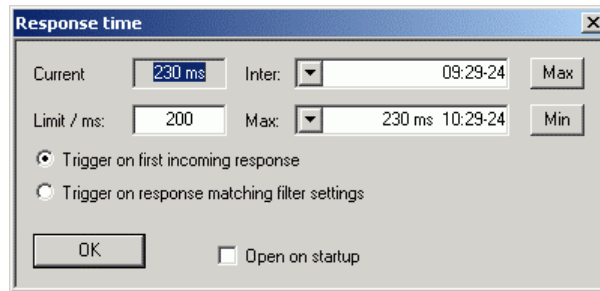
Not + Containing: Nur die Zeilen, die die eingegebene Zeichenkette nicht enthalten ...

... werden angezeigt. Die Zeilen, die das Filterkriterium nicht erfüllen, werden nicht gelöscht, sondern nur ausgeblendet.

Mit **Apply** wird der Filtervorgang gestartet. Im Feld **Linecounter** steht daraufhin die Anzahl der Zeilen, die das Filterkriterium erfüllt haben. Wählt man **Open on startup**, so wird der Dialog **Output postfilter** automatisch beim Öffnen des Terminals angezeigt. Mit **OK** wird das Dialogfenster geschlossen.

Response Time ...

Öffnet den Dialog **Response time**. Damit ist es möglich, die Antwortzeiten eines Messkopfes zu bestimmen.



Current zeigt die Zeit, die vom Senden des letzten Befehls bis zum Eintreffen der Befehlsbestätigung vom Messkopf vergangen ist. Durch Klicken auf den **Max**-Button werden die Antwortzeiten aufgezeichnet, die den Grenzwert im Feld **limit / ms** überschreiten. Durch Klicken auf den **Min**-Button werden die Antwortzeiten aufgezeichnet, die den Grenzwert im Feld **limit / ms** einhalten. Mit **Trigger on first incoming response** endet die Zeitmessung mit dem Eintreffen der ersten Antwort nach Absenden des Befehls. Mit **Trigger on response matching filter settings** endet die Zeitmessung mit dem Eintreffen einer Antwort, die das Filterkriterium im Dialog **Output postfilter** erfüllt.

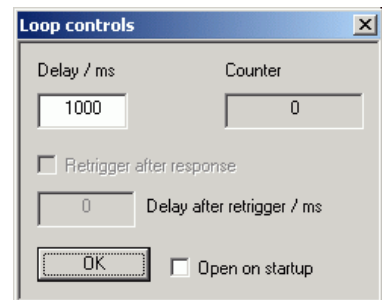
Wählt man **Open on startup**, so wird der Dialog **Response Time** automatisch beim Öffnen des Terminals angezeigt. Mit **OK** wird das Dialogfenster geschlossen.

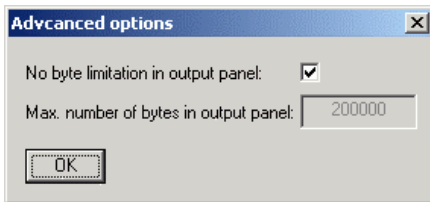
Loop ...

Öffnet den Dialog **Loop controls**. Damit ist es möglich, das zyklische Senden von Befehlen und Befehlsfolgen zu steuern.

Im Feld **Delay / ms** wird das Zeitintervall für das zyklische Senden in Millisekunden spezifiziert.

Das Feld **Counter** enthält die Anzahl der abgeschlossenen Sendezyklen. Wählt man **Open on startup**, so wird der Dialog **Response time** automatisch beim Öffnen des Terminals angezeigt. Mit **OK** wird das Dialogfenster geschlossen.



Options	Protocol Mode	In diesem Modus wird jeder Antwortblock mit einem Zeitstempel versehen.	
	Hex Mode	In diesem Modus werden die vom Messkopf kommenden Antwortblöcke im hexadezimalen Format angezeigt.	
	Auto Delete	Wenn diese Option aktiviert ist, wird das Output -Textfeld immer dann automatisch gelöscht, wenn der Send -Button betätigt wird.	
	Auto Scroll	Wenn diese Option aktiviert ist, werden die älteren Inhalte des Output -Textfeldes automatisch nach oben aus dem sichtbaren Bereich heraus verschoben, wenn Platz für neue Ausgaben benötigt wird.	
	LF at EOT	Wenn diese Option aktiviert ist, wird an jeden vom Messkopf kommenden Antwortblock ein Zeilenumbruch angehängt.	
	Delete on Start	Wenn diese Option aktiviert ist, wird das Output -Textfeld beim Start des Programmmoduls "Terminal" automatisch gelöscht.	
	Send as Hex	Wenn diese Option aktiviert ist, wird der Text im Input-Textfeld als Folge hexadezimaler Zeichen interpretiert.	
	Advanced ...	Öffnet ein Dialogfenster zum Einstellen der Puffergröße für das Ausgabefeld Output .	
			
Help	Contents	Öffnet das Inhaltsverzeichnis zur Online-Hilfe.	
	About	Zeigt u. a. Informationen zur verwendeten Programmversion an.	

Programmmodul "Firmware Update"

Das Programmmodul für das Durchführen von Firmware-Updates ist im Service-Handbuch ausführlich beschrieben.

Programmmodul "Update S-Parameters"

Grundlagen

Der Messkopf R&S NRP-Z11/Z21 bietet die Möglichkeit, den Einfluss eines beliebigen vorgeschalteten Zweitores auf den Messwert rechnerisch zu korrigieren. Die Voraussetzung dafür ist, dass im interessierenden Frequenzbereich ein vollständiger Satz der komplexen S-Parameter des Zweitores vorliegt. Im Kalibrierdatensatz des R&S NRP-Z11/-Z21 ist zu diesem Zweck eine S-Parameter-Tabelle angelegt, die bis zu 1000 Frequenzstützstellen enthalten kann. Für jede dieser Stützstellen können Realteil, Imaginärteil und Unsicherheit der S-Parameter s_{11} , s_{12} , s_{21} und s_{22} gespeichert werden. Da die Frequenzstützstellen der S-Parameter-Tabelle unabhängig von den Kalibrierfrequenzen sind, hat man die Möglichkeit, die Stützstellen so zu legen, dass der interessierende Frequenzbereich des Zweitores optimal abgedeckt wird. Zwischen den Stützstellen werden Real- und Imaginärteil linear interpoliert, während die größere Messunsicherheit der beiden angrenzenden Stützstellen für die Berechnung der Messunsicherheit des Messergebnisses zu Grunde gelegt wird. Unterhalb der ersten und oberhalb der letzten Stützstelle gelten die Werte der ersten bzw. letzten Stützstelle.

Zum Laden einer S-Parameter-Tabelle dient das Programm NrpFlashup (Menüpunkt **Update S-Parameters**). Um Kompatibilität zu einer Vielzahl von Netzwerkanalysatoren sicher zu stellen, kann NrpFlashup Messdatenfiles im S2P-Format verarbeiten. Unterstützt werden alle standardmäßig vorgesehen Frequenzeinheiten (Hz, kHz, MHz, GHz) und Darstellungsformate (Realteil-Imaginärteil, linearer Betrag und Phase, Betrag in dB und Phase). Die einzige Einschränkung besteht darin, dass als Bezugsimpedanz für die S-Parameter keine von 50 Ω verschiedenen Werte zulässig sind. Enthält das Messdatenfile zusätzlich Noise-Parameter, so werden diese nicht ausgewertet.

Das S2P-Messdatenfile ist folgendermaßen aufgebaut:

1. Kopfzeile (*option line*), mit folgendem Aufbau:

```
# [<Frequenzeinheit>] [<Parameter>] [<Format>] [<R n>]
```

Das Zeichen "#" kennzeichnet eindeutig die Kopfzeile.

<Frequenzeinheit> kann "Hz", "kHz", "MHz" oder "GHz" lauten. Ist keine Frequenzeinheit angegeben, wird implizit "GHz" angenommen.

<Parameter> muss, wenn angegeben, "S" für S-Parameter-Files lauten. Ist kein Parameter angegeben, wird implizit "S" angenommen.

<Format> kann "MA" (linearer Betrag und Phase in Grad), "DB" (Betrag in dB und Phase in Grad) oder "RI" (Realteil und Imaginärteil) lauten. Ist kein Format angegeben, wird implizit "MA" angenommen.

Es folgt optional der Buchstabe "R", gefolgt vom Wert der Bezugsimpedanz in Ω . Ist dieser Eintrag vorhanden, muss er "R 50" lauten. Ist er nicht vorhanden, wird implizit "R 50" angenommen.

Zusammengefasst muss die Kopfzeile also folgendermaßen aufgebaut sein:

```
# [HZ | KHZ | MHZ | GHZ] [S] [MA | DB | RI] [R 50]
```

2. Frequenzstützstellen, aufsteigend geordnet nach Frequenz, mit folgendem Aufbau:

$$f_i \quad s_{11}(f_i) \quad s_{21}(f_i) \quad s_{12}(f_i) \quad s_{22}(f_i).$$

Dabei steht $s_{jk}(f_i)$ für das *option line* spezifizierte Darstellungsformat:

$$|s_{jk}(f_i)| \quad \arg s_{jk}(f_i) \quad \text{(Darstellungsformat linearer Betrag und Phase in Grad) oder}$$

$$20 \cdot \lg |s_{jk}(f_i)| \quad \arg s_{jk}(f_i) \quad \text{(Darstellungsformat Betrag in dB und Phase in Grad).}$$

$$\operatorname{Re} [s_{jk}(f_i)] \quad \operatorname{Im} [s_{jk}(f_i)] \quad \text{(Darstellungsformat Realteil-Imaginärteil),}$$

3. Kommentare:

Jede Zeile, die mit einem Ausrufungszeichen (!) beginnt, wird als Kommentarzeile interpretiert.

Zur Charakterisierung der Messunsicherheit des S-Parameter-Messplatzes kann optional ein weiteres Datenfile angelegt werden. Ohne dieses Unsicherheits-Datenfile ist eine korrekte Messunsicherheitsberechnung im Messkopf nicht möglich. Das Unsicherheits-Datenfile ist ähnlich aufgebaut wie das S2P-Messdatenfile, jedoch enthält die *option line* für <Parameter> den Kennbuchstaben "U", z. B. lautet bei Frequenzangaben in Hz die *option line*: # Hz U

Die Frequenzstützstellen müssen nicht mit denen des S2P-Messdatenfiles identisch sein. In den meisten Fällen werden einige wenige Einträge genügen, um die Messunsicherheit des S-Parameter-Messplatzes zu charakterisieren. Die Unsicherheit eines S-Parameters wird dann so groß gewählt wie an den benachbarten Frequenzstützstellen des Unsicherheits-Datenfiles. Bei unterschiedlichen Werten wird der größere gewählt. Dies soll durch ein Beispiel erläutert werden:

Tabelle 3-4 Unsicherheiten des S-Parameter-Messplatzes (Beispiel)

f in GHz	unc [$s_{jk}(f)$]
0,1	0,01
1,0	0,01
1,1	0,005
10,0	0,005
10,1	0,01
40,0	0,01

Tabelle 3-5 Interpolierte Unsicherheiten der Frequenzstützstellen der S-Parameter (Beispiel)

f in GHz	unc [$s_{jk}(f)$]
0,9	0,01
0,95	0,01
1,0	0,01
1,05	0,01
1,1	0,005
1,15	0,005
1,2	0,005

Für die Frequenz 1,05 GHz wurde die größere Unsicherheit der beiden angrenzenden Stützstellen 1,0 GHz und 1,1 GHz in die S-Parameter-Tabelle übertragen. Wenn für alle Frequenzen über 1,0 GHz eine Unsicherheit von 0,005 gewünscht würde, dann müsste im Unsicherheits-Datenfile die erste Stützstelle über 1,0 GHz auf z. B. 1,000001 GHz gelegt werden.

Das Unsicherheits-Datenfile ist folgendermaßen aufgebaut:

1. Kopfzeile (*option line*), mit folgendem Aufbau:

[<Frequenzeinheit>] <Parameter> [<Format>] [<R n>]

Das Zeichen "#" kennzeichnet eindeutig die Kopfzeile.

<Frequenzeinheit> kann "Hz", "kHz", "MHz" oder "GHz" lauten. Ist keine Frequenzeinheit angegeben, wird implizit "GHz" angenommen.

<Parameter> muss bei Unsicherheits-Datenfiles "U" lauten. Ist kein Parameter angegeben, wird implizit "S" angenommen, was zu einer Fehlermeldung führt.

<Format> wird bei Unsicherheits-Messdatenfiles ignoriert und kann daher beliebig lauten.

Es folgt optional der Buchstabe "R", gefolgt vom Wert der Bezugsimpedanz in Ω . Ist dieser Eintrag vorhanden, muss er "R 50" lauten. Ist er nicht vorhanden, wird implizit "R 50" angenommen.

Zusammengefasst muss die Kopfzeile also folgendermaßen aufgebaut sein:

[HZ | KHZ | MHZ | GHZ] U [MA | DB | RI] [R 50]

2. Frequenzstützstellen, aufsteigend geordnet nach Frequenz, mit folgendem Aufbau:

f_i unc $[s_{11}(f_i)]$ unc $[s_{21}(f_i)]$ unc $[s_{12}(f_i)]$ unc $[s_{22}(f_i)]$.

Die Unsicherheiten der S-Parameter werden wie folgt übergeben:

- als erweiterte absolute Unsicherheiten ($k = 2$) für die Beträge der Anpassungsparameter s_{11} und s_{22} , also z. B. 0.015,
- als erweiterte Unsicherheiten ($k = 2$) in dB für die Beträge der Transmissionsparameter s_{21} und s_{12} , also z. B. 0.05.

3. Kommentare:

Jede Zeile, die mit einem Ausrufungszeichen (!) beginnt, wird als Kommentarzeile interpretiert.

Zwei zusätzliche Angaben, die beim Laden der S-Parameter gemacht werden müssen, sind die nominale untere und die obere Messgrenze der Messkopfe-Zweitor-Kombination, die bei aktivierter S-Parameter-Korrektur vom Messkopf bei SYSTem:INFO? gemeldet werden. Nicht immer ergeben sich diese Werte aus der unteren bzw. oberen Messgrenze des Messkopfes allein und der Dämpfung oder Verstärkung des vorgeschalteten Zweitors. Die obere Messgrenze der Messkopf-Zweitor-Kombination kann auch durch die maximale Belastbarkeit des Zweitors limitiert werden. Weiterhin kann die untere Messgrenze außer durch die Dämpfung auch durch das Eigenrauschen des Zweitors angehoben werden. Aus diesem Grunde erlaubt NrpFlashup die Eingabe dieser beiden Werte.



Die beim Laden der S-Parameter eingegebene nominale obere Messgrenze der Messkopf-Zweitor-Kombination sollte sorgfältig festgelegt werden, da möglicherweise automatisierte Messsysteme diese Angabe auswerten und ein falscher Wert zur Überlastung von Messkopf und/oder Zweitor führen könnte.

Vorgehensweise

Um eine S-Parameter-Tabelle in den Kalibrierdatensatz des Messkopfes zu laden, gehen Sie bitte folgendermaßen vor:

1. Schließen Sie den Messkopf an den USB-Port des PC an und starten Sie NrFlashup.
2. Wählen Sie den Menüpunkt **Update S-Parameters**. Dadurch wird das entsprechende Dialogfenster geöffnet (Bild 3-3).
3. Geben Sie unter **S-Parameter File** den Suchpfad und Dateinamen des S2P-Files, welches die S-Parameter enthält, ein. Betätigen Sie den Button **Browse ...**, um einen Datei-Öffnen-Dialog zu starten, mit dem das S2P-Messdatenfile bequem ausgewählt werden kann.
4. Geben Sie unter **Uncertainty File** den Suchpfad und Dateinamen des Messunsicherheits-Datenfiles, welches die Messunsicherheit des S-Parameter-Messplatzes enthält, ein. Betätigen Sie den Button **Browse ...**, um einen Datei-Öffnen-Dialog zu starten, mit dem das Messunsicherheits-Datenfile bequem ausgewählt werden kann.
5. Tragen Sie in die Felder **Lower Power Limit** und **Upper Power Limit** die nominale untere bzw. obere Messgrenze der Messkopf-Zweitor-Kombination in Watt ein.
6. Tragen Sie in das Feld **S-Parameter Device Mnemonic** einen Namen für den geladenen S-Parameter-Satz ein. Dieser Name kann später über den Befehl *SYSTEM:INFO? "SPD Mnemonic"* abgefragt werden und erscheint bei eingeschalteter S-Parameter-Korrektur im Display des R&S NRP-Grundgerätes.
7. Aktivieren Sie die Checkbox **S-Parameter Correction on by Default**, wenn bei Inbetriebnahme des Messkopfes der Schalter *SENSe:CORRection:TRANsmiission* automatisch auf *ON* gesetzt werden soll.
8. Betätigen Sie den Button **Start**, um den Ladevorgang zu starten. (Mit **OK** wird der Dialog verlassen, die eingestellten Parameter bleiben erhalten. Mit **Cancel** wird der Dialog verlassen, und alle Änderungen von Parametern werden verworfen.)

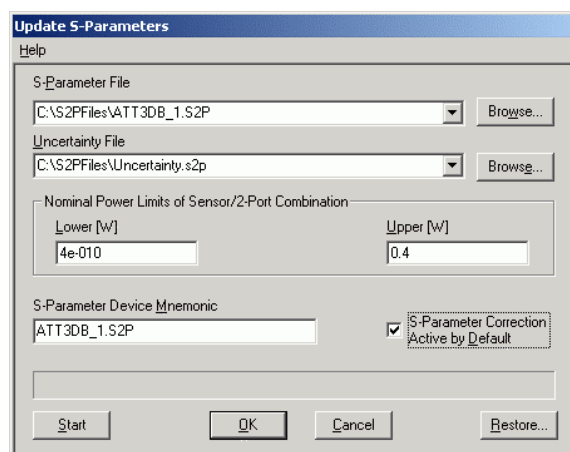


Bild 3-3 Dialogfenster für das Laden einer S-Parameter-Tabelle

Beim Ladevorgang wird der aktuelle Kalibrierdatensatz des Messkopfes überschrieben. Aus Sicherheitsgründen wird deshalb vor jedem Laden von S-Parametern automatisch eine Sicherungskopie des aktuellen Kalibrierdatensatzes angelegt. Die entsprechenden Dateien haben Namen in der Form "<Seriennummer>_<Datum><Uhrzeit>.bak", dabei ist <Seriennummer> die Seriennummer des Messkopfes, <Datum> das Datum des S-Parameter-Updates im Format jjmmtt und <Uhrzeit> die Uhrzeit des S-Parameter-Updates im Format hhmmss.



Speichern Sie die automatisch angelegten Sicherungskopien auf einem separaten Datenträger (z. B. Diskette, CD-R oder Netzlaufwerk) und versehen Sie sie gegebenenfalls mit einem aussagekräftigen Namen, um bei Bedarf darauf zugreifen zu können. Mit diesen Dateien ist es möglich, einen älteren Zustand des Messkopf-Kalibrierdatensatzes wieder herzustellen.

Um die Sicherungskopie eines Kalibrierdatensatzes wieder in den Messkopf zu laden, gehen Sie bitte folgendermaßen vor:

- Betätigen Sie den Button **Restore** Dadurch wird das **Restore S-Parameters**-Dialogfenster (Bild 3-4) geöffnet.
- Geben Sie in diesem Dialogfenster unter **Backup File** den Suchpfad und Dateinamen des Backup-Files ein. Betätigen Sie den Button **Browse** ..., um einen Datei-Öffnen-Dialog zu starten, mit dem das Backup-File bequem ausgewählt werden kann.
- Betätigen Sie den Button **OK**, um den Restore-Vorgang zu starten. (Mit **Cancel** wird das Dialogfenster verlassen, ohne dass ein Restore-Vorgang durchgeführt wird.)

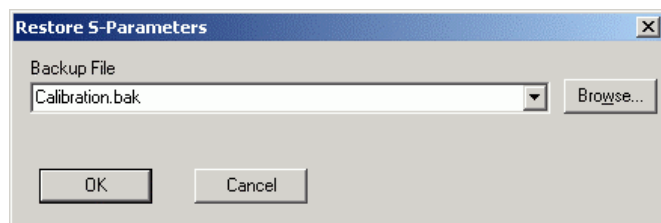


Bild 3-4 Dialogfenster zum Laden einer Kalibrierdatensatz-Sicherungskopie

Inhaltsübersicht

5 Fernbedienung – Grundlagen	5.1
USB-Treiberstack	5.1
Universelle USB-Treiber-Schnittstelle	5.2
Funktionen zum Einstellen der Messkopf-Betriebsart	5.4
NrpChannelAssignment	5.4
NrpStartApplicationMode	5.4
Funktionen zum Senden von Befehlen und Daten	5.5
NrpSendBinaryBlock	5.5
NrpSendCommand.....	5.5
Funktionen zum Abholen von Messergebnissen und Daten	5.6
NrpDataAvailable	5.6
NrpGetData	5.6
NrpGetBinaryResult	5.11
NrpGetBitfieldResult.....	5.11
NrpGetFloatArray	5.12
NrpGetFloatResult.....	5.12
NrpGetLongResult.....	5.12
NrpGetStringResult	5.13
Funktionen zur Status- und Fehlererkennung.....	5.14
NrpEmptyAllBuffers.....	5.14
NrpEmptyErrorQueue.....	5.14
NrpGetLastError	5.14
NrpGetErrorText.....	5.16
NrpIsAlive	5.16
NrpGetTriggerState.....	5.16
NrpGetTriggerStateText.....	5.17
Callback-Funktionen	5.17
Funktionen zum Setzen von Callback-Funktionen	5.19
NrpSetNotifyCallbackCommandAccepted	5.19
NrpSetNotifyCallbackDataAvailable.....	5.19
NrpSetNotifyCallbackDeviceChanged	5.19
NrpSetNotifyCallbackErrorOccured	5.19
NrpSetNotifyCallbackStateChanged.....	5.19
Windows-Messages	5.20

USB-spezifische Befehle.....	5.20
NrpOpenDriver	5.20
NrpCloseDriver	5.20
NrpClearDevice	5.20
NrpGetDeviceChangedDevName	5.20
NrpGetDeviceChangedMsgText	5.21
NrpGetDeviceID	5.22
NrpGetIDByLogicalName	5.22
NrpGetLogicalName	5.22
NrpGetManufacturerCode	5.22
NrpGetNrOfDevices	5.23
NrpGetProductID	5.23
NrpGetSerialNumber	5.23
NrpSDRGetDescriptor	5.23
NrpSelectDevice	5.24
Anwendungsbeispiele	5.25
Einfache Leistungsmittelwertmessung.....	5.25
Schnelle und mehrfache Leistungsmittelwertmessung.....	5.26
Messung eines einzelnen Bursts	5.26
Messung eines Frames mit mehreren Zeitfenstern (Timeslots)	5.27
Scopemodus zur Darstellung von periodischen Signalen	5.27
Scopemodus zur Darstellung eines einmaligen Vorgangs (Single-Shot-Messung)	5.28

Bilder

Bild 5-1 Funktionsweise der Dynamic Link Library NrpControl.dll..... 5.3

Tabellen

Tabelle 5-1 Teile des USB-Treiberstacks 5.1
Tabelle 5-2 Bedeutung der Datentypen 5.6
Tabelle 5-3 Bedeutung der Gruppen- und Parameternummern 5.7
Tabelle 5-4 Mögliche Messkopf-Fehlerzustände 5.14
Tabelle 5-5 Mögliche Messkopf-Triggerzustände 5.16
Tabelle 5-6 Beschreibung der Struktur *USB_DEVICECHANGED_HANDLER_RESULT*..... 5.21

5 Fernbedienung – Grundlagen

USB-Treiberstack

Bei der Installation des NRP-Toolkit wird u. a. ein USB-Treiberstack für die Messköpfe R&S NRP-Z11/-Z21 installiert. Dieser besteht aus folgenden Dateien:

Tabelle 5-1 Teile des USB-Treiberstacks

Datei	Funktion	Bemerkung
Z11USB.sys Z21USB.sys	USB-Device-Treiber für R&S NRP-Messköpfe	
Z11USB.inf Z21USB.inf	enthält Installationsinformationen für USB-Device-Treiber	
NrpFU.sys	USB-Device-Treiber für R&S NRP-Messköpfe	wird benötigt für Firmware-Update
NrpFU.inf	enthält Installationsinformationen für USB-Device-Treiber	wird benötigt für Firmware-Update
NrpControl.h	enthält die C-Deklarationen aller Funktionsaufrufe	muss in den Teil eines C-Projektes inkludiert werden, in dem die DLL angesprochen wird. *)
NrpControl.lib	enthält die von der DLL exportierten Symbole für den Linker	muss dem Linker der Entwicklungsumgebung bekannt gemacht werden. *)
NrpControl.dll	kapselt den USB-Device-Treiber, um eine komfortablere Kommunikation mit dem Messkopf zu ermöglichen	sollte entweder im selben Verzeichnis liegen wie die ausführbare Applikation, oder mindestens in einem Verzeichnis, das im Systempfad aufgeführt ist, z. B. %SYSTEM_ROOT%\system32; (%SYSTEM_ROOT% ist eine von Windows™ angelegte Umgebungsvariable, die das Windows-Verzeichnis enthält). Die Datei NrpControl.dll wird bei der Installation des NRP-Toolkits in das Systemverzeichnis kopiert.

Mit den mitgelieferten Bestandteilen des Treiberstacks ist es möglich, den Messkopf mit geringem Zusatzaufwand unter verschiedenen Programmiersprachen anzusprechen: In C-, C++- und CVI-Projekte müssen das Headerfile *NrpControl.h* und die Programmbibliothek *NrpControl.lib* eingebunden werden.

*) Die Dateien NrpControl.h und NrpControl.lib werden bei der Installation des NRP-Toolkits im Verzeichnis c:\Programme\Rohde&Schwarz\NRP-Toolkit\NRPControl gespeichert (Installation mit der Einstellung 'Typical' durchgeführt).

Universelle USB-Treiber-Schnittstelle

Der Messkopf R&S NRP-Z11/-Z21 wird vom Host-PC über SCPI-Befehle (siehe Abschnitt 6 dieses Benutzerhandbuchs) angesprochen. Diese Befehle werden als ASCII-Strings über den USB zum Messkopf übertragen und von diesem interpretiert. Der Messkopf sendet dagegen Messwerte, Parameter und sonstige Daten in einem Binärblock-Format.

Dem Softwareentwickler wird mit der Dynamic Link Library *NrpControl.dll* eine universelle Schnittstelle angeboten, mit der er den Messkopf R&S NRP-Z11/-Z21 fernsteuern kann. Diese Schnittstelle umfasst einen Ausgabebefehl für die SCPI-Befehle, einen Befehl zum Senden von Binärblöcken (zur Übertragung von Kalibrierdatensätzen in den Messkopf) und mehrere Befehle zum Abfragen der vom Messkopf gesendeten Daten (Messwerte, Gerätestatus, USB-spezifische Informationen usw.)

Die Interpretation des in Rückrichtung benutzten Binärblock-Formats muss in Abhängigkeit von der Art der Daten erfolgen. Sendet der Messkopf z. B. *FLOAT*-Werte, so kann es sich entweder um Messwerte oder um abgefragte Einstellparameter handeln. Sind es Einstellparameter, dann muss festgestellt werden, um welche genau es sich handelt. Die DLL stellt Mechanismen für diese Interpretation bereit. Bild 5-1 stellt die prinzipielle Funktionsweise der DLL dar.

Folgende Daten werden innerhalb der DLL in unabhängigen, dynamischen Puffern verwaltet:

- aufgetretene Fehler
- Ergebnis-Deskriptoren (nähere Informationen über die Art der gepufferten Daten, z. B. Messwerte, Parameter, Grenzwerte)
- binäre Daten
- *FLOAT*-Daten
- *FLOAT*-Array-Daten
- *LONG*-Daten
- Bitfeld-Daten

Ausgelesen und entleert werden diese Puffer über spezielle Lesefunktionen. Es existieren auch Funktionen zum gezielten Löschen dieser Puffer. Die Puffer sind als FIFO-Speicher organisiert, d. h. es werden beim Auslesen eines bestimmten Puffers stets die ältesten darin vorhandenen Daten ausgelesen und dabei aus dem Puffer entfernt.

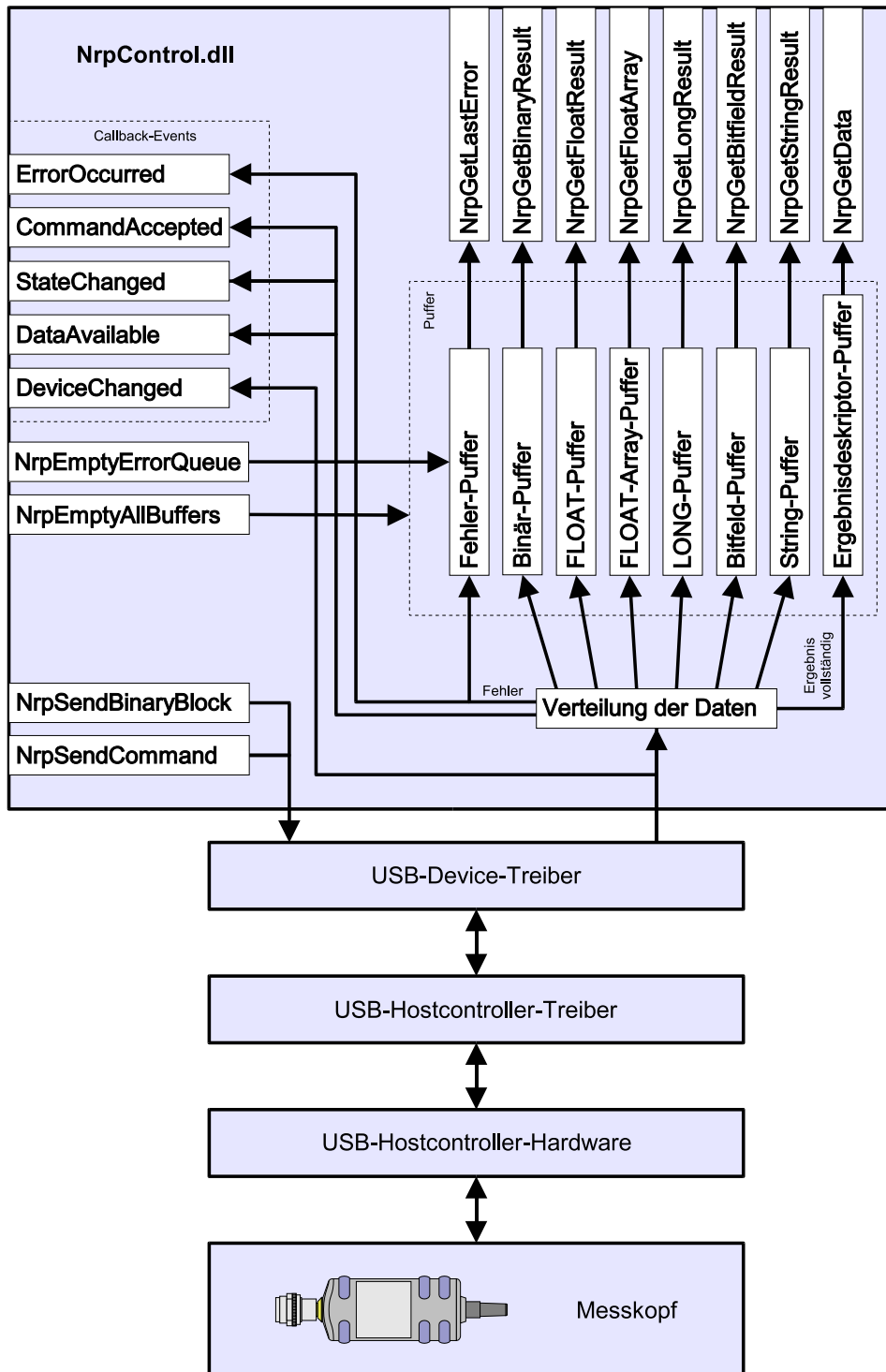


Bild 5-1 Funktionsweise der Dynamic Link Library NrpControl.dll

Funktionen zum Einstellen der Messkopf-Betriebsart

NrpChannelAssignment

Der Aufruf von *NrpChannelAssignment* öffnet einen Dialog, der es ermöglicht, Messköpfen eine leicht verständliche Bezeichnung zuzuordnen. Dies soll die Arbeit mit mehreren Messköpfen erleichtern. Dabei werden die Messköpfe mit Hilfe der Typenbezeichnung und Seriennummer eindeutig identifiziert.

Prototyp: `void NrpChannelAssignment (void);`

NrpStartApplicationMode

Mit *NrpStartApplicationMode* wird der Messkopf in den Messmodus versetzt.

Nach einem Power-on-Reset startet bei den R&S NRP-Messköpfen zunächst der Bootlader. Dies ermöglicht selbst bei einer defekten Mess-Firmware einen Firmware-Update. Nach einer Wartezeit von 10 s wechseln die Messköpfe automatisch in den Messmodus. Mit *NrpStartApplicationMode ()* erfolgt der Moduswechsel sofort.

Prototyp: `void NrpStartApplicationMode (void);`

Funktionen zum Senden von Befehlen und Daten

NrpSendBinaryBlock

Mit *NrpSendBinaryBlock* ist es möglich, einen Befehl, gefolgt von einem Block binärer Daten als Parameter, an den Messkopf zu senden.

Prototyp:

```
void NrpSendBinaryBlock (const char *i_pcCommand, void *i_pBuf,  
                        long i_wCount);
```

Parameter:

<i>i_pcCommand</i>	Zeiger auf den zu sendenden Befehl (nullterminierter String)
<i>i_pBuf</i>	Zeiger auf den Puffer, der die Binärdaten enthält
<i>i_wCount</i>	Anzahl der zu sendenden Bytes

Beispiel: `NrpSendBinaryBlock ("CALibration:DATA", caldata_ptr, 1234);`

Der Befehl *CALibration:DATA* wird zum Übertragen eines 1234 Bytes langen Kalibrierdatensatzes in den Flash-Speicher des Messkopfes genutzt. Der Zeiger *i_pBuf* zeigt auf den Anfang des Speicherbereiches, in dem die Kalibrierdaten stehen.

NrpSendCommand

Mit *NrpSendCommand* werden Befehle als ASCII-Strings an den Messkopf gesendet. Danach wartet die Funktion, bis entweder der Messkopf die Ausführung des Befehls bestätigt oder die Wartezeit die angegebene Timeout-Zeit überschreitet. Die Funktion liefert als Rückgabewert 1, wenn der Befehl erfolgreich ausgeführt wurde, und 0, wenn dies nicht der Fall ist.

Wird für *i_wTimeoutMs* der Wert 0 übergeben, so liefert die Funktion sofort den Rückgabewert 1, ohne auf eine Bestätigung vom Messkopf zu warten.

Eine Übersicht über die verfügbaren Befehle bietet Abschnitt 6 dieses Benutzerhandbuchs.

Prototyp: `long NrpSendCommand (const char *i_pcCommand, long i_wTimeoutMs);`

Parameter: *i_pcCommand* Zeiger auf den zu sendenden Befehl (nullterminierter String)

Beispiel: `NrpSendCommand ("*IDN?");`

Der Abfragebefehl **IDN?* wird an den Messkopf gesendet.

Funktionen zum Abholen von Messergebnissen und Daten

NrpDataAvailable

NrpDataAvailable liefert 1, wenn Daten vom Messkopf eingetroffen sind, und 0, wenn dies nicht der Fall ist und wenn alle internen Puffer leer sind. Wird mit einer Callback-Funktion auf das Vorhandensein von Daten reagiert, so ist die gleichzeitige Verwendung von *NrpDataAvailable* nicht sinnvoll.

Prototyp: long NrpDataAvailable (void);

NrpGetData

NrpGetData ermittelt den Typ des abzuholenden Ergebnisses (siehe Tabelle 5-2) sowie die Gruppennummer und die Parameternummer, wenn das Ergebnis ein Parameter ist. Die Gruppennummer ist die laufende Nummer einer Gruppe von zusammen gehörenden Parametern bzw. Befehlen. Von verschiedenen Messkopftypen werden in Abhängigkeit von deren Eigenschaften unterschiedliche Parameter-/Befehlsgruppen implementiert. Die Parameternummer ist die laufende Nummer eines Parameters/Befehls innerhalb seiner Gruppe. Bei Messergebnissen sind Gruppen- und Parameternummer 0.

Prototyp:

```
void NrpGetData (long *o_pDataType, long *o_pGroupNr, long *o_pParamNr);
```

Parameter:

- o_pDataType* Zeiger auf die *LONG*-Variable, die den numerischen Wert des Datentyps aufnehmen soll
- o_pGroupNr* Zeiger auf die *LONG*-Variable, die die Gruppennummer aufnehmen soll
- o_pParamNr* Zeiger auf die *LONG*-Variable, die die Parameternummer aufnehmen soll

Tabelle 5-2 Bedeutung der Datentypen

Datentyp (enum-Konstante)	num. Wert	abzuholende Daten
DATA_BINARYBLOCK	0	Eine binäre Bytefolge kann mit <i>NrpGetBinaryResult</i> abgeholt werden (Kalibrierdatensatz des Messkopfes bei Verwendung des Befehls <i>CALibration:DATA?</i>).
DATA_BITFIELDLIMIT	1	Drei Bitfeld-Werte können mit <i>NrpGetBitfieldResult</i> abgeholt werden. Der erste Wert ist die Voreinstellung, der zweite Wert ist die Bitmaske zur Darstellung aller zulässigen Zustände, der dritte Wert ist ungenutzt und mit 0 belegt.
DATA_BITFIELDPARAM	2	Drei Bitfeld-Werte zur Darstellung von diskreten Zuständen (z. B. <i>OFF</i> , <i>ON</i> , <i>ONCE</i>) können mit <i>NrpGetBitfieldResult</i> abgeholt werden. Der erste Wert ist ein Parameterwert, der zweite und der dritte Wert sind ungenutzt und mit 0 belegt.
DATA_BITFIELDFEATURE	3	Drei Bitfeld-Werte zur Darstellung von implementierten Befehlen können mit <i>NrpGetBitfieldResult</i> abgeholt werden. Der erste Wert enthält die Information (für jeden innerhalb einer Befehlsgruppe vorhandenen Befehl oder Parameter ist ein Bit gesetzt, beginnend bei Bit 0 für Parameter-Nr. 1), der zweite und der dritte Wert sind ungenutzt und mit 0 belegt.
DATA_FLOATARRAY	4	Ein Array von <i>FLOAT</i> -Werten (Leistungswerte) und ein Array von <i>LONG</i> -Werten (Indizes) können mit <i>NrpGetFloatArray</i> abgeholt werden.
DATA_FLOATLIMIT	5	Drei <i>FLOAT</i> -Werte (Voreinstellung, oberer Grenzwert und unterer Grenzwert) können mit <i>NrpGetFloatResult</i> abgeholt werden.

Datentyp (enum-Konstante)	num. Wert	abzuholende Daten
DATA_FLOATPARAM	6	Drei <i>FLOAT</i> -Werte können mit <i>NrpGetFloatResult</i> abgeholt werden. Der erste Wert ist ein Parameterwert, der zweite und der dritte Wert sind unbenutzt und mit 0.0 belegt.
DATA_FLOATRESULT	7	Drei <i>FLOAT</i> -Werte (Leistungswert und zwei sekundäre Messwerte) können mit <i>NrpGetFloatResult</i> abgeholt werden.
DATA_LONGLIMIT	8	Drei <i>LONG</i> -Werte (Voreinstellung, oberer Grenzwert und unterer Grenzwert) können mit <i>NrpGetLongResult</i> abgeholt werden.
DATA_LONGPARAM	9	Drei <i>LONG</i> -Werte können mit <i>NrpGetLongResult</i> abgeholt werden. Der erste Wert ist ein Parameterwert, der zweite und der dritte Wert sind ungenutzt und mit 0 belegt.
DATA_STRING	10	Ein String kann mit <i>NrpGetStringResult</i> abgeholt werden.

Tabelle 5-3 Bedeutung der Gruppen- und Parameternummern

Gruppennummer	Befehlsgruppe	Parameter- nummer	Parameter oder Befehl
1	CALibration	1	CALibration:DATA
		2	CALibration:ZERO:AUTO
		3	CALibration:DATA:LENGth
2	INPut (z. Z. reserviert)		
3	SENSe	1	SENSe:AVERage:COUNT
		2	SENSe:AVERage:COUNT:AUTO
		3	SENSe:CORRection:DCYClE
		4	SENSe:AVERage:STATe
		5	SENSe:AVERage:TCONtrol
		6	SENSe:CORRection:OFFSet
		7	SENSe:CORRection:OFFSet:STATe
		8	SENSe:DCYClE:OFFSet:STATe
		9	SENSe:FREQUency
		10	SENSe:FUNCTion
		11	SENSe:EUNCertainty:STATe
		12	SENSe:RANGE

Gruppennummer	Befehlsgruppe	Parameternummer	Parameter oder Befehl
		13	SENSe:RANGe:AUTO
		14	SENSe:RANGe:CLEVel
		15	SENSe:TIMing:EXCLude:STARt
		16	SENSe:TIMing:EXCLude:STOP
		17	SENSe:SAMPling
		18	SENSe:AVERage:COUNt:AUTO:MTIME
		19	SENSe:AVERage:COUNt:AUTO:RESolution
		20	SENSe:AVERage:COUNt:AUTO:SLOT
		21	SENSe:AVERage:COUNt:AUTO:NSRatio
		22	SENSe:AVERage:COUNt:AUTO:TYPE
		23	SENSe:CORRection:SPDevice:STATe
		24	SENSe:SGAMma:MAGNitude
		25	SENSe:SGAMma:PHASe
		26	SENSe:SGAMma:CORRection:STATe
		27	SENSe:SGAMma:EUNCertainty
28	SENSe:EUNCertainty:SGAMma:STATe		
4	SENSe:POWer:AVG	1	SENSe:POWer:AVG:APERture
		2	SENSe:POWer:AVG:BUFFer:SIZE
		3	SENSe:POWer:AVG:BUFFer:STATe
		4	SENSe:POWer:AVG:SMOothing:STATe
5	SENSe:POWer:TSLot:AVG	1	SENSe:POWer:TSLot:AVG:COUNt
		2	SENSe:POWer:TSLot:AVG:WIDTh
6	SENSe:POWer:BURSt:AVG	1	SENSe:POWer:BURSt:DTOLerance
7	SENSe:SWEep	1	SENSe:SWEep:AVERage:STATe
		2	SENSe:SWEep:OFFSet:TIME
		3	SENSe:SWEep:POINts
		4	SENSe:SWEep:REALtime
		5	SENSe:SWEep:TIME

Gruppennummer	Befehlsgruppe	Parameternummer	Parameter oder Befehl
		6	SENSe:SWEep:AVERage:COUNT
		7	SENSe:SWEep:AVERage:COUNT:AUTO
		8	SENSe:SWEep:AVERage:COUNT:AUTO:MTIME
		9	SENSe:SWEep:AVERage:COUNT:AUTO:RESolution
		10	SENSe:SWEep:AVERage:COUNT:AUTO:POINT
		11	SENSe:SWEep:AVERage:COUNT:AUTO:NSRatio
		12	SENSe:SWEep:AVERage:COUNT:AUTO:TYPE
		13	SENSe:SWEep:AVERage:TCONTROL
8	SYSTem	1	SYSTem:INFO
		2	SYSTem:INITialize
		3	SYSTem:TRANsaction:BEgin
		4	SYSTem:TRANsaction:END
		5	SYSTem:MINPower
9	Triggersystem-Befehle	1	ABORt
		2	INITiate:CONTinuous
		3	INITiate:IMMediate
		4	reserviert
		5	reserviert
		6	TRIGger:ATRigger:STATe
		7	TRIGger:COUNt
		8	TRIGger:DELay
		9	TRIGger:DELay:AUTO
		10	TRIGger:HOLDoff
		11	TRIGger:IMMediate
		12	TRIGger:LEVel
		13	TRIGger:SLOPe
		14	TRIGger:SOURce
		15	TRIGger:HYSTeresis

Gruppennummer	Befehlsgruppe	Parameternummer	Parameter oder Befehl
10	SERVice	1	SERVice:DITHer
		2	SERVice:SAMPle
		3	SERVice:SIMCount
		4	SERVice:UNLock
		5	SERVice:CALibration:ZERO:NEG0
		6	SERVice:CALibration:ZERO:POS0
		7	SERVice:CALibration:ZERO:NEG1
		8	SERVice:CALibration:ZERO:POS1
		9	SERVice:CALibration:ZERO:NEG2
		10	SERVice:CALibration:ZERO:POS2
		11	SERVice:CALibration:DITHer
		12	SERVice:CALibration:DITHer:DATA
		13	SERVice:CALibration:TEMP
		14	SERVice:CALibration:TEMP:DATA
		15	SERVice:PARAmeter:RTemp
		16	SERVice:PARAmeter:RNULL0
		17	SERVice:PARAmeter:RNULL1
		18	SERVice:PARAmeter:RNULL2
		19	SERVice:PARAmeter:RBAHN
		20	SERVice:PARAmeter:NREF
		21	SERVice:PARAmeter:ATHERM
		22	SERVice:PARAmeter:BTHERM
		23	SERVice:MVCorrection
		24	SERVice:CALibration:TEST
		25	SERVice:RCOunt
		26	SERVice:RESult
		27	SERVice:PARAmeter:CTHERM

Gruppennummer	Befehlsgruppe	Parameternummer	Parameter oder Befehl
		28	SERVice:PARAmeter:DThERM
		29	SERVice:PARAmeter:CJUNC
11	IEEE 488.2 Common Commands	1	*RST
		2	*TRG
		3	*IDN?
		4	*TST?
12	TEST	1	TEST:SENSor

NrpGetBinaryResult

NrpGetBinaryResult liest sämtliche Antwort-Datenblöcke, die seit dem letzten Aufruf der Funktion vom Messkopf gesendet wurden, aus dem Binärpuffer aus und kopiert sie in einen bereitgestellten Puffer. Die ausgelesenen Bytes werden aus dem Binärpuffer gelöscht. Die ältesten Antwort-Datenblöcke werden zuerst ausgelesen. Die Funktion liefert als Rückgabewert die Anzahl der tatsächlich aus dem Binärpuffer ausgelesenen Bytes.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der im Binärpuffer befindlichen Bytes.

Prototyp: `long NrpGetBinaryResult (void *o_pBuf, long i_wCount);`

Parameter: *o_pBuf* Zeiger auf den Puffer, in dem das Ergebnis hinterlegt wird
i_wCount Puffergröße in Bytes

NrpGetBitfieldResult

Parameter, die eine bestimmte Anzahl diskreter Zustände (z. B. *OFF*, *ON* und *ONCE*) annehmen können, werden als 32 Bit breite Bitfelder codiert. Jedes Bit repräsentiert einen der möglichen Zustände. Auf diese Weise ist es möglich, dass der Messkopf bei abhängigen Parametern (analog zu Unter- und Obergrenze bei numerischen Parametern) bestimmte diskrete Zustände zulassen oder ausschließen kann.

NrpGetBitfieldResult liest den ältesten der Bitfeld-Blöcke, die seit dem letzten Aufruf der Funktion vom Messkopf gesendet wurden, aus dem Bitfeldpuffer aus. Der ausgelesene Bitfeldblock wird aus dem Bitfeldpuffer gelöscht. Die Bitfelder werden in *LONG*-Variablen übergeben.

Die Funktion liefert als Rückgabewert 1, wenn das Auslesen erfolgreich war, und 0, wenn beim Auslesen ein Fehler aufgetreten ist.

Prototyp:

`long NrpGetBitfieldResult (long *o_pdwR1, long *o_pdwR2, long *o_pdwR3);`

Parameter: *o_pdwR1* Zeiger auf die erste der drei *LONG*-Variablen, die das Ergebnis aufnehmen sollen
o_pdwR2 Zeiger auf die zweite der drei *LONG*-Variablen, die das Ergebnis aufnehmen sollen
o_pdwR3 Zeiger auf die dritte der drei *LONG*-Variablen, die das Ergebnis aufnehmen sollen

NrpGetFloatArray

Befindet sich der Messkopf im sogenannten *buffered mode*, so werden Messergebnisse nicht einzeln, sondern blockweise übertragen. Dadurch lässt sich eine besonders hohe Übertragungsgeschwindigkeit erreichen. Auch in den Modi *Timeslot* und *Scope* liefert der Messkopf seine Ergebnisse in Form von *FLOAT*-Array-Blöcken. Jedem der *FLOAT*-Werte ist innerhalb des Arrays ein fester Index zugeordnet. *NrpGetFloatArray* kopiert eine bestimmte Anzahl von *FLOAT*-Werten und deren Indizes, beginnend mit den ältesten Werten, aus dem DLL-internen Float-Array-Puffer in ein *FLOAT*- bzw. *LONG*-Array. Die kopierten Werte werden aus dem Float-Array-Puffer gelöscht. Als Rückgabewert liefert die Funktion die Anzahl der tatsächlich kopierten *FLOAT*-Werte bzw. Indizes.

Übergibt man der Funktion als Argument *o_pFarr* einen Nullpointer, so liefert die Funktion die Anzahl der im Float-Array-Puffer befindlichen *FLOAT*- und Index-Werte.

Prototyp:

```
long NrpGetFloatArray (float *o_pFarr , long *o_pIarr, long i_wCount);
```

Parameter:

<i>o_pFarr</i>	Zeiger auf das <i>FLOAT</i> -Array, das die <i>FLOAT</i> -Werte aufnehmen soll
<i>o_pIarr</i>	Zeiger auf das <i>LONG</i> -Array, das die Indizes der <i>FLOAT</i> -Werte aufnehmen soll
<i>i_wCount</i>	Länge der Arrays

NrpGetFloatResult

NrpGetFloatResult kopiert die drei *FLOAT*-Werte, die in einem *FLOAT*-Block enthalten sind, in *FLOAT*-Variable. Handelt es sich bei dem *FLOAT*-Block um ein Messergebnis, so ist der erste *FLOAT*-Wert der Leistungswert, der zweite und der dritte *FLOAT*-Wert sind sekundäre Messwerte, z. B. Rauschen und Messunsicherheit. Die Funktion liefert als Rückgabewert 1, wenn das Auslesen erfolgreich war, und 0, wenn beim Auslesen ein Fehler aufgetreten ist.

Prototyp:

```
long NrpGetFloatResult(float *o_pfR1, float *o_pfR2, float *o_pfR3);
```

Parameter:

<i>o_pfR1</i>	Zeiger auf die <i>FLOAT</i> -Variable, die den ersten <i>FLOAT</i> -Wert aufnehmen soll
<i>o_pfR2</i>	Zeiger auf die <i>FLOAT</i> -Variable, die den zweiten <i>FLOAT</i> -Wert aufnehmen soll
<i>o_pfR3</i>	Zeiger auf die <i>FLOAT</i> -Variable, die den dritten <i>FLOAT</i> -Wert aufnehmen soll

NrpGetLongResult

NrpGetLongResult kopiert die drei *LONG*-Werte, die in einem *LONG*-Block enthalten sind, in *LONG*-Variable. Die Funktion liefert als Rückgabewert 1, wenn das Auslesen erfolgreich war, und 0, wenn beim Auslesen ein Fehler aufgetreten ist.

Prototyp:

```
long NrpGetLongResult (long *o_pwR1, long *o_pwR2, long *o_pwR3);
```

Parameter:

<i>o_pwR1</i>	Zeiger auf die <i>LONG</i> -Variable, die den ersten <i>LONG</i> -Wert aufnehmen soll
<i>o_pwR2</i>	Zeiger auf die <i>LONG</i> -Variable, die den zweiten <i>LONG</i> -Wert aufnehmen soll
<i>o_pwR3</i>	Zeiger auf die <i>LONG</i> -Variable, die den dritten <i>LONG</i> -Wert aufnehmen soll

NrpGetStringResult

NrpGetStringResult erlaubt den Zugriff auf die bis zum Aufruf dieser Funktion eingetroffenen String-Antworten. Eine Stringantwort wird in der Regel in mehrere String-Blöcke aufgeteilt empfangen, in der DLL zusammengesetzt und im internen String-Puffer gespeichert. Die Funktion kopiert eine bestimmte Anzahl von Zeichen aus diesem internen String-Puffer in einen bereitgestellten Puffer. Die kopierten Zeichen werden aus dem internen String-Puffer gelöscht. Als Rückgabewert liefert die Funktion die Anzahl der tatsächlich gelesenen Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der im Stringpuffer befindlichen Zeichen. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp: `long NrpGetStringResult (char *o_pBuf, long i_wCount);`

Parameter:

<i>o_pBuf</i>	Zeiger auf den Puffer, in welchem die Antwort hinterlegt werden soll
<i>i_wCount</i>	Puffergröße in Bytes

Funktionen zur Status- und Fehlererkennung

NrpEmptyAllBuffers

NrpEmptyAllBuffers löscht sämtliche DLL-internen dynamischen Puffer.

Prototyp: void NrpEmptyAllBuffers (void);

NrpEmptyErrorQueue

NrpEmptyErrorQueue löscht alle in der Fehler-Queue befindlichen Fehler.

Prototyp: void NrpEmptyErrorQueue (void);

NrpGetLastError

NrpGetLastError liest den am weitesten zurückliegenden Fehler aus der Fehler-Queue aus. Um die Fehler-Queue zu leeren, müssen entweder nacheinander alle Fehler mit *NrpGetLastError* ausgelesen werden, oder einer der Funktionen *NrpEmptyErrorQueue* oder *NrpEmptyAllBuffers* muss aufgerufen werden. Eine Zusammenstellung aller möglichen Fehlerzustände bietet Tabelle 5-4.

Prototyp: long NrpGetLastError (void);

Tabelle 5-4 Mögliche Messkopf-Fehlerzustände

Fehler (enum-Konstante)	num. Wert	Bedeutung
NRPEERROR_NOERROR	0	Dieser Wert wird von der Funktion <i>NrpGetLastError</i> zurückgegeben, wenn kein Fehler aufgetreten ist.
NRPEERROR_CALDATA_FORMAT	1	Das Format des an den Messkopf zu übertragenden Kalibrierdatensatzes entspricht nicht der Spezifikation. Entweder wird der vorhandene Kalibrierdatensatz nicht als solcher erkannt, oder er liegt in einer nicht unterstützten Version vor.
NRPEERROR_OVERRANGE	2	Die Leistung des Messsignals übersteigt den Messbereich des manuell gewählten Messkanals (<i>bei SENSE:RANGE:AUTO OFF</i>).
NRPEERROR_NOTINSERVICEMODE	3	Der an den Messkopf geschickte Befehl ist nur im Service-Modus verfügbar, der Messkopf befindet sich aber nicht im Service-Modus.
NRPEERROR_CALZERO	4	Der Nullabgleich konnte nicht durchgeführt werden, weil die anliegende HF-Leistung zu hoch ist.
NRPEERROR_TRIGGERQUEUEFULL	5	Im Modus <i>Burst Average</i> : Der Burst ist breiter als zulässig, daher wurde die Messung nach Ablauf der maximal zulässigen Burst-Dauer abgebrochen. Das so erhaltene Ergebnis kann vom korrekten Messwert abweichen.
NRPEERROR_EVENTQUEUEFULL	6	Die interne Queue für Trigger-Ereignisse ist übergelaufen. Diese Fehlermeldung deutet auf einen Software-Fehler hin, da der Messkopf normalerweise bei einem drohenden Überlauf dieser Queue keine weiteren Triggerereignisse annimmt.

Fehler (enum-Konstante)	num. Wert	Bedeutung
NRPEROR_SAMPLEERROR	7	Ein oder mehrere Samples konnten nicht verarbeitet werden, weil der Messkopf z. B. während der Abarbeitung eines Bus-Triggers nicht auf den Sample-Interrupt reagieren konnte. Normalerweise führt dies nicht zu einer Verfälschung des Messergebnisses, außer im Realtime-Modus (<i>SENSe:SWEEp:REALtime ON</i>).
NRPEROR_OVERLOAD	8	Die Leistung des Messsignals übersteigt die obere Messgrenze des Messkopfes. Es besteht die Gefahr einer irreversiblen Beschädigung des Messkopfes.
NRPEROR_HARDWARE	9	Ein Hardware-Fehler ist aufgetreten. Fehler am Temperatursensor führen während des Betriebes sofort zu dieser Fehlermeldung. Wurde bei einem der Testbefehle <i>*TST?</i> oder <i>TEST:SENSe?</i> ein Fehler festgestellt, so führt dies so lange zu dieser Fehlermeldung, bis ein erneuter Selbsttest keinen Fehler mehr feststellt. Fehler bei den Betriebsspannungen werden außerdem separat gemeldet (siehe <i>NRPEROR_VOLTAGE</i>).
NRPEROR_CHECKSUM	10	Die Überprüfung der Prüfsumme des zu ladenden Kalibrierdatensatzes ergab einen Fehler. Der Kalibrierdatensatz wurde nicht geladen.
NRPEROR_ILLEGALSERIAL	11	Es wurde versucht, einen Kalibrierdatensatz in den Messkopf zu laden, dessen Seriennummer nicht mit der Seriennummer des Messkopfes übereinstimmt. Der Kalibrierdatensatz wurde nicht geladen.
NRPEROR_FILTERTRUNCATED	12	Im Auto-Averaging-Modus: Die Messung wurde nach Ablauf der maximalen Messzeit (Parameter <i>SENSe:AVERAge:COUNt:AUTO:MTIME</i> oder <i>SENSe:SWEEp:AVERAge:COUNt:AUTO:MTIME</i>) abgebrochen, ohne dass die vorgegebene Auflösung oder das vorgegebene Signal-Rausch-Verhältnis erreicht wurden.
NRPEROR_GENERIC	128	Es ist ein nicht näher spezifizierter Fehler aufgetreten.
NRPEROR_OVERMAX	129	Der numerische Parameter ist größer als die zulässige Obergrenze.
NRPEROR_UNDERMIN	130	Der numerische Parameter ist kleiner als die zulässige Untergrenze.
NRPEROR_VOLTAGE	131	Mindestens eine der internen Versorgungsspannungen fehlt oder liegt außerhalb des zulässigen Bereiches.
NRPEROR_SYNTAX	132	Die Syntax des an den Messkopf geschickten Befehls ist fehlerhaft.
NRPEROR_MEMORY	133	Der zur Verfügung stehende Speicher reicht nicht aus. Diese Fehlermeldung deutet auf einen Firmware-Fehler hin.
NRPEROR_PARAMETER	134	Dieser Parameter ist unzulässig.
NRPEROR_TIMING	135	reserviert
NRPEROR_NOTIDLE	136	Der an den Messkopf geschickte Befehl steht während einer laufenden Messung nicht zur Verfügung.
NRPEROR_UNKNOWNCOMMAND	137	Der an den Messkopf geschickte Befehl konnte nicht interpretiert werden.
NRPEROR_OUTBUFFERFULL	138	Der Sendepuffer des Messkopfes ist übergelaufen, weil die vom Messkopf ausgegebenen Daten nicht abgeholt wurden.

Fehler (enum-Konstante)	num. Wert	Bedeutung
NRPEROR_FLASHPROG	139	Es ist ein Fehler beim Programmieren des Flash-Speichers aufgetreten.
NRPEROR_CALDATANOTPRESENT	140	Es ist kein gültiger Kalibrierdatensatz vorhanden.

NrpGetErrorText

NrpGetErrorText dient zum Abrufen der zum Fehlercode vom Typ *NRPEROR* passenden Fehlermeldung im Klartext. Diese wird als nicht nullterminierter String in einem bereitgestellten Puffer abgelegt. Als Rückgabewert liefert die Funktion die Anzahl der im Puffer abgelegten Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der im Stringpuffer befindlichen Zeichen. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp: `long NrpGetErrorText (long i_wErr, char *o_pBuf, long i_wCount);`

Parameter:

<i>i_wErr</i>	Fehlercode, der beispielsweise von <i>NrpGetLastError</i> geliefert wurde
<i>o_pBuf</i>	Zeiger auf den Puffer, in dem der Fehlertext abgelegt werden soll
<i>i_wCount</i>	Puffergröße in Bytes

NrplsAlive

NrplsAlive liefert als Rückgabewert

- 0, wenn der als aktiv gewählte Messkopf nicht betriebsbereit ist,
- 1, wenn er betriebsbereit und im Messmodus ist,
- 2, wenn der Bootloader aktiv ist.

Prototyp: `long NrpIsAlive (void);`

NrpGetTriggerState

NrpGetTriggerState liefert als Rückgabewert den numerischen Wert des aktuellen Zustandes des Triggersystems (Tabelle 5-5). Um eine kurze Klartextbeschreibung des Triggerzustandes zu erhalten, kann die Funktion *NrpGetTriggerStateText* benutzt werden.

Prototyp: `long NrpGetTriggerState (void);`

Tabelle 5-5 Mögliche Messkopf-Triggerzustände

Triggerzustand (enum-Konstante)	num. Wert	Beschreibung
SENSORHW_STATE_IDLE	0	Es läuft keine Messung, der Messkopf ist bereit.
SENSORHW_STATE_WAIT_FOR_ARM	1	Reserviert. Der R&S NRP-Z11/-Z21 hat keinen ARM-Layer.
SENSORHW_STATE_WAIT_FOR_TRIGGER	2	Eine Messung wurde gestartet, der Messkopf wartet auf ein Trigger-Ereignis.
SENSORHW_STATE_MEASURING	3	Eine Messung läuft.

NrpGetTriggerStateText

NrpGetTriggerStateText liefert eine kurze Klartextbeschreibung des z. B. mit *NrpGetTriggerState* ermittelten Triggerzustandes.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der Zeichen im beschreibenden String. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp:

```
long NrpGetTriggerStateText (long i_wState, char *o_pBuf, long i_wCount);
```

Callback-Funktionen

Um die verzögerungsfreie und asynchrone Auswertung bestimmter Ereignisse durch die Anwendung zu ermöglichen, können Callback-Funktionen festgelegt werden, die beim Auftreten des jeweiligen Ereignisses durch die DLL aufgerufen werden. Die Ereignisse, auf die reagiert werden kann, umfassen:

- *CommandAccepted*: Ein zum Messkopf gesendeter Befehl wurde von diesem quittiert.
- *DataAvailable*: Daten vom Messkopf sind eingetroffen.
- *DeviceChanged*: Ein Messkopf wurde an den Host-Rechner angeschlossen oder vom Host-Rechner getrennt.
- *ErrorOccurred*: Der Messkopf hat einen Fehler gemeldet.
- *StateChanged*: Der Triggerstatus des Messkopfes hat sich geändert.

Um Callback-Funktionen zu definieren, dienen folgende Funktionen:

- *NrpSetNotifyCallbackCommandAccepted*
- *NrpSetNotifyCallbackDataAvailable*
- *NrpSetNotifyCallbackDeviceChanged*
- *NrpSetNotifyCallbackErrorOccurred*
- *NrpSetNotifyStateChanged*

Diese Funktionen sind im Abschnitt „Funktionen zum Setzen von Callback-Funktionen“ auf Seite 5.19 näher beschrieben. Der Typ *function* ist dabei wie folgt definiert:

```
typedef void (__stdcall *function) (void *arg1, void *arg2);
```

Die zu übergebende Callback-Funktion muss die folgende Form haben:

```
void __stdcall MyCallbackFunction (void *arg1, void *arg2)
{
    ...
}
```

Ist die Callback-Funktion eine Klassenmethode, so muss sie als *static* vereinbart werden.

Im Falle der Ereignisse *CommandAccepted*, *DataAvailable*, *ErrorOccured* und *StateChanged* wird *arg1* mit *NULL* belegt und braucht nicht weiter ausgewertet zu werden. Bei *DeviceChanged* wird dagegen ein Zeiger auf eine Struktur des Typs *USB_DEVICECHANGED_HANDLER_RESULT* (siehe Beschreibung der Funktion *NrpSetNotifyCallbackDeviceChanged* auf Seite 5.19) übergeben, die für eine differenziertere Ereignisbehandlung ausgewertet werden kann.

Das Argument *arg2* dient zum Übergeben einer Kontextinformation (z. B. Referenz auf eine Klasse, wenn die Callback-Funktion eine Methode dieser Klasse aufrufen soll). Es wird beim Aufruf der *SetNotifyCallback...*-Funktion angegeben.

Folgendes Beispiel zeigt das Beispiel einer Callbackfunktion für das *DeviceChanged*-Ereignis, welche in Visual C++ als Klassenmethode implementiert ist. Es handelt sich dabei nicht um ein selbständig lauffähiges Programm. Das betrifft insbesondere die nicht dargestellte Klassenmethode *AddToProtocol* zur Ausgabe eines Strings z. B. in einem mehrzeiligen Edit-Control.

Definition im Headerfile:

```
private:
    static void __stdcall DeviceChangedCallback(void* arg1, void* arg2);
```

Anmelden im Initialisierungsteil der Klasse:

```
NrpSetNotifyCallbackDeviceChanged(DeviceChangedCallback, this);
```

Implementierung:

```
void CTestDlg::DeviceChangedCallback(void* arg1, void* arg2)
{
    CString txt, name, msg;
    long length, devices, devID, aliveState;

    USB_DEVICECHANGED_HANDLER_RESULT* hRes =
        (USB_DEVICECHANGED_HANDLER_RESULT*) arg1;

    CTestDlg* dialog = (CTestDlg*) arg2;

    length = NrpGetDeviceChangedMsgText(hRes->wEventTextHandle, NULL, 0);
    NrpGetDeviceChangedMsgText(hRes->wEventTextHandle, txt.GetBuffer(length),
        length);
    msg = "DeviceChangedMsgText:\t" + txt + "\r\n";
    dialog->AddToProtocol(msg);

    length = NrpGetDeviceChangedDevName(NULL, 0);
    NrpGetDeviceChangedDevName(name.GetBuffer(length), length);
    msg = "DeviceChangedDevName:\t" + name + "\r\n";
    dialog->AddToProtocol(msg);

    msg.Format("t\t%i Sensor(s) connected\r\n", NrpGetNrOfDevices());
    dialog->AddToProtocol(msg);
}
```

Funktionen zum Setzen von Callback-Funktionen

NrpSetNotifyCallbackCommandAccepted

NrpSetNotifyCallbackCommandAccepted legt eine nutzerspezifische Funktion fest, die aufgerufen wird, wenn ein zum Messkopf gesendeter Befehl von diesem erfolgreich abgearbeitet und quittiert wurde (siehe Abschnitt „Callback-Funktionen“ auf Seite 5.17).

Prototyp: void NrpSetNotifyCallbackCommandAccepted (function f, void *arg);

Parameter: *f* aufzurufende Funktion
 arg Argument zum Übergeben einer Kontextinformation

NrpSetNotifyCallbackDataAvailable

NrpSetNotifyCallbackDataAvailable legt eine nutzerspezifische Funktion fest, die aufgerufen wird, wenn Messwerte oder andere Daten vom Messkopf zur Abholung vorliegen (siehe Abschnitt „Callback-Funktionen“ auf Seite 5.17).

Prototyp: void NrpSetNotifyCallbackDataAvailable (function f, void *arg);

Parameter: *f* aufzurufende Funktion
 arg Argument zum Übergeben einer Kontextinformation

NrpSetNotifyCallbackDeviceChanged

NrpSetNotifyCallbackDeviceChanged legt eine nutzerspezifische Funktion fest, die aufgerufen wird, wenn ein Messkopf an den Host-Rechner angeschlossen oder vom Host-Rechner getrennt wurde (siehe Abschnitt „Callback-Funktionen“ auf Seite 5.17).

Prototyp: void NrpSetNotifyCallbackDeviceChanged (function f, void *arg);

Parameter: *f* aufzurufende Funktion
 arg Argument zum Übergeben einer Kontextinformation

NrpSetNotifyCallbackErrorOccured

NrpSetNotifyCallbackErrorOccured legt eine nutzerspezifische Funktion fest, die aufgerufen wird, wenn ein Fehler aufgetreten ist (siehe Abschnitt „Callback-Funktionen“ auf Seite 5.17).

Prototyp: void NrpSetNotifyCallbackErrorOccured (function f, void *arg);

Parameter: *f* aufzurufende Funktion
 arg Argument zum Übergeben einer Kontextinformation

NrpSetNotifyCallbackStateChanged

NrpSetNotifyCallbackStateChanged legt eine nutzerspezifische Funktion fest, die aufgerufen wird, wenn sich der Triggerzustand des Messkopfes ändert (siehe Abschnitt „Callback-Funktionen“ auf Seite 5.17).

Prototyp: void NrpSetNotifyCallbackStateChanged (function f , void *arg);

Parameter: *f* aufzurufende Funktion
 arg Argument zum Übergeben einer Kontextinformation

Windows-Messages

Die DLL versendet und empfängt Windows™-Messages. Sie ist also darauf angewiesen, dass das aufrufende Programm diese weiterleitet. In der Regel ist das aufrufende Programm eine GUI-Applikation, die auf dem Windows API aufsetzt, womit diese Bedingung immer erfüllt ist. Handelt es sich jedoch um eine einfache Konsolenapplikation, so werden von dieser automatisch keine Messages weitergeleitet. Um eine Message loop in einer Konsolenapplikation nachbauen zu können, bietet die DLL die Funktion *NrpMessageLoopBody* an. Sie muss in der Hauptschleife der Konsolenapplikationen zyklisch aufgerufen werden, um die von der DLL ein- und ausgehenden Windows™-Messages weiterzuleiten.

Prototyp: `void NrpMessageLoopBody (void);`

USB-spezifische Befehle

Folgende Kommandos betreffen nur die USB-Schnittstelle und haben keinerlei Bezug zur Funktionalität des Messkopfes:

NrpOpenDriver

NrpOpenDriver stellt die Verbindung zum USB-Device-Treiber her.

Prototyp: `void NrpOpenDriver (void);`

NrpCloseDriver

NrpCloseDriver beendet die Verbindung zum USB-Device-Treiber.

Prototyp: `void NrpCloseDriver (void);`

Hinweis: Beim Schließen der DLL wird eine bestehende Verbindung zum USB-Device-Treiber automatisch abgebaut.

NrpClearDevice

NrpClearDevice sendet den *Vendor Request SET_DEVICE_CLEAR* an den Messkopf. Dieser *Vendor Request* löscht die USB-FIFOs und initialisiert das Triggersystem.

Prototyp: `void NrpClearDevice (void);`

NrpGetDeviceChangedDevName

Die Callback-Funktion, die durch das *DeviceChanged*-Ereignis aufgerufen wird, liefert als Argument die Struktur *USB_DEVICECHANGED_HANDLER_RESULT* (siehe Tabelle 8). Eines der Strukturmitglieder ist der Zeiger *pDeviceName* auf den Namen des Gerätes, das dieses Ereignis ausgelöst hat. Da in bestimmten Programmierumgebungen, wie z. B. Visual Basic, Strings nicht direkt als Zeiger übergeben werden können, wird als Alternative die Funktion *NrpGetDeviceChangedDevName* angeboten, die den Gerätenamen in einen Puffer kopiert. Die Funktion liefert als Rückgabewert die Anzahl der tatsächlich kopierten Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der Zeichen im Device-Name-String. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp: `long NrpGetDeviceChangedDevName (char *o_pBuf, long i_wCount);`

Parameter: *o_pBuf* Zeiger auf den Puffer, der den Gerätenamen aufnehmen soll
i_wCount Puffergröße in Bytes

Tabelle 5-6 Beschreibung der Struktur *USB_DEVICECHANGED_HANDLER_RESULT*

Typ	Name	Beschreibung
long	wResult	Der Wert ist 1, wenn dem Ereignis ein Gerät zugeordnet werden konnte, und 0, wenn dies nicht der Fall ist (dann ist der String, auf den pDeviceName zeigt, ein Leerstring).
long	wEventType	Kann die Werte <i>SENSORHW_STATE_DEVICEARRIVAL</i> bis <i>SENSORHW_STATE_DEVICEUNKNOWN</i> (siehe Tabelle 5-7) einnehmen.
long	wEventTextHandle	Text-Handle, über das die Funktion <i>NrpGetDeviceChangedMsgText</i> die Klartextbeschreibung des Ereignisses ermittelt.
char*	pEventText	Zeiger auf die Klartextbeschreibung des Ereignisses. Dieser Zeiger ist nur in einer C- oder C++-Umgebung zu verwenden. Für Visual Basic müssen die Funktionen <i>NrpGetDeviceChangedMsgText</i> und <i>NrpGetDeviceChangedMsgTextLength</i> verwendet werden.
char*	pDeviceName	Der Name des Gerätes, das das Ereignis ausgelöst hat. Dieser Zeiger ist nur in einer C- oder C++-Umgebung zu verwenden. Für Visual Basic müssen die Funktionen <i>NrpGetDeviceChangedDevName</i> und <i>NrpGetDeviceChangedDevNameLength</i> verwendet werden.

NrpGetDeviceChangedMsgText

Die Callback-Funktion, die durch das *DeviceChanged*-Ereignis aufgerufen wird, liefert als Argument die Struktur *USB_DEVICECHANGED_HANDLER_RESULT* (siehe Tabelle 5-6). Eines der Strukturmitglieder ist *wEventTextHandle*, ein *LONG*-Wert, hinter dem sich ein DLL-interner String-Listen-Index verbirgt. Mit *NrpGetDeviceChangedMsgText* lässt sich die in der String-Liste abgelegte Ereignismeldung in einen Puffer kopieren. Die Funktion liefert als Rückgabewert die Anzahl der tatsächlich kopierten Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der Zeichen in der Ereignismeldung. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp:

`long NrpGetDeviceChangedMsgText (long wEventTextHandle, char *o_pBuf, long i_wCount);`

Parameter: *wEventTextHandle* String-Listen-Index
o_pBuf Zeiger auf den Puffer, der die Ereignismeldung aufnehmen soll
i_wCount Puffergröße in Bytes

NrpGetDeviceID

NrpGetDeviceID liefert die laufende Nummer des zur Zeit als aktiv konfigurierten Messkopfes. Jeder angeschlossene Messkopf erhält eine solche laufende Nummer. Die Zählung beginnt mit 0.

Prototyp: long NrpGetDeviceID (void);

NrpGetIDByLogicalName

Jeder Messkopf hat eine eindeutige Kombination aus Typenbezeichnung und Seriennummer, welcher beim ersten Anschluss an den USB automatisch ein logischer Name zugeordnet wird (Sensor1, Sensor2 usw.). Diese Zuordnungen sind in der Registry unter *HKEY_CURRENT_USER\Software\Rohde&Schwarz\NrpDII\Sensors* eingetragen. Sie können mit Hilfe der Funktion *NrpChannelAssignment* geändert werden. *NrpGetIDByLogicalName* liefert die laufende Nummer eines angeschlossenen Messkopfes, wenn der übergebene logische Name in der Registry gefunden wurde. Wurde der logische Name nicht in der Registry gefunden oder ist der betreffende Messkopf nicht angeschlossen, dann liefert die Funktion -1.

Prototyp: long NrpGetIDByLogicalName (const char *i_pLogicalName);

Parameter: *i_pLogicalName* logischer Name des Messkopfes als nullterminierter String

NrpGetLogicalName

Jeder Messkopf hat eine eindeutige Kombination aus Typenbezeichnung und Seriennummer, welcher beim ersten Anschluss an den USB automatisch ein logischer Name zugeordnet wird (Sensor1, Sensor2 usw.). Diese Zuordnungen sind in der Registry unter *HKEY_CURRENT_USER\Software\Rohde&Schwarz\NrpDII\Sensors* eingetragen. Sie können mit Hilfe der Funktion *NrpChannelAssignment* geändert werden. *NrpGetLogicalName* kopiert den logischen Namen des angeschlossenen Messkopfes mit der als Argument übergebenen laufenden Nummer in einen bereitgestellten Puffer. Als Rückgabewert liefert die Funktion die Anzahl der tatsächlich kopierten Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der Zeichen im logischen Namen. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp: long NrpGetLogicalName (char *o_pBuf, long i_wCount, long i_wID);

Parameter: *o_pBuf* Zeiger auf den Puffer, der den logischen Namen aufnehmen soll
i_wCount Puffergröße in Bytes
i_wID laufende Nummer des Messkopfes

NrpGetManufacturerCode

NrpGetManufacturerCode kopiert die Herstellerbezeichnung des angeschlossenen Messkopfes mit der als Argument übergebenen laufenden Nummer in einen bereitgestellten Puffer. Als Rückgabewert liefert die Funktion die Anzahl der tatsächlich kopierten Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der Zeichen im logischen Namen. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp:

```
long NrpGetManufacturerCode(char *o_pBuf, long i_wCount, long i_wID);
```

Parameter:

<i>o_pBuf</i>	Zeiger auf den Puffer, der die Herstellerbezeichnung aufnehmen soll
<i>i_wCount</i>	Puffergröße in Bytes
<i>i_wID</i>	laufende Nummer des Messkopfes

NrpGetNrOfDevices

NrpGetNrOfDevices liefert als Rückgabewert die Anzahl der an den USB angeschlossenen Messköpfe.

Prototyp: long NrpGetNrOfDevices (void);

NrpGetProductID

NrpGetProductID kopiert die Typenbezeichnung des angeschlossenen Messkopfes mit der als Argument übergebenen laufenden Nummer in einen bereitgestellten Puffer. Als Rückgabewert liefert die Funktion die Anzahl der tatsächlich kopierten Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der Zeichen in der Typenbezeichnung. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp:

```
long NrpGetProductID(char *o_pBuf, long i_wCount, long i_wID);
```

Parameter:

<i>o_pBuf</i>	Zeiger auf den Puffer, der die Typenbezeichnung aufnehmen soll
<i>i_wCount</i>	Puffergröße in Bytes
<i>i_wID</i>	laufende Nummer des Messkopfes

NrpGetSerialNumber

NrpGetSerialNumber kopiert den Seriennummernstring des angeschlossenen Messkopfes mit der als Argument übergebenen laufenden Nummer in einen bereitgestellten Puffer. Als Rückgabewert liefert die Funktion die Anzahl der tatsächlich kopierten Zeichen.

Übergibt man der Funktion als Argument *o_pBuf* einen Nullpointer, so liefert die Funktion die Anzahl der Zeichen im Seriennummernstring. Dabei wird das terminierende Nullzeichen nicht mitgezählt. Soll das terminierende Nullzeichen mit ausgelesen werden, so muss folglich ein um 1 größerer Puffer übergeben werden.

Prototyp:

```
long NrpGetProductID(char *o_pBuf, long i_wCount, long i_wID);
```

Parameter:

<i>o_pBuf</i>	Zeiger auf den Puffer, der den Seriennummernstring aufnehmen soll
<i>i_wCount</i>	Puffergröße in Bytes
<i>i_wID</i>	laufende Nummer des Messkopfes

NrpSDRGetDescriptor

NrpSDRGetDescriptor sendet den *Standard Device Request GetDescriptor* an den Messkopf. Dieser *Standard Device Request* dient zum Auslesen von Device-, Configuration- und String-Deskriptoren. Interface- und Endpoint-Deskriptoren können nicht direkt, sondern nur zusammen mit dem Configuration Descriptor ausgelesen werden. Als Rückgabewert liefert die Funktion den Fehlerstatus nach Tabelle 5-4. Der Aufbau der Deskriptoren ist in der USB-Fachliteratur detailliert beschrieben.

Prototyp:

```
long NrpSDRGetDescriptor (void *o_pBuffer, long *io_wByteCount,
                          long i_wRecipient, long i_wDescrType,
                          long i_wDescrIndex, long i_wLangID);
```

Parameter:	<i>o_pBuffer</i>	Zeiger auf den Puffer, der die Deskriptordaten aufnehmen soll bei Aufruf: Puffergröße in Bytes
	<i>io_wByteCount</i>	nach Aufruf: Anzahl der tatsächlich im Puffer abgelegten Bytes
	<i>i_wRecipient</i>	0 = Device, 1 = Interface, 2 = Endpoint, 3 = Andere
	<i>i_wDescrType</i>	Typ des Deskriptors (1 = Device Descriptor, 2 = Configuration Descriptor inkl. zugehörige Interface-, Klassen- und Endpoint-Deskriptoren, 3 = String Descriptor)
	<i>i_wDescrIndex</i>	nur bei String Descriptor: Index des Strings für Herstellerbezeichnung, Produktbezeichnung oder Seriennummer
	<i>i_wLangID</i>	nur bei String Descriptor: Sprachenindex (optional, 0x0409 = Englisch), sonst 0

NrpSelectDevice

NrpSelectDevice selektiert den angeschlossenen Messkopf mit der als Argument übergebenen laufenden Nummer.

Prototyp: void NrpSelectDevice (long i_wID);

Parameter: *i_wID* laufende Nummer des Messkopfes

Anwendungsbeispiele

Die folgenden Beispiele demonstrieren die Funktionalität des Messkopfs. Zum Kennenlernen der Befehle eignet sich das Programmmodul **USB-Terminal** des NRP-Toolkits. Sofort, nachdem der Messkopf an den USB-Anschluss des PC angeschlossen wurde, ist er einsatzbereit.

Befehle müssen dem Messkopf mit Hilfe der Funktion *NrpSendCommand* übermittelt werden. Der Übersichtlichkeit halber werden im Folgenden nur die SCPI-Befehlsfolgen dargestellt. Es wird generell die Kurzform der Befehle verwendet. Für nähere Erläuterungen zu den einzelnen Befehlen siehe Befehlsreferenz in Abschnitt 6 dieses Benutzerhandbuchs.

Vor der Messung sollte mit dem Messkopf bei abgeschalteter Leistung ein Nullabgleich durchgeführt werden.

```
cal:zero:auto once           oder   cal:zero:auto on
```

Zusätzlich muss bei jeder Messung die HF-Trägerfrequenz (im folgenden Beispiel 500 MHz) dem Sensor mitgeteilt werden.

```
sens:freq 500e6
```

Einfache Leistungsmittelwertmessung

Um den Leistungsmittelwert zu messen, sind im einfachsten Fall nur folgende Befehle notwendig:

```
*rst           setzt den Messkopf auf die Voreinstellungen zurück
init:imm       startet die Messung
```

In allen Messmodi steht ein Mittelungsfilter zur Verfügung. Dieses Filter kann entweder mit einem manuell eingestellten oder einem vom Messkopf automatisch ermittelten Averaging-Faktor betrieben werden. Bei Nutzung der Voreinstellung ermittelt der Messkopf den Averaging-Faktor automatisch. Zur Ermittlung des Averaging-Faktors kann zwischen zwei Algorithmen gewählt werden: Beim klassischen Verfahren wird die gewünschte Auflösung vorgegeben, und der Messkopf ermittelt die Filterlänge anhand der anliegenden Leistung, wobei eine Obergrenze für die Messzeit (Parameter *SENSe:AVERage:COUNT:AUTO:MTIME*) eingehalten wird. Mit dem neuen „Fixed Noise“-Verfahren (*SENSe:AVERage:COUNT:AUTO:TYPE NSRatio*) wird dagegen ein genau definierter Rauschanteil im Messergebnis eingehalten.

Zwei Verhaltensweisen des Mittelungsfilters werden unterschieden. Der Parameter *SENSe:AVERage:TCONtrol* kann die Werte *MOVing* und *REPeat* annehmen.

- Im Fernsteuerbetrieb ist in der Regel die Einstellung *REPeat* vorteilhaft. Hier wird der Mittelwert erst ausgegeben, wenn das Filter vollständig gefüllt ist. Im Modus *Continuous Average* wird zum Füllen des Mittelungsfilters nur ein Triggerereignis zum Start der Messung benötigt, die restlichen erforderlichen Messungen werden automatisch durchgeführt. In den Modi *Burst Average*, *Timeslot* und *Scope* ist wegen des Zeitbezugs ein Triggerereignis pro Messwert notwendig.
- Bei der Einstellung *MOVing* wird jedes Mal, wenn ein Messwert in das Filter geschoben wird, der Mittelwert gebildet und ausgegeben. Dieses Filterverhalten lässt es zu, frühzeitig eine Tendenz im angezeigten Mittelwert zu erkennen. Es kommt auch bei der Handbedienung über das Grundgerät zum Einsatz.

Beispiel für eine Konfiguration mit manuellem Averaging und *REPeating*-Verhalten:

```
sens:aver:coun:auto off
sens:aver:coun 8
sens:aver:tcon rep
```

Beispiel für eine Konfiguration mit Auto-Averaging mit „Fixed Noise“-Verfahren (Rauschanteil: 0,01 dB, Obergrenze für die Messzeit: 30 s):

```
sens:aver:coun:auto on
sens:aver:coun:auto:type nsr
sens:aver:coun:auto:nsr 0.01
sens:aver:count:auto:mtim 30
```

Die Einstellungen des Mittelungsfilters sind für die Modi *Continuous Average*, *Burst Average* und *Timeslot* gleich und werden beim Moduswechsel übernommen. Für den Modus *Scope* wird das Filter gesondert konfiguriert, die Befehle beginnen mit *SENSe:SWEep:AVERage*: ...

Schnelle und mehrfache Leistungsmittelwertmessung

Für schnelle aufeinanderfolgende Messungen stellt der Messkopf einen Messwertpuffer mit einstellbarer Größe zur Verfügung. Erst wenn dieser Puffer komplett mit Messwerten gefüllt ist, wird sein Inhalt zum Steuergerät (Grundgerät/PC) übertragen. Durch die Übertragung in *FLOAT*-Array-Blöcken wird eine besonders zeitsparende Übertragung möglich. Folgendes Beispiel zeigt die Einstellungen für den schnellstmöglichen Betrieb:

*rst	setzt den Messkopf auf die Voreinstellungen zurück
sens:freq 500e6	HF-Trägerfrequenz einstellen (hier 500 MHz)
sens:aver:stat off	schaltet das Mittelungsfiler aus
sens:pow:avg:aper 100e-6	kürzestes Messfenster einstellen (hier 100 µs)
sens:pow:avg:buff:size 100	Messwertpuffer soll 100 Messwerte aufnehmen
sens:pow:avg:buff:stat on	Messwertepufferung einschalten
trig:sour imm	freilaufender Trigger
trig:coun 100	100 Messwerte aufnehmen
init:imm	Messung starten

Der Befehl `trig:count 100` sorgt dafür, dass nach einem einmaligen Messstart mit `init:imm` 100 Messungen durchgeführt werden, um den bereitgestellten Puffer zu füllen. Würde dagegen hier die Voreinstellung `trig:count 1` verwendet, müsste für dieselbe Anzahl von Messergebnissen 100 Mal nacheinander `init:imm` aufgerufen werden.

Messung eines einzelnen Bursts

Der Messkopf ist im Modus *Burst Average* in der Lage, einzelne Bursts selbständig zu erkennen und ihren Leistungsmittelwert zu messen. In diesem Messmodus wartet der Messkopf auf eine steigende Triggerflanke und misst so lange, bis er eine fallende Flanke erkennt. Sollte innerhalb von 50 ms kein Burstende erkannt werden, bricht der Messkopf die Messung ab und gibt die Fehlermeldung *NRPERERROR_TRIGGERQUEUEFULL* aus.

*rst	setzt den Messkopf auf die Voreinstellungen zurück
sens:func "pow:burs:avg"	Modus <i>Burst Average</i> einstellen
trig:lev 1e-6	Triggerschwelle auf 1 µW einstellen
sens:pow:burs:dtol 50e-6	Dropout-Toleranz einstellen
init:imm	Messung starten

Zusätzlich können noch Bereiche zu Beginn und am Ende des Bursts ausgeschlossen werden (z. B. um einen Einfluss von flachen Signalfanken oder Überschwingern auf das Messergebnis auszuschließen). Im folgenden Beispiel werden die ersten 20 μs und die letzten 30 μs des Bursts von der Messung ausgeschlossen:

```
sens:tim:excl:star 20e-6
sens:tim:excl:stop 30e-6
```

Messung eines Frames mit mehreren Zeitfenstern (Timeslots)

Für die Messung der Leistung in mehreren aufeinanderfolgenden Zeitfenstern (Timeslots) in einem festen Zeitrahmen stellt der Messkopf den Modus *Timeslot* zur Verfügung. Für die Messung müssen Anzahl und Dauer der Zeitfenster eingestellt werden. Getriggert wird eine Messung durch das Messsignal selbst (*TRIGger:SOURce INTernal*) oder durch ein von außen zugeführtes Triggersignal (*TRIGger:SOURce EXTernal*). Um Messfehler durch unpräzise Triggerung zu vermeiden, ist einer externen Triggerquelle der Vorzug zu geben. Eine Triggerung über den USB (*TRIGger:SOURce BUS*) ist zwar prinzipiell möglich, wird aber aufgrund der ungenauen zeitlichen Auflösung nicht empfohlen.

Im folgenden Beispiel wird gezeigt, wie die Leistung in den acht Timeslots eines GSM-/EDGE-Signals gemessen werden kann:

<code>*rst</code>	setzt den Messkopf auf die Voreinstellung zurück
<code>sens:func "pow:tsl:avg"</code>	Modus <i>Timeslot</i> wählen
<code>sens:freq 500e6</code>	HF-Trägerfrequenz einstellen (hier 500 MHz)
<code>sens:aver:coun:auto off</code>	Auto-Averaging deaktivieren
<code>sens:aver:coun 8</code>	Averaging-Faktor 8 einstellen
<code>trig:sour ext</code>	externe Triggerquelle wählen (Hier kann auch <i>int</i> für interne Triggerung angegeben werden, dann muss zusätzlich die Triggerschwelle mit <i>TRIGger:LEVel</i> festgelegt werden.)
<code>trig:slop pos</code>	Triggerung auf steigende Flanke des Triggersignals
<code>sens:pow:tsl:avg:coun 8</code>	Anzahl der aufeinanderfolgenden Timeslots (hier 8)
<code>sens:pow:tsl:avg:widt 570e-6</code>	Breite eines Timeslots in Sekunden (hier 570 μs = Breite eines GSM-Timeslots)
<code>init:imm</code>	Messung starten

Wie im Modus *Burst* können auch im Modus *Timeslot* Start- und Endbereiche der Timeslots von der Messung ausgenommen werden. Die Einstellparameter sind dieselben wie im Modus *Burst Average*:

```
sens:tim:excl:star 50e-6    die ersten 50  $\mu\text{s}$  jedes Timeslots werden ausgeblendet
sens:tim:excl:stop 80e-6   die letzten 80  $\mu\text{s}$  jedes Timeslots werden ausgeblendet
```



Die Parameter *SENSe:TIMing:EXCLude:STARt* und *-STOP* gelten sowohl für den Modus *Burst Average* als auch für den Modus *Timeslot*, d. h. die in einem der beiden Modi gemachten Einstellungen werden beim Wechsel in den jeweils anderen Modus übernommen.

Scopemodus zur Darstellung von periodischen Signalen

Um den zeitlichen Verlauf der Leistung ähnlich wie mit einem Oszilloskop darzustellen, bietet der Messkopf den Modus *Scope*. Wie im Modus *Timeslot* ist eine Triggerung durch das Messsignal selbst oder ein von außen zugeführtes Triggersignal möglich. Um Messfehler durch unpräzise Triggerung zu vermeiden, ist auch hier einer externen Triggerquelle der Vorzug zu geben.

Im Modus *Scope* stehen separate Parameter für das Mittelungsfilter zur Verfügung.

Um den Messkopf für eine *Scope*-Messung zu konfigurieren, muss angegeben werden, welche Zeit durch die *Scope*-Darstellung abgedeckt werden soll, und in wie viele Intervalle die Darstellung aufgelöst werden soll. Die kleinste zeitliche Auflösung des Messkopfes beträgt 10 µs. Es ist daher zu beachten, dass bei der Einstellung der o. g. Parameter die Intervallbreite (= Zeit / Anzahl der Intervalle) diese Zeit nicht unterschritten wird, da sonst die *Scope*-Darstellung verfälscht wird.

Das folgende Beispiel zeigt die Nutzung des Modus *Scope*, wobei der Trigger aus dem Messsignal abgeleitet wird:

<code>*rst</code>	setzt den Messkopf auf die Voreinstellung zurück
<code>sens:func "xtim:pow"</code>	Modus <i>Scope</i> wählen
<code>sens:freq 500e6</code>	HF-Trägerfrequenz einstellen (hier 500 MHz)
<code>sens:swe:aver:coun:auto off</code>	Auto-Averaging deaktivieren
<code>sens:swe:aver:coun 8</code>	Averaging-Faktor 8 einstellen
<code>trig:sour int</code>	Triggerung durch das Messsignal wählen
<code>trig:slop pos</code>	Triggerung auf steigende Flanke des Messsignals
<code>trig:lev 1e-6</code>	Triggerschwelle einstellen (hier 1 µW)
<code>trig:hyst 3</code>	Triggerhysterese einstellen (hier 3 dB, d. h. eine erneute Triggerung durch Überschreiten der Triggerschwelle wird erst möglich, nachdem der Pegel des Messsignals die Triggerschwelle um mindestens 3 dB unterschritten hat)
<code>sens:swe:time 20e-3</code>	Zeit in Sekunden, die von der <i>Scope</i> -Darstellung abgedeckt werden soll (hier 20 ms)
<code>sens:swe:poin 200</code>	Anzahl der Intervalle, in die die <i>Scope</i> -Darstellung aufgelöst werden soll (hier 200, d. h. Auflösung = 20 ms / 200 = 100 µs)
<code>init:imm</code>	Messung starten

Scopemodus zur Darstellung eines einmaligen Vorgangs (Single-Shot-Messung)

Ein einmaliges, nichtperiodisches Signal kann im Modus *Scope* erfasst werden, wenn der Parameter *SENSe:SWEp:REALtime ON* gesetzt wird. Eine Mittelung ist bei einmaligen Vorgängen natürlich nicht möglich. Die Einstellungen des Mittelungsfilters werden daher ignoriert (und auch nicht verändert). Anders als in den letzten beiden Betriebsarten ist hier eine Bustringerung möglich, da nicht mehrere Messungen durch die Mittelung überlagert werden.

Das folgende Beispiel zeigt die Einstellungen für die Single-Shot-Messung:

<code>*rst</code>	setzt den Messkopf auf die Voreinstellung zurück
<code>sens:func "xtim:pow"</code>	Modus <i>Scope</i> wählen
<code>sens:freq 500e6</code>	HF-Trägerfrequenz einstellen (hier 500 MHz)
<code>sens:swe:time 20e-3</code>	Zeit in Sekunden, die von der <i>Scope</i> -Darstellung abgedeckt werden soll (hier 20 ms)
<code>sens:swe:poin 200</code>	Anzahl der Intervalle, in die die <i>Scope</i> -Darstellung aufgelöst werden soll (hier 200, d. h. Auflösung = 20 ms / 200 = 100 µs)
<code>sens:swe:real on</code>	Single-Shot-Modus aktivieren
<code>trig:sour ext</code>	externe Triggerquelle wählen (Hier kann auch <i>int</i> für interne Triggerung angegeben werden, dann muss zusätzlich die Triggerschwelle mit <i>TRIGger:LEVel</i> festgelegt werden.)
<code>trig:slop pos</code>	Triggerung auf steigende Flanke des Triggersignals
<code>init:imm</code>	Messung starten

Für eine busgetriggerte Single-Shot-Messung sind folgende Befehle notwendig:

<code>trig:sour hold</code>	alle Triggerereignisse außer <i>TRIGger:IMMEDIATE</i> ignorieren
<code>init:imm</code>	Messung initialisieren, Triggersystem wartet auf Triggerereignis
<code>trig:imm</code>	Single-Shot-Messung triggern

Hier ist zu beachten, dass durch den Befehl *TRIGger:IMMEDIATE* die Triggerverzögerungszeit (Parameter *TRIGger:DElay*) ignoriert wird.

Inhaltsübersicht

6	Fernbedienung – Befehle	6.1
	Notation	6.1
	Befehle nach IEEE 488.2	6.2
	*IDN? – IDentification Query	6.2
	*RST – Reset	6.2
	*TRG – Trigger	6.2
	*TST? – Self Test Query	6.2
	SCPI-Befehle	6.3
	CALibration (Kalibrierung).....	6.3
	CALibration:DATA[?] <Kalibrierdatensatz als <i>definite length block</i> >	6.3
	CALibration:DATA:LENGth?	6.3
	CALibration:ZERO:AUTO[?] OFF ON ONCE	6.4
	SENSe (Messkopf-Konfiguration)	6.5
	SENSe:AVERAge:COUNt[?] 1 ... 65536	6.7
	SENSe:AVERAge:COUNt:AUTO[?] OFF ON ONCE	6.7
	SENSe:AVERAge:COUNt:AUTO:MTIMe[?] 1.0 ... 999.99	6.7
	SENSe:AVERAge:COUNt:AUTO:NSRatio[?] 0.0001 ... 1.0.....	6.8
	SENSe:AVERAge:COUNt:AUTO:RESolution[?] 1 ... 4	6.8
	SENSe:AVERAge:COUNt:AUTO:SLOT[?] 1 ... <SENSe:POWer:TSLot:AVG:COUNt>	6.8
	SENSe:AVERAge:COUNt:AUTO:TYPE[?] RESolution NSRatio	6.8
	SENSe:AVERAge:STATe[?] OFF ON.....	6.9
	SENSe:AVERAge:TCONtroll[?] MOVing REPeat.....	6.9
	SENSe:CORRection:DCYClE[?] 0.001 ... 99.999	6.9
	SENSe:CORRection:DCYClE:STATe[?] OFF ON	6.9
	SENSe:CORRection:OFFSet[?] -100.0 ... 100.0.....	6.10
	SENSe:CORRection:OFFSet:STATe[?] OFF ON.....	6.10
	SENSe:CORRection:SPDev:STATe[?] OFF ON	6.10
	SENSe:FREQUency[?] 0.0 ... 100.0e9	6.10
	SENSe:FUNcTION[?] <sensor_function>	6.11
	SENSe:POWer:AVG:APERture[?] 0.0001 ... 0.1	6.12
	SENSe:POWer:AVG:BUFFer:SIZE[?] 1 ... 1024.....	6.12
	SENSe:POWer:AVG:BUFFer:STATe[?] OFF ON.....	6.12
	SENSe:POWer:AVG:SMOothing:STATe[?] OFF ON	6.13
	SENSe:POWer:BURSt:DTOLerance[?] 0.0 ... 0.003	6.13
	SENSe:POWer:TSLot:AVG:COUNt[?] 1 ... 128.....	6.14
	SENSe:POWer:TSLot:AVG:WIDTh[?] 0.0001 ... 0.1	6.14
	SENSe:RANGe[?] 0 ... 2.....	6.14
	SENSe:RANGe:AUTO[?] OFF ON	6.15
	SENSe:RANGe:AUTO:CLEVel[?] -20.0 ... 0.0	6.15
	SENSe:SAMPLing[?] FREQ1 FREQ2.....	6.15
	SENSe:SGAMma:CORRection:STATe[?] OFF ON.....	6.15
	SENSe:SGAMma:MAGNitude[?] 0.0 ... 1.0.....	6.15

SENSe:SGAMma:PHASe[?]	-360.0 ... 360.0	6.16
SENSe:SWEep:AVERAge:COUNt[?]	1 ... 65536	6.16
SENSe:SWEep:AVERAge:COUNt:AUTO[?]	OFF ON ONCE	6.16
SENSe:SWEep:AVERAge:COUNt:AUTO:MTIME[?]	1.0 ... 999.99	6.16
SENSe:SWEep:AVERAge:COUNt:AUTO:NSRatio[?]	0.0001 ... 1.0	6.17
SENSe:SWEep:AVERAge:COUNt:AUTO:RESolution[?]	1 ... 4	6.17
SENSe:SWEep:AVERAge:COUNt:AUTO:POINt[?]	1 ... <SENSe:SWEep:POINts>	6.17
SENSe:SWEep:AVERAge:COUNt:AUTO:TYPE[?]	RESolution NSRatio	6.17
SENSe:SWEep:AVERAge:STATe[?]	OFF ON	6.18
SENSe:SWEep:AVERAge:TCONtroll[?]	MOVing REPeat	6.18
SENSe:SWEep:OFFSet:TIME[?]	0.0 ... <TRIGger:DELay> + 0.005	6.18
SENSe:SWEep:POINts[?]	1 ... 1024	6.18
SENSe:SWEep:REALtime[?]	OFF ON	6.18
SENSe:SWEep:TIME[?]	0.0001 ... 0.3	6.19
SENSe:TIMing:EXCLude:STARt[?]	0.0 ... 0.1	6.19
SENSe:TIMing:EXCLude:STOP[?]	0.0 ... 0.003	6.19
SYSTem		6.21
SYSTem:INFO? [Item]		6.21
SYSTem:INITialize		6.22
SYSTem:MINPower?		6.22
SYSTem:TRANsaction:BEgIn		6.23
SYSTem:TRANsaction:END		6.23
TEST		6.24
TEST:SENSor?		6.24
TRIGger		6.25
ABORt		6.25
INITiate:CONTInuous[?]	OFF ON	6.25
INITiate:IMMediate		6.26
TRIGger:ATRigger:STATe[?]	OFF ON	6.26
TRIGger:COUNt[?]	1 ... 2×10^9	6.26
TRIGger:DELay[?]	x ... 100.0	6.26
TRIGger:DELay:AUTO[?]	OFF ON	6.27
TRIGger:HOLDoff[?]	0.0 ... 10.0	6.27
TRIGger:HYSteresis[?]	0.0 ... 10.0	6.27
TRIGger:IMMediate		6.27
TRIGger:LEVel[?]	100.0e-9 ... 0.2	6.27
TRIGger:SLOPe[?]	POSitive NEGative	6.28
TRIGger:SOURce[?]	BUS EXTernal HOLD IMMediate INTernal	6.28
Liste der Fernsteuer-Befehle		6.29

Bilder

Bild 6-1	Wirkung von <i>SENSe:POWer:BURSt:DTOLerance</i>	6.14
Bild 6-2	Wirkung von <i>SENSe:TIMing:EXCLude:STARt</i> und <i> -:END</i> im <i>Burst Average</i> -Modus....	6.19
Bild 6-3	Wirkung von <i>SENSe:TIMing:EXCLude:STARt</i> und <i> -:END</i> im <i>Timeslot</i> -Modus.....	6.20

Tabellen

Tabelle 6-1	Befehle des Befehlssystems <i>CALibration</i>	6.3
Tabelle 6-2	Befehle des Befehlssystems <i>SENSe</i>	6.5
Tabelle 6-3	Messmodi	6.11
Tabelle 6-4	Optimale Wahl der Größe des Sampling Window (N = 1, 2, 3, ...).....	6.12
Tabelle 6-5	Befehle des Befehlssystems <i>SYSTEM</i>	6.21
Tabelle 6-6	Bedeutung des <i>Item</i> beim Befehl <i>SYSTEM:INFO?</i>	6.21
Tabelle 6-7	Befehle des Befehlssystems <i>TEST</i>	6.24
Tabelle 6-8	Befehle des Befehlssystems <i>TRIGger</i>	6.25
Tabelle 6-9	Liste der Fernsteuer-Befehle	6.29

6 Fernbedienung – Befehle

Notation

In den folgenden Abschnitten werden alle im Messkopf realisierten Befehle nach Befehlssystemen getrennt zuerst tabellarisch aufgelistet und dann ausführlich beschrieben. Die Schreibweise entspricht dabei weitgehend der des SCPI-Normenwerks.

Befehlstabellen Den Beschreibungen der Befehlssysteme ist eine Tabelle vorangestellt, die einen schnellen Überblick über die einzelnen Befehle liefert. Diese Tabellen enthalten die folgenden vier Spalten:

- Befehl:** Die Befehle und ihre hierarchische Anordnung.
Parameter: Die möglichen Parameter.
Einheit: Die Grundeinheit der physikalischen Parameter (darf nicht mitgesendet werden).
Bemerkung: Kennzeichnung aller Befehle,
- für die keine Abfrageform existiert,
 - die nur als Abfragebefehl existieren.

Einrückungen Die verschiedenen Ebenen der SCPI-Befehlshierarchie sind in der Tabelle durch Einrücken nach rechts dargestellt. Je tiefer die Ebene liegt, desto weiter wird nach rechts eingerückt. Es ist zu beachten, dass die vollständige Schreibweise des Befehls immer auch die höheren Ebenen mit einschließt.

Beispiel:

SENSe:AVERage:COUNt ist in der Tabelle so dargestellt:

```
SENSe      erste Ebene
  :AVERage  zweite Ebene
    :COUNt   dritte Ebene
```

In der individuellen Beschreibung ist der Befehl in seiner gesamten Länge dargestellt. Ein Beispiel zu jedem Befehl befindet sich am Ende der individuellen Beschreibung.

[?]
?

Ein Fragezeichen in eckigen Klammern am Ende eines Befehls zeigt an, dass dieser Befehl nicht nur als Einstellbefehl (ohne Fragezeichen), sondern auch als Abfragebefehl (mit Fragezeichen) eingesetzt werden kann. Steht das Fragezeichen nicht in eckigen Klammern, dann ist der Befehl ein reiner Abfragebefehl.

Beispiel:

SENSe:POWer:AVG:APERture[?]

SENSe:POWer:AVG:APERture 1e-3 stellt die Länge des Abtastfensters auf 1 ms ein.

SENSe:POWer:AVG:APERture? liefert als Antwort die aktuell eingestellte Länge.

**IDN?* erfragt den Identifikationsstring des Messkopfes, der sich verständlicherweise nicht ändern lässt. Daher existiert dieser Befehl nur in der Abfrageform.

Sonderzeichen | bei Parametern Ein senkrechter Strich zwischen Parametern kennzeichnet die verschiedenen Möglichkeiten, die hier zur Auswahl stehen (Oder-Verknüpfung).

Beispiel:

NITiate:CONTinuous OFF | ON

Als Parameter lässt sich entweder *OFF* oder *ON* angeben.

{numerischer Ausdruck}

Geschweifte Klammern um einen numerischen Ausdruck bedeuten eine Rundung auf den nächstliegenden ganzzahligen Wert.

<Parameter>
<Variable>

Dreieckige Klammern um einen Parameter oder eine Variable bedeuten dessen bzw. deren aktuellen Wert.

Befehle nach IEEE 488.2

Der Messkopf unterstützt eine Untermenge der möglichen Einstellbefehle und Abfragen (*Common Commands and Queries*) nach IEEE 488.2.

*IDN? – IDentification Query

*IDN? liefert einen String, mit dem der Messkopf Auskunft über seine Identität gibt (Geräteerkennung). Außerdem wird die Versionsnummer der installierten Firmware angegeben. Der String ist für einen Messkopf vom Typ R&S NRP-Z21 folgendermaßen aufgebaut:

ROHDE&SCHWARZ,NRP-Z21,<Seriennummer>,<Firmware-Version>

<Seriennummer>: ASCII-Darstellung der Seriennummer

<Firmware-Version>: ASCII-Darstellung der Versionsnummer der Firmware

*RST – Reset

*RST versetzt den Messkopf in den Grundzustand, d. h. die Voreinstellungen für alle Messparameter werden geladen.

*TRG – Trigger

*TRG löst eine Messung aus. Dazu muss sich der Messkopf im Zustand *WAIT_FOR_TRIGGER* befinden und die Quelle für das Triggerereignis auf *BUS* eingestellt sein (*TRIGger:SOURce BUS*).

*TST? – Self Test Query

*TST? startet einen Selbsttest und liefert als Ergebnis 0 (kein Fehler festgestellt) oder 1 (es ist ein Fehler aufgetreten). Der Selbsttest umfasst folgende Funktionen:

- RAM-Speichertest,
- Betriebsspannungen,
- Temperaturmessung,
- Kalibrierdatensatz,
- Rauschen,
- Nullpunktoffsets.

SCPI-Befehle

Die Messköpfe R&S NRP-Z11/-Z21 werden über die Befehlsgruppen

- CALibration (Nullabgleich),
- SENSE (Messkonfigurationen),
- SYSTem,
- TRIGger,
- SERVice

gesteuert.

CALibration (Kalibrierung)

Tabelle 6-1 Befehle des Befehlssystems *CALibration*

Befehl	Parameter	Einheit	Bemerkung
CALibration			
:DATA[?]	<Kalibrierdatensatz als definite length block>		
:LENGth?		Bytes	nur Abfrage
:ZERO			
:AUTO[?]	OFF ON ONCE		

CALibration:DATA[?] <Kalibrierdatensatz als *definite length block*>

CALibration:DATA dient zum Schreiben eines Kalibrierdatensatzes in den Flash-Speicher des Messkopfes.

Der Abfragebefehl liefert den aktuell im Flash-Speicher befindlichen Kalibrierdatensatz als *definite length block*.

CALibration:DATA:LENGth?

CALibration:DATA:LENGth? liefert die Länge des aktuell im Flash-Speicher befindlichen Kalibrierdatensatzes in Bytes. Diese Angabe kann von Programmen, die den Kalibrierdatensatz auslesen, verwendet werden, um die Größe des dafür benötigten Pufferspeichers zu ermitteln.

CALibration:ZERO:AUTO[?] OFF | ON | ONCE

Die Befehle *CALibration:ZERO:AUTO ON* oder *CALibration:ZERO:AUTO ONCE* führen einen Nullabgleich für die drei Messpfade des Sensors durch. Dazu muss das Messsignal abgeschaltet oder der Messkopf von der Signalquelle getrennt werden. Das Vorhandensein größerer Messleistungen erkennt der Messkopf selbständig, was zum Abbruch des Nullabgleiches und der Ausgabe der Fehlermeldung *NRPEROR_CALZERO* führt. Der Befehl *CALibration:ZERO:AUTO OFF* wird ignoriert. Ein Nullabgleich dauert mindestens vier Sekunden, mindestens aber so lange, wie das gewählte Mittelungsfilter zum Einschwingen braucht (nur Festfiltermodus).



Der Nullabgleich ist zu wiederholen

- *in der Aufwärmphase nach dem Einschalten bzw. Anstecken,*
- *nach plötzlichen Änderungen der Umgebungstemperatur,*
- *nach dem Anschrauben des Messkopfes an einen HF-Anschluss mit erhöhter Temperatur,*
- *generell nach einigen Stunden Betrieb,*
- *wenn Signale mit sehr geringer Leistung gemessen werden sollen, beispielsweise weniger als 10 dB über der unteren Messgrenze.*

Zum Nullabgleich sollte möglichst das Messsignal abgeschaltet und nicht der Messkopf von der Signalquelle abgeschraubt werden. Abgesehen von der Beibehaltung des thermischen Gleichgewichts hat dies den Vorteil, dass dem Messsignal überlagertes Rauschen (z. B. von einem Breitbandverstärker) beim Nullabgleich erfasst werden kann und nicht das Messergebnis verfälscht.

Der Abfragebefehl liefert immer 1 (= OFF).

Voreinstellung

Nach einem Power-on-Reset werden bis zum ersten Nullabgleich die im Rahmen der letzten Kalibrierung ermittelten Nullpunkt-Offsets verwendet. Daher ist generell mit sehr kleinen Nullpunktabweichungen bei eingelaufenem Messkopf zu rechnen. Eine Initialisierung durch **RST* oder *SYSTEM:INITialize* hat keinen Einfluss auf die aktuellen Nullpunktabweichungen.

SENSe (Messkopf-Konfiguration)

Mit den Befehlen der Gruppen *SENSe* und *TRIGger* wird der Messkopf konfiguriert.

Tabelle 6-2 Befehle des Befehlssystems *SENSe*

Befehl	Parameter	Einheit	Bemerkung
SENSe			
:AVERage			
:STATe[?]	OFF ON		
:TCONtrol[?]	MOVing REPeat		
:COUNt[?]	1 ... 65536		
:AUTO[?]	OFF ON ONCE		
:TYPE[?]	RESolution NSRatio		
:MTIME[?]	1.0 ... 999.99	s	
:NSRatio[?]	0.0001 ... 1.0	dB	
:RESolution[?]	1 ... 4		
:SLOT[?]	1 ... <SENSe:POWer:TSLot:AVG:COUNt>		
:CORRection			
:OFFSet[?]	-100.0 ... 100.0	dB	
:STATe[?]	OFF ... ON		
:DCYCLE[?]	0.001 ... 99.999	%	
:STATe[?]	OFF ... ON		
:SPDevice:STATe[?]	OFF ... ON		
:FREQuency[?]	0.0 ... 100.0e9	Hz	
:FUNction[?]	"POWer:AVG" "POWer:TSLot:AVG" "POWer:BURSt:AVG" "XTIME:POWer"		
:POWer			
:AVG			
:APERture[?]	1.0e-6 ... 100.0e-3	s	
:BUFFer			
:STATe[?]	OFF ON		

Befehl	Parameter	Einheit	Bemerkung
:SIZE[?]	1 ... 1024		
:SMOothing:STATe[?]	OFF ON		
:BURSt:DTOLerance[?]	0.0 ... 3.0e-3	s	
:TSLot:AVG			
:COUNT[?]	1 ... 128		
:WIDTh[?]	10e-6 ... 100e-3	s	
:RANGe [?]	0 ... 2		
:AUTO[?]	OFF ON		
:CLEVel[?]	0.0 ... 20.0	dB	
:SAMPling[?]	FREQ1 FREQ2		
:SGAMma			
:CORRection:STATe[?]	OFF ON		
:MAGNitude[?]	0.0 ... 1.0		
:PHASe[?]	-360.0 ... 360.0	Grad	
:SWEep			
:AVERAge			
:STATe[?]	OFF ON		
:TCONtrol[?]	MOVing REPeat		
:COUNT[?]	1 ... 65536		
:AUTO[?]	OFF ON ONCE		
TYPE[?]	RESolution NSRatio		
:MTIME[?]	1.0 ... 999.99	s	
:NSRatio[?]	0.0001 ... 1.0	dB	
:RESolution[?]	1 ... 4		
:POINt[?]	1 ... <SENSe:SWEep:POINts>		
:OFFSet:TIME[?]	- (<TRIGger:DELay> + 0.005) ... 100.0	s	
:POINts[?]	1 ... 1024		
:REALtime[?]	OFF ON		

Befehl	Parameter	Einheit	Bemerkung
:TIME[?]	0.0001 ... 0.3	s	
:TIMing			
:EXCLude			
:STAR[?]	0.0 ... 3.0e-3	s	
:STOP[?]	0.0 ... 3.0e-3	s	

SENSE:AVERage:COUNT[?] 1 ... 65536

Mit *SENSE:AVERage:COUNT* wird eingestellt, wie viele Messwerte zur Bildung des Messergebnisses in den Modi *Continuous Average*, *Burst Average* oder *Timeslot* gemittelt werden sollen. Je höher dieser Mittelungsfaktor gewählt wird, desto weniger schwanken die Messwerte, und desto länger ist die Messzeit. Der Parameter wird auf die nächste Zweierpotenz auf- oder abgerundet.

Der Abfragebefehl liefert den in den Modi *Continuous Average*, *Burst Average* oder *Timeslot* verwendeten Averaging-Faktor.



Damit der eingestellte Mittelungsfaktor wirksam wird, muss die Mittelungsfunktion mit *SENSE:AVERage:STATE ON* eingeschaltet werden.

Voreinstellung: 4

SENSE:AVERage:COUNT:AUTO[?] OFF | ON | ONCE

Mit *SENSE:AVERage:COUNT:AUTO* wird die automatische Bestimmung des Mittelungsfaktors in den Modi *Continuous Average*, *Burst Average* oder *Timeslot* aktiviert (Auto-Averaging) oder deaktiviert (Festfilter-Modus). Wird das Auto-Averaging aktiviert, dann wird der Mittelungsfaktor fortlaufend in Abhängigkeit von der Höhe der Leistung und anderer Parameter neu bestimmt und eingestellt.

Mit *SENSE:AVERage:COUNT:AUTO ON* wird das Auto-Averaging ein-, mit *SENSE:AVERage:COUNT:AUTO OFF* ausgeschaltet. Beim Ausschalten wird der zuletzt automatisch ermittelte Mittelungsfaktor in den Festfiltermodus übernommen. Das Kommando *SENSE:AVERage:COUNT:AUTO ONCE* bewirkt, dass von der Filterautomatik unter den momentanen Messbedingungen einmalig ein neuer Mittelungsfaktor ermittelt und in den Festfiltermodus übernommen wird.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *OFF*

SENSE:AVERage:COUNT:AUTO:MTIME[?] 1.0 ... 999.99

SENSE:AVERage:COUNT:AUTO:MTIME setzt die Obergrenze für die Einschwingzeit des Mittelungsfilters im Auto-Averaging-Modus für die Modi *Continuous Average*, *Burst Average* oder *Timeslot* und begrenzt damit auch dessen Länge.

Der Abfragebefehl liefert die aktuelle Obergrenze für die Einschwingzeit des Mittelungsfilters im Auto-Averaging-Modus für die Modi *Continuous Average*, *Burst Average* oder *Timeslot*.

Voreinstellung: 30.0 [s]

SENSe:AVERAge:COUNT:AUTO:NSRatio[?] 0.0001 ... 1.0

SENSe:AVERAge:COUNT:AUTO:NSRatio legt den relativen Rauschanteil im Messergebnis für die Modi *Continuous Average*, *Burst Average* oder *Timeslot* fest, wenn das Auto-Averaging im entsprechenden Modus (*SENSe:AVERAge:COUNT:AUTO:TYPE NSRatio*) betrieben wird. Der Rauschanteil ist definiert als der Betrag der durch das Eigenrauschen des Messkopfes bewirkten Pegelschwankung in dB (zwei Standardabweichungen).

Der Abfragebefehl liefert den relativen Rauschanteil im Messergebnis für die Modi *Continuous Average*, *Burst Average* oder *Timeslot*.

Voreinstellung: 0.01 [dB]

SENSe:AVERAge:COUNT:AUTO:RESolution[?] 1 ... 4

SENSe:AVERAge:COUNT:AUTO:RESolution stellt den Auflösungsindex für das automatische Mittelungsfiler in den Modi *Continuous Average*, *Burst Average* oder *Timeslot* ein, wenn es im Modus *RESolution* betrieben wird. Der Auflösungsindex ist gleich der Anzahl der Nachkommastellen, die bei einer Weiterverarbeitung des Messergebnisses in dBm, dB μ V oder dB berücksichtigt werden sollen. Der Normal-Modus ist so ähnlich wie bei den Vorgängergeräten R&S NRVS und R&S NRVD bzw. anderen handelsüblichen Leistungsmessern ausgelegt. Je höher der Index gewählt wird, umso besser ist das Messergebnis gefiltert, ohne dass damit erreicht würde, dass die letzte signifikante Stelle (0,01 dB bei einem Index von 3) auch wirklich steht. Zu empfehlen ist stattdessen die Einstellung *NSRatio*.

Der Abfragebefehl liefert den Auflösungsindex für die Modi *Continuous Average*, *Burst Average* oder *Timeslot*.

Voreinstellung: 3

SENSe:AVERAge:COUNT:AUTO:SLOT[?] 1 ... <SENSe:POWer:TSLot:AVG:COUNT>

SENSe:AVERAge:COUNT:AUTO:SLOT definiert das Zeitfenster (Timeslot), auf dessen Leistung sich das Auto-Averaging im *Timeslot*-Modus bezieht. Der Timeslot wird über seine Nummer adressiert, wobei die Zählung mit 1 beginnt. Die Timeslot-Nummer darf die Anzahl der aktuell eingestellten Timeslots nicht übersteigen. Wird zuerst eine gültige Timeslot-Nummer eingestellt und danach die Anzahl der Timeslots auf einen Wert reduziert, der kleiner ist als die Timeslot-Nummer, dann wird diese automatisch gleich der neuen Timeslot-Anzahl gesetzt, d. h. das Auto-Averaging bezieht sich auf den letzten Timeslot.

Der Abfragebefehl liefert die Nummer des aktuellen Zeitfensters (Timeslot's), auf dessen Leistung sich das Auto-Averaging im *Timeslot*-Modus bezieht.

Voreinstellung: 1

SENSe:AVERAge:COUNT:AUTO:TYPE[?] RESolution | NSRatio

SENSe:AVERAge:COUNT:AUTO:TYPE definiert den Modus des automatischen Mittelungsfilters in den Modi *Continuous Average*, *Burst Average* oder *Timeslot*. Mit dem Parameter *RESolution* wird der bei Leistungsmessern übliche Modus eingestellt; mit *NSRatio* kann die Einhaltung eines genau definierten Rauschanteils vorgegeben werden.

Der Abfragebefehl liefert

- 1 für *RESolution*,
- 2 für *NSRatio*.

Voreinstellung: *RESolution*

SENSe:AVERage:STATe[?] OFF | ON

SENSe:AVERage:STATe schaltet das Mittelungfilter für die Modi *Continuous Average*, *Burst Average* oder *Timeslot* aus oder ein.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *ON*

SENSe:AVERage:TCONtrol[?] MOVing | REPeat

SENSe:AVERage:TCONtrol (*terminal control*) legt das Verhalten des Mittelungsfilters in den Modi *Continuous Average*, *Burst Average* oder *Timeslot* fest. Sobald ein neuer Messwert in das FIR-Filter geschoben wird, steht am Filterausgang auch ein neuer Mittelwert zur Verfügung, der aus dem neu hinzugekommenen Messwert und den restlichen im Filter befindlichen Werten gewonnen wird.

Der Parameter *MOVing* legt fest, dass jeder neue Mittelwert als Messergebnis ausgegeben wird. Auf diese Weise können Tendenzen im Messergebnis schon während des Messvorgangs erkannt werden.

Der Parameter *REPeat* legt fest, dass erst dann, wenn das FIR-Filter komplett mit neuen Messwerten aufgefüllt wurde, ein neues Messergebnis ausgegeben wird. Dadurch wird sichergestellt, dass redundante Informationen nicht ausgegeben werden.

Der Abfragebefehl liefert

- 1 für *MOVing*,
- 2 für *REPeat*.

Voreinstellung: *MOVing*

SENSe:CORRection:DCYClE[?] 0.001 ... 99.999

SENSe:CORRection:DCYClE stellt zur Korrektur pulsmodulierter Signale das Tastverhältnis (*duty cycle*) in Prozent ein. Bei aktivierter Korrektur berechnet der Messkopf daraus und aus der mittleren Leistung die Impulsleistung (*pulse power*) des Signals. Weil das Tastverhältniss nur im *Continuous-Average-Modus* nur im *Normal-Modus* Sinn macht, wird es auch nur dort ausgewertet.

Der Abfragebefehl liefert das aktuelle Tastverhältnis in Prozent.

Voreinstellung: *1.0 [%]*

SENSe:CORRection:DCYClE:STATe[?] OFF | ON

SENSe:CORRection:DCYClE:STATe ON aktiviert die Tastverhältnis-Korrektur und damit die Messung der Impulsleistung, *SENSe:CORRection:DCYClE:STATe OFF* deaktiviert sie.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *OFF*

SENSe:CORRection:OFFSet[?] -100.0 ... 100.0

SENSe:CORRection:OFFSet definiert einen festen Offset in dB, mit dem der Messwert korrigiert werden kann (bei logarithmischer Darstellung wird der Offset zum Messwert addiert, daher rührt die Bezeichnung).

Mit einem positiven Offset lässt sich z. B. die Dämpfung eines vor dem Messkopf liegenden Dämpfungsgliedes oder die Auskoppeldämpfung eines Richtkopplers berücksichtigen, d. h. der Messkopf berechnet dann die Leistung am Eingang des Dämpfungsgliedes oder Richtkopplers. Mit einem negativen Offset lässt sich der Einfluss einer vorgeschalteten Verstärkung korrigieren.

Der Abfragebefehl liefert den eingestellten Offset in dB.

Voreinstellung: 0.0 [dB]

SENSe:CORRection:OFFSet:STATe[?] OFF | ON

SENSe:CORRection:OFFSet:STATe ON aktiviert die Offsetkorrektur, *SENSe:CORRection:OFFSet:STATe OFF* deaktiviert sie.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *OFF*

SENSe:CORRection:SPDev:STATe[?] OFF | ON

SENSe:CORRection:SPDevice:STATe ON aktiviert den S-Parameter-Datensatz für eine dem Messkopf vorgeschaltete Komponente (Dämpfungsglied, Richtkoppler). Mit dem Parameter *OFF* wird er deaktiviert.

Die Verwendung von S-Parametern anstelle eines festen Offsets (siehe Befehlsgruppe *SENSe:CORRection:OFFSet*) ermöglicht genauere Messungen, da die Wechselwirkungen zwischen Messkopf, Quelle und dazwischengeschalteter Komponente berücksichtigt werden können. (Näheres zum Laden von S-Parameter-Datensätzen siehe Abschnitt 3). Bei Erstauslieferung des Messkopfes ist noch kein S-Parameter-Datensatz geladen. In diesem Zustand führt der Befehl *SENSe:CORRection:SPDevice:STATe ON* zu einer Fehlermeldung.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung:

Bei Auslieferung des Messkopfes ist die Voreinstellung *OFF*. Beim Laden einer S-Parameter-Tabelle kann die Voreinstellung neu festgelegt werden (siehe Abschnitt 3).

SENSe:FREQuency[?] 0.0 ... 100.0e9

SENSe:FREQuency übergibt die Trägerfrequenz des zu messenden HF-Signals, welche zur Frequenzgangkorrektur des Messergebnisses verwendet wird. Bei breitbandigen Signalen (*Spread-spectrum*-Signale, Mehrträgersignale) stellt man die Mittenfrequenz ein.

Der Abfragebefehl liefert die eingestellte Trägerfrequenz in Hz.

Voreinstellung: 50.0e6 [Hz]

SENSe:FUNCTION[?] <sensor_function>

SENSe:FUNCTION <sensor_function> versetzt den Messkopf in einen der folgenden Messmodi:

Tabelle 6-3 Messmodi

<sensor_function>	Bezeichnung des Messmodus
"POWer:AVG"	<p>Continuous Average Nach dem einmaligen Auftreten des Triggerereignisses wird die mittlere Leistung in einem Zeitintervall (Sampling Window) gemessen, dessen Breite mit SENSe:POWer:AVG:APERture festgelegt wird. Die Einzelmessungen erfolgen grundsätzlich paarweise, um durch Differenzbildung eine höhere Genauigkeit des Messergebnisses zu erreichen. Bei aktivierter Mittelungsfunktion (Averaging) wird dieser Vorgang so oft, wie durch den Averaging-Faktor spezifiziert wurde, wiederholt. Die effektive Messzeit beträgt also bei aktivierter Mittelungsfunktion $2 \times \langle \text{SENSe:AVERage:COUNT} \rangle \times \langle \text{SENSe:POWer:AVG:APERture} \rangle$ und bei deaktivierter Mittelungsfunktion $2 \times \langle \text{SENSe:POWer:AVG:APERture} \rangle$. Triggerereignisse starten im Modus Continuous Average eine oder mehrere Messungen (in Abhängigkeit vom Parameter TRIGger:COUNT).</p>
"POWer:TSLot:AVG"	<p>Timeslot Die Leistung wird sequentiell in einer definierten Anzahl aufeinanderfolgender Zeitfenster (Timeslots) gemessen. Die bis zu 128 Timeslots haben alle die gleiche Breite, welche durch SENSe:POWer:TSLot:WIDTh festgelegt wird. Ihre Anzahl wird durch SENSe:POWer:TSLot:COUNT definiert. Das Messergebnis ist ein Array, das so viele Elemente enthält, wie es Timeslots gibt. Jedes Element repräsentiert den Leistungsmittelwert in einem der Timeslots. Nach jedem Auftreten des Triggerereignisses wird in allen Zeitfenstern eine Messung durchgeführt. Die Einzelmessungen erfolgen auch in diesem Modus grundsätzlich paarweise, um durch Differenzbildung eine höhere Genauigkeit des Messergebnisses zu erreichen. Bei aktivierter Mittelungsfunktion (Averaging) wird dieser Vorgang so oft, wie durch den Averaging-Faktor spezifiziert wurde, wiederholt. Die effektive Messzeit pro Timeslot beträgt also bei aktivierter Mittelungsfunktion $2 \times \langle \text{SENSe:AVERage:COUNT} \rangle \times \langle \text{SENSe:POWer:TSLot:AVG:WIDTh} \rangle$ und bei deaktivierter Mittelungsfunktion $2 \times \langle \text{SENSe:POWer:TSLot:AVG:WIDTh} \rangle$. Zu beachten ist, dass auch entsprechend viele Triggerereignisse vorliegen müssen.</p>
"POWer:BURSt:AVG"	<p>Burst Average Dieser Messmodus dient z. B. zur Messung des Leistungsmittelwerts repetierender Einfach-Bursts. Das Zeitintervall (Integrationszeit), in welchem die mittlere Leistung gemessen wird, wird nicht vorgegeben. Stattdessen erkennt ein spezieller Algorithmus Anfang und Ende des Bursts und sorgt für eine Synchronisation des Sampling Window. Mit SENSe:POWer:BURSt:DTOLerance muss der Anwender allerdings die Dropout-Toleranz, ein zeitliches Kriterium für das Erkennen des Puls- oder Burstendes, festlegen. Im Modus Burst Average werden externe Triggerereignisse (unabhängig von der Einstellung des Parameters TRIGger:SOURce) sowie der Parameter TRIGger:DELay ignoriert.</p>
"XTIME:POWer"	<p>Scope Der Scope-Modus ähnelt dem Timeslot-Modus. Da jedoch bis zu 1024 aufeinanderfolgende Zeitfenster (hier „Points“ genannt) möglich sind, ist der maximale Averaging-Faktor aus Gründen des Speicherplatzes auf 8192 begrenzt. Im Scope-Modus gelten außerdem andere Grenzen für die Breite der Zeitfenster als im Timeslot-Modus. Mit Hilfe von Mit SENSe:SWEep:REALtime ON lässt sich die Zeitauflösung maximieren. Dazu werden die Differenzbildung und das Mittelungsfilter umgangen.</p>

In den Messmodi *Burst Average* und *Timeslot* können am Anfang und Ende des Sampling Window bzw. Timeslots Zeitintervalle eingestellt werden, die von der Messung ausgeschlossen bleiben sollen. (*SENSe:TIMing:EXCLude:STARt* und *-:STOP*).

Der Abfragebefehl liefert

- 1 für "*POWer:AVG*",
- 2 für "*POWer:TSLot:AVG*",
- 4 für "*POWer:BURSt:AVG*",
- 8 für "*XTIME:POWer*".

Voreinstellung: "*POWer:AVG*"

SENSe:POWer:AVG:APERture[?] 0.0001 ... 0.1

SENSe:POWer:AVG:APERture legt für den *Continuous Average*-Modus das Zeitintervall (Sampling Window) fest, in welchem kontinuierlich Messwerte aufgenommen werden. Im manuellen Betrieb ist die Default-Einstellung von 20 ms in Kombination mit aktiviertem Smoothing (siehe *SENSe:POWer:AVG:SMOothing:STATE*) meist ausreichend. Ein anderer, i. a. größerer Wert ist dann erforderlich, wenn das Messergebnis modulationsbedingt Schwankungen aufweist. Speziell bei sehr niederfrequenter Modulation ist es sinnvoll, die Größe des Sampling Window genau an die Modulationsperiode anzupassen, was zu einer optimal beruhigten Anzeige führt:

Tabelle 6-4 Optimale Wahl der Größe des Sampling Window (N = 1, 2, 3, ...)

Smoothing	Optimale Größe des Sampling Window
OFF	$N \times \text{Modulationsperiode} / 2$
ON	$N \times \text{Modulationsperiode} \times 2$

Die theoretisch kürzeste Messzeit kann danach nur bei abgeschaltetem Smoothing erreicht werden. Je mehr Modulationsperioden in ein Sampling Window passen, umso unkritischer ist es, ob N ganzzahlig ist oder nicht. Bei eingeschaltetem Smoothing reichen ca. 5 Perioden, um modulationsbedingte Schwankungen auf ein akzeptables Maß zu drücken, bei mehr als 9 Perioden sind sie nicht mehr wahrnehmbar. Bei ausgeschaltetem Smoothing sind die Verhältnisse deutlich ungünstiger: Hier werden statt 5 bereits 300 Perioden benötigt, und erst ab 3000 Perioden sind die Schwankungen völlig verschwunden.

Der Abfragebefehl liefert die aktuell eingestellte Breite des Sampling Window in Sekunden.

Voreinstellung: 0.02 [s]

SENSe:POWer:AVG:BUFFer:SIZE[?] 1 ... 1024

SENSe:POWer:AVG:BUFFer:SIZE stellt die Puffergröße für den gepufferten *Continuous Average*-Modus ein.

Der Abfragebefehl liefert die aktuelle Puffergröße für den gepufferten *Continuous Average*-Modus.

Voreinstellung: 1

SENSe:POWer:AVG:BUFFer:STATe[?] OFF | ON

Mit *ON* wird der gepufferte *Continuous Average*-Modus aktiviert, mit *OFF* deaktiviert. In diesem Modus werden die durch die Triggerereignisse erzeugten Messergebnisse so lange im Messkopf gesammelt, bis der Puffer gefüllt ist. Anschließend erfolgt die Übertragung aller Ergebnisse als Blockdaten. Dadurch

wird eine höhere Messrate erzielt als im ungepufferten *Continuous Average*-Modus. Die höchste Messrate wird durch Kombination des gepufferten Modus mit Mehrfachtriggerung (siehe Parameter *TRIGger:COUNT*) Die Größe des Messwertpuffers wird mit dem Befehl *SENSe:POWer:AVG:BUFFer:SIZE* eingestellt.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *OFF*

SENSe:POWer:AVG:SMOothing:STATe[?] OFF | ON

Der Parameter *ON* aktiviert im *Continuous Average*-Modus ein Glättungsfilter für modulierte Signale, *OFF* deaktiviert es. Das Glättungsfilter ist ein steiflankiges digitales Tiefpassfilter zur Unterdrückung von Messwertschwankungen durch niederfrequente Modulation. Dieser Parameter sollte zur Reduzierung modulationsbedingter Messwertschwankungen immer dann aktiviert sein, wenn die Größe des Sampling Window nicht genau an die Modulationsperiode angepasst werden kann oder soll. Wenn das Sampling Window 5 ... 9 mal so groß wie eine Modulationsperiode gewählt wird, sind die Anzeigeschwankungen i. a. ausreichend reduziert. Bei ausgeschaltetem Smoothing werden 300 bis 3000 Perioden für denselben Effekt benötigt.

Bei ausgeschaltetem Smoothing werden die Abtastwerte innerhalb eines Sampling Window als gleichwertig betrachtet und gemittelt, was zu einem integrierenden Verhalten des Messgeräts führt. Wie oben beschrieben, kann damit eine optimale Unterdrückung modulationsbedingter Schwankungen im Messergebnis erreicht werden, wenn die Größe des Sampling Window genau an die Modulationsperiode angepasst ist. Wenn dies nicht der Fall ist, kann die Modulation erheblich durchschlagen, selbst wenn das Sampling Window um ein Vielfaches größer als die Modulationsperiode ist. Dieses Verhalten lässt sich erheblich verbessern, wenn die Abtastwerte vor der Mittelung einer Wichtung (raised-von-Hann-Fenster) unterworfen werden, was einer Video-Filterung entspricht. Genau dies passiert bei aktiviertem Smoothing.

Da das Glättungsfilter das Eigenrauschen des Messkopfes um etwa 20 % erhöht, sollte es ausgeschaltet bleiben, wenn keine Notwendigkeit für seinen Einsatz besteht.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *ON*

SENSe:POWer:BURSt:DTOLerance[?] 0.0 ... 0.003

SENSe:POWer:BURSt:DTOLerance definiert die Dropout-Toleranz, einen Parameter zum sicheren Erkennen des Burst-Endes im Modus *Burst Average* bei modulierten Signalen (z. B. bei den digitalen Standards NADC, PDC, PHS etc.) Die Dropout-Toleranz ist größer als der längste Amplitudeneinbruch und kleiner als die Lücke zwischen zwei aufeinanderfolgenden Bursts zu wählen. Für alle gängigen digitalen Kommunikationsstandards genügt der Default-Wert.

Der Abfragebefehl liefert die Dropout-Toleranz für den *Burst Average*-Modus.

Voreinstellung: *0.0001 [s]*

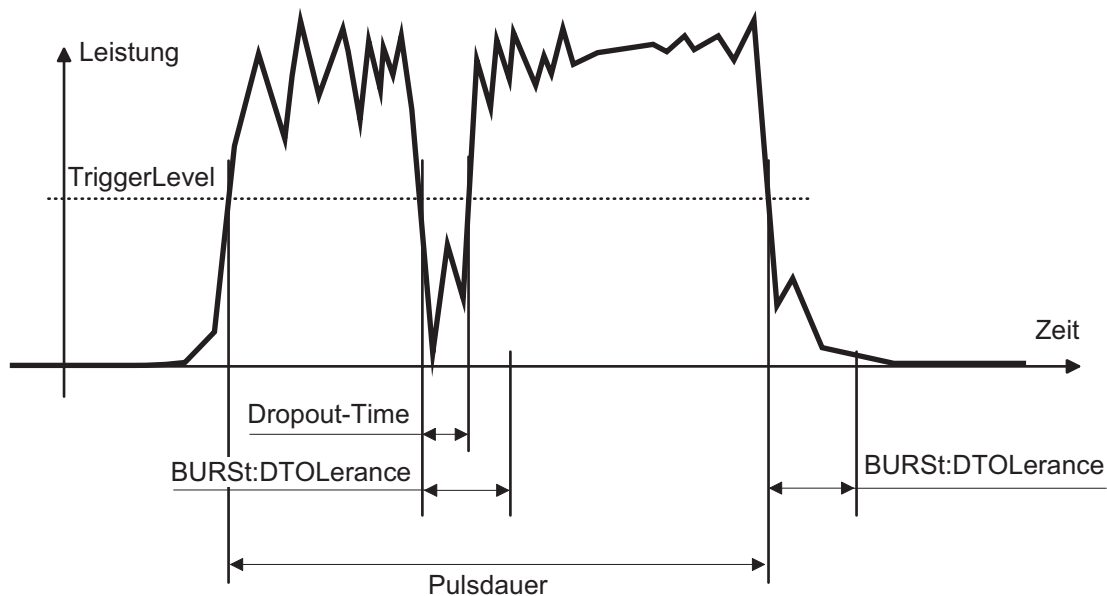


Bild 6-1 Wirkung von SENSE:POWER:BURSt:DTOLerance

SENSE:POWER:TSLot:AVG:COUNT[?] 1 ... 128

SENSE:POWER:TSLot:AVG:COUNT stellt die Anzahl der aufeinanderfolgenden Zeitfenster (Timeslots) für den Timeslot-Modus ein, die nach jedem Triggerereignis abgearbeitet werden sollen.

Der Abfragebefehl liefert die Anzahl der aufeinanderfolgenden Zeitfenster (Timeslots).

Voreinstellung: 1

SENSE:POWER:TSLot:AVG:WIDTH[?] 0.0001 ... 0.1

SENSE:POWER:TSLot:AVG:WIDTH stellt die Länge eines Zeitfensters (Timeslots) in Sekunden für den Timeslot-Modus ein.

Der Abfragebefehl liefert die Länge eines Zeitfensters (Timeslots) in Sekunden für den Timeslot-Modus.

Voreinstellung: 0.001 [s]

SENSE:RANGe[?] 0 ... 2

SENSE:RANGe wählt den Messpfad des Messkopfes. Der Messkopf hat drei separate Messpfade. Pfad 1 ist der empfindlichste, Pfad 2 der mittlere und Pfad 3 der unempfindlichste Messpfad. SENSE:RANGe 0 wählt Pfad 1, SENSE:RANGe 1 wählt Pfad 2 und SENSE:RANGe 2 wählt Pfad 3.

Die Aussteuergrenzen dieser Messpfade sind temperatur- und exemplarabhängig. Als Richtwerte können 40 μ W (-14 dBm) für den empfindlichsten, 4 mW (6 dBm) für den mittleren und 400 mW (26 dBm) für den unempfindlichsten Messpfad gelten.

Der Abfragebefehl liefert

- 0 für Pfad 1,
- 1 für Pfad 2,
- 2 für Pfad 3.

Bei manueller Wahl des Messpfades (SENSE:RANGe:AUTO OFF) wird der aktuell gewählte Messpfad ausgegeben. Bei automatischer Wahl wird der zuletzt manuell eingestellte Pfad ausgegeben. Er wird deshalb nach Ausschalten der Automatik sofort wieder eingestellt.

Voreinstellung: 2 (unempfindlichster Pfad)

SENSe:RANGe:AUTO[?] OFF | ON

SENSe:RANGe:AUTO ON aktiviert die automatische Wahl des Messpfads, *SENSe:RANGe:AUTO OFF* deaktiviert sie.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *ON*

SENSe:RANGe:AUTO:CLeVel[?] -20.0 ... 0.0

Mit *SENSe:RANGe:AUTO:CLeVel* hat man die Möglichkeit, den Übergangsbereich zwischen den Messpfaden 1 und 2 bzw. 2 und 3 um den angegebenen Wert (in dB) abzusenken. Diese Maßnahme kann bei Signalen mit einem hohen Verhältnis *peak-to-average* zu einer Verbesserung der Messgenauigkeit führen, da die Aussteuerungsreserve für Modulationsspitzen größer wird. Der Nachteil besteht in einer Verringerung des Signal-Rauschabstandes an den Untergrenzen der Übergangsbereiche.

Der Abfragebefehl liefert die Verschiebung der Übergangsbereiche zwischen den Messkanälen 1 und 2 bzw. 2 und 3.

Voreinstellung: *0.0 [dB]*

SENSe:SAMPling[?] FREQ1 | FREQ2

Mit *SENSe:SAMPling* ist es möglich, Einfluss auf die Abtastfrequenz der Analog-Digital-Wandler im Messkopf zu nehmen. Mit dem Parameter *FREQ1* beträgt die Abtastfrequenz 133,358 kHz, mit dem Parameter *FREQ2* beträgt sie 119,467 kHz. Zweck dieser Maßnahme ist die Unterdrückung störender niederfrequenter Mischprodukte aus Signalanteilen und der Abtastfrequenz.

Der Abfragebefehl liefert

- 1 für *FREQ1*,
- 2 für *FREQ2*.

Voreinstellung: *FREQ1*

SENSe:SGAMma:CORRection:STATe[?] OFF | ON

SENSe:SGAMma:CORRection:STATe ON veranlasst, dass der mit *SENSe:SGAMma:MAGNitude* und *SENSe:SGAMma:PHASe* definierte komplexe Reflexionsfaktor der Quelle zur Korrektur der Wechselwirkungen zwischen Messkopf, Quelle und dazwischengeschalteter Komponente (*siehe SENSe:CORRection:SPDevice:STATe*) benutzt wird. Auf diese Weise lässt sich die Quelltor-Fehlanpassung (source mismatch), die oft einen beträchtlichen Anteil an der Messunsicherheit hat, kompensieren.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *OFF*

SENSe:SGAMma:MAGNitude[?] 0.0 ... 1.0

SENSe:SGAMma:MAGNitude definiert den Betrag des komplexen Reflexionsfaktors der Quelle. Ein Wert von *0.0* entspricht einer ideal angepassten Quelle, ein Wert von *1.0* entspricht Totalreflexion.

Der Abfragebefehl liefert den eingestellten Betrag.

Voreinstellung: 0.0

SENSe:SGAMma:PHASe[?] -360.0 ... 360.0

SENSe:SGAMma:PHASe definiert den Phasenwinkel (in Grad) des komplexen Reflexionsfaktors der Quelle.

Der Abfragebefehl liefert den eingestellten Phasenwinkel.

Voreinstellung: 0.0 [°]

SENSe:SWEep:AVERage:COUNT[?] 1 ... 65536

Mit *SENSe:SWEep:AVERage:COUNT* wird eingestellt, wie viele Messwerte zur Bildung des Messergebnisses im *Scope*-Modus gemittelt werden sollen. Je höher dieser Mittelungsfaktor gewählt wird, desto weniger schwanken die Messwerte, und desto länger ist die Messzeit. Der Parameter wird auf die nächste Zweierpotenz auf- oder abgerundet.

Der Abfragebefehl liefert den im *Scope*-Modus verwendeten Averaging-Faktor.



*Damit der eingestellte Mittelungsfaktor wirksam wird, muss die Mittelungsfunktion mit *SENSe:SWEep:AVERage:STATe ON* eingeschaltet werden.*

Voreinstellung: 4

SENSe:SWEep:AVERage:COUNT:AUTO[?] OFF | ON | ONCE

Mit *SENSe:SWEep:AVERage:COUNT:AUTO* wird die automatische Bestimmung des Mittelungsfaktors im *Scope*-Modus aktiviert (Auto-Averaging) oder deaktiviert (Festfilter-Modus). Wird das Auto-Averaging aktiviert, dann wird der Mittelungsfaktor fortlaufend in Abhängigkeit von der Höhe der Leistung und anderer Parameter neu bestimmt und eingestellt.

Mit *SENSe:SWEep:AVERage:COUNT:AUTO ON* wird das Auto-Averaging ein-, mit *SENSe:SWEep:AVERage:COUNT:AUTO OFF* ausgeschaltet. Beim Ausschalten wird der zuletzt automatisch ermittelte Mittelungsfaktor in den Festfiltermodus übernommen. Das Kommando *SENSe:SWEep:AVERage:COUNT:AUTO ONCE* bewirkt, dass von der Filterautomatik unter den momentanen Messbedingungen einmalig ein neuer Mittelungsfaktor ermittelt und in den Festfiltermodus übernommen wird.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *OFF*

SENSe:SWEep:AVERage:COUNT:AUTO:MTIME[?] 1.0 ... 999.99

SENSe:SWEep:AVERage:COUNT:AUTO:MTIME setzt die Obergrenze für die Einschwingzeit des Mittelungsfilters im Auto-Averaging-Modus für den *Scope*-Modus und begrenzt damit auch dessen Länge.

Der Abfragebefehl liefert die aktuelle Obergrenze für die Einschwingzeit des Mittelungsfilters im Auto-Averaging-Modus für den *Scope*-Modus.

Voreinstellung: 30.0 [s]

SENSe:SWEep:AVERAge:COUNT:AUTO:NSRatio[?] 0.0001 ... 1.0

SENSe:SWEep:AVERAge:COUNT:AUTO:NSRatio legt den relativen Rauschanteil im Messergebnis für den *Scope*-Modus fest, wenn das Auto-Averaging im entsprechenden Modus (*SENSe:SWEep:AVERAge:COUNT:AUTO:TYPE NSRatio*) betrieben wird. Der Rauschanteil ist definiert als der Betrag der durch das Eigenrauschen des Messkopfes bewirkten Pegelschwankung in dB (zwei Standardabweichungen).

Der Abfragebefehl liefert den relativen Rauschanteil im Messergebnis für den *Scope*-Modus.

Voreinstellung: 0.01 [dB]

SENSe:SWEep:AVERAge:COUNT:AUTO:RESolution[?] 1 ... 4

SENSe:SWEep:AVERAge:COUNT:AUTO:RESolution stellt den Auflösungsindex für das automatische Mittelungsfiler im *Scope*-Modus ein, wenn es im Modus *RESolution* betrieben wird. Der Auflösungsindex ist gleich der Anzahl der Nachkommastellen, die bei einer Weiterverarbeitung des Messergebnisses in dBm, dBµV oder dB berücksichtigt werden sollen. Der Normal-Modus ist so ähnlich wie bei den Vorgängergeräten R&S NRVS und R&S NRVD bzw. anderen handelsüblichen Leistungsmessern ausgelegt. Je höher der Index gewählt wird, umso besser ist das Messergebnis gefiltert, ohne dass damit erreicht würde, dass die letzte signifikante Stelle (0,01 dB bei einem Index von 3) auch wirklich steht. Zu empfehlen ist stattdessen die Einstellung *NSRatio*.

Der Abfragebefehl liefert den Auflösungsindex für den *Scope*-Modus.

Voreinstellung: 3

SENSe:SWEep:AVERAge:COUNT:AUTO:POINT[?] 1 ... <SENSe:SWEep:POINTS>

SENSe:SWEep:AVERAge:COUNT:AUTO:SLOT definiert den „Punkt“, auf dessen Leistung sich das Auto-Averaging im *Scope*-Modus bezieht. Der „Punkt“ wird über seine Nummer adressiert, wobei die Zählung mit 1 beginnt. Die Nummer des „Punktes“ darf die Anzahl der aktuell eingestellten „Punkte“ nicht übersteigen. Wird zuerst eine gültige „Punkt“-Nummer eingestellt und danach die Anzahl der „Punkte“ auf einen Wert reduziert, der kleiner ist als die „Punkt“-Nummer, dann wird diese automatisch gleich der neuen „Punkt“-Anzahl gesetzt, d. h. das Auto-Averaging bezieht sich auf den letzten „Punkt“.

Der Abfragebefehl liefert die Nummer des aktuellen „Punktes“, auf dessen Leistung sich das Auto-Averaging im *Scope*-Modus bezieht.

Voreinstellung: 1

SENSe:SWEep:AVERAge:COUNT:AUTO:TYPE[?] RESolution | NSRatio

SENSe:SWEep:AVERAge:COUNT:AUTO:TYPE definiert den Modus des automatischen Mittelungsfilters im *Scope*-Modus. Mit dem Parameter *RESolution* wird der bei Leistungsmessern übliche Modus eingestellt; mit *NSRatio* kann die Einhaltung eines genau definierten Rauschanteils vorgegeben werden.

Der Abfragebefehl liefert

- 1 für *RESolution*,
- 2 für *NSRatio*.

Voreinstellung: *RESolution*

SENSe:SWEEp:AVERAge:STATe[?] OFF | ON

SENSe:SWEEp:AVERAge:STATe schaltet das Mittelungsfiler für den Scope-Modus aus oder ein.

Der Abfragebefehl liefert

- 1 für OFF,
- 2 für ON.

Voreinstellung: ON

SENSe:SWEEp:AVERAge:TCONtrol[?] MOVing | REPeat

SENSe:SWEEp:AVERAge:TCONtrol (*terminal control*) legt das Verhalten des Mittelungsfilters im Scope-Modus fest. Sobald ein neuer Messwert in das FIR-Filter geschoben wird, steht am Filterausgang auch ein neuer Mittelwert zur Verfügung, der aus dem neu hinzugekommenen Messwert und den restlichen im Filter befindlichen Werten gewonnen wird.

Der Parameter *MOVing* legt fest, dass jeder neue Mittelwert als Messergebnis ausgegeben wird. Aus diese Weise können Tendenzen im Messergebnis schon während des Messvorgangs erkannt werden.

Der Parameter *REPeat* legt fest, dass erst dann, wenn das FIR-Filter komplett mit neuen Messwerten aufgefüllt wurde, ein neues Messergebnis ausgegeben wird. Dadurch wird sichergestellt, dass redundante Informationen nicht ausgegeben werden.

Der Abfragebefehl liefert

- 1 für *MOVing*,
- 2 für *REPeat*.

Voreinstellung: *MOVing*

SENSe:SWEEp:OFFSet:TIME[?] – (<TRIGger:DELay> + 0.005) ... 100.0

Mit diesem Befehl ist es möglich, das Zeitfenster für die Messwernerfassung im Modus *Scope* entlang der Zeitachse zu verschieben, ohne den Wert von *TRIGger:DELay* zu verändern. Positive Werte bewirken eine zusätzliche Verzögerung, negative Werte eine entsprechend frühere Messwertaufnahme.

Der Abfragebefehl liefert die eingestellte Zeit in Sekunden.

Voreinstellung: 0.0 [s]

SENSe:SWEEp:POINts[?] 1 ... 1024

Mit diesem Befehl wird die zeitliche Auflösung des Messergebnisses definiert. Jeder „Punkt“ repräsentiert ein Zeitintervall, dessen Dauer sich aus der Länge des Zeitfensters (Befehl *SENSe:SWEEp:TIME*) geteilt durch die Anzahl der „Punkte“ ergibt. Das Messergebnis für einen „Punkt“ ist gleich dem Leistungsmittelwert über das zugehörige Zeitintervall.

Der Abfragebefehl liefert die Zahl der eingestellten „Punkte“.

Voreinstellung: 100

SENSe:SWEEp:REALtime[?] OFF | ON

SENSe:SWEEp:REALtime ON unterdrückt im Scope-Modus die paarweise Aufnahme von Messwerten, so dass in dieser Einstellung einmalige Vorgänge aufgezeichnet werden können. Weil das Mittelungsfiler des Messkopfes nicht benutzt wird, wird *SENSe:AVERAge:STATe* ignoriert, bleibt aber unbeeinflusst.

Der Abfragebefehl liefert

- 1 für OFF,
- 2 für ON.

Voreinstellung: OFF

SENSE:SWEep:TIME[?] 0.0001 ... 0.3

SENSE:SWEep:TIME stellt die Dauer des Zeitfensters im Modus *Scope* ein. Dieses Zeitfenster wird in eine Anzahl gleich großer Intervalle unterteilt, in denen jeweils der Leistungsmittelwert bestimmt wird. Die Zahl der Intervalle ist gleich der Zahl der Messpunkte, die mit dem Befehl *SENSE:SWEep:POINTS* eingestellt wird.

Der Abfragebefehl liefert die Dauer des Zeitfensters im Modus *Scope* (in s).

Voreinstellung: 0.01 [s]

SENSE:TIMing:EXCLude:START[?] 0.0 ... 0.1

SENSE:TIMing:EXCLude:START definiert die Ausschlusszeit am Anfang des Messfensters in den Modi *Burst Average* (Bild 6-2) und *Timeslot* (Bild 6-3).

Der Abfragebefehl liefert die Ausschlusszeit am Anfang des Messfensters.

Voreinstellung: 0.0 [s]

SENSE:TIMing:EXCLude:STOP[?] 0.0 ... 0.003

SENSE:TIMing:EXCLude:STOP definiert die Ausschlusszeit am Ende des Messfensters in den Modi *Burst Average* (Bild 6-2) und *Timeslot* (Bild 6-3).

Der Abfragebefehl liefert die Ausschlusszeit am Ende des Messfensters.

Voreinstellung: 0.0 [s]

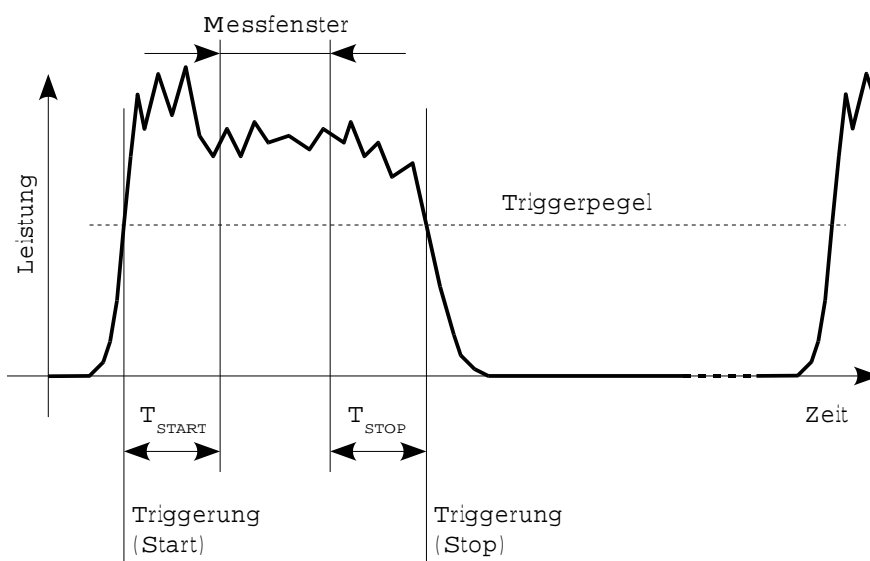


Bild 6-2 Wirkung von *SENSE:TIMing:EXCLude:START* und *-:END* im *Burst Average*-Modus

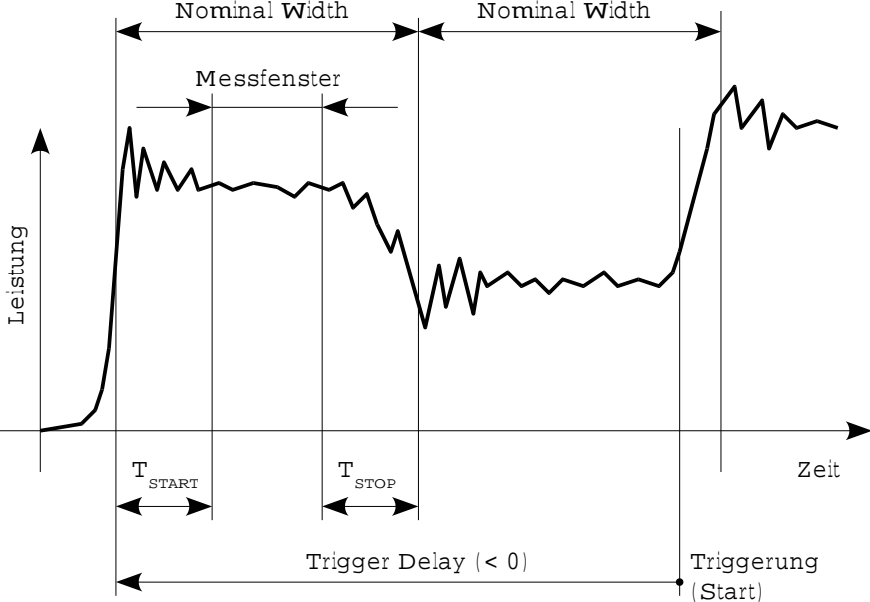


Bild 6-3 Wirkung von *SENSe:TIMing:EXCLude:STARt* und *-:END* im *Timeslot*-Modus

SYSTEM

Über das Befehlssystem *SYSTEM* können administrative Geräteeinstellungen vorgenommen bzw. abgefragt werden. Hierzu gehören detaillierte Informationen über den Messkopf und dessen Initialisierung einschließlich Übertragung der verfügbaren Befehle und ihrer Parametergrenzen.

Tabelle 6-5 Befehle des Befehlssystems *SYSTEM*

Befehl	Parameter	Einheit	Bemerkung
SYSTEM			
:INFO? [Item]			nur Abfrage
:INITialize			keine Abfrage möglich
MINPower?		W	nur Abfrage
:TRANsaction			
:BEGIN:			keine Abfrage möglich
:END			keine Abfrage möglich

SYSTEM:INFO? [Item]

SYSTEM:INFO? liefert einen String, der detailliertere Informationen als der Identifikations-String, den der Messkopf als Antwort auf **IDN?* liefert, enthält. Ist kein *Item* angegeben, dann ist der Antwort-String eine Folge von durch *CR* und *LF* (in C-Notation: $\backslashr\backslashn$) getrennten Einträgen der Form *Item:Informations-String*. Mit dem optional an den Befehl angehängten *Item* lässt sich gezielt der Eintrag zum gewünschten *Item* abfragen. Der Antwort-String ist nullterminiert, d. h. seine Endekennung ist ein Nullbyte (in C-Notation: $\backslash0$).

Tabelle 6-6 Bedeutung des *Item* beim Befehl *SYSTEM:INFO?*

Item	Informations-String	Bemerkung
„MANUFACTURER“	„Rohde & Schwarz GmbH & Co. KG“	Hersteller
„TYPE“	„NRP-Z11“ oder „NRP-Z21“	Typenbezeichnung
„STOCK NUMBER“	„1138.3004.02“ oder „1137.6000.02“	Materialnummer
„SERIAL“	„<Seriennummer>“	6-stellige Seriennummer
„HWVERSION“	„000000000“	Hardware-Version (Standard)
„HWVARIANT“	„000000000“	Hardware-Variante (Standard)
„SW BUILD“	„<Build-Nummer>“	Versionsnummer der Messkopf-Firmware
„TECHNOLOGY“	„3 Path Diode“	Verwendete Detektor-Technologie

Item	Informations-String	Bemerkung
"FUNCTION"	"Power Terminating"	Beim R&S NRP-Z11/-Z21 handelt es sich um einen Abschluss-Leistungsmesskopf.
"MINPOWER"	"<nominale untere Messgrenze in W>"	Beim R&S NRP-Z11/-Z21 liegt die nominale untere Messgrenze bei 200 pW, d. h. der Messkopf liefert bei deaktivierter S-Parameter-Korrektur als Antwort auf "SYSTem:INFo? "MINPOWER" den Informations-String "2e-10". Bei aktivierter S-Parameter-Korrektur richtet sich der Informationsstring nach der nominalen unteren Messgrenze der Messkopf-Zweitor-Kombination.
"MAXPOWER"	"<nominale obere Messgrenze in W>"	Beim R&S NRP-Z11/-Z21 liegt die nominale obere Messgrenze bei 200 mW, d. h. der Messkopf liefert bei deaktivierter S-Parameter-Korrektur als Antwort auf "SYSTem:INFo? "MAXPOWER" den Informations-String "0.2". Bei aktivierter S-Parameter-Korrektur richtet sich der Informationsstring nach der nominalen oberen Messgrenze der Messkopf-Zweitor-Kombination.
"IMPEDANCE"	"50"	Die nominale Eingangsimpedanz des HF-Eingangs beträgt beim R&S NRP-Z11/-Z21 50 Ω .
"COUPLING"	"AC/DC"	Der HF-Eingang des R&S NRP-Z11/-Z21 ist zwar gleichspannungsgekoppelt, jedoch werden dem HF-Signal überlagerte Gleichspannungen durch den Messverstärker unterdrückt.
"CAL. ABS."	"<Datum>"	Datum der Absolutkalibrierung im Format JJJJ-MM-TT. Bei ungültigem Datumseintrag wird "Invalid Calibration Date" zurück gegeben.
"CAL. REFL."	"<Datum>"	Datum der Reflexionsfaktor-Kalibrierung im Format JJJJ-MM-TT. Bei ungültigem Datumseintrag wird "Invalid Calibration Date" zurück gegeben.
"CAL. S PARA."	"<Datum>"	Datum der S-Parameter-Kalibrierung im Format JJJJ-MM-TT. Ist kein S-Parameter-Satz geladen, liefert der Messkopf den String "not applicable". Bei ungültigem Datumseintrag wird "Invalid Calibration Date" zurück gegeben.
"CAL. MISC."	"<Datum>"	Datum der Kalibrierung sonstiger Parameter im Format JJJJ-MM-TT. Bei ungültigem Datumseintrag wird "Invalid Calibration Date" zurück gegeben.
"SPD MNEMONIC"	"<Mnemonic-String>"	Klartextbezeichnung der dem Sensor vorgeschalteten Komponente

SYSTem:INITialize

SYSTem:INITialize versetzt den Messkopf in den Standardzustand, d. h. die Voreinstellungen für alle Messparameter werden genau wie bei *RST geladen. Danach gibt der Messkopf eine komplette Liste aller unterstützten Befehle und Parameter aus. Der Befehl ermöglicht es, dass die Fernsteuersoftware sich automatisch an die Möglichkeiten verschiedener Messkopftypen mit unterschiedlichem Funktionsumfang anpassen kann.

SYSTem:MINPower?

SYSTem:MINPower? liefert die untere Messgrenze des Messkopfes oder der Kombination aus Messkopf und vorgeschalteter Komponente, wenn der Parameter *SENSe:CORRection:SPDevice* den

Wert *ON* hat. Dieser Abfragebefehl kann z. B. verwendet werden, um eine sinnvolle Auflösung für die Anzeige des Messwertes in der Nähe der unteren Messgrenze zu ermitteln.

SYSTEM:TRANSACTION:BEGIN

SYSTEM:TRANSACTION:BEGIN markiert den Anfang einer Folge von Einstellbefehlen, zwischen denen keine Überprüfung der Parametergrenzen erfolgen soll. Auf diese Weise werden Fehlermeldungen verhindert, wenn ein Einstellbefehl einen Konflikt verursacht, welcher durch einen folgenden Einstellbefehl aufgelöst wird. Siehe *SYSTEM:TRANSACTION:END*.

SYSTEM:TRANSACTION:END

SYSTEM:TRANSACTION:END markiert das Ende einer Folge von Einstellbefehlen, zwischen denen keine Überprüfung der Parametergrenzen erfolgen soll. Im Anschluss an diesen Befehl wird eine Überprüfung der Parametergrenzen durchgeführt.

TESTTabelle 6-7 Befehle des Befehlssystems *TEST*

Befehl	Parameter	Einheit	Bemerkung
TEST:SENSor?			nur Abfrage

TEST:SENSor?

TEST:SENSor? löst einen Selbsttest des Messkopfes aus. Im Unterschied zu **TST* liefert dieser Befehl detaillierte Ausgaben, die z. B. für die Fehlersuche nützlich sein können.

TRIGger

Tabelle 6-8 Befehle des Befehlssystems TRIGger

Befehl	Parameter	Einheit	Bemerkung
ABORt			keine Abfrage möglich
INITiate			
:CONTInuous[?]	OFF ON		
:IMMediate			keine Abfrage möglich
TRIGger			
:ATRigger:STATe[?]	OFF ON		
:COUNt[?]	1 ... 2×10^9		
:DELay[?]	x ... 100.0	s	
:AUTO[?]	OFF ON		
:HOLDoff[?]	1.0e-9 ... 10.0	s	
:HYSTeresis[?]	0.0 ... 3.0	dB	
:IMMediate			keine Abfrage
:LEVel[?]	x ... y	W	
:SLOPe[?]	POSitive NEGative		
:SOURce[?]	BUS EXTernal HOLD IMMediate INTernal		

ABORt

ABORt bricht die gerade laufende Messung ab und bringt den Messkopf in den *IDLE*-Zustand (Normalfall). Wenn sich der Messkopf allerdings im freilaufenden Messmodus befindet (Einstellung *INITiate:CONTInuous ON*), wird der Zustand *IDLE* sofort wieder verlassen, und der Messkopf geht in den Zustand *WAIT_FOR_TRIGGER*.

INITiate:CONTInuous[?] OFF | ON

INITiate:CONTInuous ON aktiviert den freilaufenden Messmodus. In dieser Betriebsart wird nach Beendigung einer Messung automatisch eine neue gestartet. Dabei geht der Messkopf zunächst in den Zustand *WAIT_FOR_TRIGGER* und beginnt mit dem Messvorgang, sobald die Triggerbedingung erfüllt ist. Nach Abschluss der Messung wird wieder der Zustand *WAIT_FOR_TRIGGER* eingenommen. Fortlaufende Triggerereignisse vorausgesetzt, wird der Messkopf also kontinuierlich messen.

Im Gegensatz dazu muss nach Senden des Befehls *INITiate:CONTInuous OFF* jeder Messzyklus explizit mit dem Befehl *INITiate:IMMediate* gestartet werden. Nach Triggerung und Durchführung des Messvorgangs geht der Messkopf in den Zustand *IDLE* und verharrt dort bis zu einem neuen Messstart mit dem Befehl *INITiate:IMMediate*.

Der Abfragebefehl liefert

- 1 für OFF,
- 2 für ON.

Voreinstellung: OFF

INITiate:IMMediate

INITiate:IMMediate startet einen einzelnen Messzyklus. Ausgehend vom Zustand *IDLE* geht der Messkopf zunächst in den Zustand *WAIT_FOR_TRIGGER* und beginnt mit dem Messvorgang, sobald die Triggerbedingung erfüllt ist. Nach Abschluss der Messung wird wieder der Zustand *IDLE* eingenommen. Weil der Befehl während einer laufenden Messung ignoriert wird, ist er im freilaufenden Modus (Einstellung *INITiate:CONTinuous ON*) generell ohne Wirkung.

TRIGger:ATRigger:STATe[?] OFF | ON

TRIGger:ATRigger:STATe ON bewirkt, dass im Messkopf ein künstliches Triggerereignis ausgelöst wird, wenn mehr als 10 s nach Start des Messzyklus kein Trigger registriert wurde (nur im Modus *Scope*). *TRIGger:ATRigger:STATe OFF* deaktiviert die Trigger-Automatik.

Der Abfragebefehl liefert

- 1 für OFF,
- 2 für ON.

Voreinstellung: OFF

TRIGger:COUNt[?] 1 ... 2×10^9

Diese Einstellung ist für jene Anwendungen gedacht, bei denen durch einmaliges Senden des Befehls *INITiate:IMMediate* mehrere aufeinanderfolgende Messungen durchgeführt werden sollen, z. B. zum Erzielen einer höheren Messgeschwindigkeit. Damit wird die Lücke zwischen einer einmaligen Messung und dem freilaufenden Messmodus geschlossen. Die Anzahl der Messungen wird mit dem Parameter zum Befehl *TRIGger:COUNt* definiert. Diese Zahl ist auch gleich der Anzahl der Messergebnisse, die der Messkopf am Schluss zur Verfügung stellt.



Der Befehl *TRIGger:COUNt* definiert nicht die Zahl der Triggerereignisse, die zur Durchführung der gesamten Messaufgabe erforderlich sind. Je nach Messmodus können das unterschiedlich viele sein.

Eine weitere Erhöhung der Messgeschwindigkeit kann durch Kombination mit dem gepufferten Modus erreicht werden. Dabei werden die Messergebnisse nicht sequenziell, sondern erst am Schluss der Messfolge als Block zur Verfügung gestellt (siehe Befehlsgruppe *SENSe:POWer:AVG:BUFFer*).

Der Abfragebefehl liefert die Anzahl an Messungen, die nach einem Messstart mit dem Befehl *INIT:IMMediate* durchgeführt werden.

Voreinstellung: 1

TRIGger:DELay[?] x ... 100.0

TRIGger:DELay legt die zeitliche Verzögerung (in Sekunden) vom Auftreten des Triggerereignisses bis zum Beginn des eigentlichen Messvorgangs für die Modi *Timeslot* und *Scope* fest. Im Modus *Burst Average* wird dieser Parameter ignoriert. Pre-Triggerung wird durch negative Werte des Parameters

erreicht, bei bus-getriggerten Messungen (siehe *TRIGger:SOURce*) sollte der Parameter auf positive Werte oder Null eingestellt sein, um Messfehler zu vermeiden.

Der Abfragebefehl liefert den eingestellten Trigger-Delay für die Modi *Timeslot* und *Scope* (in Sekunden).

Untergrenze x des Parameters

Modi *Continuous Average*, *Burst Average* und *Timeslot*: $x = -0.005$

Modus *Scope*: $x = - (<SENSe:SWEEp:OFFSet:TIME> + 0.005)$

Voreinstellung: *0.0 [s]*

TRIGger:DELAy:AUTO[?] OFF | ON

TRIGger:DELAy:AUTO ON stellt durch eine automatisch ermittelte Wartezeit sicher, dass ein Messvorgang erst begonnen wird, wenn der Messkopf eingeschwungen ist. Dies ist vor allem bei thermischen Messköpfen wichtig. Die automatisch ermittelte Wartezeit wird ignoriert, wenn über *TRIGger:DELAy* eine längere Zeit eingestellt wurde. Der Wert von *TRIGger:DELAy* wird dadurch nicht überschrieben. *TRIGger:DELAy:AUTO OFF* deaktiviert diese Funktion.

Der Abfragebefehl liefert

- 1 für *OFF*,
- 2 für *ON*.

Voreinstellung: *ON*

TRIGger:HOLDoff[?] 0.0 ... 10.0

TRIGger:HOLDoff bewirkt ein Ausblenden von Triggerereignissen innerhalb der eingestellten Holdoff-Zeit (in s), gerechnet vom Zeitpunkt der letzten erfolgreichen Triggerung.

Der Abfragebefehl liefert die eingestellte Holdoff-Zeit (in s).

Voreinstellung: *0.0 [s]*

TRIGger:HYSteresis[?] 0.0 ... 10.0

TRIGger:HYSteresis stellt die Hysterese der internen Triggerschwelle (Parameter *TRIGger:LEVel*) ein. Unter Hysterese versteht man den Betrag (in dB), um den der Pegel des Triggersignals die Triggerschwelle unterschreiten muss (bei positiver Triggerflanke), damit eine erneute Triggerung möglich wird. Bei negativer Triggerflanke sind die Verhältnisse genau umgekehrt. Die Einstellung der Trigger-Hysterese ist nur für die Triggerquelle *INTernal* relevant.

Der Abfragebefehl liefert die Trigger-Hysterese in dB.

Voreinstellung: *0.0 [dB]*

TRIGger:IMMediate

TRIGger:IMMediate löst ein generisches Triggerereignis aus, welches bewirkt, dass der Messkopf sofort – unabhängig von Triggerquelle und Trigger-Delay – den Zustand *WAIT_FOR_TRIGGER* verlässt und mit dem Messvorgang beginnt. Dieser Befehl ist die einzige Möglichkeit, einen Messvorgang zu starten, wenn die Triggerquelle auf *HOLD* steht.

TRIGger:LEVel[?] x ... y

TRIGger:LEVel stellt die Triggerschwelle für die interne, vom Messsignal abgeleitete Triggerung ein (in W). Diese Einstellung ist für alle anderen Triggerquellen ohne Bedeutung.

Der Abfragebefehl liefert die Triggerschwelle in Watt.

Untergrenze x und Obergrenze y des Parameters

SENSe:CORRection:OFFSet:STATe OFF: $x = 500 \times \langle \text{untere Messgrenze} \rangle$
 $y = \langle \text{obere Messgrenze} \rangle$

SENSe:CORRection:OFFSet:STATe ON: $x = 500 \times \langle \text{untere Messgrenze} \rangle \times$
 $10^{\langle \text{SENSe:CORRection:OFFSet} \rangle / 10}$
 $y = \langle \text{obere Messgrenze} \rangle \times$
 $10^{\langle \text{SENSe:CORRection:OFFSet} \rangle / 10}$

$\langle \text{untere Messgrenze} \rangle$: 200.0e-12 (bei *SENSe:CORRection:SPDevice:STATe OFF*) oder eingegebene untere Messgrenze der Messkopf-Zweiter-Kombination (bei *SENSe:CORRection:SPDevice:STATe ON*)

$\langle \text{obere Messgrenze} \rangle$: 0.2 (bei *SENSe:CORRection:SPDevice:STATe OFF*) oder eingegebene obere Messgrenze der Messkopf-Zweiter-Kombination (bei *SENSe:CORRection:SPDevice:STATe ON*)

Voreinstellung: $10 \times x$

TRIGger:SLOPe[?] POSitive | NEGative

TRIGger:SLOPe definiert die Flanke des Triggerereignisses bei interner oder externer Triggerung in den Modi *Timeslot* und *Scope*. Positiv bedeutet in diesem Zusammenhang steigende Hüllkurvenleistung (bei interner Triggerung) bzw. steigende Spannung (bei externer Triggerung). In Kombination mit den Triggerquellen *BUS*, *HOLD* und *IMMEDIATE* ist dieser Befehl ebenso ohne Wirkung wie im Modus *Burst Average*.

Der Abfragebefehl liefert

- 1 für *POSitive*,
- 2 für *NEGative*.

Voreinstellung: *POSitive*

TRIGger:SOURce[?] BUS | EXTernal | HOLD | IMMEDIATE | INTernal

TRIGger:SOURce stellt die Triggerquelle ein.

- *BUS*: Triggerung durch die Befehle **TRG* oder *TRIGger:IMMEDIATE*.
- *EXTernal*: Triggerung über den USB-Adapter R&S NRP-Z3. Relevante Trigger-Parameter: *TRIGger:DELay* und *TRIGger:SLOPe*.
- *HOLD*: Triggerung nur mit dem Befehl *TRIGger:IMMEDIATE*.
- *IMMEDIATE*: Automatische Triggerung ohne explizites Ereignis.
- *INTernal*: Triggerung durch das Messsignal. Relevante Trigger-Parameter: *TRIGger:LEVel*, *TRIGger:DELay* und *TRIGger:SLOPe* (nicht im Modus *Burst Average*).

Der Abfragebefehl liefert

- 1 für *BUS*,
- 2 für *EXTernal*,
- 4 für *HOLD*,
- 8 für *IMMEDIATE*,
- 16 für *INTernal*.

Voreinstellung: *IMMEDIATE*

Liste der Fernsteuer-Befehle

Die Fernsteuerbefehle des R&S NRP-Z11/-Z21 haben eine Syntax in Anlehnung an die Norm SCPI 1999.0, entsprechen dieser jedoch nur eingeschränkt.

Tabelle 6-9 Liste der Fernsteuer-Befehle

Befehl	Parameter	Einheit	Voreinstellung	Seite
*-Befehle				
*IDN?				6.2
*RST				6.2
*TRG				6.2
*TST?				6.2
CALibration-Befehle				
CALibration:DATA[?]	<Kalibrierdatensatz als definite length Block>			6.3
CALibration:DATA:LENGth?		Bytes		6.3
CALibration:ZERO:AUTO[?]	OFF ON ONCE		OFF (fest)	6.4
SENSe-Befehle				
SENSe:AVERage:COUNT[?]	1 ... 65536		4	6.7
SENSe:AVERage:COUNT:AUTO[?]	OFF ON ONCE		ON	6.7
SENSe:AVERage:COUNT:AUTO:MTIME[?]	1.0 ... 999.99	s	30.0	6.7
SENSe:AVERage:COUNT:AUTO:NSRatio[?]	0.0001 ... 1.0	dB	0.01	6.8
SENSe:AVERage:COUNT:AUTO:RESolution[?]	1 ... 4		3	6.8
SENSe:AVERage:COUNT:AUTO:SLOT[?]	1 ... <SENSe:POWer:TSLot:AVG :COUNT>		1	6.8
SENSe:AVERage:COUNT:AUTO:TYPE[?]	RESolution NSRatio		RESolution	6.8
SENSe:AVERage:STATe[?]	OFF ON		ON	6.9
SENSe:AVERage:TCONtrol[?]	MOVing REPeat		REPeat	6.9
SENSe:CORRection:DCYClE[?]	0.001 ... 99.999	%	1.0	6.9
SENSe:CORRection:DCYClE:STATe[?]	OFF ON		OFF	6.9
SENSe:CORRection:OFFSet[?]	-100.0 ... 100.0	dB	0.0	6.10
SENSe:CORRection:OFFSet:STATe[?]	OFF ON		OFF	6.10

Befehl	Parameter	Einheit	Voreinstellung	Seite
SENSe:CORRection:SPDev:STATe[?]	OFF ON		OFF (kann vom Anwender geändert werden)	6.10
SENSe:FREQuency[?]	0.0 ... 100.0e9	Hz	50.0e6	6.10
SENSe:FUNcTION[?]	"POWer:AVG" "POWer:TSLot:AVG" "POWer:BURSt:AVG" "XTIME:POWer"		"POWer:AVG"	6.10
SENSe:POWer:AVG:APERture[?]	0.0001 ... 0.1	s	0.02	6.12
SENSe:POWer:AVG:BUFFer:SIZE[?]	1 ... 1024		1	6.12
SENSe:POWer:AVG:BUFFer:STATe[?]	OFF ON		OFF	6.12
SENSe:POWer:AVG:SMOothing:STATe[?]	OFF ON		ON	6.13
SENSe:POWer:BURSt:DTOLerance[?]	0.0 ... 0.003	s	0.0001	6.13
SENSe:POWer:TSLot:AVG:COUNt[?]	1 ... 128		8	6.14
SENSe:POWer:TSLot:AVG:WIDTh[?]	0.0001 ... 0.1	s	0.001	6.14
SENSe:RANGe[?]	0 ... 2		2	6.14
SENSe:RANGe:AUTO[?]	OFF ON		ON	6.15
SENSe:RANGe:AUTO:CLeVel[?]	-20.0 ... 0.0	dB	0.0	6.15
SENSe:SAMPLING[?]	FREQ1 FREQ2		FREQ1	6.15
SENSe:SGAMma:CORRection:STATe[?]	OFF ON		OFF	6.15
SENSe:SGAMma:MAGNitude[?]	0.0 ... 1.0		0.0	6.15
SENSe:SGAMma:PHASe[?]	-360.0 ... 360.0	Grad	0.0	6.16
SENSe:SWEep:AVERAge:COUNt[?]	1 ... 65536		4	6.16
SENSe:SWEep:AVERAge:COUNt:AUTO[?]	OFF ON ONCE		ON	6.16
SENSe:SWEep:AVERAge:COUNt:AUTO:MTIME[?]	1.0 ... 999.99	s	30.0	6.16
SENSe:SWEep:AVERAge:COUNt:AUTO:NSRatio[?]	0.0001 ... 1.0	dB	0.01	6.17
SENSe:SWEep:AVERAge:COUNt:AUTO:POINt[?]	1 ... <SENSe:SWEep:POINts>		1	6.17
SENSe:SWEep:AVERAge:COUNt:AUTO:RESolution[?]	1 ... 4		3	6.17
SENSe:SWEep:AVERAge:COUNt:AUTO:TYPE[?]	RESolution NSRatio		RESolution	6.17
SENSe:SWEep:AVERAge:STATe[?]	OFF ON		ON	6.18

Befehl	Parameter	Einheit	Voreinstellung	Seite
SENSe:SWEep:AVERAge:TCONtrol[?]	MOVing REPeat		REPeat	6.18
SENSe:SWEep:OFFSet:TIME[?]	– (<TRIGger:DELay> + 0.005) ... 100.0	s	0.0	6.18
SENSe:SWEep:POINts[?]	1 ... 1024		100	6.18
SENSe:SWEep:REALtime[?]	OFF ON		OFF	6.18
SENSe:SWEep:TIME[?]	0.0001 ... 0.3	s	0.01	6.19
SENSe:TIMing:EXCLude:STARt[?]	0.0 ... 0.003	s	0.0	6.19
SENSe:TIMing:EXCLude:STOP[?]	0.0 ... 0.003	s	0.0	6.19
SYSTEM-Befehle				
SYSTEM:INFO? [Item]				6.21
SYSTEM:INITialize				6.22
SYSTEM:MINPower?		W		6.22
SYSTEM:TRANsaction:BEgIn				6.22
SYSTEM:TRANsaction:END				6.23
Test-Befehle				
TEST:SENSor?				6.24
Triggersystem-Befehle				
ABORt				6.25
INITiate:CONTinuous[?]	OFF ON		OFF	6.25
INITiate:IMMediate				6.26
TRIGger:ATRigger:STATe[?]	OFF ON		OFF	6.26
TRIGger:COUNt[?]	1 ... 2×10 ⁹		1	6.26
TRIGger:DELay[?]	x ... 100.0	s	0.0	6.26
TRIGger:DELay:AUTO[?]	OFF ON		OFF	6.27
TRIGger:HOLDoff[?]	1.0e-9 ... 10.0	s	0.0	6.27
TRIGger:HYSTeresis[?]	0.0 ... 10.0	dB	0.0	6.27
TRIGger:IMMediate				6.27
TRIGger:LEVel[?]	x ... y	W	10 × x	6.27
TRIGger:SLOPe[?]	POSitive NEGative		POSitive	6.28

Befehl	Parameter	Einheit	Voreinstellung	Seite
TRIGger:SOURce[?]	BUS EXTernal HOLD IMMEDIATE INTernal		IMMEDIATE	6.28
SERVICE-Befehle				
SERVICE:CALibration:DITHer	ONCE		OFF	
SERVICE:CALibration:DITHer:DATA?			0	
SERVICE:CALibration:TEMP	ONCE		OFF	
SERVICE:CALibration:TEMP:DATA?		K	0.0	
SERVICE:CALibration:TEST[?]			-1	
SERVICE:CALibration:ZERO:NEG0?			1	
SERVICE:CALibration:ZERO:POS0?			2	
SERVICE:CALibration:ZERO:NEG1?			3	
SERVICE:CALibration:ZERO:POS1?			4	
SERVICE:CALibration:ZERO:NEG2?			5	
SERVICE:CALibration:ZERO:POS2?			6	
SERVICE:DITHer	OFF ON		ON	
SERVICE:MVCORrection[?]	0 ... 63		63	
SERVICE:PARAmeter:RTEMP[?]	<Float-Wert>	K	0.0	
SERVICE:PARAmeter:RNULL1[?]	<Float-Wert>	Ω	0.0	
SERVICE:PARAmeter:RNULL2[?]	<Float-Wert>	Ω	0.0	
SERVICE:PARAmeter:RNULL3[?]	<Float-Wert>	Ω	0.0	
SERVICE:PARAmeter:RBAHN[?]	<Float-Wert>	Ω	0.0	
SERVICE:PARAmeter:NREF[?]	<Float-Wert>		0.0	
SERVICE:PARAmeter:ATHERM[?]	<Float-Wert>	K^{-1}	0.0	
SERVICE:PARAmeter:BTHERM[?]	<Float-Wert>	K	0.0	
SERVICE:PARAmeter:CTHERM[?]	<Float-Wert>	K^{-1}	0.0	
SERVICE:PARAmeter:DTHERM[?]	<Float-Wert>	K^{-1}	0.0	
SERVICE:PARAmeter:CJUNC[?]	<Float-Wert>	F	0.0	
SERVICE:RCOUNT[?]	1 ... 32767		0	
SERVICE:RESult[?]	0.0 ... 1.0e6	W	0.0	

Befehl	Parameter	Einheit	Voreinstellung	Seite
SERVice:SAMPle[?]	0 ... 99999999		1000	
SERVice:UNLock	1234			

